جهاد
واحد صنعتی اصفهان

اصول عملی
آنالیز داده‌های توالی‌یابی RNA

تألیف:
ایجا کورپلاینن
جارنو تویمالا
پانو سامرویو
میکائیل هاس
گری وانگ

ترجمه:
فرجاد رفیعی، حبیب‌اله سمیع‌زاده لاهیجی

۲-۳۴۸

اصول عملی
آنالیز داده‌های توالی‌یابی RNA

تألیف:

ایجا کورپلاینن، جارنو تویمالا، پانو سامرویو،
میکائیل هاس، گری وانگ

ترجمه:

فرجاد رفیعی
استادیار گروه بیوتکنولوژی دانشکده علوم کشاورزی
دانشگاه گیلان

حبیب‌اله سمیع‌زاده لاهیجی
استاد گروه بیوتکنولوژی دانشکده علوم کشاورزی
دانشگاه گیلان

۱۳۹۸

سخ

پیشگفتار مولفان

توالی‌یابی RNA اطلاعات بی‌نظیری را در مورد ترانسکریپتوم فراهم آورده است. ولی بهره‌برداری از این اطلاعات با استفاده از ابزارهای بیوانفورماتیک معمولاً چالش‌برانگیز است. هدف از نگارش این کتاب آن است که توانایی خوانندگان در آنالیز داده‌های توالی‌یابی RNA را ارتقا ببخشد. چندین موضوع به طور مفصل مورد بحث و بررسی قرار گرفته و گردش کار کامل آنالیز داده‌ها از کنترل کیفیت، مکان‌یابی و اسمبل کردن تا آزمون آماری و آنالیز مسیر پوشش داده شده است. هدف این بوده است که به جای حداقل نمودن همپوشانی با کتاب‌های مرجع موجود، مطالب به صورت جامع‌تر و کاربردی‌تر ارائه گردد.

کتاب حاضر این امکان را برای محققین فراهم می‌آورد که تفاوت بیان در سطوح ژن، اگزون و رونوشت را مورد بررسی قرار داده و ژن‌ها، رونوشت‌ها و ترانسکریپتوم‌های جدید را شناسایی کنند. با توجه به اهمیت نقش تنظیمی RNA های نارمزگر کوچک، بخش‌هایی از این کتاب به طور کامل به شناسایی و آنالیز عملکردی آنها اختصاص داده شده است.

گروه هدف این کتاب دانشجویان و محققین در سطوح تکمیلی و عالی هستند. مثال‌های عملی به گونه‌ای انتخاب شده‌اند که نه تنها متخصصین بیوانفورماتیک بلکه متخصصین آزمایشگاهی که به برنامه‌نویسی آشنا نیستند نیز بتوانند از آنها بهره گرفته و همین امر نیز این کتاب را برای طیف وسیعی از محققین با پشتوانه علمی متفاوت از علوم زیست‌شناسی، پزشکی، ژنتیک و رایانه مفید و قابل استفاده می‌سازد. این کتاب را می‌توان به عنوان یک کتاب مرجع در مقاطع تحصیلات تکمیلی و نیز دوره‌ها و کارگاه‌های آموزشی کوتاه‌مدت به کار گرفت. این کتاب می‌تواند به عنوان یک مرجع جامع برای روش‌های آنالیز داده‌های توالی‌یابی RNA و چگونگی استفاده عملی از آنها به کار گرفته شود.

در این کتاب، بین مطالب نظری و عملی تعادل برقرار شده است. بدین منظور هر فصل با ارائه مطالب نظری شروع شده و سپس با توصیف ابزارهای معمول آنالیز ادامه یافته و سپس مثال‌هایی از کاربرد ابزارهای مزبور ارائه می‌گردد. چون هدف این بوده است که کتاب حاضر یک مجموعه جامع باشد که برای متخصصین آزمایشگاهی ناآشنا به برنامه‌نویسی رایانه‌ای نیز قابل استفاده باشد، لذا نرم‌افزار گرافیکی Chipster نیز علاوه بر ابزارهای خط فرمان مورد بررسی قرار گرفته است. کلیه نرم‌افزارهای مورد استفاده در مثال‌ها، از نوع متن باز بوده و به صورت رایگان در دسترس هستند.

در فصل‌های اول و دوم به عنوان بخش مقدماتی، کاربردهای توالی‌یابی RNA از شناسایی ژن‌ها و رونوشت‌ها تا آنالیز افتراقی بیان و شناسایی جهش‌ها و ژن‌های تلفیقی مورد بحث و بررسی واقع

شده است. این فصل‌ها دیدگاهی کُلی از آنالیز داده‌های توالی‌یابی RNA ارائه کرده و جنبه‌های مهم طراحی آزمایشات را مورد بررسی قرار می‌دهند.

سپس ابتدا در مورد مکان‌یابی خوانش‌ها روی مرجع و اسمبل کرن از نو بحث می‌شود. چون هر دو مورد فوق به شدت تحت تاثیر کیفیت خوانش‌ها واقع می‌شوند، لذا یک فصل به کنترل کیفیت و پیش‌پردازش اختصاص یافته است. در فصل سوم چندین چالش مرتبط با کیفیت داده‌های توالی‌یابی پُربروندا و ابزارهای شناسایی و رفع آنها مورد بررسی قرار می‌گیرند. در فصل چهارم چالش‌های مکان‌یابی خوانش‌های توالی‌یابی RNA با یک مرجع مورد بحث قرار گرفته و برخی از همردیف‌سازهای معمول همراه با مثال‌های عملی و نیز ابزارهای دست‌ورزی فایل‌های همردیفی و مرورگرهای ژنومی برای مصورسازی خوانش‌ها در زمینه ژنومی معرفی می‌شوند. در فصل پنجم عناصر اسمبل کردن ترانسکریپتوم و ارتباط بین مراحل پردازش داده‌ها نظیر پاکسازی، پیرایش و تصحیح خطا در اسمبل کردن توالی‌یابی RNA تشریح می‌شوند. مفاهیم پایه نظیر نمودار پیرایش، نمودار دوبران و گذرگاه مسیر در یک نمودار اسمبل توضیح داده شده و تفاوت‌های بین اسمبل کردن ژنوم و ترانسکریپتوم نیز مورد بحث قرار می‌گیرد. دو روش بازسازی طول کامل رونوشت‌ها شامل اسمبل کردن مبتنی بر مکان‌یابی و اسمبل کردن از نو تشریح می‌گردد. برای هر دو روش مزبور نیز مثال‌های عملی ارائه می‌شود.

در بخش بعدی کتاب بر آنالیزهای آماری که عمدتاً توسط نرم‌افزار R همراه با ابزارهای افزوده شده توسط پروژه Bioconductor انجام می‌شوند، تمرکز می‌گردد. در فصل ششم روش‌ها و ابزارهای مختلف کمی‌سازی همراه با معیارهای کیفی مبتنی بر حاشیه‌نگاری معرفی می‌گردند. در فصل هفتم چارچوب‌های مبتنی بر R و Bioconductor برای آنالیز داده‌های توالی‌یابی RNA و نحوه وارد نمودن داده‌ها در R بررسی می‌شوند. تفاوت‌های اصلی بین ابزارهای آماری و بیوانفورماتیکی نیز تشریح می‌گردند. فصل‌های هشتم و نهم شامل گزینه‌های مختلف برای آنالیز افتراقی بیان ژن‌ها، رونوشت‌ها و اگزون‌ها بوده و نحوه اجرای این آنالیزها با استفاده از ابزارهای R یا Bioconductor و برخی از ابزارهای دیگر را نشان می‌دهند. فصل دهم راه‌هایی را برای حاشیه‌نگاری نتایج معرفی کرده و در فصل یازدهم نیز روش‌های مختلفی برای مصورسازی و نمایش نتایج ارائه می‌گردد.

آخرین بخش کتاب بر آنالیز RNA های نارمزگر کوچک با کمک ابزارهای تحت وب یا قابل دانلود رایگان متمرکز است. فصل دوازدهم انواع مختلف RNA های نارمزگر کوچک و خصوصیات عملکردی، فراوانی و توالی آنها را مورد بررسی قرار می‌دهد. فصل سیزدهم الگوریتم‌های مختلف برای شناسایی RNA های نارمزگر کوچک با استفاده از مجموعه داده‌های توالی‌یابی نسل جدید را معرفی کرده و روش عملی شناسایی و حاشیه‌نگاری RNA های نارمزگر کوچک را همراه با چند

مثال ارائه می‌دهد. علاوه بر این در این فصل ابزارهای پایین دستی که می‌توانند برای توضیح عملکرد RNA های نارمزگر کوچک مورد استفاده قرار گیرند، تشریح می‌شوند.

از استادان انتشارات CRC که امکان نگارش یک کتاب مرجع در زمینه توالی‌یابی RNA را فراهم آورده‌اند، تشکر می‌نماییم. در این بین به ویژه راهنمایی‌های Sunil Nair ، Sarah Gelson و Stephanie Morket در طی فرآیند نگارش، صبوری بیش از حد و پاسخگویی سریع‌شان بسیار راهگشا بوده است. همچنین از همکاران‌مان در آزمایشگاه‌های مختلف، VuokkoAarnio ، LiisaHeikkinen و JuhaniPeltonen ، که وقت ارزشمند خود را صرف خواندن و ارائه نظر در فصل‌های مختلف کردند، عمیقاً سپاسگزاری می‌نماییم. Tommy Kissa در مراحل پایانی نگارش این اثر بدون هیچ توقعی مشارکت نموده و الهام‌بخش اعضای گروه بوده است. در پایان از همسران و فرزندان خود، Lily ، Philippe ، Stefan ، Sanna و Merjaکه به عنوان دستیار پژوهشی، داور، هنرمندگرافیک و پشتیبانی رایانه، هتلدار، رستوران‌دار، خدمتکار و روانکاو نقش ایفا کرده تشکر نموده و این کتاب را با عشق به محضرشان تقدیم می‌کنیم.

ایجا کورپلاینن

(متخصص بیوانفورماتیک در CSC-IT Center for Science)

جارنو تویمالا

(متخصص آمار زیستی و بیوانفورماتیک در
صلیب سرخ فنلاند و CSC-IT Center for Science)

پانو سامرویو

(محقق در دانشگاه هلسینکی، فنلاند)

میکائیل هاس

(متخصص بیوانفورماتیک در SciLifeLab's WABI استکهلم، سوئد)

گری وانگ

(استاد زیست‌پزشکی در دانشگاه ماکائو، چین)

پیشگفتار مترجمان

ای دهـــنـده‌ی عـقـل‌هـا فـریـــاد رس / گـــر نـخـواهـی تـو نخـواهـد هیـچ‌کـس
هم طلـب از توست و هم آن نیکـــویی / مـا کـی‌ایـم، اول تـــویـی، آخـر تـــویـی
دو دهـان داریـم گـویـا هـمـچـو نـــی / یـک دهـــان پنهانسـت در لـب‌هـای وی
یـک دهـان نـالان شـده سـوی شـــما / هـــای و هـــویـــی در فـکـنـده در هـــوا
لیـک دانـد هـر کـه او را منـظرســـت / کـه فـغـان این سـری هـم زان سـرست
دمـدمـه ایـن نـای از دم‌هـای اوســت / هـای و هـــوی روح از هـیـهـای اوسـت
گـر نـبـودی بـا لـبـش نـی را ســمـر / نـــی جـهـان را پـــر نـکـردی از شـکـر
گـــرد ایـن بـام و کـبـوتـرخـانـه مـــن / چـــون کـبـوتـر پـر زنـم مـسـتانه مـن
جـبـرئـیـل عـــشـقـم و سـدره‌م تـویی / مـن سـقیـمـم عیـسـی مـریـم تـــویی
جـــوش ده آن بـــحـــر گـوهـربـــار را / خـوش بـپـرس امـــروز ایـن بـــیـمار را

«مولوی»

در پی پیشرفت روز افزون دانش بشری در زمینه زیست‌شناسی مولکولی و مهندسی ژنتیک، پیدایش نسل جدید روش‌های توالی‌یابی اسیدهای نوکلئیک به مثابه انقلابی عظیم در این حوزه وارد شده و مباحث مربوط به آن نیز به نحوی چشمگیر رشد و گسترش یافته است. کتاب حاضر، یکی از منابع تقریباً منحصر به فرد در رابطه با آنالیز داده‌های توالی‌یابی RNA بوده که در سال‌های اخیر، همگام با سایر مجامع علمی جهان، در دانشگاه‌های ایران و به خصوص در دوره‌های تحصیلات تکمیلی، مورد توجه و تدریس قرار گرفته است. بدین منظور برای رفع کمبود منابع نوشتاری تخصصی در این زمینه، نسبت به ترجمه این کتاب ارزشمند اقدام گردید. در این کتاب، مباحث نظری همراه با مثال‌های عملی و کاربردی در زمینه آنالیز داده‌های توالی‌یابی RNA مورد بحث و بررسی قرار گرفته و از این نظر یکی از منابع جامع و کامل محسوب می‌گردد.

مترجمان بر این امر معترفند که با وجود دقت فراوان، ترجمه حاضر خالی از ایراد و اشکال نیست. لذا از کلیه اهالی فضل و دانش مصرانه تمنا می‌شود که هرگونه نقد و نظر در رابطه با مطالب کتاب را برای مترجمین ارسال دارند تا امکان جبران کاستی‌ها در چاپ‌های بعدی فراهم گردد.

فرجاد رفیعی[1] - حبیب‌اله سمیع‌زاده لاهیجی[2]

بهار ۱۳۹۸

۱- استادیار گروه بیوتکنولوژی دانشکده علوم کشاورزی دانشگاه گیلان farjad.rafeie@guilan.ac.ir
۲- استاد گروه بیوتکنولوژی دانشکده علوم کشاورزی دانشگاه گیلان hsamizadeh@yahoo.com

هو الستّار

به جهان خرم از آنم که جهان خرم از اوست

عاشقم بر همه عالم که همه عالم از اوست

«سعدی»

ترجمه کتاب حاضر جهت استفاده دانش پژوهان این مرز و بوم، وقف عام می گردد. از همه بهره‌مندان تمنا می شود که دعای خیر خود را بدرقه راه کلیه کسانی کنند که دَینی بر گردن این بنده عاصی دارند تا ان شاء الله از این طریق صاحبان دَین از این حقیر نیز درگذرند.

هر گونه بهره برداری انتفاعی از این ترجمه به هر نحو (اعم از چاپ، تکثیر و نظایر آن) ممنوع بوده و متخلف به ضمان شرعی و قانونی می گردد. هر آن کس که در نشر رایگان این اثر سهمی داشته باشد، ان شاء الله از منافع معنوی آن نیز برخوردار گردد.

هر گونه استفاده عام المنفعه و رایگان و نیز نقل مطالب این اثر با یا بدون ذکر منبع آزاد بوده و تنها تمنای دعای خیر از همه بهره‌مندان می شود.

«فرجاد رفیعی - حبیب اله سمیع زاده لاهیجی»

فهرست مطالب

فصل اول: مقدمه‌ای بر توالی‌یابی RNA

فصل دوم: مقدمه‌ای بر آنالیز داده‌های توالی‌یابی RNA

فصل سوم: کنترل کیفیت و پیش‌پردازش

فصل چهارم: همردیف‌سازی خوانش‌ها با مرجع

فصل پنجم: اسمبل کردن ترانسکریپتوم

فصل ششم: کمی‌سازی و کنترل کیفیت مبتنی بر حاشیه‌نگاری

فصل هفتم: چارچوب آنالیز توالی‌یابی RNA در R و Bioconduictor

فصل هشتم: آنالیز افتراقی بیان

فصل نهم: آنالیز افتراقی استفاده از اگزون

فصل دهم: حاشیه‌نگاری نتایج

فصل یازدهم: مصورسازی

فصل دوازدهم: RNA های نارمزگر کوچک

فصل سیزدهم: آنالیز محاسباتی داده‌های توالی‌یابی RNA های نارمزگر کوچک

فصل اول

مقدمه‌ای بر توالی‌یابی RNA

۱–۱ مقدمه

توالی‌یابی RNA[1] شامل مجموعه‌ای از روش‌های آزمایشگاهی و محاسباتی برای تعیین ماهیت و فراوانی توالی‌های RNA در نمونه‌های زیستی است. در نتیجه‌ی این کار، ترتیب هر آدنوزین، سیتوزین، گوانین و یوراسیل موجود در یک مولکول RNA تک رشته‌ای مشخص می‌گردد. روش‌های آزمایشگاهی شامل استخراج و جداسازی RNA از سلول، بافت یا نمونه‌های کامل حیوانی، ایجاد کتابخانه‌های نشان دهنده‌یانواع RNA در نمونه‌ها، توالی‌یابی شیمیایی واقعی کتابخانه و آنالیز بیوانفورماتیکی نهایی است. تفاوت اصلی بین توالی‌یابی RNA و روش‌های قبلی نظیر ریزآرایه‌ها[2] عبارت از کارآمدی بسیار بالاتر پلتفرم‌های[3] فعلی توالی‌یابی RNA ، حساسیت بیشتر حاصل از کاربرد فناوری‌های جدیدتر، قابلیت شناسایی رونوشت[4] جدید، مدل‌های ژنی و انواع RNA نارمزگر کوچک[5] است.

روش‌های توالی‌یابی RNA از تغییرات نسل به نسل در فناوری‌های توالی‌یابی حاصل شده‌اند. منظور از نخستین نسل توالی‌یابی پُربروندداد[6] (کارآمد)، همان توالی‌یابی دی‌دئوکسی سَنگِر[7] است. با به کارگیری موبین الکتروفورز[8] در سنجش طول قطعات اسید نوکلئیک، یک اجرای[9] استاندارد می‌تواند از ۹۶ لوله‌ی موبین استفاده کرده و در هر لوله‌ی موبین نیز یک توالی با طول ۶۰۰ تا ۱۰۰۰ باز و به طور تقریبی مجموعاً ۱۰۰۰۰۰ باز را تعیین توالی نماید. نسل دوم توالی‌یابی که تحت عنوان توالی‌یابی نسل جدید[10] (NGS) نیز شناخته می‌شود، شامل روش‌هایی است که در آنها توالی‌یابی توسط سنتز شیمیایی تک نوکلئوتیدها و در یک قالب گسترده به صورت موازی انجام می‌شود. این کار به گونه‌ای صورت می‌گیرد که میلیون‌ها واکنش توالی‌یابی در یک اجرای واحد

1- RNA-Seq
2- Microarray
3- Platform
4- Transcript
5- Small noncoding RNA
6- High-throughput
7- Sanger dideoxy sequencing
8- Capillary electrophoresis
9- Run
10- Next Generation Sequencing (NGS)

می‌تواند انجام گردد. یک اجرای معمول NGS می‌تواند شامل ۶۰۰۰ میلیون واکنش توالی‌یابی ۱۰۰ نوکلئوتیدی بوده و اطلاعات ۶۰۰ میلیارد باز موجود در توالی را فراهم آورد. توالی‌یابی نسل سوم نیز شامل روش‌هایی مبتنی بر واکنش‌های موازی و استفاده از توالی‌یابی بر مبنای سنتز شیمیایی است. ولی در این روش‌ها از تک مولکول‌های DNA یا RNA به عنوان الگو بهره گرفته می‌شود. در پلتفرم‌های توالی‌یابی نسل سوم، از حدود چند میلیون واکنش توالی‌یابی کمتر در هر اجرا استفاده می‌شود. ولی طول توالی در هر واکنش می‌تواند بزرگ‌تر بوده و به راحتی در محدوده‌ی ۱۵۰۰ نوکلئوتید اجرا شود.

داده‌های حاصل از یک آزمایش توالی‌یابی RNA می‌توانند در تولید دانش نوین نقش ایفا کنند. به عنوان مثال محدوده‌ی این دانش‌ها از شناسایی پروتئین‌های جدید رمز شده توسط رونوشت‌ها در سلول‌های بنیادی تا شناسایی رونوشت‌های فرابیان[1] در رده‌های سلولی سرطان پوست گسترده است. سوالاتی که در این حوزه می‌توانند طرح شوند، عبارتند از: سطح بیان ژن در سلول‌های طبیعی و سرطانی چه تفاوتی دارد؟ در رده‌های سلولی که یک ژن سرکوبگر تومور از بین رفته است، سطوح بیان ژن چگونه است؟ بیان ژن در رده‌ی سلولی مورد نظر قبل و بعد از تیمار جهش‌زایی چه تفاوتی نشان می‌دهد؟ در طی تکامل مغز، کدام ژن‌ها دچار افزایش بیان[2] می‌شوند؟ چه رونوشت‌هایی در پوست حضور داشته ولی در ماهیچه دیده نمی‌شوند؟ در طی تنش اکسیداتیو، پیرایش[3] ژن چگونه تغییر می‌کند؟ چه miRNA های جدیدی را می‌توان در یک نمونه‌ی سلول بنیادی جنینی کشف نمود؟ همان‌گونه که ملاحظه می‌شود، دامنه پرسش‌هایی که می‌توانند طرح شوند، وسیع است.

توقعات از ترانسکریپتومیک[4] هنگامی افزایش یافت که فناوری‌های توالی‌یابی RNA نشان دادند که دانش فعلی از ساختار ژن و حاشیه‌نگاری[5] عمومی ژن‌ها، از جانداران مدل تک‌سلولی تا سلول‌های انسانی کاملاً ضعیف است. داده‌های جدید حاصل از پلتفرم‌های توالی‌یابی RNA تنوع بسیار وسیعی در ساختار ژن‌ها نشان داده (که تحت عنوان ژن‌های ناشناخته جدید از آنها یاد می‌شود) و نوری را بر رونوشت‌های نارمزگر با طول‌های کوتاه و بلند تابانیده‌اند. مطالعات بعدی داده‌های زیادی را از بسیاری از گونه‌های جدید، که اطلاعات موجود از توالی‌های رونوشتی آنها بسیار محدود بوده است، فراهم آورد. سرعت پژوهش‌ها به نحوی بوده است که یک قیاس شناخته

1- Over-expressed
2- Up-regulated
3- Splicing
4- Transcriptomics
5- Annotation

شده در جامعه توالی‌یابی این شده است که هزینه توالی‌یابی با نرخی سریع‌تر از قانون مور[1] کاهش می‌یابد. این مزیت اقتصادی دستاوردهای بزرگی در بهره‌وری و حتی فراتر از انتظار ایجاد می‌کند.

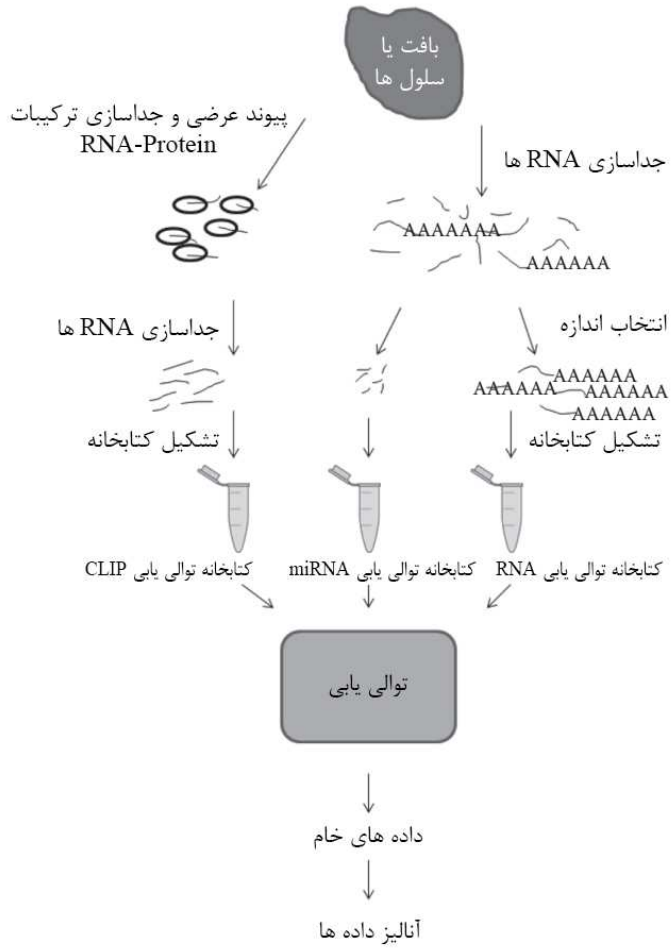
این کتاب بر اصول عملی روش‌های آنالیز داده‌ها در توالی‌یابی RNA تاکید می‌نماید. با این حال نمی‌توان اصول مزبور را بدون ارائه روش‌های آزمایشگاهی تشریح نمود. در این فصل، مباحث مقدماتی ضروری ارائه گردیده، برخی از دستورالعمل‌های[2] اولیه تشریح شده، یک گردش کار[3] ارائه گردیده و سرانجام چند نرم‌افزار مفید معرفی می‌شوند. در انتهای این فصل خواننده درک بهتری از کُل فرآیند و مراحل آن، از مفاهیم پروژه تا تجسم و تفسیر نتایج، خواهد داشت.

یک گردش کار معمول در توالی‌یابی RNA در نگاره‌ی ۱-۱ ارائه گردیده است. شروع این گردش کار با فرآیندهای آزمایشگاهی بوده و انتهای آن به مدیریت داده‌ها و آنالیز ختم می‌شود.

۱-۲ جداسازی RNA

معمولاً RNA از نمونه‌های بافت یا سلول‌های تازه برداشت شده یا منجمد و با استفاده از کیت‌های تجاری در دسترس نظیر RNAEasy (کایاژن، هایلدن، آلمان[4])، TRIZOL (لایف تکنولوژیز، کارلزبد، کالیفرنیا[5]) یا RiboPure (امبیون، اوستین، تگزاس[6]) جدا[7] (استخراج) می‌شود. مزیت این کیت‌ها، آسان بودن استفاده از آنها و تولید مقادیر زیاد RNA کُل در صورت کاربرد صحیح آنها است. سامانه‌های کارآمدی نیز برای جداسازی RNA وجود دارند که عمدتاً بر مبنای اتصال RNA به ذرات مغناطیسی استوار هستند. این ذرات، شستشو و جداسازی RNA را تسهیل می‌نمایند. علاوه بر این می‌توان RNA را از بافت‌های تثبیت شده در فرمالین و قالب‌گیری شده در پارافین نیز جداسازی نمود، هرچند که روشی ایده‌آل نمی‌باشد. برای پرهیز از تخریب RNA ، نمونه‌ها را می‌توان در مواد نگهدارنده‌ی RNA نظیر RNAlater (امبیون) نگهداری کرده یا ابتدا به صورت جزیی فرآوری نموده و سپس در یک امولسیون فنلی (TRIZOL) ذخیره نمود. همچنین در این مرحله می‌توان نمونه‌های RNA را برای رده‌های اندازه‌ای خاص نظیر RNA های کوچک[8] با کمک سامانه‌های ستونی (miRVana ، امبیون) غنی‌سازی نمود. علاوه بر این می‌توان نمونه‌ها را

1- Moor's law
2- Protocol
3- Workflow
4- Qiagen, Hilden, Germany
5- Life Technologies, Carlsbad, CA
6- Ambion, Austin, TX
7- Isolation
8- Small RNA



نگاره‌ی ۱-۱: نمای کُلی از آزمایشات توالی‌یابی RNA . این گردش کار از بافت تا داده‌های حاصل از روش توالی‌یابی RNA را با گزینه‌هایی برای توالی‌یابی CLIP ، توالی‌یابی miRNA و توالی‌یابی RNA کُل نشان می‌دهد.

ابتدا به صورت RNA کُل استخراج نموده و سپس با استفاده از الکتروفورز روی ژل پُلی‌اکریلامید[1] بر اساس اندازه جداسازی و انتخاب نمود.

تقریباً در کلیه‌ی مواردِ جداسازی RNA کُل، نمونه با DNA ژنومی آلوده می‌شود. این موضوع اجتناب‌ناپذیر بوده و حتی اگر این آلودگی بسیار ناچیز باشد، در نهایت حساسیت و کارآمدی

1- Polyacrylamide gel electrophoresis

توالی‌یابی RNA تحت تاثیر قرار خواهد گرفت. بنابراین معمول‌ترین کار این است که برای هضم آلودگی DNA پیش از آماده‌سازی کتابخانه، نمونه‌های RNA کُل جدا شده، با DNase تیمار شوند. اکثر کیت‌های DNase حاوی موادی هستند که DNase را در پایان مرحله‌ی حذف DNA، غیرفعال می‌کنند. مقدار RNA کُل مورد نیاز برای تشکیل کتابخانه‌ی توالی‌یابی RNA متفاوت است. دستورالعمل‌های استاندارد برای تشکیل کتابخانه نیازمند ۰/۱ تا ۱۰ میکروگرم RNA کُل بوده و دستورالعمل‌های با حساسیت بالا نیز می‌توانند با مقادیر ناچیز در حد ۱۰ پیکوگرم RNA نیز کتابخانه را تشکیل دهند. با رایج شدن جداسازی RNA از تک سلول‌ها، دسترسی به کیت‌های اختصاصی برای چنین کاربری‌هایی نیز در حال متداول شدن است.

۱-۳ کنترل کیفیت RNA

بهترین کار این است که کیفیت RNA ها از نظر تجزیه شدن، خلوص و کمیت قبل از تشکیل کتابخانه بررسی شوند. برای این مرحله چندین پلتفرم موجود است. نانودراپ[1] و ابزارهای مشابه آن میزان جذب فلورسنت[2] نمونه‌های اسید نوکلئیک را در ۲۶۰ و ۲۸۰ نانومتر اندازه‌گیری می‌کنند. این کار نیازمند تنها مقدار اندکی در حدود کمتر از یک میکرولیتر از مایع مورد نظر برای اندازه‌گیری بوده و بدین ترتیب می‌توان یک نمونه را رقیق کرده و یا چند نانولیتر از نمونه‌ی اولیه را مورد استفاده قرار داد. استفاده از این ابزار بسیار آسان بوده و برای به دست آوردن یک خوانش تنها چند ثانیه زمان نیاز داشته و به طور همزمان نمونه‌های زیادی را می‌توان به کار گرفت. علی‌رغم اینکه این دستگاه‌ها می‌توانند میزان جذب نمونه را اندازه بگیرند، ولی نمی‌توانند بین RNA و DNA تمایز قائل شوند و در نتیجه نمی‌توانند نشان دهند که آیا نمونه‌ی RNA آلوده به DNA است یا خیر. علاوه بر این RNA تخریب شده نیز خوانشی مشابه RNA سالم داشته و در نتیجه نمی‌توان کیفیت تقریبی نمونه را به دست آورد. ولی نسبت جذب ۲۶۰ به ۲۸۰ اطلاعاتی را در مورد آلودگی پروتئینی ارائه می‌دهد.

چون پایپت[3] کردن نمونه‌ها در دامنه‌ی نانولیتر در محدوده‌ی پایپتورهای[4] آزمایشگاهی رایج می‌باشد، صحت اندازه‌گیری در پایین‌ترین دامنه‌های غلظت (ng/μL) ممکن است چالش‌برانگیز باشد. کیبِت‌فلورومتر[5] (لایف تکنولوژیز) و سامانه‌های مشابه که میزان فلورسنت محصولات مشتق

1- Nanodrop
2- Fluorescent absorbance
3- Pipetting
4- Pipettor
5- QubitFluorometer

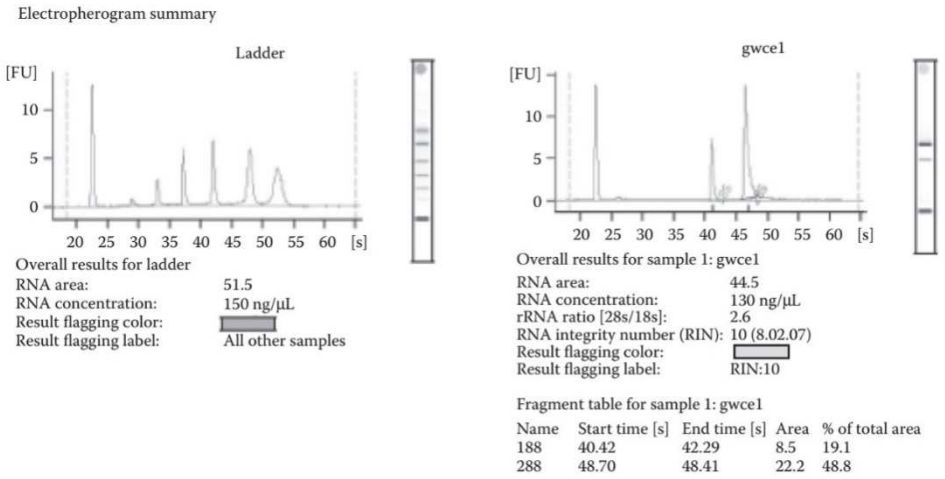
شده از اسیدهای نوکلئیک را اندازه‌گیری می‌کنند، سنجش RNA یا DNA در نمونه‌ها را به صورت مستقیم‌تری انجام می‌دهند. با استفاده از سنجش استانداردهای با غلظت پایین همراه با به کارگیری مقادیر فلورسنت روی یک خط رگرسیون کالیبراسیون از استانداردها و متعاقب آن نقطه‌یابی سنجش‌های فلورسنت نمونه روی این خط رگرسیون، سبب می‌شود که سنجش‌های اختصاصی‌تر، صحیح‌تر و در دامنه‌ی دینامیک وسیع‌تری از مقادیر RNA ها به دست آید. علاوه بر این، مقادیر اندک کمتر از یک میکرولیتر برای سنجش کافی بوده و حتی می‌توان این مقادیر را نیز رقیق نمود. علی‌رغم سادگی استفاده از این روش، این سامانه‌ها نیز نمی‌توانند مقدار تجزیه‌شدگی RNA را نشان دهند. برای رفع این مشکل، بایستی از ابزرهای دیگری استفاده نمود.

اجیلنت بایوآنالایزر[۱] یک سامانه‌ی میکروفلوییدی[۲] مبتنی بر مویین الکتروفورز[۳] برای سنجش اسیدهای نوکلئیک است. این سامانه میانگین حساسیت و حجم‌های اندک را با هم تلفیق می‌کند. الکتروفورز نیز برای اندازه‌بندی[۴] نمونه‌های اسید نوکلئیک به کار گرفته می‌شود. وقتی که استانداردهای اندازه نیز در کنار نمونه‌ها الکتروفورز می‌شوند، اندازه‌بندی و کمی‌سازی RNA ها در نمونه، علاوه بر اطلاعات حیاتی در مورد غلظت اسیدهای نوکلئیک، کیفیت آنها را نیز نشان می‌دهد. RNA های تجزیه شده به صورت اسمیرهایی[۵] با وزن مولکولی پایین ظاهر می‌شوند. ولی RNA کُل سالم اوج‌های[۶] تیزی[۷] در ۲۸S و ۱۸S نشان می‌دهد. این سامانه‌ی بایوآنالایزر حاوی میکروچیپی است که با کنترل‌های اندازه بارگیری[۸] شده و فضای لازم برای ۱۲ نمونه به صورت همزمان را دارد. نمونه‌ها با یک پُلیمر و یک رنگ فلورسنت مخلوط شده و سپس از طریق حرکت مویین الکتروفورزی بارگیری گردیده و مورد سنجش واقع می‌شوند. همزمان با این کار، آنالیز داده‌ها روی دستگاه انجام شده و داده‌های الکتروفورز به صورت یک تصویر شبیه ژل ارائه می‌گردد تا از این طریق کار کردن با آنها برای کاربرانی که به الکتروفورز روی ژل معمولی عادت دارند، ساده‌تر گردد. نمونه‌ای از یک اجرای بایوآنالایزر در نگاره‌ی ۱-۲ ارائه شده است.

۱-۴ آماده‌سازی کتابخانه

پیش از توالی‌یابی، RNA های موجود در یک نمونه به یک کتابخانه‌ی cDNA که نماینده‌ی

1- Agilent Bioanalyzer
2- Microfluidics
3- Capillary Electrophoresis
4- Sizing
5- Smear
6- Peak
7- Sharp
8- Load



نگاره‌ی ۱-۲: خروجی اجیلنت بایوآنالایزر برای سنجش کیفیت RNA. هم خروجی اجرای لَدِر (خط‌کش ژنی استاندارد) و هم خروجی اجرای نمونه نمایش داده شده است.

کلیه‌ی مولکول‌های RNA موجود در نمونه است، تبدیل می‌شوند. دلیل اجرای این مرحله آن است که در عمل مولکول‌های RNA مستقیماً توالی‌یابی نشده و به جای آنها DNA ها توالی‌یابی می‌شوند. زیرا مولکول DNA پایداری شیمیایی بهتری داشته و نیز تبعیت بیشتری از روش شیمیایی توالی‌یابی و دستورالعمل‌های موجود در هر کدام از پلتفرم‌های توالی‌یابی دارد. بدین ترتیب، در آماده‌سازی کتابخانه دو هدف دنبال می‌شود. نخست اینکه کتابخانه آیینه‌ی تمام نمای RNA های موجود در نمونه بوده و دوم اینکه RNA به DNA تبدیل شود. هر پلتفرم توالی‌یابی RNA (نظیر اِلومِنا[1]، سولاید[2]، آیون تورنت[3]) دستورالعمل اختصاصی خود را داشته و بنابراین نیازی به تهیه‌ی دستورالعمل جداگانه برای هر کدام از آنها نیست. دستورالعمل‌های کتابخانه برای هر پلتفرم تجاری همراه باکیت‌های مربوط به آنها در وب‌سایت شرکت‌های ارائه دهنده‌ی خدمات مزبور، در دسترس هستند (جدول ۱-۱).

کیت‌های تجاری برای آماده‌سازی کتابخانه نیز در دسترس بوده و با موفقیت استفاده می‌شوند. همچنین می‌توانید با استفاده از موادی که در آزمایشگاه‌های زیست‌شناسی مولکولی به طور معمول به کار برده می‌شوند، برای خود کیت شخصی تولید کنید. البته چنین کیت‌هایی از سهولت کاربرد، بهینه‌سازی و پشتیبانی محصولات تجاری برخوردار نیستند. در اینجا مراحل دستورالعمل تهیه‌ی

1- Illumina
2- SOLID
3- Ion Torrent

جدول ۱-۱: پلتفرم‌های اصلی توالی‌یابی RNA و ویژگی‌های عمومی آنها

پلتفرم	روش شیمیایی توالی‌یابی	اصول شیمیایی تشخیص	لینک وبسایت
اِلومِنا	توالی‌یابی سنتزی	فلورسنت	www.illumina.com
سولاید	توالی‌یابی پایان‌دهی	فلورسنت	www.invitrogen.com
رُش ۴۵۴	توالی‌یابی سنتزی	لومینسانس	www.454.com
آیون تورنت	توالی‌یابی سنتزی	آزادسازی پروتون	www.iontorrent.com
پاسیفیک بایوساینسز	توالی‌یابی سنتزی تک‌مولکولی	فلورسنت به‌هنگام (لحظه‌ای)	www.pacifcbiosciences.com
آکسفورد نانوپور	توالی‌یابی سنتزی تک‌مولکولی	تفاوت جریان الکتریکی با عبور هر نوکلئوتید از یک منفذ	www.nanoporetech.com

کتابخانه برای توالی‌یابی RNA در پلتفرم اِلومِنا ارائه می‌گردد. مراحل کار در نگاره‌ی ۱-۳ نیز نشان داده شده است.

مراحل اصلی تهیه‌ی کتابخانه عبارت است از:

۱- حدود ۱ تا ۱۰ میکروگرم RNA کُل خالص و سالم که کیفیت آن نیز بررسی شده است برداشته می‌شود. حجم دقیق مورد نیاز به کاربرد آن و پلتفرم مورد استفاده بستگی دارد.

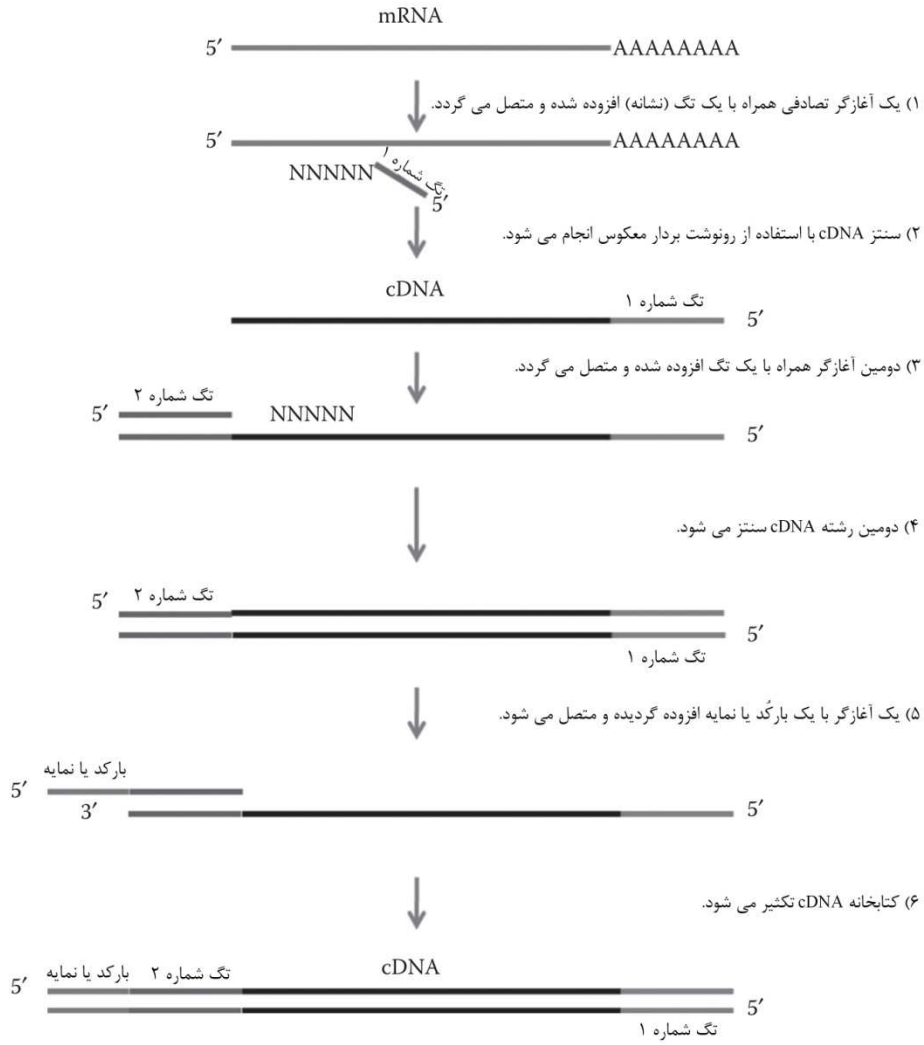
۲- از RNA کُل، mRNA تخلیص می‌گردد. به طور معمول برای این کار RNA کُل به مُهره‌های مغناطیسی اُلیگو-dT متصل می‌شود. برای اطمینان از رفع شدن RNA های ریبوزمی غیراختصاصی و سایر RNA ها از اُلیگو-dT، این مرحله از تخلیص دوبار اجرا می‌گردد. در پایان mRNA ها از مُهره‌های اُلیگو-dT جدا می‌شوند.

۳- mRNA خالص با مواد برش دهنده انکوباسیون می‌شود تا قطعه قطعه گردد. در پایان این مرحله، رشته‌های mRNA به چندین قطعه‌ی کوچک شکسته می‌شوند.

۴- آغازگرهای شش نوکلئوتیدی (هگزامر) تصادفی به ابتدای قطعات mRNA چسبانده می‌شوند.

۵- با کمک آنزیم رونوشت‌بردار معکوس[1] از روی قطعات mRNA رونویسی معکوس[2] صورت گرفته و در نتیجه cDNA ها تولید می‌شوند.

1- Reverse transcriptase
2- Reverse transcription



نگاره‌ی ۱-۳: مراحل تهیه‌ی کتابخانه‌ی توالی‌یابی RNA

۶- رشته‌ی دوم (مقابل) cDNA تولید گردیده و RNA از بین می‌رود. در پایان این مرحله، cDNA دو رشته‌ای (ds cDNA) حاصل می‌شود.

۷- ds cDNA از نوکلـئوتیدها، آنزیم‌ها، بافـرها و RNA تخلیص و پاکسازی می‌شود. به عـنوان

مثال می‌توان این کار را با کمک اتصال DNA به مُهره‌های تثبیت کننده برگشت‌پذیر فاز جامد[1] (SPRI) انجام داد. مزیت استفاده از این مُهره‌های پارامغناطیسی در این است که بعد از اتصال، مُهره‌ها را می‌توان شست تا ds cDNA که روی مُهره‌ها باقی می‌مانند تخلیص شوند. پس از شستشو می‌توان ds cDNA را برای مرحله‌ی بعدی از مُهره‌ها جدا نمود.

۸- روی ds cDNA جدا شده‌ی خالص، عملیات ترمیم انتها[2] صورت می‌گیرد.

۹- ds cDNA که انتهای آن ترمیم شده است، تخلیص می‌گردد. این کار می‌تواند روی مُهره‌های SPRI انجام شود.

۱۰- انتهای '3 در ds cDNA هایی که انتهای آنها ترمیم شده است، آدنیله[3] می‌شوند.

۱۱- آداپتورها[4] (سازگارسازها) به ds cDNA که انتهای آن ترمیم شده است، متصل[5] می‌گردد. آداپتورها به هر دو انتهای ds cDNA متصل می‌شوند. این آداپتورها می‌توانند برای هر واکنشی در کتابخانه نمایه[6] شوند. به عبارت دیگر هر آداپتوری می‌تواند شش نوکلئوتید مختلف در توالی خود داشته باشد. استفاده از یک نمایه‌ی متفاوت برای هر واکنش در کتابخانه اجازه می‌دهد که بعداً کتابخانه‌ها برای توالی‌یابی تلفیق شده ولی همچنان این امکان وجود دارد که با ردیابی توالی آداپتورها به کتابخانه‌ی اولیه‌ی خود منسوب شوند.

۱۲- ds cDNA هایی که انتهای‌شان تعمیر شده و آداپتور به آنها متصل گردیده است، تخلیص می‌گردند. این کار نیز می‌تواند با استفاده از مُهره‌های SPRI انجام شود.

۱۳- کتابخانه با استفاده از واکنش زنجیره‌ای پلیمراز[7] (PCR) غنی می‌گردد. با کمک توالی آداپتورها به عنوان آغازگر، تکثیر توالی‌های موجود با استفاده از تعداد محدودی چرخه (۱۲ تا ۱۶ چرخه) انجام می‌شود.

۱۴- ds cDNA هایی که انتهای‌شان ترمیم شده، آداپتور به آنها متصل گردیده و توسط PCR غنی شده‌اند، تخلیص می‌گردند. این مرحله نیز می‌تواند با استفاده از مُهره‌های SPRI صورت گیرد. در پایان این مرحله، محتویات کتابخانه نشان دهنده mRNA های اولیه‌ی موجود در نمونه خواهند بود.

1- Solid-Phase Reversible Immobilization (SPRI) bead
2- End-repair
3- Adenylate
4- Adaptor
5- Ligate
6- Index
7- Polymerase Chain Reaction (PCR)

۱۵- فرآیند اعتبارسنجی[1] و کنترل کیفیت[2] کتابخانه انجام می‌شود. این کار می‌تواند به چند طریق انجام شود: ۱) تکثیر انتخابی از طریق ژن‌های اختصاصی PCR[3] که بایستی در کتابخانه حضور داشته باشند، ۲) کمی‌سازی[4] محصول ds cDNA موجود در کتابخانه، ۳) آشکارسازی[5] توزیع فراوانی و اندازه‌ی کتابخانه با استفاده از الکتروفورز روی ژل پُلی‌اکریل‌آمید یا مویین الکتروفورز روی اجیلنت بایوآنالایزر.

۱۶- کتابخانه‌ها نرمال[6] گردیده و تلفیق[7] می‌شوند. چون ظرفیت توالی‌یابی در یک سلول جریان خیلی بالاست، لذا می‌توان چندین کتابخانه را توالی‌یابی نمود (تا ۲۴ کتابخانه به ازای هر چاهک سلول جریان امکان‌پذیر است. ولی معمولاً ۶ تا ۱۲ کتابخانه در هر چاهک در نظر گرفته می‌شود). نرمال‌سازی با هدف به دست آوردن مقدار ds cDNA موجود در هر کتابخانه صورت می‌گیرد. به عنوان مثال، همه‌ی کتابخانه‌ها می‌توانند تا ۱۰ نانو مولار ds cDNA رقیق شده و سپس در حجم‌های مساوی تلفیق شوند. بدین ترتیب همه‌ی کتابخانه‌ها مساوی خواهند بود.

۱۷- کتابخانه‌هایی که نرمال و تلفیق شده‌اند، برای ایجاد خوشه و دستورالعمل توالی‌یابی که بستگی به پلتفرم اختصاصی دارد (نظیر: اِلومِنا، سولاید، ۴۵۴ و . . .) به دستگاه توالی‌یابی ارسال می‌شوند.

۱-۵ پلتفرم‌های اصلی توالی‌یابی RNA

۱-۵-۱ اِلومِنا

این پلتفرم یکی از متداول‌ترین پلتفرم‌های توالی‌یابی مبتنی بر سنتز شیمیایی در یک چیدمان وسیع شیمیایی است. پس از ساخت کتابخانه، ds cDNA از یک سلول جریان[8] گذر کرده و با مولکول‌هایی بر مبنای مکمل بودن با توالی‌های آداپتور هیبرید می‌گردد. توالی‌های هیبرید شده که توسط سلول جریان در هر دو انتهای آداپتور نگه داشته می‌شوند، به صورت یک پُل تکثیر می‌گردند. این توالی‌های تازه تولید شده، به محدوده سلول جریان هیبرید شده و بعد از اجرای

1- Validate
2- Quality control
3- PCR-specific gene
4- Quantifying
5- Visualizing
6- Normalize
7- Pool
8 -Flow cell

چندین چرخه، یک ناحیه از سلول جریان حاوی چندین کُپی از ds cDNA اصلی خواهد بود. کُل این فرآیند، تشکیل خوشه[1] نامیده می‌شود. پس از تشکیل خوشه‌ها و حذف یک زنجیره از ds cDNA، ترکیبات لازم برای توالی‌یابی مبتنی بر سنتز[2] از مسیر سلول جریان عبور داده می‌شود. توالی‌یابی مبتنی بر سنتز واکنشی است که در آن و در طی هر دور سنتز یک نوکلئوتید (A، C، G یا T) افزوده شده و همزمان یک سیگنال فلورسنت تشخیص داده شده و تصویربرداری گردیده و بدین ترتیب مکان و نوع نوکلئوتید مزبور مشخص شده، ذخیره گردیده و آنالیز می‌شود. بازسازی توالی این افزونه‌ها در یک نقطه‌ی معین از سلول جریان که متناظر با یک خوشه‌ی ds cDNA تشکیل شده است، توالی دقیق نوکلئوتیدی برای یک قطعه‌ی اصلی از ds cDNA را فراهم می‌آورد.

به عنوان مثال، تعداد دورهای سنتز می‌تواند کمتر از ۵۰ نوکلئوتید (nt) تا ۱۵۰ نوکلئوتید باشد. همچنین دو وضعیت[3] برای توالی‌یابی متصور است. اگر توالی‌یابی تنها در یک انتهای ds cDNA انجام گیرد، خوانش تکی[4] نامیده شده ولی اگر توالی‌یابی در هر دو انتها صورت گیرد، وضعیت خوانش جفت انتهایی[5] (هر دو انتها) نامیده می‌شود. این دو نوع خوانش و طول آنرا در شکل خلاصه به صورت SR50 و PE100 نشان می‌دهند که به ترتیب نشان دهنده‌ی خوانش تکی ۵۰ نوکلئوتیدی و خوانش جفت انتهایی ۱۰۰ نوکلئوتیدی است. چون هر چرخه نیازمند شستشوی ترکیبات مورد استفاده و وارد کردن ترکیبات جدید است، لذا یک اجرای توالی‌یابی روی دستگاه می‌تواند بسته به مدل دستگاه و طول توالی، بین ۳ تا ۱۲ روز طول بکشد. اِلومِنا طیف وسیعی از دستگاه‌ها با بروندادهای مختلف را عرضه می‌کند. دستگاه Hi-Seq 2500 تا ۶ میلیارد خوانش جفت انتهایی در یک اجرا تولید می‌نماید. در مقیاس PE100 این عدد معادل ۶۰۰ گیگابایت داده است. این حجم از داده‌ی توالی‌یابی بسیار بزرگ‌تر از داده‌ی مورد نیاز برای یک مطالعه است و به همین دلیل این کتابخانه‌ها نمایه‌گذاری شده و چندین کتابخانه نرمال‌سازی و تلفیق گردیده و روی یک سلول جریان اجرا می‌شوند. معمولاً در مجموع ۱۰۰ کتابخانه روی یک سلول جریان ۱۶ خانه‌ای اجرا می‌شود. چون این ظرفیت توالی‌یابی برای یک آزمایشگاه خیلی زیاد است، لذا اِلومِنا یک دستگاه توالی‌یابی کوچک‌تر با برونداد کمتر نیز عرضه کرده است. سامانه‌ی MiSeq می‌تواند ۳ مگا خوانش تولید کند که در مقیاس PE100 معادل ۸/۵ گیگابایت داده در یک اجرای دو روزه است.

1- Cluster generation
2- Sequencing by synthesis
3- Mode
4- Single read mode
5- Paired-end read mode

۱-۵-۲ سولاید

سولاید برای توالی‌یابی از طریق اتصال[1] اُلیگونوکلئوتیدها و تشخیص آنها به کار گرفته شده و پلتفرمی است که توسط اپلاید بیوسیستمز[2] (کالزبد[3]، کالیفرنیا) تجاری‌سازی شده است. همان‌گونه که از نام این پلتفرم نیز برمی‌آید، اصول شیمیایی توالی‌یابی در این روش، بیش از آنکه از سنتز تبعیت کنید، مبتنی بر اتصال است. در پلتفرم سولاید، کتابخانه‌ای از قطعات DNA (که از مولکول‌های RNA منشا گرفته‌اند)، به مُهره‌های[4] مغناطیسی متصل می‌شوند (هر مولکول به یک مُهره). سپس DNA موجود روی هر مُهره در یک امولسیون تکثیر می‌گردد. این کار به گونه‌ای صورت می‌گیرد که محصولات تکثیر شده همراه با مُهره باقی می‌مانند. پس از آن محصولات تکثیر شده به صورت کووالانسی به یک اسلاید شیشه‌ای پیوند می‌یابند. با استفاده از چندین آغازگر[5] که به یک آغازگر عمومی[6] هیبرید[7] شده‌اند، کاوشگرهای[8] دوبازی دارای برچسب[9] فلورسنت به صورت رقابتی به آغازگر متصل می‌شوند. اگر بازهای موجود در نخستین و دومین موقعیت کاوشگر دوبازی مکمل توالی باشند، واکنش اتصال رُخ داده و برچسب نیز یک سیگنال ارسال می‌کند. آغازگرها توسط یک نوکلئوتید پنج بار بازنشانی[10] می‌شوند. بدین ترتیب در انتهای چرخه، به دلیل وجود کاوشگرهای دو نوکلئوتیدی، حداقل چهار نوکلئوتید دو بار و نوکلئوتید پنجم حداقل یک بار مورد بررسی قرار می‌گیرند. اتصال کاوشگرهای دو نوکلئوتیدی بعدی سبب می‌شود که نوکلئوتید پنجم برای بار دوم مورد بررسی قرار گرفته و بعد از بازنشانی‌های پرایمر پنجم، پنج نوکلئوتید بیشتر از حداقل دوبار مورد بررسی واقع خواهند شد. مراحل اتصال تا زمانی که توالی در حال خوانده شدن است، ادامه می‌یابد.

فرآیند شیمیایی منحصر به فرد اتصال سبب می‌شود که موقعیت یک نوکلئوتید دو بار کنترل گردیده و در نتیجه صحت توالی‌یابی به بیش از ۹۹/۹۹ درصد افزایش خواهد یافت. علی‌رغم اینکه چنین صحت بالایی برای مواردی همچون آنالیز افتراقی بیان[11] لازم نیست، ولی برای شناسایی

1- Ligation
2- Applied Biosystems
3- Carlsbad, CA
4- Bead
5- Primer
6- Universal
7- Hybrid
8- Probe
9- Label
10- Reset
11- Differential expression analysis

چندشکلی‌های تک نوکلئوتیدی[1] (SNP) ضروری است. جدیدترین دستگاه‌ها نظیر 5500 W این کار را بدون تکثیر روی مُهره انجام داده و از تراشه‌های جریان[2] برای تکثیر الگوها استفاده می‌کنند. برونداد چنین دستگاه‌هایی تا ۳۲۰ گیگابایت داده از دو تراشه‌ی جریان افزایش می‌یابد. همانند سایر پلتفرم‌ها، نمایه‌گذاری یا بارکُد زدن[3] نیز می‌تواند برای کتابخانه‌های چند عضوی[4] به کار گرفته شود. این کار سبب می‌شود که صدها نمونه‌ی کتابخانه‌ای را بتوان به طور همزمان روی دستگاه اجرا نمود.

۱-۵-۳ رُش ۴۵۴

این پلتفرم نیز بر مبنای توالی‌یابی با استفاده از آداپتور متصل شده به کتابخانه‌ی ds DNA و بر اساس اصول شیمیایی سنتز بنا نهاده شده است. ds DNA روی مُهره‌ها تثبیت شده و در یک امولسیون آب – روغن تکثیر می‌گردد. سپس مُهره‌ها در پلیت‌های پیکوتیتر[5] قرار داده شده و در آنجا واکنش‌های توالی‌یابی صورت می‌گیرد. تعداد بسیار زیاد چاهک‌ها در پلیت‌های پیکوتیتر امکان انجام تعداد بسیار زیاد واکنش موازی را که لازمه‌ی NGS است، فراهم می‌آورد. روش تشخیص در اینجا با سایر پلتفرم‌های مبتنی بر اصول شیمیایی سنتز که شامل شناسایی یک نوکلئوتید افزوده شده از طریق یک واکنش دو مرحله‌ای است، تفاوت دارد.

در نخستین مرحله، نوکلئوتید تری‌فسفات پس از افزوده شدن، شکسته شده و پیروفسفات آزاد می‌گردد. در مرحله‌ی دوم پیروفسفات با واسطه‌ی آنزیم ATP سولفوریلاز به آدنوزین تری‌فسفات (ATP) تبدیل می‌شود. در مرحله‌ی سوم ATP تازه سنتز شده، برای کاتالیزِ واکنشِ تبدیلِ لوسیفرین به اُکسی‌لوسیفرین به کار برده شده و این واکنش یک کوانتای نوری تولید می‌کند که با کمک یک دوربین مجهز به شارژ[6] از پلیت پیکوتیتر ثبت می‌گردد. پس از هر مرحله افزودن، نوکلئوتیدهای آزاد و ATP های واکنش نداده، با استفاده از آنزیم پیراز تجزیه می‌گردند. این مراحل تا دستیابی به یک تعداد از پیش تعیین شده‌ی واکنش‌ها، تکرار می‌گردند. ثبت نور تولید شده و چاهک مربوط به آن بعد از افزودن هر نوکلئوتید سبب می‌شود که بازسازی[7] مشخصه‌ی نوکلئوتیدی و توالی هر چاهک امکان‌پذیر گردد.

1- Single Nucleotide Polymorphism (SNP)
2- Flow chip
3- Barcoding
4- Multiplex
5- Picotiter plate
6- Chargecoupled camera
7- Reconstruction

این روش پیروسکوئنسینگ[1] نیز نامیده شده و مزیت فرآیند شیمیایی آن در این است که امکان خوانش‌های بلندتر را در مقایسه با سایر پلتفرم‌ها فراهم می‌آورد. در این پلتفرم می‌توان به طول‌های خوانش تا ۱۰۰۰ باز نیز دست یافت. کمپانی رُش[2] مالک این پلتفرم بوده و در حال حاضر سامانه‌ی GS FLX+ و سامانه‌ی کوچک‌تر GS junior را بر مبنای این پلتفرم عرضه می‌کند. با در نظر گرفتن یک میلیون خوانش در هر اجرا و متوسط ۷۰۰ نوکلئوتید به ازای هر خوانش، می‌توان به ۷۰۰ مگابایت داده‌ی توالی‌یابی در کمتر از یک روز اجرا دست یافت.

۱-۵-۴ آیون تورنت

این پلتفرمِ جدیدتر از کتابخانه‌ی متصل شده به آداپتور بهره برده و به دنبال آن از اصول شیمیایی توالی‌یابی مبتنی بر سنتر که در سایر پلتفرم‌ها نیز به کار گرفته می‌شوند، استفاده می‌نماید. ولی این پلتفرم یک ویژگی منحصر به فرد دارد. در این پلتفرم به جای تشخیص فوتون‌ها یا سیگنال‌های فلورسنت، تغییرات در pH محلول موجود در چاهک همزمان با افزودن نوکلئوتیدها و تولید پروتون، تشخیص داده شده و ثبت می‌گردد. این تغییرات بسیار اندک هستند. ولی تجهیزات آیون تورنت از تکنولوژی‌هایی بهره می‌برند که در صنایع نیمه‌رسانا برای دستیابی به آشکارسازهای با حساسیت کافی و سنجه‌های مناسب برای توالی‌یابی اسیدهای نوکلئیک، بسط و توسعه داده شده‌اند. یکی از محدودیت‌هایی که برای این پلتفرم ذکر شده است این است که ممکن است خوانش هموپلیمرها دشوار باشد. زیرا اگر نوکلئوتیدهای متوالی، یکسان باشند، هیچ راهی برای توقف افزودن تنها یک نوکلئوتید وجود ندارد. در چنین مواردی، آیون تورنت می‌تواند یک تغییر بزرگ‌تر را در pH تشخیص داده و این مقدار را برای خوانش مناطق پلیمری به کار گیرد.

به طور کلی در مقایسه با سایر پلتفرم‌ها، این پلتفرم خوانش‌های کمتری در یک اجرا تولید می‌کند. به عنوان مثال، ۶۰ تا ۸۰ مگا خوانش در ۲۰۰ باز به ازای هر خوانش روی دستگاه Proton در یک اجرا امکان‌پذیر بوده و ۱۰ گیگابایت داده تولید می‌کند. ولی در اینجا زمان اجرا تنها ۲ تا ۴ ساعت است. در حالی که همین اجرا روی پلتفرم‌های دیگر، ۱ تا ۲ هفته به طول می‌انجامد. چون در این پلتفرم نیازی به نوکلئوتیدهای تغییر یافته و تجهیزات سنجش نوری نیست، لذا هزینه‌ی خرید دستگاه و مواد شیمیایی مرتبط با آن پایین‌تر است. دستگاه مزبور جای کمی اشغال کرده، می‌توان آنرا در زمان‌هایی که مورد استفاده واقع نمی‌شود، خاموش نموده و در هنگام نیاز به آسانی روشن کرد و نیاز به نگهداری اندکی دارد. با توجه به کاربری آسان، اندازه و سرعت، این دستگاه

1- Pyrosequencing
2- Roche

کاربردهای وسیعی در توالی‌یابی میکروبی، ژنومیک محیطی و کاربردهای کلینیکی که زمان در آنها اهمیت حیاتی دارد، یافته است. همچنین این پلتفرم برای توالی‌یابی آمپلیکون‌ها[1] و کاربرد پنل‌های آغازگری در توالی‌یابی آنها با استفاده از رابط‌های کاربری خاص به طور وسیعی به کار گرفته شده است. هزینه‌ی پایین و اندازه‌ی کوچک این دستگاه نیز آنرا برای آزمایشگاه‌هایی که تمایل دارند دستگاه توالی‌یابی شخصی خودشان را داشته باشند، جذاب ساخته است.

۱-۵-۵ پاسیفیک بایوساینسز

این پلتفرم یک پلتفرم شاخص در نسل سوم است. اصول شیمیایی این پلتفرم مشابه پلتفرم‌های نسل دوم بوده و از سامانه‌ی توالی‌یابی مبتنی بر سنتز استفاده می‌کند. ولی یک تفاوت مهم در این است که این پلتفرم نیازمند تنها یک مولکول بوده و خوانش نوکلئوتیدهای افزوده شده به‌هنگام[2] انجام می‌شود. به همین دلیل اصول شیمیایی این روش را اصطلاحاً SMRT می‌نامند که خلاصه‌ی تک مولکول به‌هنگام (Single Molecule Real-Time) است. تک مولکول بدان معناست که هیچ تکثیری در این روش صورت نمی‌گیرد. باید توجه داشت که این پلتفرم، مولکول‌های DNA را توالی‌یابی می‌کند.

در دستگاه‌های پاسیفیک بایوساینسز[3] که از SMRT بهره می‌گیرند، از موجبَرهای حالت صفر[4] (ZMW) به عنوان فناوری پایه استفاده می‌شود. ZMW ها اتاقک‌های[5] با فضای محدود هستند که امکان هدایت انرژی نور و مواد در حجم‌های بسیار کوچک (در حد زِپتولیتر[6] (10^{-21} لیتر)) را فراهم می‌آورند. در قالب پلتفرم پاسیفیک بایوساینسز، این مجموعه یک اتاقک است که حاوی یک مولکول DNA پُلی‌مراز و یک مولکول DNA که به‌هنگام تعیین توالی خواهد شد، است. با استفاده از نوکلئوتیدهای تری‌فسفات خاص، افزوده شدن A ، C ، G یا T به زنجیره‌ی نوکلئوتیدی را می‌توان همزمان با سنتز شدن، تشخیص داد. مزیت بزرگ این پلتفرم، سرعت بسیار زیاد آن است. چون یک دستگاه به‌هنگام، افزوده شدن نوکلئوتیدها را همزمان با وقوع آن تشخیص می‌دهد، زمان اجرا می‌تواند در حد ۱ تا ۲ ساعت کاهش یابد. متوسط طول خوانش می‌تواند ۵۰۰۰ نوکلئوتید باشد. با اصلاحاتی که در آنزیم و فرآیند شیمیایی سنتز صورت گرفته است، خوانش‌های معمول به ۱۰۰۰۰ نوکلئوتید و طولانی‌ترین خوانش به ۳۰۰۰۰ نوکلئوتید ارتقا یافته است. در حال حاضر

1- Amplicon
2- Real-time
3- Pacific Biosciences
4- Zeromode waveguide (ZMW)
5 - Chamber
6- Zeptoliter

مدل PacBio RS II از این دستگاه‌ها در بازار عرضه شده است که می‌تواند در هر اجرا تا ۲۵۰ مگابایت توالی بدون کاهش بُروندا عرضه نماید.

یکی از مواردی که در توالی‌یابی مستقیم تک مولکول DNA ذکر گردیده این است که تغییرات اسید نوکلئیک نظیر ۵-متیل سیتوزین سبب بروز تاخیرهای پایدار و تجدیدپذیر در کینتیک آنزیم DNA پُلی‌مراز توالی‌یابی می‌گردد. از این موضوع در پلتفرم مزبور برای توالی‌یابی تغییرات DNA بهره‌برداری شده است. در حال حاضر گفته می‌شود که با این پلتفرم امکان تشخیص ۲۵ تغییر در بازها فراهم گردیده است.

۱-۵-۶ فناوری‌های نانوپور

علی‌رغم برونداد خوب و هزینه‌ی پایین به ازای هر باز در توالی‌یابی‌های کنونی، تلاش‌ها برای بهبود فناوری‌های توالی‌یابی ادامه دارد. هر چند که فناوری‌های نانوپور[1] در حال تکامل هستند ولی تاکنون تاثیر اندکی بر مطالعات توالی‌یابی RNA داشته‌اند. ولی تاثیر این فناوری‌ها در آینده بیشتر خواهد شد. توالی‌یابی نانوپور یک تکنیک تک مولکولی نسل سوم است که در آن یک آنزیم برای جداسازی یک زنجیره‌ی DNA و هدایت آن از میان یک منفذ پروتئینی واقع در یک غشا استفاده می‌شود. همزمان با آن، یون‌ها از این منفذ عبور کرده و یک جریان الکتریکی قابل اندازه‌گیری ایجاد می‌شود. این جریان به نوکلئوتیدهای خاصی که از طریق این منفذ عبور می‌کنند، حساس بوده و بدین ترتیب A ، C ، G یا T به نحو متفاوتی مانع از عبور جریان شده و سیگنالی تولید می‌کنند که در منفذ مزبور اندازه‌گیری می‌شود. مزیت این سامانه در این است که به نحو ساده‌ای منجر به کاهش اندازه‌ی پلتفرم و دستگاه شده (به عنوان مثال، ادعاهای اولیه حاکی از آن بود که این دستگاه به اندازه‌ی یک فلش USB خواهد بود) ولی به دلیل نیاز به اندازه‌گیری تغییرات بسیار اندک جریان در مقیاس یک مولکول، از نظر تکنیکی دچار چالش می‌شود. تلاش‌ها برای تجاری‌سازی این فناوری منجر به ابداع سامانه‌ی آکسفورد نانوپور[2] گردید. شرکت اِلومِنا نیز در حال بسط و گسترش توالی‌یابی نانوپور است. فناوری‌های آکسفورد نانوپور به گونه‌ای طراحی شده‌اند که DNA ، RNA یا پروتئین را مستقیماً و در حین عبور از منفذ ایجاد شده مورد سنجش قرار می‌دهند. علی‌رغم اینکه این فناوری به صورت وسیعی در سطح تجاری در دسترس نمی‌باشد، ولی بسیار نویدبخش ظاهر شده است.

1- Nanopore technologies
2- Oxford Nanopore

۱-۶ کاربردهای توالی‌یابی RNA

اهداف اصلی توالی‌یابی RNA عبارتند از: شناسایی توالی، ساختار و فراوانی مولکول‌های RNA در یک نمونه‌ی مشخص. منظور از توالی‌یابی، تعیین ترتیب نوکلئوتیدهای A ، C ، G و U است. تعیین ساختار به معنای شناسایی ساختار ژن (یعنی موقعیت پروموتور، اتصالات اینترون – اگزون، مناطق ناترجمان[1](UTR) در '3 و '5 و محل پُلی‌A) می‌باشد. ساختار ثانویه نشان دهنده مکان‌هایی است که جفت نوکلئوتیدها مکمل هم بوده و سبب ایجاد ساختارهای سنجاق‌سری یا برآمدگی می‌شوند. ساختار سوم نیز نشان دهنده‌ی شکل سه بُعدی مولکول است. تعیین فراوانی نیز بدان معناست که مقادیر کمی هر توالی به صورت مقادیر مطلق و نرمال‌سازی شده تعیین گردد. توالی حاصل می‌تواند برای شناسایی ژن‌های شناخته شده‌ی رمزگر[2] پروتئین، ژن‌های جدید یا RNA های نارمزگر بلند[3] مورد استفاده قرار گیرد. وقتی که تعیین توالی صورت گرفت، تاخوردگی مولکول برای تشکیل ساختارهای ثانویه می‌تواند نوع مولکول (نظیر tRNA یا miRNA) را مشخص نماید. فراوانی خوانش‌ها برای هر گونه از RNA را می‌توان بین نمونه‌های حاصل از مراحل مختلف تکامل، بخش‌های مختلف بدن یا گونه‌های نزدیک به هم مقایسه نمود. در زیر برخی از کاربردهای معمول توالی‌یابی RNA برای طیفی از سوالاتی که می‌توان مطرح کرد و به آنها پاسخ گفت، ارائه می‌گردد. در موارد مناسب، مثال‌هایی از مقالات علمی نیز ارائه می‌شود.

۱-۶-۱ شناسایی ساختار ژن‌های رمزگر پروتئین‌ها

روش‌های قدیمی‌تر مطالعه‌ی ترانسکریپتوم نظیر کلون کردن و توالی‌یابی کتابخانه‌های cDNA به روش سَنگِر، آنالیز بیان ریزآرایه و آنالیز سریالی بیان ژن (SAGE) همراه با پیش‌بینی محاسباتی در کنار توالی‌های ژنومی، ساختارهای ژنی را ارائه داده‌اند. این ساختارها در پایگاه‌های داده بایگانی شده و یک منبعِ با دسترسیِ آسان برای مقایسه‌ی داده‌های خام توالی‌یابی RNA با ژن‌های رمزگر پروتئین‌ها هستند. نخستین مرحله که اهمیت بالایی نیز دارد این است که ابتدا خوانش‌های توالی‌یابی RNA اغلب به ژن‌های شناخته شده‌ی رمزگر پروتئین مکان‌یابی می‌شوند. علاوه بر تعیین محدوده‌های اگزون – اینترون، داده‌های توالی‌یابی RNA می‌توانند شواهدی برای محدوده‌های کوتاه‌تر و بلندتر اگزون همراه با شناسایی اگزون‌های کاملاً جدید ارائه دهند. مجموعه‌ی اگزون‌ها و اینترون‌های سازنده‌ی یک ژن، مدل ژنی نامیده می‌شود. چون توالی‌یابی

1- Untranslated Region (UTR)
2- Coding
3- Long non-coding RNA

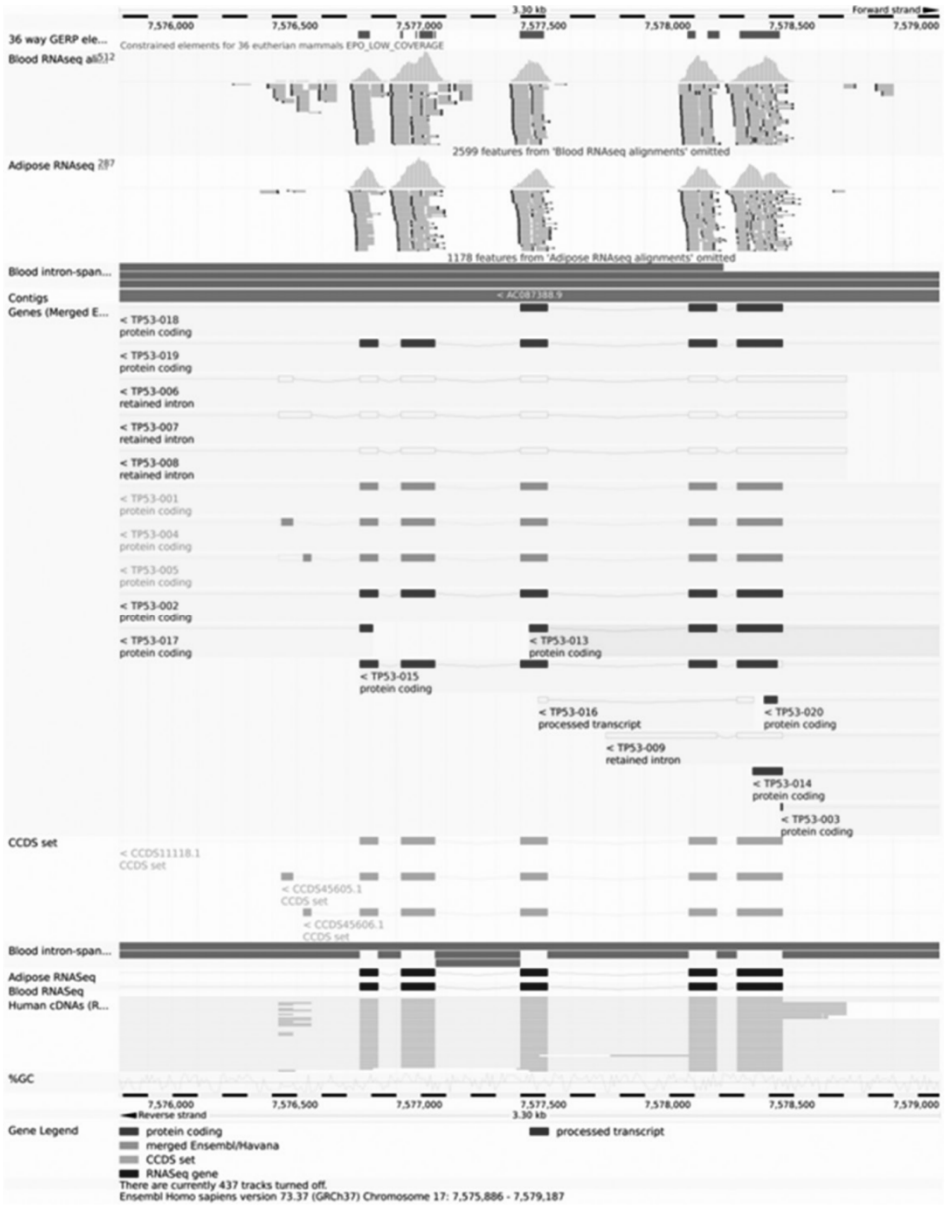
RNA یک روش کمی است، لذا می‌تواند کاربرد دسته‌ای از محدوده‌های اگزون یا اگزون جایگزین را نشان دهد (به عنوان مثال، وقتی که یک اگزون خاص پنج بار بیشتر از اگزون دیگر مورد استفاده قرار می‌گیرد). به طور مشابه، محل آغاز رونویسی[1] (TSS) در '5 می‌تواند به طور دقیق مکان‌یابی گردد. همچنین جایگزین TSS '5 نیز می‌تواند شناسایی شود. در انتهای '3 مولکول، می‌توان UTR '3 را دقیقاً شناسایی کرده و این کار به نحوی صورت می‌گیرد که مکان پُلی آدنیلاسیون در خوانش‌های توالی‌یابی RNA را بتوان مشاهده نمود. مکان‌های جایگزین پُلی آدنیلاسیون و فراوانی مربوط به آنرا نیز می‌توان با روشی مشابه TSS جایگزین مشاهده کرد. چون توالی‌یابی RNA به میزان بسیار زیادی موازی است، لذا خوانش‌ها این امکان را فراهم می‌آورند که این ساختارهای ژنی و جایگزین‌هایشان برای هر ژن احتمالی رمزگر پروتئین در یک ژنوم مکان‌یابی گردند. بدین ترتیب، توالی‌یابی RNA می‌تواند TSS '5 ، UTR '5 ، محدوده‌های اگزون – اینترون، UTR '3 ، مکان پُلی آدنیلاسیون و در صورت امکان استفاده‌ی جایگزین از هر کدام از موارد فوق را نشان دهد. یک مثال گرافیکی از یک ساختار ژنی و آنچه که توالی‌یابی RNA می‌تواند شناسایی کند، در نگاره‌ی ۱-۴ ارائه شده است.

۱-۶-۲ ژن‌های جدید رمزگر پروتیئن‌ها

حاشیه‌نگاری‌های اولیه از ژن‌های رمزگر پروتئین‌ها مبتنی بر پیش‌بینی‌های محاسباتی بر مبنای توالی‌های ژنومی بودند. این روش تا زمانی که داده‌های ژنومی در دسترس بودند، عناصر مدل ژنی با اندازه‌ی مورد انتظار معمول انطباق داشتند و داده‌های ترانسکریپتومی در قالب مجموعه‌ی داده‌های نشانه‌ی توالی‌های بیان شده[2] (EST) یا داده‌های اُرتولوژی برای تایید پیش‌بینی‌ها در دسترس قرار داشتند، مناسب بود. ولی با بررسی‌های علمی به راحتی دیده می‌شد که این معیارها تنها در تعداد بسیار معدودی از جانداران به خوبی انطباق دارند. بنابراین توالی‌یابی RNA به دلیل برونداد بالا می‌تواند بسیاری از پیش‌بینی‌های قبلی را تایید نماید. همچنین در جاهایی که هنوز هیچ پیش‌بینی وجود ندارد، قادر است ژن‌های جدید رمزگر پروتئین را شناسایی کند. این موضوع به ویژه در مواردی که هیچ توالی ژنومی در دسترس نبوده و ترانسکریپتوم یک جاندار به طور کامل از داده‌های توالی‌یابی RNA ایجاد می‌شود، می‌تواند مفید واقع گردد. یک نمونه‌ی جدیداز این کاربرد در توالی‌یابی تیلاپیای سیاه چینی (یک ماهی مهاجم دارای منشا آفریقایی با منابع ژنومی بسیار کمیاب) بوده است (۵). مثال دیگر در این زمینه، ترانسکریپتوم یولاف (*Avena sativa* L.)

1 -Transcription Start Site (TSS)

2- Expressed Sequence Tag (EST)



نگاره‌ی ۱-۴: تصویری از مدل ساختار ژنی برای ژن TP53 انسان از مرورگر ژنوم ENSEMBL . خوانش‌های توالی‌یابی RNA حاصل از خون و بافت چربی به عنوان پشتیبان این مدل نشان داده شده‌اند.

می‌باشد. علی‌رغم لذیذ بودن، ویژگی‌های تغذیه‌ای و اهمیت اقتصادی، این ژنوم آلوهگزاپلوییید برای مکان‌یابی ژنتیکی، توالی‌یابی و تعیین خصوصیات، بسیار چالش برانگیز بوده است. یک مطالعه‌ی جدید توالی‌یابی RNA توانست ۱۳۴ مگا خوانش جفت انتهایی ۱۰۰ نوکلئوتیدی تولید کرده و تعداد توالی‌های EST در دسترس را سه برابر نمود (۶).

۱-۶-۳ کمی‌سازی و مقایسه‌ی بیان ژن

وقتی که توالی و ساختار ژنی مشخص شده باشد، به طور منطقی می‌توان مقادیر فراوانی برای هر ژن و انواع ویژگی‌ها در ساختارشان را تعیین نمود. چون هدفِ تعداد زیادی از مطالعات مقایسه‌ی فراوانی رونوشت‌های RNA حاصل از افراد سالم با بیمار، درمان نشده با درمان شده یا نقطه‌ی زمانی صفر با یک است، لذا منطقی است که مطالعات مقایسه‌ای در این زمینه انجام شود. دامنه و نوع مطالعات مقایسه‌ای عملاً نامحدود است. بنابراین ارائه‌ی فهرستی از آنها در اینجا امکان‌پذیر نیست. در عوض برخی از مطالعات توالی‌یابی RNA ارائه می‌گردند تا خواننده با کاربرد این تکنیک و آنچه که از آن به دست می‌آید، آشنا شود.

در یکی از اولین مطالعات توالی‌یابی RNA ، رونوشت‌های حاصل از مغز کبد و ماهیچه‌ی اسکلتی در موش بالغ، توالی‌یابی و مقایسه گردید (۷). بیش از ۴۰ مگا خوانش تک انتهایی در ۲۵ نوکلئوتید روی یک پلتفرم اِلومِنا توالی‌یابی گردیده و محققین TSS ها، اگزون‌های جایگزین و 3' UTR های جدید کشف نمودند. این مطالعه نشان داد که حاشیه‌نگاری‌های قبلی از ساختار ژن عمیق نبوده و بدین ترتیب نشان داد که چگونه وسعت و عمق حاشیه‌نگاری‌های فراهم آمده توسط فناوری توالی‌یابی RNA می‌تواند دیدگاه موجود در مورد ساختار ژن را تغییر دهد. بنابراین این نتایج راه را برای مطالعات بعدی توالی‌یابی RNA هموار نمود.

دو سال بعد، بارلی از توالی‌یابی RNA برای مطالعه‌ی بیان رونوشت‌های RNA حاصل از سلول‌های C2C12 ماهیچه‌ی اسکلتی موش در طی تمایز بعد از ۶۰ ساعت، ۵ روز و ۷ روز استفاده کرد (۸). این فناوری به گونه‌ای بهبود یافت که بیش از ۴۳۰ مگا خوانش جفت انتهایی در ۷۵ نوکلئوتید برای شناسایی بیش از ۳۷۰۰ رونوشت که قبلاً حاشیه‌نگاری نشده بودند، به کار گرفته شد. همچنین TSS ها در بیش از ۳۰۰ ژن درگیر در تمایز تغییر کردند. این موضوع حاکی از آن است که وسعت اطلاعات رونوشت اضافی حاصل از توالی‌یابی RNA می‌تواند در یک سامانه‌ی کشت سلولی نسبتاً شناخته شده آشکار گردد.

مطالعه‌ی رونوشت‌های RNA در حیوانات کامل نیز امکان‌پذیر است. به عنوان مثال، *Caenorhabditis elegans* ، یک کرم لوله‌ای (نماتد) آزادزی بوده که در اتانول ۰/۲ مولار یا آب

مرحله‌ی جنینی تا رسیدن به مرحله‌ی لارو نهایی قبل از بلوغ رشد می‌کند. RNA کُل حاصل از حیوانات کامل جدا و تخلیص شده و توالی‌یابی گردید (۹). بیش از ۳۰ مگا خوانش از حیوانات رشد یافته در آب یا اتانول به دست آمد. قرار گرفتن در معرض اتانول منجر به افزایش رونوشت‌های RNA مربوط به ژن‌های آنزیم‌های سم‌زدایی و کاهش رونوشت‌های دخیل در تنش‌های شبکه‌ی اندوپلاسمی شد. مطالعات مشابهی نیز روی انواع جانداران مدل از طریق قرار گرفتن در معرض انواع سموم،از سرطان‌زاها (نظیر آفلاتوکسین) و بنزوپیرِن تا آلاینده‌های محیطی (نظیر متیل جیوه)، انجام گرفته است.

در یک مطالعه که اخیراً انجام شده است، میگوی آب شیرین (*Macrobrachium rosenbergii*) به عنوان یک جاندار مدل که کاربرد تجاری نیز دارد، با توالی‌یابی RNA مورد مطالعه قرار گرفت (۱۰). RNA غنی شده با پُلی A حاصل از RNA کُل مستخرج از کبد - لوزالمعده (هپاتوپانکراس)، آبشش و ماهیچه، ۸۶ مگا خوانش جفت انتهایی ۷۵ نوکلئوتیدی ایجاد نمود. چون ژنوم این جاندار قبلاً توالی‌یابی نشده بود، لذا این داده‌ها برای ایجاد یک ترانسکریپتوم مورد استفاده قرار گرفت که دارای بیش از ۱۰۲۰۰۰ UniGenes بوده و ۲۴ درصد آن می‌توانست توسط پایگاه‌های داده NCBI nr ، Swissprot ، KEGG و COG مکان‌یابی گردد.

۱-۶-۴ بیان جایگاه‌های ژنی صفات کمی (eQTL)

مطالعات توالی‌یابی RNA آنقدر فراگیر گردیده‌اند که برای مطالعه صفات کمی نیز به کار گرفته شده‌اند. به طور معمول، جایگاه‌های ژنی صفات کمی (QTL) در قالب مطالعات پیوستگی گستره‌ی ژنوم[1](GWAS) مورد مطالعه قرار می‌گیرند. در این مطالعات، پیوند SNP ها با یک صفت کمی نظیر طول قد، وزن، سطح کلسترول یا خطر ابتلا به دیابت نوع دوم مورد بررسی قرار می‌گیرد. eQTL تغییراتی در بیان ژن را نشان می‌دهد که توسط SNP ها کنترل می‌شود (۱۱). مبنای این همبستگی می‌تواند یک فعالیت منطقه‌ای باشد که اصطلاحاً eQTL سیس[2] نامیده می‌شود (به عنوان مثال یک SNP که روی یک ناحیه‌ی تقویت کننده واقع شده و می‌تواند بیان را تغییر دهد)، یا یک فعالیت از دور داشته باشد که اصطلاحاً eQTL ترانس[3] نامیده می‌شود (به عنوان مثال یک SNP که ساختار یک فاکتور رونویسی را تغییر داده و سبب می‌شود که فاکتور مزبور نتواند بیش از آن بر ژن هدف تاثیر بگذارد).

1- Genome-Wide Association Study (GWAS)
2 - cis-eQTL
3- trans-eQTL

بنابراین سطوح بیان ژن که توسط توالی‌یابی RNA تعیین می‌شوند، می‌توانند از طریق همبستگی‌شان با SNP ها به فنوتیپ پیوند داده شوند. همین ایده برای بررسی همبستگی بین مکان‌های پیرایش ژن و SNP ها بسط داده شد. این روش که sQTL نامیده می‌شود، حاکی از آن است که پیرایش نقشی مهم در تنظیم کلی بیان ژن ایفا می‌کند (۱۲). این روش، علاوه بر تحقیق در مورد بیماری‌های انسان، در زمینه‌هایی نظیر اصلاح نباتات که در آنها صفات کمی از اهمیت بالایی برخوردار هستند، نیز به کار گرفته می‌شود.

۱-۶-۵ توالی‌یابی RNA تک سلول

توالی‌یابی RNA تک سلول یکی از انواع توالی‌یابی‌های RNA است که در آن منبع RNA کُل برای توالی‌یابی تنها از یک سلول منشاء می‌گیرد. در این روش RNA کُل استخراج نشده ولی سلول‌ها به صورت منفرد از منبع‌شان برداشت شده و رونویسی معکوس می‌گردند. روش تهیه‌ی کتابخانه مشابه توالی‌یابی RNA است. RNA به cDNA رونویسی معکوس شده، آداپتور متصل گردیده، بارکُدها برای هر سلول افزوده شده و ds cDNA تکثیر می‌گردد. به دلیل پیچیدگی کمتر گونه‌های RNA ، گاهی اوقات پیش از توالی‌یابی، تک سلول‌های جدا شده یا کتابخانه‌های منفرد با هم ممزوج می‌شوند. به عنوان مثال، یک بلاستودرم موش جدا گردیده و از محتویات آن توالی‌یابی RNA صورت گرفت. محققین پی بردند که ۵۰۰۰ ژن بیان شده و بیش از ۱۷۰۰ اتصال پیرایشی جایگزین جدید شناسایی گردید. این موضوع هم نشان دهنده‌ی توانمندی این روش و هم پیچیدگی پیرایش در یک سلول است (۱۳). مثال دیگر در این مورد، تک سلول‌های برداشت شده در کرم لوله‌ای *C. elegans* در آغاز مرحله‌ی تکامل چند سلولی است که پس از جداسازی، کتابخانه‌هایی از RNA های کُل تهیه گردید. محققین توانستند از طریق تهیه‌ی پروفایل رونوشت سلول‌های منفرد، رونویسی جدید ژن‌ها را در هر کدام از مراحل تکامل رصد کنند (۱۴).

۱-۶-۶ ژن‌های تلفیقی

همگام با افزایش طول و تعداد خوانش‌ها و در دسترس قرار گرفتن توالی‌یابی جفت انتهایی، امکان شناسایی رونوشت‌های نادر ولی بالقوه مهم افزایش یافت. مثال بارز در این زمینه، ژن‌های تلفیقی[1] هستند که در آنها رونوشت‌ها از تلفیق دو ساختار ژنی که قبلاً جدای از هم بودند، ایجاد می‌شوند. اجزای تلفیق شده می‌توانند در 5'UTR ، نواحی رمزگر و سیگنال‌های3' پُلی‌آدنیلاسیون اشتراک داشته باشند. شرایط لازم برای وقوع چنین حالتی در طی بازآرایی ژنومی در بافت‌ها و

1- Fusion genes

سلول‌های سرطانی فراهم می‌آید. اختلالات سیتوژنتیکی نظیر تکثیرهای ژنومی، جابجایی و حذف می‌توانند دو ساختار ژنی مستقل را با یکدیگر تلفیق نمایند. به عنوان مثال ۲۴ ژن تلفیقی جدید و ۳ ژن تلفیقی شناخته شده از طریق توالی‌یابی جفت انتهایی کتابخانه‌های با طول ۱۰۰ تا ۲۰۰ نوکلئوتید، در ۳ خط سلولی سرطان سینه شناسایی شدند (۱۵). یکی از این ژن‌های تلفیقی VAPB-IKZF3 بوده که محققین پی بردند در آزمایشات رشد سلولی نقش دارد. مطالعات توالی‌یابی اخیر نشان می‌دهند که ژن‌های تلفیقی در بافت‌های معمولی نیز حضور دارند. این موضوع نشان می‌دهد که ژن‌های تلفیقی می‌توانند عملکرد زیستی معمول نیز داشته باشند.

۱-۶-۷ تنوع‌های ژنی

با افزایش حجم داده‌های توالی‌یابی RNA ، امکان داده‌کاوی برای تنوع ژنتیکی فراهم شده است. این حوزه بسیار فعال است. زیرا داده‌های حاصل از پروژه‌های با مقیاس بزرگ و مقالات منتشر شده اجازه می‌دهند که داده‌ها به صورت عمومی انتشار یافته و حتی نیاز دارند که چنین انتشاری صورت گیرد. اکثر مطالعات بیوانفورماتیکی از دانلود داده‌هایی که به صورت عمومی در دسترس هستند، بهره گرفته و از آنها برای پویش SNP ها در داده‌های ترانسکریپتومی استفاده کرده‌اند. در یک مطالعه، ۸۹ درصد از SNP ها حاصل از داده‌های توالی‌یابی RNA در پوشش[1] ۱۰x به عنوان واریانت‌های صحیح شناسایی شدند (۱۶). شناسایی SNP نیز می‌تواند مستقیماً از داده‌های توالی‌یابی RNA حاصل آید. یک گروه از محققین توالی‌یابی RNA را روی ماهیچه‌ی گاوهای لیموزین اجرا نمودند (۱۷). آنها توانستند بیش از ۸۰۰ SNP با کیفیت بالا را از بیش از ۳۰ مگا خوانش جفت انتهایی شناسایی کنند. یک زیرمجموعه از این SNP ها برای تعیین ژنوتیپ ۹ نژاد اصلی گاو در فرانسه استفاده شد که نشان دهنده‌ی یکی از کاربردهای این تکنیک نیز است.

یکی از کاربردهای NGS که به تازگی ارائه شده است، شناسایی تنوع در توالی‌های ژنی رمزگر پروتئین حاصل از نمونه‌های DNA ژنومی است. این تکنیک که اصطلاحاً توالی‌یابی اگزوم[2] یا شناسایی اگزوم نامیده می‌شود، توالی‌یابی RNA نیست. زیرا این روش مبتنی بر توالی‌یابی DNA ژنومی قطعه قطعه شده که از طریق هیبرید شدن با توالی‌های اگزونی برای اگزون‌ها غنی‌سازی شده‌اند، است. این کار از مطالعات بیماری‌های انسانی که در آنها لازم است تنوع‌ها (اغلب SNP ها) از گروه بزرگی از افراد شناسایی شوند، شروع شد. حتی امروزه نیز توالی‌یابی گروه‌های هزاران نفری از افراد، پرهزینه بوده و بنابراین تنها راه میانبر برای این نوع توالی‌یابی استفاده از توالی‌های اگرونی

1- Coverage
2- Exome sequencing

یک فرد است. چون اگزون‌ها عمدتاً در ژن‌های رمزگر پروتئین‌ها واقع شده‌اند، لذا از مزیت یافتن تنوع‌هایی که تاثیر مستقیم بر ساختار پروتئین داشته باشند، برخوردارند. این تکنیک یکی از عمومی‌ترین کاربردهای NGS بوده و در قالب کیت‌های تجاری در دسترس، در سطح وسیعی بسط و گسترش یافته است.

۱-۶-۸ RNA های نارمزگر بلند

کاربرد دیگر توالی‌یابی RNA ، یافتن رونوشت‌هایی است که حضور داشته ولی ژنی را رمز نمی‌کنند. RNA های نارمزگر بلند (lncRNA)، پیش از ابداع فناوری‌های توالی‌یابی RNA شناسایی شده بودند. ولی تا زمانی که روش‌های توالی‌یابی RNA گونه‌های بسیار متفاوت آنها را در سلول‌های زنده آشکار نکرده بودند، وسعت و فراگیری حضورشان کاملاً مشخص نشده بود. معمولاً lncRNA ها به عنوان رونوشت‌هایی شناخته می‌شوند که متفاوت از RNA های نارمزگر دیگر نظیر tRNA ها، RNA های ریبوزومی و RNA های کوچک بوده، با یک اگزون رمزگر پروتئین همپوشانی نداشته و بیش از ۱۰۰ نوکلئوتید طول دارند (۱۸). lncRNA ها می‌توانند به عنوان تقویت کننده[1] (eRNA) به شیوه‌ی اپی‌ژنتیکی و از طریق اتصال و جایگزینی عملکرد پروتئین‌های هیستون، به عنوان رقیب برای ماشین پردازش RNA (RNA درون‌زاد رقابتی[2] (ceRNA)) و یا به عنوان یک اختلال[3] که به صورت تصادفی تولید شده است، ترانسکریپتوم را کنترل نمایند. در حال حاضر مشخص شده است که lncRNA ها می‌توانند در بیماری‌هایی نظیر آلزایمر نقش ایفا نمایند (۱۹).

۱-۶-۹ RNA های نارمزگر کوچک (توالی‌یابی miRNA)

توالی‌یابی RNA می‌تواند برای شناسایی توالی، ساختار، عملکرد و فراوانی RNA های نارمزگر کوچک نیز به کار گرفته شود. شناخته شده‌ترین مثال از این دست، توالی‌یابی miRNA ها است. ولی سایر RNA های نارمزگر کوچک نظیر RNA های هسته‌ای کوچک[4] (snRNA)، RNA های تعدیل کننده‌ی ریز RNA ها[5] (moRNA) و RNA های سرکوبگر درون‌زاد[6] (endo-siRNA) نیز

1- Enhancer RNA (eRNA)
2- Competitive Endogenous RNA (ceRNA)
3 - Noise
4- Small nuclear RNAs (snRNA)
5- MicroRNA off-set RNAs (moRNA)
6- Endogenous silencing RNAs (endo-siRNAs)

می‌توانند با استفاده از روش‌های توالی‌یابی miRNA مورد مطالعه قرار گیرند. روش‌های توالی‌یابی miRNA مشابه روش‌های توالی‌یابی RNA است. ماده آغاز کننده می‌تواند RNA کُل یا RNA های کوچک بخش‌بندی شده یا انتخاب شده بر مبنای اندازه[1] باشد. اکثر پلتفرم‌های توالی‌یابی معمول RNA های کوچکی را که قبلاً به ds cDNA تبدیل شده‌اند، توالی‌یابی خواهند نمود. بنابراین بیشتر تفاوت‌ها در دستورالعمل‌های آزمایشگاهی قبل از توالی‌یابی است. این تفاوت‌ها به طور مفصل در فصل‌های بعدی مورد بررسی قرار می‌گیرند. تعیین مشحصات این مولکول‌ها کاربردهای فراوانی در بیوشیمی، فیزیولوژی، ژنتیک پایه و زیست‌شناسی تکامل و نیز در علم پزشکی به عنوان یک ابزار تشخیصی در سرطان یا فرآیندهای پیری دارد. در یک مطالعه که اخیراً روی کرم لوله‌ای *Panagrellus redivivus* صورت گرفته است، بیش از ۲۰۰ miRNA جدید و توالی‌های پیشروی سنجاق‌سری‌شان شناسایی شده و همراه با مدل‌های ساختار ژنی، حاشیه‌نگاری‌های ژن‌های رمزگر پروتئین و توالی‌های ژنی در قالب یک مقاله ارائه شده است (۲۰).

۱-۶-۱۰ توالی‌یابی محصولات تکثیر (توالی‌یابی Ampli)

گاهی اوقات نیازی به توالی‌یابی ترانسکریپتوم کامل نبوده و تنها تعداد اندکی از ژن‌ها مورد مطالعه واقع می‌شوند. البته همواره می‌توانید یک زیرمجموعه از ژن‌های مورد نظر را از یک آنالیز توالی ترانسکریپتوم کامل به دست آورید. ولی این روش نیازمند کار، زمان و منابع مالی زیادی بوده و الزامی هم برای آن وجود ندارد. با کمک یک پنل آغازگرهای PCR شامل ۱۰ تا ۲۰ جفت آغازگر، می‌توان PCR رونویسی معکوس (RT-PCR) را اجرا نموده و به جای کلون کردن هر محصول منفرد و جداسازی DNA پلازمید برای توالی‌یابی با روش سَنگِر، مجموعه‌ی محصولات PCR را توالی‌یابی نمود. این روش در جاهایی که تعداد نمونه زیاد و تعداد ژن‌ها اندک است، کاربرد دارد.

۱-۷ انتخاب پلتفرم توالی‌یابی RNA

حال که پلتفرم‌ها تشریح شده و برخی از کاربردهای معمول توالی‌یابی RNA توضیح داده شده‌اند، این سوال پیش می‌آید که کدام پلتفرم برای یک کاربری خاص بایستی انتخاب شود؟ یک راه حل ساده این است که یک مقاله‌ی مرجع با کاربری مشابه یا یکسان در PubMed پیدا شده و بر مبنای همان تجربه‌ی منتشر شده، انتخاب صورت گیرد. البته مراجعه به مقالات قبلی پیش از شروع یک مطالعه‌ی علمی برای بررسی نحوه‌ی برخورد آنها با مشکلات جاری، همواره توصیه

1- Size-selected/fractionated small RNA

می‌گردد. ولی یک نقطه ضعف در پیروی از تجربیات گذشته این است که به طور کلی توالی‌یابی NGS و به طور خاص توالی‌یابی RNA هم از نظر طراحی آزمایش‌ها و هم از نظر نحوه‌ی اجرا سریعاً در حال تغییر هستند. به دلیل همین تکامل سریع فناوری، شاید منصفانه‌تر باشد که گفته شود که هیچ پاسخ صحیح واحدی برای یک مشکل خاص وجود ندارد. علاوه بر این، اکثر پروژه‌های توالی‌یابی RNA چندین هدف را دنبال می‌کنند. به عنوان مثال ممکن است شما درصدد شناسایی رونوشت‌های جدید تلفیق ژن در یک نمونه، کمی‌سازی فراوانی ژن‌های شناخته شده‌ی کنونی و شناسایی SNP ها در ژن‌های شناخته شده باشید.

بنابراین منطقی‌تر این است که راهنماهایی بر مبنای اصول کلی طراحی مطالعات ارائه گردد تا از این طریق کاربران بتوانند هم پروژه را با اطمینان از خروجی‌های مورد انتظار طراحی کرده و هم علت برخی از انتخاب‌ها و تصمیمات اخذ شده را درک کنند. ممکن است که لازم باشد بین عمق پوشش و تعداد پلتفرم‌های به کار برده شده در یک مطالعه تعادل برقرار شود و چون آزمایشگاه‌ها منابع مالی محدودی دارند، در اغلب مواقع برقراری این تعادل و موازنه امری اجتناب‌ناپذیر به نظر می‌رسد.

۱-۷-۱ هشت قانون کُلی برای انتخاب پلتفرم توالی‌یابی RNA و وضعیت توالی‌یابی

۱-۷-۱-۱ صحت: توالی‌یابی بایستی چقدر صحت داشته باشد؟

اگر هدف شناسایی SNP ها یا وقایع ویرایشی تک نوکلئوتیدی در گونه‌های RNA است، بایستی پلتفرمی انتخاب شود که نرخ خطای کمتری داشته و در عمل قادر به تشخیص و تمییز بین SNP های اصلی و خطاهای توالی‌یابی باشد. فراوانی SNP در ژنوم انسان، تقریباً ۱ مورد در هر ۸۰۰ نوکلئوتید است. این فراوانی معادل با نرخ صحت ۹۹/۹ درصد می‌باشد. تنها پلتفرم سولاید ادعا می‌کند که نرخ صحتی فراتر از این سطح داشته و برخی از پلتفرم‌ها نیز پایین‌تر از این سطح هستند. با این حال باید این نکته را در ذهن داشت که پایین بودن صحت را می‌توان با خوانش‌های بیشتر جبران نمود. بدین ترتیب ۱۰ خوانش از یک قطعه‌ی یکسان RNA با صحت ۹۹/۹ درصد می‌تواند به طور موثری سبب ایجاد صحت ۹۹/۹ درصدی گردد.

اگر هدف شناسایی ژن‌های شناخته شده‌ی رمزگر پروتئین و بهبود حاشیه‌نگاری مدل ساختار ژنی همراه با کمی‌سازی رونوشت‌ها و احتمالاً کشف ژن‌های جدید است، صحت بسیار اندکی مورد نیاز خواهد بود. برنامه‌هایی برای مکان‌یابی خوانش‌ها برای مدل‌های ژنی شناخته شده وجود دارند

که یک یا حتی دو عدم تطابق[1] را نادیده می‌گیرند. اگر خوانش‌ها ۵۰ نوکلئوتیدی بوده و یک عدم تطابق نادیده گرفته شود، صحت ۹۸ درصد خواهد بود. در این سطح از صحت، اکثر پلتفرم‌های معمول نظیر سولاید، اِلومِنا، ۴۵۴ و آیون تورنت می‌توانند به کار گرفته شوند.

۱-۷-۱-۲ خوانش‌ها: چه مقدار خوانش مورد نیاز است؟

محاسبه‌ی آماره‌ی پوشش در مطالعات توالی‌یابی RNA امری مطلوب است. ژنوم انسان ۳۰۰۰ مگا نوکلئوتید دارد که تقریباً $\frac{1}{30}$ آن برای ژن‌های رمزگر پروتئین استفاده می‌شود. این بدان معناست که حدوداً ۱۰۰ مگا نوکلئوتید RNA برای توالی‌یابی وجود دارد. اگر از خوانش تکی ۱۰۰ نوکلئوتیدی (یا خوانش جفت انتهایی ۵۰ نوکلئوتیدی) استفاده شود، هر ۱ مگا خوانش، ۱۰۰ مگا نوکلئوتید داده‌ی توالی‌یابی ایجاد کرده که معادل پوشش 1x است. بنابراین ۳۰ مگا خوانش که خروجی خوانش حاصل از پلتفرم‌های معمول است، پوشش 30x ایجاد می‌نماید. در نتیجه با ۳۰ مگا خوانش می‌توان انتظار داشت که حجم بسیار زیادی از خوانش‌ها برای ژن‌های با بیان بالا و پوشش خوب برای اکثر ژن‌ها وجود داشته باشد و ممکن است که تعداد بسیار اندکی از ژن‌های با بیان پایین یا ژن‌هایی که به ندرت بیان می‌شوند، از دست بروند. برای محاسبه‌ی احتمال اینکه یک خوانش بتواند یک ژن خاص را مکان‌یابی کند، فرض می‌شود که متوسط اندازه‌ی یک ژن ۴۰۰۰ نوکلئوتید (۱۰۰ مگا نوکلئوتید تقسیم بر ۲۵۰۰۰ ژن) است. در ۳۰ مگا خوانش معادل با پوشش 30x، در خوانش تکی با طول ۱۰۰ نوکلئوتید (یا خوانش جفت انتهایی ۵۰ نوکلئوتیدی)، می‌توان انتظار داشت که به طور متوسط یک خوانش تکی برای مکان‌یابی ۱۲۰۰ بار بیان شود:

$$\frac{\text{طول ژن ۴۰۰۰ نوکلئوتیدی} \times \text{پوشش ۳۰}}{\text{۱۰۰ نوکلئوتید}} = ۱۲۰۰$$

بنابراین اگر ژن در سطح $\frac{1}{1200}$ در مقایسه با متوسط ژن بیان شود، احتمال اینکه یک خوانش آنرا مکان‌یابی کند، ۵۰ : ۵۰ خواهد بود. در عمل ۳۰ مگا خوانش برای اکثر کاربری‌ها مناسب است. البته احتمالاً همه‌ی ژن‌های بیان شده در یک نمونه را در بر نمی‌گیرد. چون اکثر پلتفرم‌ها می‌توانند تا ۳۰ مگا خوانش را تولید کنند، معمولاً محدودیتی از این نظر ایجاد نمی‌شود. در مواردی که پوشش بهتر مورد نیاز بوده و داده‌ها برای استفاده‌ی اگزون جایگزین و جزییات سایر مدل‌های ژنی یا وقایع نادر مورد استفاده واقع می‌گردند، پلتفرم‌هایی که می‌توانند به آسانی خوانش‌های بیشتری را تولید کنند، ترجیح داده می‌شوند. از روشی که اخیراً ابداع شده و توالی‌یابی تسخیری[2] نامیده می‌شود، می‌توان برای غنی‌سازی RNA ها در تعداد کمتری از جایگاه‌های ژنوم انسان

1- Mismatch
2- Capture-seq

استفاده نمود. این روش از یک ریزآرایه‌ی نیمبلِگِن[1] چاپ شده برای تسخیر RNA های حاصل از تعداد محدودی از جایگاه‌ها استفاده می‌کند (۲۱). در این مثال، محققین به طور متوسط ۵۰ جایگاه ژنی شامل ژن‌های رمزگر پروتئین‌ها و RNA های نارمزگر بلند را شناسایی کردند. با این استراتژی آنها توانستند به طور موثر بیش از ۴۶۰۰ برابر پوشش برای جایگاه‌های ژنی به دست آورده و اگزون‌ها و الگوهای پیرایشی حاشیه‌نگاری شده را حتی برای ژن‌هایی که به خوبی مطالعه شده‌اند، کشف نمایند. به طور ساده می‌توان نتیجه گرفت که ممکن است هیچگاه پوشش کافی برای به دست آوردن رونوشت تکی از یک جایگاه ژنی نداشته باشید.

راه دیگر برای نگرش به این مساله این است که بررسی شود که چه مقدار خوانش برای اثبات وجود یک رونوشت کافی است. هیچ آمار و ارقامی برای این موضوع در دسترس نیست. برخی از مقالات معتقدند که یک خوانش نیز برای اثبات وجود یک مولکول کافی بوده و در مقابل هم برخی از مقالات ادعا می‌کنند که کمتر از ۱۰ خوانش برای اثبات این موضوع کافی نمی‌باشد. این موضع اکثراً به زمینه‌ی مطالعه، معیارهای مجله یا پایگاه داده و اهداف کُلی مطالعه بستگی دارد.

۱-۷-۱-۳ طول: طول خوانش‌ها باید چقدر باشد؟

برای مکان‌یابی خوانش‌ها در مدل‌های ژنی شناخته شده روی یک جاندار، ۱۴ نوکلئوتید نیز کافی است. با این حال چون ممکن است برخی از خوانش‌ها برای بیش از یک موقعیت مکان‌یابی گردند، خوانش‌های بلندتری مورد نیاز است. در ۵۰ نوکلئوتید، درصد پایینی از خوانش‌ها در بیش از یک موقعیت مکان‌یابی می‌گردند. ولی این تعداد معمولاً کم (کمتر از ۰/۰۱ درصد) و به گونه‌ای است که این طول خوانش به شما اجازه‌ی انجام مطالعات افتراقی بیان و تعریف مدل‌های ژنی بهتر را می‌دهد. ولی در بسیاری از موارد نظیر حاشیه‌نگاری ژن‌های جدید در گونه‌ای که داده‌های توالی‌یابی دیگری (نظیر داده‌های ژنومی، EST یا cDNA های بلند) از آن در دسترس نیست، نیاز به خوانش‌های بلندتر است. در اختیار داشتن توالی‌های بلندتر به جای تلاش برای پیش‌بینی مدل‌های ژنی بر مبنای مکان‌یابی خوانش‌های ناپیوسته‌ی ۵۰ نوکلئوتیدی، یک مزیت متمایز است. رُش ۴۵۴ مسیر خاصی برای این نوع کاربری‌ها ایجاد کرده است. پاسیفیک بایوساینسز، به ویژه در دستگاه‌ها و کیت‌های نسل جدیدتر معمولاً قادر به تولید خوانش‌های بلند تا حد ۱۰۰۰۰ نوکلئوتید یا بیشتر است.

۱-۷-۱-۴ SR یا PE: خوانش تکی یا جفت انتهایی؟

اگر هیچ‌گونه اُریبی در هیچ‌یک از مراحل تهیه‌ی کتابخانه (قطعه‌بندی RNA ، اتصال به

1- Nimblegen

آداپتورها و جهت‌گیری رشته‌ها) وجود نداشته باشد و سنتز cDNA به طور کامل قطعات تصادفی که نمونه‌ی RNA را نمایندگی کنند، تولید نماید، آنگاه می‌توان اطلاعات توالی مشابهی از SR و PE به دست آورد. با این حال اُریبی‌هایی در مراحل تهیه‌ی کتابخانه وجود دارد. یک راه برای افزایش تصادفی کردن قطعات جهت تعیین توالی عبارت از توالی‌یابی هر دو انتهای یک کلون کتابخانه است. این کار هدف دوگانه‌ای داشته که در آن توالی‌های PE حاصل از قطعات کوتاه به شرط تایید اضافی یک توالی، می‌توانند همپوشانی کنند. در حال حاضر اکثر برنامه‌های آنالیز داده‌ها قادر به کار کردن با هر دو دسته داده‌ی SR و PR هستند. بنابراین مانعی در آنالیزهای پایین‌دستی[1] ایجاد نمی‌شود. متاسفانه همه‌ی پلتفرم‌ها اجازه‌ی توالی‌یابی در هر دو انتها را نداده و بنابراین ضرورتاً و در صورت امکان، بهتر است که از توالی‌یابی جفت انتهایی استفاده شود.

۱-۷-۱-۵ RNA یا DNA: RNA توالی‌یابی گردد یا DNA؟

همان‌گونه که قبلاً نیز ذکر شد، اکثر پلتفرم‌ها cDNA دو رشته‌ای حاصل از رونویسی معکوس و تکثیر PCR مولکول‌های RNA در یک نمونه را توالی‌یابی می‌کنند. نمونه‌هایی از توالی‌یابی RNA وجود دارد که در آن RNA ترجیحاً به صورت مستقیم توالی‌یابی می‌شود (به عنوان مثال، در پروژه‌هایی که تغییر در ساختارهای RNA نظیر کلاهک‌گذاری mRNA اهمیت دارد).

۱-۷-۱-۶ ماده: چه مقدار نمونه مورد نیاز است؟

در حال حاضر، توالی‌یابی RNA کُل حاصل از یک سلول منفرد نیز امکان‌پذیر است. به همین دلیل ممکن است این سوال پیش آید که آیا حد پایین‌تری برای نمونه‌ی مورد نیاز وجود دارد؟ پلتفرم‌های توالی‌یابی که از cDNA دو رشته‌ای تکثیر شده استفاده می‌کنند، الزاماً از مقدار نمونه‌ی کمتری بهره نخواهند گرفت. البته این بدان معنا نیست که بایستی پلتفرم توالی‌یابی را با حداقل مقدار نمونه فراهم آورید. افزایش مقدار نمونه بایستی حضور گونه‌های مختلف RNA را در نمونه افزایش دهد. اکثر پلتفرم‌های توالی‌یابی مبتنی بر سنتز، کیت‌های تخصصی برای ساخت کتابخانه از مقادیر نانوگرمی RNA کُل دارند. پلتفرم‌های تک مولکولی تنها نیازمند یک مولکول برای توالی‌یابی هستند. بنابراین این موضوع محدودیتی برای پلتفرم‌های مختلف ایجاد نمی‌کند.

۱-۷-۱-۷ هزینه‌ها: چه مقدار می‌توان هزینه نمود؟

چون هزینه‌های توالی‌یابی در طی ۱۰ سال گذشته به شدت کاهش یافته است، لذا هزینه نباید

1- Downstream analysis

چندان اهمیت داشته باشد. ولی واقعیت این است که الزامات انتشار مقاله و استانداردهای کیفی همچنان بالا بوده و در نتیجه همواره مشکل هزینه‌ها وجود خواهد داشت. ارسال کتابخانه‌های توالی‌یابی RNA به آزمایشگاه‌های NGS تجاری، ملی یا محلی یک روش خوب برای کاهش هزینه‌ها است. اگر از پشتیبانی مالی خوبی برخوردار هستید، در حال حاضر خرید یک دستگاه توالی‌یابی آزمایشگاهی شخصی امکان‌پذیر است. در حقیقت اِلومِنا با تولید MiSeq و آیون تورنت با تولید Personal Genome Machine و Ion Proton دستگاه‌های توالی‌یابی آزمایشگاهی شخصی عرضه کرده‌اند که حتی برای آزمایشگاه‌هایی که از پشتیبانی مالی خوبی نیز برخوردار نباشند، قابل تهیه است. هنوز پایین‌ترین قیمت‌ها در این بازار به دست نیامده است. بنابراین شما می‌توانید برای انتخاب‌های بیشتر در پلتفرم‌های توالی‌یابی که هنوز از نظر تجاری چندان جذاب نیستند، جستجو نمایید. در واقع حجم بالای درخواست‌ها برای دریافت نمونه توسط مراکز تجاری و غیرانتفاعی حاکی از آن است که فشار قیمت‌گذاری همچنان رو به پایین است.

۱-۷-۱-۸ زمان: چقدر زمان لازم است تا کار تکمیل شود؟

یک ضرب‌المثل می‌گوید: «کاری که ضروری بوده باید دیروز انجام می‌شده است». ژنومیک یک زمینه‌ی به سرعت در حال رشد است. در حالت ایده‌آل، نمونه‌ها آماده شده، کتابخانه‌ها ساخته شده و توالی‌یابی بدون هرگونه درنگ و معطلی انجام می‌شود. در حقیقت، اکثر پلتفرم‌ها (اِلومِنا، سولاید، ۴۵۴) نه به دلیل در حال اجرا بودن دستگاه، بلکه به دلیل ناکافی بودن کتابخانه جهت پُر کردن یک سلول جریان برای راه‌اندازی و اجرای یک اجرای توالی‌یابی، دچار معطلی و درنگ می‌شوند. کافی است به این نکته توجه شود که در عمل، تعلل در کارها ممکن است ناشی از دستگاه نبوده بلکه به دلیل آماده‌سازی کار در ایجاد کتابخانه و جمع‌آوری تعداد کافی از کتابخانه‌ها برای شروع اجرای دستگاه باشد. در سوی دیگر ماجرا و پس از آماده شدن داده‌های توالی‌یابی، کار آنالیز داده‌ها شروع می‌شود. مرحله‌ی آنالیز داده‌ها ممکن است چند روز، چند ماه یا چند سال در پروژه‌های بزرگ طول بکشد. در این موارد، اگر زمان آنالیز با زمان مرحله‌ی توالی‌یابی مقایسه شود، مشخص می‌گردد که زمان اجرای دستگاه توالی‌یابی نسبتاً کوتاه بوده است.

۱-۸ خلاصه

به طور خلاصه، شما می‌توانید ملاحظه نماید که انتخاب‌های بسیار زیادی در اجرای آزمایشات توالی‌یابی RNA در اختیار دارید. هر پلتفرمی ویژگی‌های خاص خود را داشته که آنرا از سایر پلتفرم‌ها متمایز می‌کند. فهرست پلتفرم‌های اصلی توالی‌یابی RNA، اصول شیمیایی توالی‌یابی و

تشخیص و لینک وبسایت‌ها در جدول ۱-۱ ارائه شده است. اگر خوش شانس باشید، چند پلتفرم برای انتخاب کردن در اختیار خواهید داشت. در واقع برخی از مطالعات بهترین خصوصیات هر پلتفرم را ارائه کرده و در نتیجه می‌توان آنها را برای اهداف متفاوت مورد بررسی و قضاوت قرار داد. به عنوان مثال، خوانش‌های اِلومِنا می‌توانند برای پوشش، سولاید برای صحت و رُش ۴۵۴ یا پاسیفیک بایوساینسز می‌توانند برای طول مورد توجه قرار گیرند. به راحتی می‌توان آینده‌ای را تصور نمود که استفاده از چند پلتفرم برای یک پروژه‌ی خاص امکان‌پذیر باشد. عوامل دخیل در انتخاب یک پلتفرم، چند وجهی هستند. ولی تعیین مناسب‌ترین پلتفرم برای یک کاربری خاص امری ناممکن نیست. با به کارگیری اطلاعات موجود در این بخش و با آگاهی از ویژگی‌های به روز شده‌ی دستگاه‌ها و قیمت‌های فعلی می‌توان تصمیم آگاهانه‌ای برای انتخاب پلتفرم مناسب و وضعیت استفاده از آن برای آزمایشات توالی‌یابی RNA اتخاذ نمود.

منابع

1. Nagalakshmi U., Wang Z., Waern K. et al. Te transcriptional landscape ofthe yeast genome defned by RNA sequencing. Science 320(5881):1344–1349,2008.
2. Sultan M., Schulz M.H., Richard H. et al. A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. Science 321(5891):956–960, 2008.
3. Wilhelm B.T., Marguerat S., Watt S. et al. Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. Nature453(7199):1239–1243, 2008.
4. Wang Z., Gerstein M., and Snyder M. RNA-Seq: A revolutionary tool for transcriptomics. Nature Reviews in Genetics 10(1):57–63, 2009.
5. Avarre J.C., Dugué R., Alonso P. et al. Analysis of the black-chinned tilapia Sarotherodon melanotheron heudelotii reproducing under a wide range ofsalinities: From RNA-seq to candidate genes. Molecular Ecology Resources14(1):139–149, 2014.
6. Gutierrez-Gonzalez J.J., Tu Z.J., and Garvin D.F. Analysis and annotation of the hexaploid oat seed transcriptome. BMC Genomics 14:471, 2013.
7. Mortazavi A., Williams B.A., McCue K. et al. Mapping and quantifying mammalian transcriptomes by RNA-seq. Nature Methods 5(7):621–628, 2008.
8. Trapnell C., Williams B.A., Pertea G. et al. Transcript assembly and quantifcation by RNA-seq reveals unannotated transcripts and isoform switchingduring cell differentiation. Nature Biotechnology 28(5):511–515, 2010.
9. Peltonen J., Aarnio V., Heikkinen L. et al. Chronic ethanol exposure increases cytochrome P-450 and decreases activated in blocked unfolded protein response gene family transcripts in Caenorhabditis elegans. Journal of Biochemical Molecular Toxicology 27(3):219–228, 2013.
10. Mohd-Shamsudin M.I., Kang Y., Lili Z. et al. In-depth transcriptomic analysis on giant freshwater prawns. PLoS ONE 8(5):e60839, 2013.

11. Majewski J. and Pastinen T. Te study of eQTL variations by RNA-seq: From SNPs to phenotypes. Trends in Genetics 27(2):72–79, 2011.
12. Lalonde E., Ha K.C., Wang Z. et al. RNA sequencing reveals the role of splicing polymorphisms in regulating human gene expression. Genome Research21(4):545–554, 2011.
13. Tang F., Barbacioru C., Wang Y. et al. mRNA-seq whole-transcriptome analysis of a single cell. Nature Methods 6:377–382, 2009.
14. Hashimshony T., Wagner F., Sher N. et al. CEL-Seq: Single-cell RNA-seq bymultiplexed linear amplifcation. Cell Reports 2(3):666–673, 2012.
15. Edgren H., Murumagi A., Kangaspeska S. et al. Identifcation of fusion genes in breast cancer by paired-end RNA-sequencing. Genome Biology 12(1):R6,2011.
16. Quinn E.M., Cormican P., Kenny E.M. et al. Development of strategies for SNP detection in RNA-seq data: Application to lymphoblastoid cell lines and evaluation using 1000 Genomes data. PLoS ONE 8(3):e58815, 2013.
17. Djari A., Esquerré D., Weiss B. et al. Gene-based single nucleotide polymorphism discovery in bovine muscle using next-generation transcriptomicsequencing. BMC Genomics 14(1):307, 2013.
18. Ilott N.E. and Ponting C.P. Predicting long non-coding RNAs using RNAsequencing. Methods 63(1):50–59, 2013.
19. Faghihi M.A., Modarresi F., Khalil A.M. et al. Expression of a noncoding RNA is elevated in Alzheimer's disease and drives rapid feed-forward regulation of beta-secretase. Nature Medicine 14(7):723–730, 2008.
20. Srinivasan J., Dillman A.R., Macchietto M.G. et al. Te draf genome and transcriptome of Panagrellus redivivus are shaped by the harsh demands of a free-living lifestyle. Genetics 193(4):1279–1295, 2013.
21. Mercer T.R., Gerhardt D.J., Dinger M.E. et al. Targeted RNA sequencing reveals the deep complexity of the human transcriptome. NatureBiotechnology 30(1):99–104, 2011.

فصل دوم

مقدمه‌ای بر آنالیز داده‌های توالی‌یابی RNA

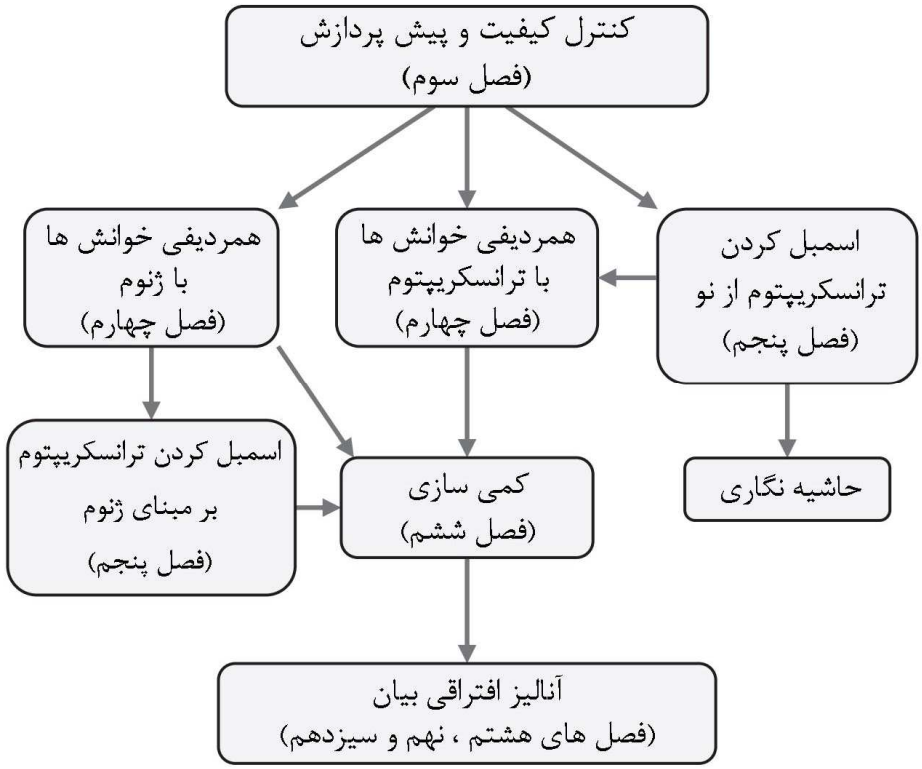
۲-۱ مقدمه

پس از اینکه میلیون‌ها خوانش از یک آزمایش توالی‌یابی RNA به دست آمد، آنالیز داده‌ها شروع می‌شود. همان‌گونه که در فصل اول تشریح گردید، توالی‌یابی RNA یک فناوری قدرتمند با کاربردهای متعدد (از جستجوی واریانت ژن و پیرایش تا آنالیز افتراقی بیان و شناسایی ژن‌های تلفیقی، واریانت‌ها و ویرایش RNA) است. به همین دلیل روش‌های متعدد برای آنالیز این داده‌ها نیز وجود داشته و نمی‌توان آنها را تنها در یک گردش کار واحد نشان داد. نگاره‌ی ۲-۱ مراحل اصلی اکثر آنالیزهای معمول را نشان می‌دهد. انتخاب راه‌های مختلف به وجود یا عدم وجود ژنوم یا ترانسکریپتوم مرجع بستگی دارد. مراحل آنالیز داده‌ها توسط برنامه‌های جداگانه انجام شده و نیازمند استفاده از فرمت‌های خاص برای داده‌ها و فایل‌های خارجی است. چون آنالیز داده‌های توالی‌یابی RNA یک زمینه‌ی فعال تحقیقاتی است که در آن سریعاً روش‌ها و ابزارهای جدیدی ابداع می‌شوند، لذا برنامه‌های جایگزین زیادی برای هر مرحله از آنالیز وجود دارد. باز نگه داشتن مسیر گزینه‌های موجود و انتخاب مناسب‌ترین برنامه می‌تواند یک چالش محسوب گردد. ولی خوشبختانه مقایسه‌های زیادی از ابزارهای کامل منتشر شده است و در فصل‌های بعدی به این مقالات ارزیابی ارجاع داده می‌شود.

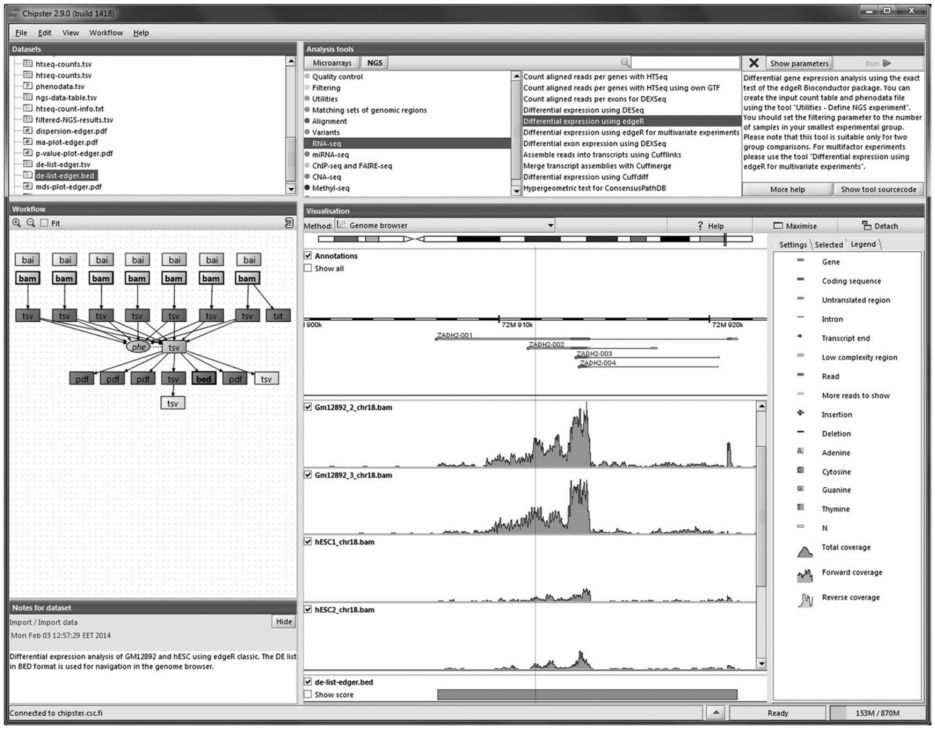
آنالیز داده‌های توالی‌یابی RNA چگونه شروع می‌شود؟ این موضوع به این بستگی دارد که چه نوع آنالیزی انجام شده و چه زمینه‌ای وجود دارد. اگر با کار کردن روی خط فرمان[1] و استفاده از یونیکس[2] و R راحت نیستید، می‌توانید از یک رابط کاربری گرافیکی جذاب نظیر Galaxy (galaxyproject.org/) (۱) یا Chipster (chipster.csc.f) (۲) استفاده نمایید. این رابط‌های کاربری ابزارهایی یکپارچه و انعطاف‌پذیر بوده که می‌توانند از خوانش‌های خام توالی‌یابی RNA تا نتایج آزمایش را همراهی کرده و از دیدگاه عملی بسیار مفید هستند. مثالی از رابط کاربری Chipster در نگاره‌ی ۲-۲ ارائه شده است.

با این حال، اکثر کاربران تمایل دارند که روی آنالیز داده‌هایشان کنترل کامل اعمال کرده، حداکثر انعطاف‌پذیری را برای تغییر پارامترها داشته، از همه‌ی گزینه‌ها استفاده نموده و ورود و

1- Command line
2- Unix



نگاره‌ی ۲-۱: روش‌های ممکن در آنالیز داده‌های توالی‌یابی RNA . در ابتدا، کیفیت خوانش‌ها کنترل شده و در صورت لزوم برای رفع داده‌های با کیفیت پایین و ورساخته‌ها، خوانش‌ها پیش پردازش می‌شوند. سپس در صورت دسترسی به ژنوم مرجع، منشا این خوانش‌ها از طریق همردیفی آنها با آن ژنوم مرجع شناسایی می‌شود. ژن‌ها و رونوشت‌های جدید با استفاده از اسمبل کردن ترانسکریپتوم مبتنی بر ژنوم شناسایی شده و بیان ژن و رونوشت کمی‌سازی می‌گردد. ممکن است که شناسایی ژن و رونوشت نادیده گرفته شده و بیان تنها برای ژن‌ها و رونوشت‌های شناخته شده کمی‌سازی گردد. اگر ژنوم مرجع در دسترس نباشد، به جای آن می‌توان با استفاده از یک ترانسکریپتوم مرجع، خوانش‌ها را همردیف و کمی‌سازی نمود. اگر یک ترانسکریپتوم در دسترس نباشد، می‌توان آنرا از خوانش‌ها و با استفاده از اسمبل کردن ترانسکریپتوم از نو تولید نمود. وقتی که فراوانی‌ها با استفاده از یکی از مسیرهای فوق برآورد شدند، تفاوت در بیان بین گروه‌های نمونه را می‌توان با استفاده از آزمون‌های آماری آنالیز نمود. جزییات هر مرحله در فصل‌هایی که در پرانتزها ذکر شده‌اند، ارائه گردیده است.



نگاره‌ی ۲-۲: نرم‌افزار متن باز Chipster از طریق یک رابط کاربری گرافیکی بصری، مجموعه‌ای از ابزارهای آنالیز برای داده‌های توالی‌یابی RNA را فراهم آورده است. پنل گردش کار (سمت چپ پایین) روابط بین فایل‌های نتایج را نشان می‌دهد. این تصویر از صفحه‌ی نمایش، یک آنالیز افتراقی بیان از سلول‌های GM12892 و hESC که به عنوان یک مثال در این کتاب مورد استفاده واقع شده است، را نشان می‌دهد. برآوردهای بیان در سطح سلولی با استفاده از خوانش‌های همردیف شده ژنوم و ابزار HTSeq به دست آمده و تفاوت در بیان با استفاده از بسته‌ی نرم‌افزاری edgeR Bioconductor آنالیزشده است. ژن‌هایی که بیان متفاوت دارند، با تغیر فولد بیشتر پاکسازی شده و در مرورگر ژنومی مصورسازی می‌گردند.

خروج داده‌ها به ابزارهای مختلفی که برخی از آنها استاندارد بوده و برخی نیز استاندارد نیستند، را انجام دهند. برای این کاربران ضرورت دارد که با محیط خط فرمان بیشتر آشنا شوند. داشتن دانش کافی از فرمان‌های یونیکس مفید است. منابع در دسترس عالی روی اینترنت برای این منظور وجود دارد (نظیر: www.ks.uiuc.edu/Training/Tutorials/Reference/unixprimer.html). تعداد زیادی

از برنامه‌های آنالیز داده‌های توالی‌یابی RNA در جاوا[1]، پرل[2]، C++ یا پایتون[3] نوشته شده‌اند. اجرای این برنامه‌ها نیازمند دانش برنامه‌نویسی نیست. ولی آگاهی از برنامه‌نویسی مفید است. در واقع امروزه اکثر ابزارهای مورد استفاده، راهنماهای بسیار خوبی برای نصب و اجرای برنامه‌ها دارند. برای مثال‌های موجود در این کتاب و ابزارهایی که در حال حاضر مورد استفاده قرارمی‌گیرند، برخی که با R مرتبط هستند، از اهمیت بالایی برخوردارند. برای افرادی که با R آشنا نیستند، پیشنهاد می‌شود که از منابع اینترنتی رایگان نظیر http://www.ats.ucla.edu/stat/r/استفاده کنند. کسانی هم که مایل هستند فقط روش آنالیز داده‌های توالی‌یابی RNA را بدون وارد شدن به آزمایشات و آنالیزها فرا بگیرند، تنها نیاز به ذهنی باز و انعطاف‌پذیر دارند.

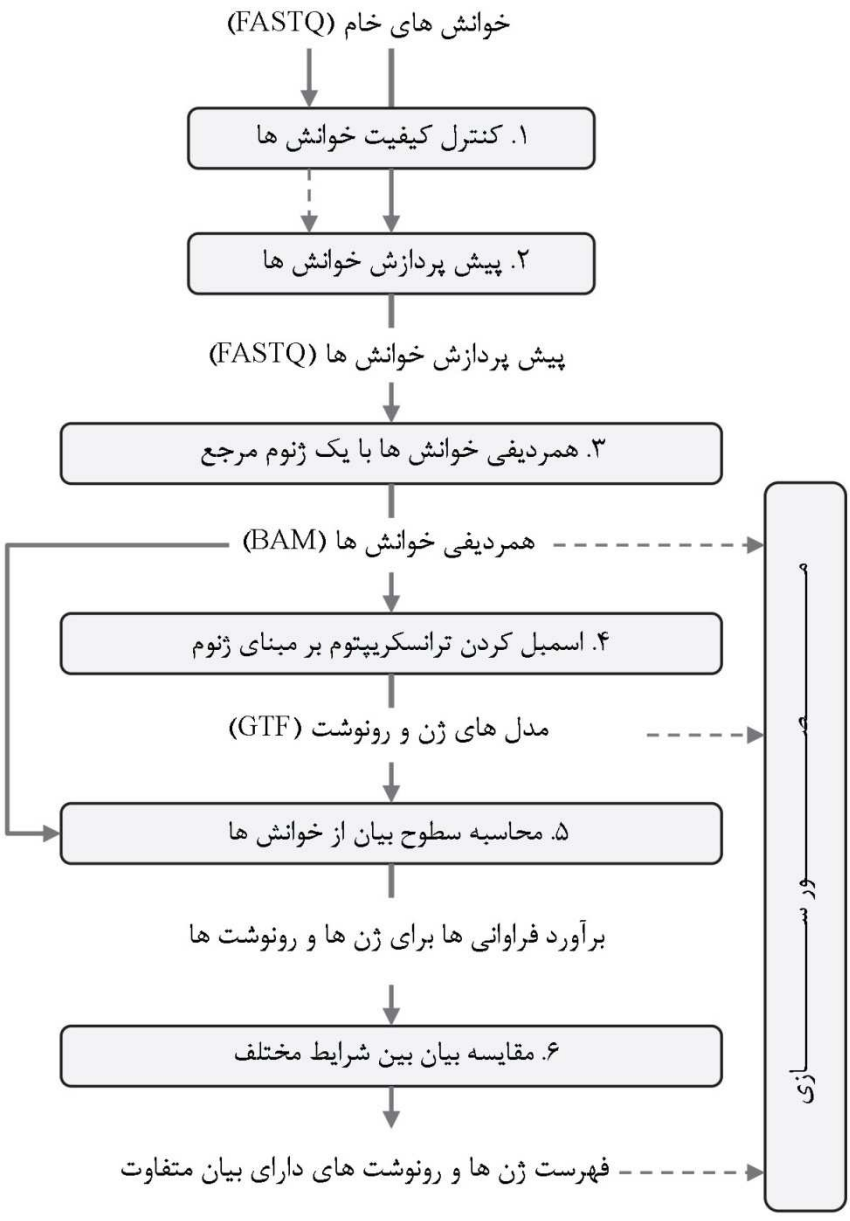
۲-۲ گردش کار آنالیز افتراقی بیان

در بخش قبل، تصویری کلی از آنالیز داده‌ها ارائه گردید. در این بخش توضیحات کوتاهی در رابطه با مراحل اصلی آنالیز افتراقی بیان که معمول‌ترین آنالیز برای داده‌های توالی‌یابی RNA است، ارائه می‌گردد. در گردش کار مثال، فرض می‌شود که یک ژنوم مرجع در دسترس است. برای هر مرحله از اهداف آنالیز، برخی از گزینه‌ها، ورودی‌ها و خروجی‌ها شرح داده شده و فصلی از کتاب که مرحله‌ی مزبور در آن به طور کامل تشریح گردیده است ذکر می‌شود. هدف از این بخش فراهم آوردن یک دیدگاه کُلی از کُل فرآیند آنالیز داده‌ها بوده تا از این طریق کاربر بتواند پی ببرد که چگونه مراحل جداگانه به یکدیگر پیوند می‌یابند. نگاره‌ی ۲-۳ شمای کلی از مراحل کار با داده‌های مزبور را ارائه می‌کند. برنامه‌های جداگانه برای هر مرحله وجود داشته ولی برخی از ابزارها می‌توانند تعدادی از آنها را با هم تلفیق نمایند.

۲-۲-۱ مرحله‌ی اول: کنترل کیفیت خوانش‌ها

آنالیز با خوانش‌های خام توالی‌یابی که معمولاً در فرمت FASTQ هستند، شروع می‌شود. البته گاهی اوقات فرمت‌های دیگر نیز می‌توانند مورد استفاده قرار گیرند. اگر سایر فرمت‌ها توسط برنامه پشتیبانی نگردد، خوانش‌ها بایستی مجدداً به فرمت FASTQ تبدیل گردند. در مرحله‌ی نخست، یک آنالیز عمومی کنترل کیفیت صورت می‌گیرد. این آنالیز کیفیت کُلی میلیون‌ها خوانش را مورد بررسی قرار می‌دهد. همان‌گونه که در فصل سوم تشریح می‌گردد، خوانش‌ها از نظر وجود بازهای با

1 - Java
2- Perl
3- Python



نگاره‌ی ۲-۳: گردش کار آنالیز افتراقی بیان که شامل چندین مرحله‌ی وابسته به هم است. فرمت فایل‌های خروجی معمول در داخل پرانتزها ذکر شده است.

اطمینان پایین[1]، اُریبی در ترتیب نوکلئوتیدها، آداپتورها، مضاعف‌شدگی‌ها[2] و غیره پویش می‌گردند. خروجی این مرحله آماره‌های پایه نظیر تعداد خوانش‌ها و اطلاعات کیفی که شما را در تصمیمات پیش پردازش در مرحله‌ی بعد راهنمایی می‌کنند، است.

۲-۲-۲ مرحله‌ی دوم: پیش پردازش خوانش‌ها

هدف از پیش پردازش[3]، حذف بازهای با کیفیت پایین و ورساخته‌ها[4] نظیر آداپتورها یا توالی‌های ساخت کتابخانه حاصل از خوانش‌های منفرد است. ورساخته‌های آزمایشی را نیز می‌توان حذف نمود. به عنوان مثال دُم‌های پُلی A را می‌توان حذف کرد. زیرا این توالی‌ها با مراحل بعدی آنالیز تداخل دارند. منبع دیگر ورساخته‌ها میکروبیوم[5] است که در بدن اکثر جانداران حاضر هستند. حذف توالی‌های *Escherichia coli* از RNA نمونه‌های بافت انسانی می‌تواند در مراحل پایین دستی بعدی مفید واقع شود. خوانش‌ها به دلیل اندازه‌شان می‌توانند پیرایش[6] گردند. به عنوان مثال، توالی‌های microRNA بالغ ۲۱ تا ۲۲ نوکلئوتید طول داشته ولی طول خوانش می‌تواند ۵۰ نوکلئوتید باشد. پیش پردازش با استفاده از پیرایش و پاکسازی[7]در فصل سوم مورد بحث قرار گرفته و تصحیح خطاها نیز در فصل پنجم و در قالب اسمبل کردن[8] ترانسکریپتوم مورد بررسی واقع می‌شود. بعد از پیش پردازش، داده‌ها در شکل تمیز و آراسته بوده و می‌توانند به مرحله‌ی بعدی آنالیز وارد شوند.

۲-۲-۳ مرحله‌ی سوم: همردیفی خوانش‌ها با ژنوم مرجع

هدف از این مرحله یافتن نقطه‌ی مرجع برای هر خوانش است. اگر هنوز ژنوم مرجع[9] در دسترس نباشد، می‌توان خوانش‌ها را با یک ترانسکریپتوم (که همان‌گونه که در فصل پنجم تشریح می‌شود، در صورت لزوم می‌توان آنرا از خوانش‌ها و به شیوه‌ی از نو[10]، ساخت) مکان‌یابی نمود.

1- Lowconfidence
2- Duplicate
3- Preprocessing
4- Artifact
5- Microbiome
6- Trimme
7- Filter
8- Assembly
9- Reference genome
10- *De novo*

وقتی که یک خوانش با یک مرجع مکان‌یابی می‌گردد، یک همردیفی[1] ایجاد می‌شود. در این مرحله لازم است که علاوه بر فایل خوانش‌های پیش پردازش شده، یک توالی مرجع به عنوان یکی از فایل‌های ورودی در اختیار باشد. مکان‌یابی از نظر محاسباتی یک فرآیند وسیع و سنگین است. زیرا میلیون‌ها خوانش برای مکان‌یابی وجود داشته، ژنوم بزرگ بوده و خوانش‌های پیرایش شده بایستی به صورت غیرمستقیم مکان‌یابی شوند. بنابراین اغلب برای تسریع در مکان‌یابی، توالی ژنوم به یک نمایه تبدیل و فشرده‌سازی می‌گردد. معمول‌ترین تبدیلی که مورد استفاده قرار می‌گیرد، تبدیل باروز - ویلر[2] است. خروجی این مرحله یک فایل همردیفی است که در آن خوانش‌های مکان‌یابی شده و موقعیت‌های مکان‌یابی آنها در مرجع، فهرست شده است. علاوه بر آنالیزهای پایین دستی، خوانش‌های همردیف شده می‌توانند با استفاده از بینندگان ژنوم[3]، در زمینه‌ی ژنومی مصورسازی[4] شوند. مکان‌یابی و مصورسازی ژنومی در فصل چهارم که برخی از ابزارهای دست‌ورزی فایل‌های همردیفی مورد بحث واقع می‌شوند، بررسی می‌گردند.

۲-۲-۴ مرحله‌ی چهارم: اسمبل کردن ترانسکریپتوم بر مبنای ژنوم

اگر خوانش‌ها با ژنوم همردیف شوند، همردیفی‌ها می‌توانند برای شناسایی واریانت‌های ژن‌ها و ویرایش‌های جدید مورد استفاده قرار گیرند. ژن‌ها در مقایسه با خوانش‌های توالی، بزرگ هستند. به عنوان مثال، اندازه‌یmRNA بالغ حاصل از گونه‌های پستانداران معمولاً ۱/۵ کیلوباز است. ولی خوانش‌های توالی‌یابی RNA بین ۱۰۰ تا ۲۵۰ نوکلئوتید طول دارند. بنابراین به طور معمول نمی‌توان ساختار دقیق (نظیر محل شروع رونویسی، سازماندهی اینترون – اگزون و محل پُلی A) یک رونوشت حاصل از یک خوانش را معلوم نمود. پلتفرم‌های خوانش بلندتر نظیر سامانه‌های پاسیفیک بایوساینسز می‌توانند به طور واقعی از طریق یک رونوشت کامل توالی‌یابی کنند. ولی داده‌های خوانش کوتاه‌تر هنوز بر آنالیز غلبه دارند. اکثر اگزون‌ها کمتر از ۲۰۰ جفت باز بوده و بنابراین لازم است که استفاده و ترتیب اگزون‌های جایگزین از مکان‌یابی با ژنوم بازسازی شده و همردیف‌های حاصل از یک ناحیه به ناحیه‌ی دیگر پیوند داده شوند. این موضوع روی کاغذ ساده به نظر می‌رسد. ولی از نظر محاسباتی سخت و سنگین بوده و بنابراین نیازمند مهارت‌هایی در زمینه اسمبل کردن ترانسکریپتوم است که با جزییات در فصل پنجم مورد بحث و بررسی واقع می‌شوند. خروجی این مرحله مدل‌های ژن و رونوشت است. رونوشت‌های اسمبل شده‌ی حاصل از نمونه‌های

1- Alignment
2- Burrows-Wheeler transform
3- Genome viewer
4- Visualization

مختلف با هم تلفیق شده و برای تولید مدل‌های ژنی پیچیده‌تر که می‌توانند برای کمی‌سازی بیان در مرحله‌ی بعد به کار گرفته شوند، با حاشیه‌نگاری مرجع ترکیب می‌گردند.

۲-۲-۵ مرحله‌ی پنجم: محاسبه‌ی سطوح بیان

جدول کلیدی داده‌ها که توسط این آنالیز تولید می‌شود، جدولی است که تعداد خوانش‌ها به ازای هر ژن و رونوشت را نشان می‌دهد. در این مرحله، یک خوانش تکی بر مبنای موقعیت مکان‌یابی‌اش با یک ژن مرتبط می‌گردد. بیان ژن‌ها و رونوشت‌های جدید را می‌توان با استفاده از مدل‌های ژنی حاصل از مرحله‌ی قبل کمی‌سازی نمود. وقتی که با داده‌های حاصل از یک جاندار با حاشیه‌نگاری خوب نظیر انسان کار می‌شود، به جای آن می‌توان حاشیه‌نگاری مرجع را مورد استفاده قرار داد. بدین ترتیب کمی‌سازی تنها به ژن‌ها و رونوشت‌های شناخته شده محدود می‌گردد. برآوردهای فراوانی را می‌توان در قالب شمارش[1] خوانش‌های خام یا در قالب واحدهای نرمال شده نظیر RPKM (تعداد خوانش در هر هزار نوکلئوتید در رونوشت به ازای هر یک میلیون خوانش) یا FPKM (تعداد قطعات در هر هزار نوکلئوتید به ازای هر یک میلیون خوانش مکان‌یابی شده) گزارش نمود. اطلاعات تعداد خوانش‌های مکان‌یابی شده به ژن‌ها و انواع مختلف ویژگی‌های ژنومی که معیارهای مهمی در کنترل کیفیت نیز هستند، در فصل ششم تشریح می‌گردد. در این مرحله، داده‌ها به سادگی در قالب یک جدول از ژن‌ها و شمارش خوانش‌های آنها یا مقادیر RPKM/FPKM ارائه می‌شوند.

۲-۲-۶ مرحله‌ی ششم: مقایسه‌ی بیان ژن بین شرایط مختلف

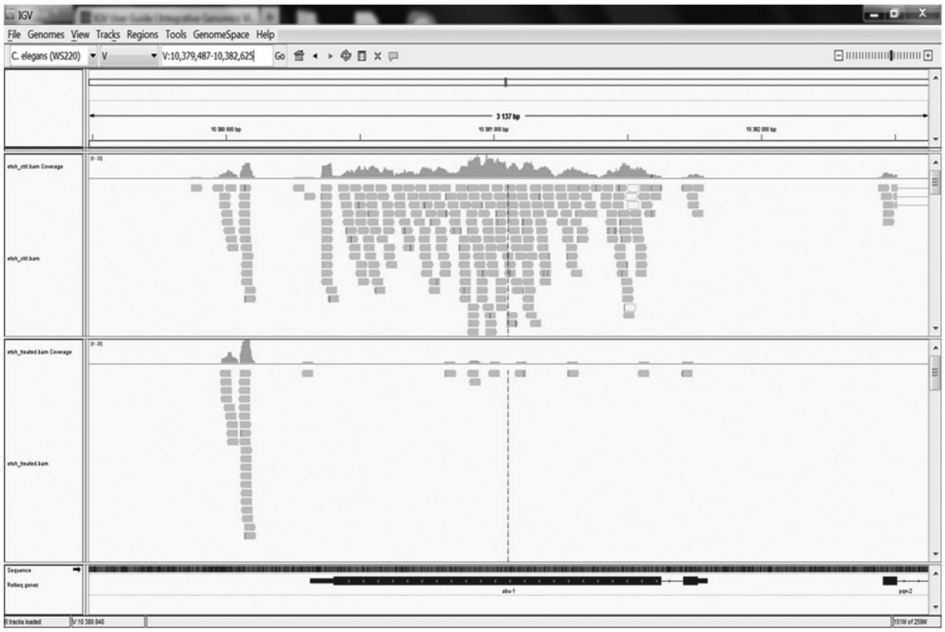
هنگامی که اطلاعات بیان موجود باشد، می‌توان مقادیر بیان بین گروه‌های نمونه‌ها را با استفاده از آزمون‌های آماری مقایسه نمود. به دلیل وجود اختلاف‌های ممکن در تعداد خوانش و ترکیب ترانسکریپتوم، لازم است که نرمال‌سازی صورت گیرد. اکثر ابزارهای آماری، روش‌های نرمال‌سازی را نیز در مجموعه‌ی خود دارند. روش‌های آماری برای نرمال‌سازی در فصل‌های هشتم و نهم ارائه شده‌اند.

۲-۲-۷ مرحله‌ی هفتم: مصورسازی داده‌ها در زمینه‌ی ژنومی

مصورسازی خوانش‌ها و نتایج در یک زمینه‌ی ژنومی در طی مراحل مختلف آنالیز و با هدف

1- Count

رسیدن به دیدگاه‌هایی از ساختار ژن و رونوشت و برداشتی کلی از فراوانی‌ها، از اهمیت زیادی برخوردار است. تعداد زیادی از مرورگرهای ژنومی در دسترس بوده و این مرورگرها می‌توانند با داده‌های شخصی خود یا داده‌های از پیش بارگذاری شده مورد استفاده قرار گیرند. یک مثال از این مرورگرها Integrative Genomic Viewer (IGV) است که این امکان را فراهم می‌آورد که شما توالی‌یابی RNA را همراه با سایر داده‌های ژنومی مشاهده کنید (۳). در نگاره‌ی ۲-۴ فایل‌های همردیفی خوانش برای دو نمونه (یک نمونه‌ی شاهد و یک نمونه‌ی تیمار شده با اتانول) بارگذاری شده است. باید توجه شود که تعداد خوانش‌های مکان‌یابی شده حاصل از نمونه‌ی شاهد بیشتر است. مرورگرهای ژنومی در فصل چهارم مورد بحث و بررسی قرار می‌گیرند.



نگاره‌ی ۲-۴: پنجره‌ی Integrative Genomic Viewer (IGV) که خوانش‌های توالی‌یابی RNA حاصل از ژن *C. elagans* *abu-1* را نشان می‌دهد. پنل بالایی نشان دهنده‌ی خوانش‌های حاصل از شاهد و پنل پایینی نشان دهنده‌ی خوانش‌های حاصل از حیوانات تیمار شده با اتانول است.

۲-۳ آنالیزهای پایین‌دستی

۲-۳-۱ حاشیه‌نگاری ژن

یکی از خروجی‌های معمول مطالعات ترانسکریپتوم که برای دستیابی به آن توالی‌یابی صورت می‌گیرد، فهرستی از ژن‌های بیان شده‌ی معلوم است. خوانش‌ها می‌توانند اطلاعات بیشتری در رابطه با ساختار ژن نظیر محل‌های آغاز جایگزین برای رونویسی و اگزون‌های جدید فراهم آورند. ژن‌های جدید نیز ممکن است شناسایی شوند. معمولاً خروجی برای ژن‌های جدید شامل یک شناساگر[1] ژن احتمالی و یک توالی است. سپس کاربر بایستی با کمک ابزارهایی نظیر BLAST توالی مزبور را با ژن‌های شناخته شده مقایسه کند تا از این طریق عملکرد آن ژن شناسایی گردد. علاوه بر ژن‌های رمزگر پروتئین، سایر انواع رونوشت‌های جدید نظیر RNA های نارمزگر بلند نیز می‌توانند شناسایی شوند.

در مواردی که داده‌های توالی‌یابی RNA برای نخستین بار ژن‌های یک جاندار را توصیف می‌کنند، یک مسیر[2] خودکار برای حاشیه‌نگاری این ژن‌ها در نظر گرفته می‌شود. حاشیه‌نگاری روی ژن‌هایی که به صورت محاسباتی پیش‌بینی شده‌اند، از خوانش‌های همردیف شده برای ساختن رونوشت‌های بلند، ایجاد می‌شود. ابتدا مولکول‌های RNA ساختاری (tRNA ها، RNA های ریبوزومی، snoRNA ها و غیره) حاشیه‌نگاری و حذف می‌شوند. سپس ژن‌های رمزگر پروتئین‌ها با پایگاه‌های داده‌ی ژن‌های شناخته شده همردیف گردیده و عملکرد آنها از مشابهت توالی‌ها استنباط می‌شود.

۲-۳-۲ آنالیز غنی‌سازی مجموعه‌ی ژنی

به طور معمول خروجی آنالیز افتراقی بیان فهرستی از ژن‌ها و تفاوت‌های سطوح بیان آنها بین دو یا چند گروه مختلف است. کاربر می‌تواند آستانه‌های برشی مختلف نظیر تفاوت دو برابری[3] یا مقدار p[4] کمتر از ۰/۰۱ را به عنوان معیاری برای کوتاه کردن فهرست مزبور تا حدی که تعداد ژن‌های آن قابل کار کردن باشد، استفاده نماید. حتی در صورت استفاده از معیارهای سخت‌گیرانه نیز ممکن است در انتها کاربر با صدها ژن مواجه شده و حس کند که کار کردن با داده‌های حاصل کاری سخت و دشوار است. آنالیز غنی‌سازی مجموعه‌ی ژنی[5] ابزاری را فراهم می‌آورد که به

1- Identifier
2- Pipeline
3- 2-fold difference
4- *p*-value
5- Gene set enrichment analysis

وسیله‌ی آن ژن‌های موجود در یک مجموعه‌ی داده می‌توانند بر مبنای حاشیه‌نویسی‌شان گروه‌بندی شده و برای فزون‌نمایی[1] در یک گروه در مقایسه با یک زمینه نظیر کلیه‌ی ژن‌ها، مورد آزمون واقع شوند. به عنوان مثال، یک فهرست ژنی متشکل از ۲۰۰ ژن ممکن است که حاوی ۲۰ فاکتور رونویسی باشد. اگر ژنومی که دارای ۲۲۰۰۰ ژن است، حاوی ۸۰۰ فاکتور رونویسی باشد، آیا وضعیت فوق معنی‌دار است؟ معمول‌ترین حاشیه‌نگاری مورد استفاده برای گروه‌بندی ژن‌ها، هستی‌شناسی ژن[2] می‌باشد (۴). هستی‌شناسی ژن یک فرآیند آشیانه‌ای[3] است، طوری که می‌توانید یک ژن را با جزییات مختلف حاشیه‌نویسی کرده و یک ژن هم می‌تواند چندین حاشیه‌نگاری داشته باشد. به عنوان مثال یک فاکتور رونویسی می‌تواند یک فاکتور هسته‌ای، یک گیرنده و یک پروتئین متصل شونده به DNA نیز باشد. آنالیز غنی‌سازی مجموعه‌ی ژنی فهرستی از عملکردهای مولکولی فزون‌نمایی شده، فرآیندهای زیستی و موقعیت‌های سلولی فراهم می‌آورد که می‌توانند برای آزمون فرضیه مبنی بر اینکه آیا ژن‌های موجود در یک مسیر بیوشیمیایی یا سلولی دچار اختلال در تنظیم[4] شده‌اند یا خیر، مورد استفاده قرار گیرند.

۲-۴ گردش کارها و مسیرهای خودکار

در اغلب موارد، مدیریت خودکار مراحل چندگانه‌ی آنالیز داده‌ها امری مطلوب است. برای آنالیز معمول توالی‌یابی RNA می‌توان مسیر آنالیز را در قالب مراحل، ورودی‌ها، خروجی‌ها و پارامترهای هر مرحله جهت استفاده مجدد ترسیم نمود. به دلیل پیچیدگی آنالیز و تعداد زیاد ابزارهای مورد استفاده در آن، ارائه‌ی همه‌ی مراحل این آنالیز در قالب یک رابط کاربری گرافیکی واحد چالش برانگیزتر است. با این حال، پیشرفت‌های زیادی در این زمینه صورت گرفته است و در حال حاضر می‌توان معمول‌ترین آنالیز توالی‌یابی RNA را در یک ابزار نرم‌افزاری واحد اجرا نمود. این ابزارها از برنامه‌هایی استفاده می‌کنند که می‌توان آنها را در قالب یک گردش کار در یک رابط کاربری گرافیکی واحد گردآوری کرد. همان‌گونه که قبلاً نیز اشاره شد، دو مثال از این ابزارها که به کاربر اجازه می‌دهند گردش کارهای شخصی خود را برای توالی‌یابی نسل بعد ایجاد نمایند، عبارت از Galaxy (galaxyproject.org/) (۱) و Chpster (chipster.csc.f) (۲) هستند. علاوه بر این ابزارها، ابزارهای تجاری نیز در دسترس می‌باشند. این ابزارها یک رابط کاربری گرافیکی سریع و سودمند ارائه کرده و به آسانی قابل استفاده هستند. ولی استفاده‌ی از آنها مستلزم هزینه است.

1- Overrepresentation
2- Gene Ontology
3- Hierarchical
4- Dysregulate

۲–۵ ملزومات سخت‌افزاری

لازم است بدانید که توالی‌یابی RNA حجم بسیار بالایی از داده‌ها را تولید می‌کند. تنها یک نمونه می‌تواند ۶۰ مگا خوانشِ ۱۰۰ بازی تولید کند (۰/۶ گیگا باز توالی) که نیازمند چندین گیگا بایت (GB) حافظه برای ذخیره‌سازی است. ملزومات سخت‌افزاری به اندازه و نوع آزمایش بستگی دارد. اگر هیچ اسمبل ترانسکریپتوم از نو انجام نشده باشد، یک مطالعه‌ی در مقیاس کوچک می‌تواند روی یک رایانه‌ی مجهز به ۴ گیگا بایت رَم (حافظه‌ی با دسترسی تصادفی)، ۲۰۰ گیگا بایت فضای خالی هارد دیسک و ۲/۵ گیگا هرتز سرعت ریزپردازنده انجام شود. در چنین شرایطی مکان‌یابی خوانش‌ها روی یک ژنوم حداقل یک شب به طول می‌انجامد. چون طرح‌های آزمایشی پیچیده‌تر دارای شرایط پیچیده‌تری بوده و در آنها از تکرار استفاده می‌شود، زمان‌های اجرای هفتگی برای مکان‌یابی امری رایج است. در این سطح، اگر شما بخواهید اجرا را روی رایانه‌ی شخصی انجام دهید، باید حداقل ۱۶ گیگا بایت رَم، ۲ ترا بایت هارد خارجی یا ۴۸ ترا بایت سِرورِ و ۳/۶ گیگا هرتز سرعت ریزپردازنده در اختیار داشته باشید.

وقتی که یک هسته یا مرکز خدمات مسئول آنالیز باشد، رایانه‌ی استاندارد برای مدیریت این برونداد که نیازمند زمان بسیار زیادی است، کافی نیست. برای یک هسته‌ی خدماتی آنالیز داده‌های توالی‌یابی RNA ، یک شبکه‌ی لینوکس با بیش از ۲۰۰ پردازنده و یک پتا بایت حافظه توصیه می‌گردد. با این حال باید از راه حل‌های دیگر نظیر سرویس‌های ابری که ضرورتاً حجم نامحدودی از حافظه و توان محاسباتی را در اختیار دارند، نیز آگاه شوید. یکی از بهترین ویژگی‌های سامانه‌ی ابری، قابلیت رهن کردن فضای ذخیره‌ی داده و دریافت توان پردازش بر مبنای نیاز است.

همچنین شما باید از نرخ انتقال داده از محل نگهداری داده‌های خام توالی‌یابی RNA به محیط آنالیز آگاه باشید. حتی با بهترین زیرساخت‌های فیبر نوری با پهنای باند یک گیگا بیت بر ثانیه (۱۲۵ مگا بایت بر ثانیه)، ممکن است انتقال داده‌های خام به واسطه‌ی تراکم، نگارش روی هارد دیسک یا محدودیت‌های ایجاد شده توسط مدیران سامانه، ساعت‌ها یا روزها به طول انجامد. از دیدگاه عملی، کاربران معمول توالی‌یابی RNA کنترل زیادی روی پهنای باند در دسترس ندارند. ولی آنها می‌توانند زمان مورد نیاز برای انتقال ساده‌ی داده‌ها را محاسبه کرده و در نظر بگیرند. در برخی از موارد راه حل آسان‌تر و سریع‌تر آن است که از آزمایشگاه تولید کننده‌ی داده‌ها خواسته شود که هارد دیسک‌های حاوی اطلاعات را از طریق پُست برای کاربر ارسال نماید.

۲–۶ مثال‌های موجود در کتاب حاضر

این کتاب حاوی چندین مثال در مورد چـگونگی استفاده از ابزارهای آنالیز مختلف است. توصیه

می‌شود که شما نیز مثال‌ها را با استفاده از همان مجموعه‌ی داده‌ها اجرا نمایید. فایل داده‌های مزبور برای دانلود در وبسایت کتاب به آدرس http://rnaseq-book.blogspot.fi/ در دسترس است. برای اینکه هم متخصصین بیوانفورماتیک و هم محققین آزمایشگاهی که با برنامه‌نویسی آشنایی ندارند، بتوانند مثال‌ها را دنبال کنند، دو مجموعه‌ی راهنما برای هر کار فراهم شده است. یکی از این مجموعه‌ها ابزارهای خط فرمان و Rرا مورد استفاده قرار داده و مجموعه‌ی دیگر، از نرم‌افزار Chipster که یک رابط کاربری گرافیکی دارد، استفاده می‌نماید. همه‌ی نرم‌افزارهای مورد استفاده در مثال‌ها، متن باز بوده و به صورت رایگان در دسترس هستند.

۲-۶-۱ استفاده از ابزارهای خط فرمان و R

به خوانندگان کتاب پیشنهاد می‌شود که یک سیتم عامل از توزیع لینوکس[1] نظیر اوبونتو[2] را نصب نمایند. زیرا اکثر ابزارهای آنالیز خط فرمان که در این کتاب نشان داده می‌شود، روی سیستم عامل لینوکس اجرا می‌گردد. اگر شما یک کاربر ویندوز بوده و نمی‌خواهید که به طور کامل به لینوکس مهاجرت کنید، می‌توانید یک پارتیشن دیسک بسازید تا از این طریق هم لینوکس و هم ویندوز را روی رایانه‌ی خود اجرا کنید. برای جزییات بیشتر در مورد نحوه‌ی دانلود اوبونتو می‌توان به http://www.ubuntu.com/download/desktop مراجعه کرد.

این مثال‌ها تعداد زیادی از ابزارهای آنالیز را پوشش داده و ابزارهای نصب برای همه‌ی آنها فراتر از مطالب این کتاب است. ولی جزییات راهنمای هر ابزار روی وبسایت آن ابزار در دسترس است. یک راه حل جایگزین برای دانلود و نصب تک تک این ابزارها، استفاده از ماشین مجازی Chipster است که بر مبنای اوبونتو کار کرده و حاوی اکثر ابزارهای آنالیز و مجموعه‌ی داده‌های مرجع است. همان‌گونه که در زیر تشریح می‌شود، شما می‌توانید از طریق یک رابط کاربری گرافیکی، از این ابزارها استفاده کرده و یا اینکه وارد ماشین مجازی شده و از آنها روی خط فرمان استفاده نمایید. راهنمای نصب ماشین مجازی در بخش بعد ارائه می‌گردد.

یادداشت‌برداری از کُدهای نوشته شده مفید واقع می‌شود. زیرا این امکان را فراهم می‌آورد که بتوان بعداً مراحل آنالیز را مجدداً بازتولید نمود. علاوه بر ویرایشگرهای ساده متنی نظیر Notepad در ویندوز و nano در لینوکس، ویرایشگرهای تخصصی کُد نظیر Notepad++ و Rstudio برای R در ویندوز و emacs (با افزونه‌های اضافی) در لینوکس نیز وجود دارند. استفاده از این ابزارهای

1- Linux
2- Ubuntu

تخصصی شدیداً توصیه می‌شود. زیرا ویرایش کُد را با استفاده از رنگی نمودن بخش‌های متفاوت کُد، آسان‌تر کرده و بدین ترتیب مشاهده‌ی دستورات[1] و برهان‌ها[2] تسهیل می‌گردد.

هرگاه که شما با کُد یا دستورات جدید روبرو می‌شوید، مایل هستید که اطلاعات بیشتری در رابطه با گزینه‌های موجود و کارهای داخلی دریافت کنید. برای R ، می‌توان با استفاده از دستور «?» و سپس نام دستور، راهنمایی‌های موجود را جستجو نمود. به عنوان مثال،lm? صفحه‌ی راهنمای دستور lm که مدل‌های خطی را برای داده‌ها برازش می‌دهد، باز می‌نماید. برای دستورات یونیکس، می‌توان با دستور man به صفحه‌ی راهنما دسترسی یافت. به عنوان مثال، man less صفحه‌ی راهنمای دستور less را باز می‌کند. راهنماهای مربوط به نرم‌افزارهای آنالیز خط فرمان، اغلب در صفحه‌ی نخست همان نرم‌افزارها موجودند. همچنین انجمن‌های فعالی نظیر SEQanswers (http://seqanswers.com/) Biostar . (http://www.biostars.org/) نیز وجود دارند که می‌توان سوالات مربوط به آنالیز داده‌ها را در آنها مطرح کرد.

۲-۶-۲ استفاده از نرم افزار Chipster

اگر خوانندگان می‌خواهند مثال‌های موجود در این کتاب را آنالیز نموده و در عین حال مایل نیستند که با ابزارهای خط فرمان و R/BioConductor کار کنند، می‌توانند همان آنالیزها را با استفاده از نرم‌افزار Chipster اجرا نمایند. Chipster یک نرم‌افزار متن باز بوده و به صورت رایگان در دسترس است. این نرم‌افزار یک مجموعه‌ی جامع از ابزارهای آنالیز داده‌ها برای کاربردهای مختلف توالی‌یابی نسل جدید و از جمله برای توالی‌یابی RNA فراهم آورده است. این ابزارها کلیه‌ی مراحل از کنترل کیفیت تا آزمون‌های آماری و آنالیز مسیر[3] را پوشش می‌دهند. شما می‌توانید آنالیزها را از هر نقطه‌ای نظیر وارد کردن خوانش‌های خام (FASTQ)، همردیف‌ها (BAM) یا جدول شمارش‌ها شروع کنید.

از نظر تکنیکی، Chipster یک سامانه‌ی کلاینت – سِروِر بر مبنای جاوا است که به صورت یک تصویر ماشین مجازی در آدرس http://chipster.sourceforge.net/downloads.shtml قابل دسترسی است. ماشین مجازی حاوی کلیه‌ی ابزارهای آنالیز و مجموعه‌ی داده‌های مرجع بوده و آماده‌ی استفاده است (ولی نسبتاً بزرگ می‌باشد). لازم است که از یک نرم‌افزار مجازی‌ساز نظیر VMware یا Vitual Box برای اجرای ماشین مجازی Chipster در ویندوز، مَک یا لینوکس استفاده شود. اگر شما تجربه‌ی قبلی در زمینه‌ی کار با ماشین‌های مجازی ندارید، توصیه می‌شود که از یک

1- Command
2- Argument
3- Pathway analysis

متخصص برای نصب کمک بگیرید. همچنین می‌توان از سِروِر Chipster در کشور فنلاند نیز بهره گرفت (http://chipster.csc.fi/). البته این روش، به دلیل زمان انتقال داده، روشی بهینه نیست. در صورتی که بخواهید بخش‌هایی از آنالیز آماده را جستجو کنید یا از مرورگر ژنوم استفاده نمایید، می‌توانید با نام کاربری مهمان وارد شوید. ارزیابی رایگان اکانت‌ها نیز امکان‌پذیر است.

چند راهنمایی کُلی برای استفاده از Chipster در زیر ارائه شده است. البته نکات مربوط به مراحل هر آنالیز به طور جداگانه در فصل مربوط به آن آنالیز ارائه گردیده است. تصویری از صفحه‌ی رابط کاربری Chipster نیز در نگاره‌ی ۲-۲ نمایش داده شده است.

- با انتخاب Import files ، داده‌ها وارد می‌شوند. فایل‌ها هم در پنجره‌ی Datasets (بالا و چپ) و هم در پنجره‌ی Workflow (پایین و چپ) ظاهر می‌گردند.
- ابزارهای آنالیز در پنجره‌ی Analysis tools به صورت دسته‌بندی شده نمایش داده شده است (بالا و راست). هر ابزاری دارای یک متن راهنمای کوچک بوده و با کلیک روی دکمه‌ی More help می‌توان به راهنمای هر ابزار دست یافت.
- با انتخاب دکمه‌ی Show tool source code می‌توان کُد منبع یک ابزار را ملاحظه نمود. ابزارها می‌توانند با پارامترهای پیش فرض اجرا شوند. ولی توصیه می‌شود که مقادیر پارامترهای مزبور از نظر مناسب بودن با داده‌ها، کنترل گردند.
- برای اجرای یک آنالیز، بایستی فایل مربوط به آن، دسته‌ی ابزار و خود ابزار را انتخاب نمود. بعد از کنترل و تغییر احتمالی پارامترها، باید روی Run کلیک کرد. می‌توان وضعیت اجرا را با کلیک روی مثلث کوچک در پنل پایینی رصد نمود.
- وقتی که یک اجرای آنالیز تکمیل می‌شود، فایل‌های نتایج در پنجره‌ی Datasets و پنجره‌ی Workflow ظاهر می‌شوند. با انتخاب یک فایل و روش مصورسازی مناسب از پنل Visualization (پایین و راست) می‌توان نتایج را مصورسازی نمود.
- برای ذخیره‌سازی آنالیز، باید File/Save session را انتخاب کرد. با این کار، همه‌ی فایل‌ها، ارتباطاتشان و فرا داده‌ها[1] (اطلاعات مربوط به ابزارها و پارامترهایی که در ایجاد یک فایل نتایج به کار گرفته شده‌اند)، در قالب یک فایل فشرده‌ی zip ذخیره می‌گردد. همچنین می‌توان یک گردش کار، که امکان اجرای مجدد مراحل آنالیز با یک کلیک روی مجموعه‌ی داده‌های متفاوت فراهم می‌آورد، را ذخیره نمود.
- نرم افزار Chipster یک lab book از آنچه که انجام شده است، نگهداری می‌نماید. با کلیک کردن روی آیکونی که به شکل یک کاغذ کوچک در پنل Workflow است، می‌توان یک

1- Metadata

گزارش متنی تهیه کرد که در آن فهرست کلیه‌ی مراحلی که منجر به تولید فایل مورد نظر شده‌اند (شامل ابزارها و تنظیمات پارامترهایشان)، ارائه گردیده است.

- اگر برای یک قابلیت خاص نیاز به راهنمایی و کمکی نظیر مرورگر ژنومی باشد، می‌توان از راهنمای موجود در http://chipster.csc.f/manual/ استفاده کرده یا پرسش مورد نظر را به فهرست پست الکترونیکی Chipster ارسال نمود.

۲-۶-۳ مجموعه داده‌های مثال

مثال‌ها از داده‌های ENCODE حاصل از رده‌های سلولی GM12878 و H1-hESC استفاده می‌کنند. GM12878 یک رده‌ی سلولی لیمفوبلاستوییید است که از خون یک اهدا کننده‌ی ماده با استفاده از انتقال EBV به دست آمده و H1-hESC نیز سلول‌های بنیادی جنینی انسان هستند. این داده‌ها در دانشگاه کالتچ تولید شده و شامل خوانش‌های جفت انتهایی ۷۵ بازی با طول الحاق[1] ۲۰۰ هستند. این داده‌ها از یک پلتفرم قدیمی‌تر اِلومِنا به دست آمده و رمزگذاری[2] کیفیت باز، phred64 است.

سه نمونه‌ی GM12878 و چهار نمونه‌ی H1-hESC وجود دارد. خوانش‌های حاصل از دو تکرار GM12878 در فصل‌های سوم، چهارم و ششم استفاده شده‌اند. فصل پنجم از خوانش‌های حاصل از یک تکرار H1-hESC که روی خوانش‌های مکان‌یابی شده با کروموزوم شماره‌ی ۱۸ متمرکز شده است، استفاده می‌نماید. در فصل‌های ۷ تا ۱۰ نیز از خوانش‌هایی که با کروموزوم شماره‌ی ۱۸ حاصل از کلیه‌ی نمونه‌ها مکان‌یابی شده‌اند، استفاده شده است.

این فایل‌ها را می‌توان در آدرس زیر یافت:
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCaltechRnaSeq/
فایل‌های FASTQ زیر انتخاب شده‌اند:

wgEncodeCaltechRnaSeqGm12892R2x75Il200FastqRd1Rep2V2.fastq.gz
wgEncodeCaltechRnaSeqGm12892R2x75Il200FastqRd2Rep2V2.fastq.gz

فایل‌های BAM زیر نیز انتخاب گردیده‌اند:

wgEncodeCaltechRnaSeqGm12892R2x75Il200AlignsRep1V2.bam
wgEncodeCaltechRnaSeqGm12892R2x75Il200AlignsRep2V2.bam
wgEncodeCaltechRnaSeqGm12892R2x75Il200AlignsRep3V2.bam
wgEncodeCaltechRnaSeqH1hescR2x75Il200AlignsRep1V2.bam

1- Insert
2- Encoding

wgEncodeCaltechRnaSeqH1hescR2x75Il200AlignsRep2V2.bam
wgEncodeCaltechRnaSeqH1hescR2x75Il200AlignsRep3V2.bam
wgEncodeCaltechRnaSeqH1hescR2x75Il200AlignsRep4V2.bam

در یک جفت از مثال‌ها، مجموعه‌ای از داده‌های توالی‌یابی RNA مربوط به یک مطالعه روی کشت‌های اولیه‌ی حاصل از تومورهای پاراتیرویید استفاده شده‌اند. داده‌های خام (همراه با مقادیر برآورد شده‌ی بیان) از Gene Expression Omnibus (شماره‌ی دسترسی GEO : GSE37211) در دسترس بوده ولی در اینجا از یک بسته‌ی نرم‌افزاری R/BioConductor موسوم به parathyroid که توسط Michael Love توسعه داده شده و حاوی یک ویرایش آماده‌ی آنالیز از مجموعه‌ی داده‌ها می‌باشد، استفاده شده است. با این کار می‌توان هم داده‌های بیان و هم فرادادەهای مرتبط با آنرا در R به سادگی بارگذاری نمود. این مجموعه‌ی داده‌ها حاوی سنجه‌های[1] توالی‌یابی RNA مربوط به سطوح mRNA در تومور کشت داده شده از چهار بیمار مختلف است. برای هر بیمار، دو کشت با دو ماده‌ی شیمیایی مختلف (دی‌آریل پروپیونیتریل (DNP) و ۴-هیدروکسی‌تاموکسی‌فن (OHT)) تیمار شده و یک کشت نیز به عنوان شاهد نگهداری شده است. همچنین هر کشت در دو زمان مختلف نمونه‌گیری شده و در نتیجه شش سنجه به ازای هر بیمار موجود است. در مورد یکی از بیماران، تهیه‌ی کتابخانه موفقیت‌آمیز نبوده و در نتیجه توالی قابل استفاده‌ای به دست نیامد.

۲-۷ خلاصه

توالی‌یابی RNA یک فناوری قدرتمند با کاربردهای متعدد بوده و بدین ترتیب مسیرهای آنالیز داده‌ی زیادی نیز برای آن وجود دارد. حتی معمول‌ترین آنالیزها نیز نیازمند مراحل جداگانه‌ی متعددی بوده که به هم وابسته هستند. علی‌رغم اینکه این مسیر در ابتدا پیچیده به نظر می‌رسد، ولی سرانجام می‌توان منطقی موجود در پشت مراحل و نحوه‌ی ارتباط آنها با یکدیگر را مشاهده نمود. به طور خلاصه، مراحل آنالیز داده‌های توالی‌یابی RNA شامل مکان‌یابی، ساخت رونوشت و کمی‌سازی بیان است (۵). تلاش شده است که این مراحل به مراحل کوچک‌تر تقسیم شده و در نتیجه خواننده بتواند هم خروجی داده‌ها را تولید کرده و هم نحوه‌ی به دست آوردن آنها را دریابد. امید می‌رود که در فصل‌های آینده بتوان زمینه، تئوری و اجرای عملی هر کدام از مراحل موجود در توالی‌یابی RNA را نشان داد. چون آنالیز داده‌های توالی‌یابی RNA یک زمینه فعال تحقیقاتی است که همواره در حال تولید روش‌ها و ابزارهای جدید می‌باشد، توصیه می‌گردد که مقالات و انجمن‌های بحث و گفتگو نظیر SEQ answers و Biostar فعالانه پیگیری شوند.

1- Measurement

منابع

1. Goecks J., Nekrutenko A., Taylor J. et al. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biology 11(8):R86, 2010.
2. Kallio M.A., Tuimala J.T., Hupponen T. et al. Chipster: User-friendly analysis sofware for microarray and other high-throughput data. BMC Genomics12:507, 2011.
3. Torvaldsdóttir H., Robinson J.T., and Mesirov J.P. Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. Briefngs in Bioinformatics 14(2):178–192, 2013.
4. Ashburner M., Ball C.A. et al. Te Gene Ontology Consortium. Gene ontology: Tool for the unifcation of biology. Nature Genetics 25(1):25–29, 2000.
5. Garber M., Grabherr M.G., Guttman M. et al. Computational methodsfor transcriptome annotation and quantifcation using RNA-seq. NatureMethods 8(6):469–477, 2011.

فصل سوم

کنترل کیفیت و پیش‌پردازش

۳–۱ مقدمه

معمولاً منشا مشکلات کیفی یا در خود توالی‌یابی و یا در مراحل ماقبل تهیه‌ی کتابخانه است. مشکلات کیفی شامل بازهای با اطمینان پایین، اُریبی مختص توالی، اُریبی موقعیت '5 / '3 ، ورساخته‌های واکنش زنجیره‌ای پلیمراز (PCR)، آداپتورهای پیرایش نشده و آلودگی توالی است. این مشکلات می‌توانند به طور جدی بر مکان‌یابی روی مرجع، اسمبل کردن و برآوردهای بیان تاثیر بگذارند. ولی خوشبختانه بیشتر آنها را با استفاده از پاکسازی، پیرایش، تصحیح خطا یا تصحیح اُریبی می‌توان تصحیح نمود. برخی از مشکلات نمی‌توانند تصحیح شوند. ولی حداقل بایستی از آنها در زمان تفسیر نتایج آگاه بود.

در این فصل کنترل کیفیت خوانش‌های خام که همان فایل‌های FASTQ هستند، پوشش داده می‌شود (۱). اگر خوانش‌ها قبلاً با یک ژنوم مرجع همردیف شده باشند، همان‌گونه که در فصل ششم بررسی می‌شود، می‌توان معیارهای کیفی اضافی را بر اساس اطلاعات موقعیت بررسی نمود. این معیارها شامل یکنواختی پوشش در طول رونوشت‌ها، اشباع شدگی عمق توالی‌یابی، محتوای RNA ریبوزومی و توزیع خوانش بین اگزون‌ها، اینترون‌ها و مناطق بین ژنی[1] هستند. سرانجام اگر خوانش‌های همردیف شده به ازای هر ژن شمارش شده باشند، روابط نمونه‌ها و اثرات دسته[2] را می‌توان با کمک نقشه‌های حرارتی[3] و نمودارهای PCA مصورسازی نمود. این کنترل کیفیت در سطح آزمایش همراه با آزمون‌های آماری در فصل هشتم مورد بحث و بررسی واقع می‌شوند.

در این فصل علاوه بر کنترل کیفیت، پیرایش و پاکسازی داده‌ها که معمول‌ترین روش پیش‌پردازش برای حل کردن مسائل و مشکلات کیفی هستند، نیز پوشش داده می‌شود. تصحیح خطا به عنوان سومین روش پیش‌پردازش نیز همراه با اسمبل کردن ترانسکریپتوم از نو در فصل پنجم مورد بحث واقع می‌گردد.

1- Intergenic region
2- Batch effect
3- Heatmap

۳-۲ نرم‌افزارهای کنترل کیفیت و پیش‌پردازش

ابزارهای زیادی برای کنترل کیفیت خوانش‌ها و پیش‌پردازش توسعه داده شده‌اند. ابزارهای کنترل کیفیت خوانش‌ها شامل FastQC (۲) و PRINSEQ (۳) هستند که چندین معیار کیفی را بررسی کرده و گزارش‌هایی با مصورسازی‌های گویا و عمیق ارائه می‌دهند. بسته‌ی PRINSEQ عملکرد پاکسازی و ویرایش نیز دارد. ابزارهای پیش‌پردازش دیگری نیز وجود دارند که به عنوان مثال می‌توان به نام‌هایی همچون Trimmomatic (۴)، Cuadapt (۵) و FastX (۶) اشاره کرد. در اینجا FastQC ، PRINSEQ و Trimmomatic معرفی شده و جزییات بیشتری از ویژگی‌های‌شان در رابطه با مسائل مختلف کیفی در بخش‌های بعدی همین فصل مورد بررسی قرار می‌گیرد. همان‌گونه که در فصل دوم اشاره شد، مثال‌های مورد استفاده، خوانش‌های جفت انتهایی حاصل از رده‌ی سلولی GM12892 (دو تکرار نمونه) است.

FastQC ۱-۲-۳

FastQC به صورت یک برنامه‌ی مستقل جاوا با یک رابط کاربری گرافیکی (GUI) در دسترس بوده و استفاده از آن روی خط فرمان نیز آسان است. همچنین این برنامه در پلتفرم‌های Galaxy (۷) و Chipster (۸) نیز ادغام شده و یک GUI با تعداد زیادی ابزارهای آنالیز فراهم می‌آورد. FastQC نسبتاً سریع بوده و تنها چند دقیقه برای اجرا شدن ده‌ها میلیون خوانش زمان لازم دارد. فایل‌های ورودی می‌توانند شامل فایل‌های FASTQ (غیرفشرده یا فشرده) یا فایل‌های SAM/BAM باشند (۹). علاوه بر فهرست تعداد خوانش‌ها و رمزگذاری کیفیت آنها، FastQC اطلاعات مربوط به کیفیت و محتوای بازها، طول خوانش و محتوای k-مِر و همچنین حضور بازهای مبهم، توالی‌های فزون‌نمایی شده و مضاعف‌شدگی‌ها را گزارش داده و مصورسازی می‌نماید.

با دستور زیر یک گزارش کیفیت تولید شده و یک فولدر تحت عنوان reads_fastqc برای فایل‌های نتایج ایجاد می‌شود:

```
fastqc reads.fastq.gz
```

فایل fastqc_report.html اطلاعات موجود در fastqc_data.txt را مصورسازی می‌نماید. FastQC علاوه بر گزارش چندین معیار کیفیت، قضاوتی از آنها نیز ارائه می‌دهد. این قضاوت به صورت متن در summary.txt (pass (موفق بودن)، warn (هشدار دادن) و fail (مردود شدن)) و به صورت چراغ‌های راهنمایی در گزارش html ارائه می‌شود (نگاره‌ی ۳-۱). به این نکته باید توجه نمود که این گزارش بر مبنای آستانه‌های عمومی ارائه شده و نمی‌تواند برای داده‌های شما قابل

نگاره‌ی ۳-۱: شروع گزارش کیفی FastQC که شامل آماره‌های پایه (راست) و قضاوت در مورد جنبه‌های مختلف کیفی مورد سنجش (چپ) است.

استفاده باشد. به عنوان مثال، داده‌های استفاده شده در اینجا، در بخش Sequence Duplication Levels مردود می‌شوند. ولی همان‌گونه که در بخش‌های بعدی این فصل بحث خواهد شد، سطوح مضاعف‌شدگی بالا برای داده‌های توالی‌یابی RNA می‌تواند معمول باشد. همچنین FastQC برخی از اطلاعات عمومی مربوط به داده‌ها نظیر تعداد و طول خوانش‌ها و رمزگذاری کیفی مورد استفاده را نیز گزارش می‌نماید (توجه شود که اگر FastQC با استفاده از خوانش‌های همردیف شده اجرا گردد (یعنی فایل BAM)، تعداد خوانش‌های گزارش شده عملاً همان تعداد همردیف‌ها خواهد بود).

PRINSEQ ۳-۲-۲

PRINSEQ به صورت یک نرم‌افزار تحت وب در دسترس بوده (۱۰) و یک ویرایش مستقل آن نیز برای استفاده از خط فرمان ارائه شده است. همچنین این نرم‌افزار در رابط کاربری گرافیکی Chipster نیز قابل دسترس است. در کنترل کیفیت PRINSEQ تعداد خوانش‌ها و توزیع طول آنها، توزیع کیفیت باز، پیچیدگی توالی و محتوای GC همراه با وجود N ها، دُم‌های پُلیA/T ، مضاعف‌شدگی‌ها و آداپتورها گزارش می‌شود. اگر در هرکدام از این موارد مشکلی تشخیص داده شود، گزینه‌های پیرایش و پاکسازی PRINSEQ انواع مختلفی از راه‌های مدیریت آنها را نیز پیشنهاد می‌کند. PRINSEQ فایل‌های FASTQ غیر فشرده، FASTA و QUAL را می‌پذیرد. گزارش کیفیت، پیرایش و پاکسازی با برنامه‌ی پرل prinseq-lite.pl انجام می‌شود. تعداد زیادی از

گزینه‌های پیرایش و پاکسازی را می‌توان در این فرمان ترکیب نمود. ترتیب پردازش آنها به چگونگی فهرست نمودن‌شان در فرمان بستگی ندارد. زیرا این فرمان در PRINSEQ اختصاصی[1] است. این ترتیب در منوی help که به صورت زیر قابل دسترس است، تشریح شده است:

```
prinseq-lite.pl -help
```

PRINSEQ می‌تواند گزارش‌های کیفی را هم در قالب متنی و هم در قالب html تولید کند. برای ساخت یک گزارش html ، دو دستور لازم است. نخستین دستور، یک فایل نمودار موقت تولید می‌نماید:

```
prinseq-lite.pl -fastq reads.fastq -phred64 -out_goodnull -
out_bad null -graph_data graph
```

چون هیچ‌گونه پیش‌پردازشی صورت نمی‌گیرد و بنابراین خوانش‌های پذیرفته شده یا حذف شده وجود نخواهند داشت، لذا فایل‌های خروجی برای آنها (-out_good و -out_bad) روی null تنظیم می‌گردد. توصیف کننده‌ی -phred64 نیز افزوده شده است. زیرا داده‌های مثال از رمزگذاری کیفی قدیمی‌تر اِلومِنا استفاده می‌کنند. اجرای این نخستین دستور، می‌تواند چندین ساعت به طول انجامد. نکته‌ای که باید مورد توجه قرار گیرد این است که برای کاهش مصرف حافظه و زمان اجرا شدن، می‌توان تنها یک مجموعه از آماره‌های کیفی را درخواست نمود. به عنوان مثال، افزودن -graph_stats ld,gc,qd,ns,pt,ts,de سبب می‌شود که محاسبه‌ی پیچیدگی توالی و دو نوکلدوتیدی نادیده گرفته شده و تنها مضاعف‌شدگی‌های دقیق (به جای گزارش مضاعف‌شدگی‌های '5 و '3) گزارش گردند.

فایل نمودار برای ایجاد فایل html به کار گرفته می‌شود. پارامتر -o پیشوند این فایل را می‌دهد و بنابراین این دستور یک فایل QCreport.html تولید می‌نماید:

```
prinseq-graphs.pl -i graph -html_all -o QCreport
```

Trimmomatic ۳-۲-۳

Trimmomatic یک ابزار همه‌کاره‌ی مبتنی بر جاوا برای پیش‌پردازش خوانش‌ها است. از این ابزار می‌توان هم روی خط فرمان و هم از طریق رابط کاربری گرافیکی Galaxy یا Chipster استفاده نمود. Trimmomatic می‌تواند از راه‌های مختلف مبتنی بر کیفیت، آداپتورها را حذف کرده و خوانش‌ها را پیرایش نماید. همچنین این ابزار می‌تواند خوانش‌ها را بر مبنای کیفیت و طول

1- Hard-coded

پاکسازی کرده و کیفیت بازهای حاصل از سامانه‌های رمزگذاری دیگر را تبدیل کند. با استفاده از یک فرمان و از طریق فهرست کردن مراحل بر اساس ترتیب مورد نظر، می‌توان چندین مرحله را اجرا نمود. ورودی‌ها و خروجی‌ها از نوع فایل‌های FASTQ بوده که می‌توانند فشرده نیز شوند. Trimmomatic یک ابزار چند مسیری[1] بوده و بنابراین خیلی سریع اجرا می‌گردد.

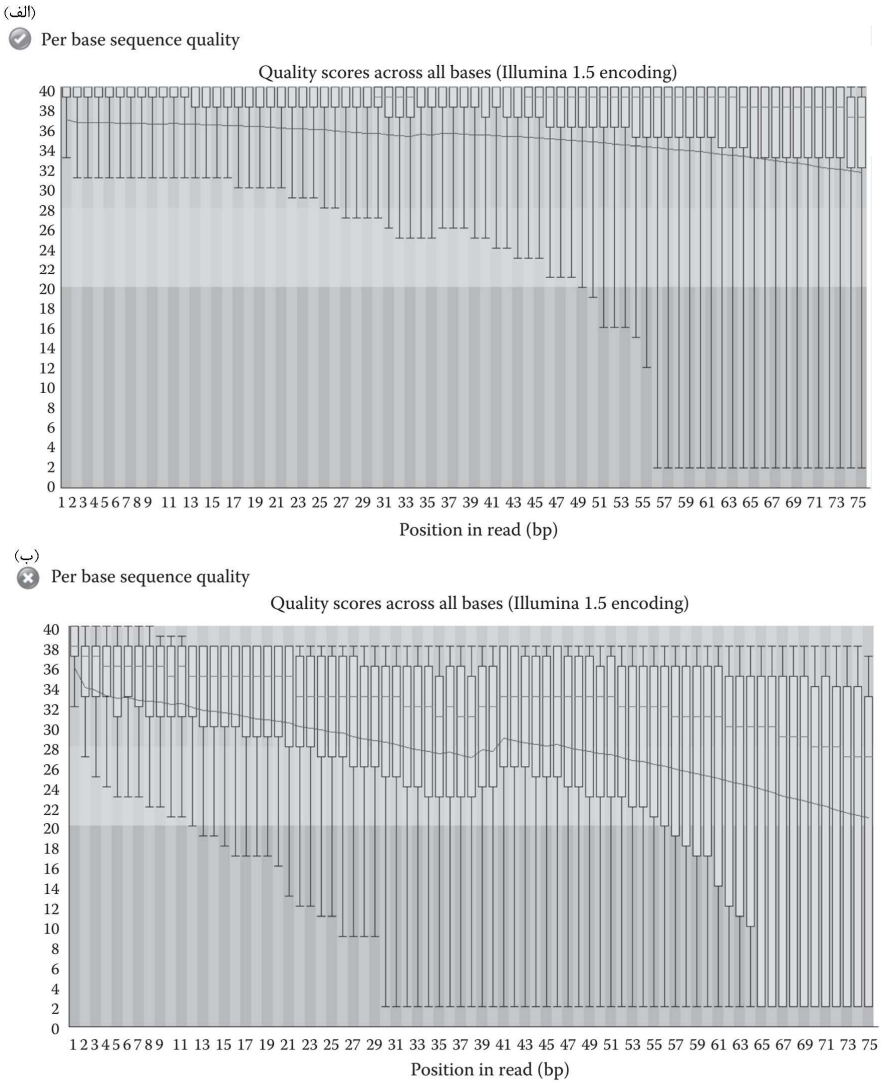
۳-۳ مسائل مرتبط با کیفیت خوانش‌ها

۳-۳-۱ کیفیت باز

کیفیت باز میزان اطمینان به خوانش باز را نشان می‌دهد. این کیفیت در مقیاس Phred ارائه می‌شود. برای محاسبه‌ی Phred ابتدا لگاریتم احتمال اشتباه بودن باز محاسبه شده و سپس عدد حاصل در ۱۰- ضرب می‌شود. به عنوان مثال، اگر شانس اشتباه بودن باز برابر با ۰/۰۱ باشد، کیفیت باز عبارت از $q = -۱۰ \times \log(۰/۰۱) = ۲۰$ خواهد بود. معمولاً مقادیر کیفیت از ۰ تا ۴۰ متغیر هستند. در فایل‌های FASTQ برای کاهش فضای ذخیره‌سازی، به جای اعداد از رمزگذاری با کمک کاراکترهای ASCII استفاده می‌شود. فایل‌های FASTQ فعلی از رمزگذاریِ Sanger استفاده می‌کنند. در این رمزگذاری برای سی و سومین کاراکتر ASCII از صفر استفاده می‌شود. لازم به ذکر است که در ویرایش‌های قبل از 1.8 نرم‌افزار اِلومِنا، فایل‌های FASTQ به گونه‌ای تولید می‌شدند که شصت و چهارمین کاراکتر ASCII صفر بوده است. برای آگاهی از جزییات سامانه‌های مختلف رمزگذاری کیفیت، می‌توان به توضیحات فرمت FASTQ مراجعه نمود (۱). اگر نمی‌دانید که رمزگذاری کیفی داده‌هایتان چیست، FastQC می‌تواند آنرا تشخیص دهد. اگر نیاز به تبدیل فایل‌های FASTQ از یک رمزگذاری کیفی به رمزگذاری کیفی دیگر است، Trimmomatic می‌تواند این کار را انجام دهد.

معمولاً مقادیر کیفیت باز در چرخه‌های بعدی توالی‌یابی کاهش می‌یابد. این موضوع به آسانی در نمودارهای جعبه‌ای که کیفیت باز را در طول خوانش‌ها نشان می‌دهند، دیده می‌شود. هر دو نرم‌افزار FastQC و PRINSEQ این نوع نمودارها را در گزارش‌های‌شان ارائه می‌دهند. نگاره‌ی ۳-۲ نمودارهای کیفیت توالی‌یابی به ازای هر باز در FastQC برای خوانش‌های جفت انتهایی داده‌های مثال (تکرار ۲ رده‌ی سلولی GM12892) را نمایش می‌دهد. همان‌گونه که در این نمودارها ملاحظه می‌شود، خوانش‌های مستقیم[2] کیفیت بالایی داشته ولی خوانش‌های معکوس[3] کیفیت پایین‌تری دارند. همچنین با نزدیک شدن به انتهای خوانش‌ها، کیفیت کاهش می‌یابد.

1- Multithreated
2- Forward
3- Reverse



نگاره‌ی ۳-۲: نمودار کیفیت توالی‌یابی به ازای هر باز از نرم‌افزار FastQC برای خوانش‌های مستقیم (بالا) و معکوس (پایین) مربوط به داده‌های مثال. این نمودار خلاصه‌ی کیفیت باز در هر موقعیت بازی در طول خوانش‌ها را نشان می‌دهد. محور y امتیازات کیفی را نشان داده و جعبه‌های زرد دامنه‌ی چارک میانی (۲۵ تا ۷۵ درصد) مقادیر کیفیت باز برای هر موقعیت بازی را نشان می‌دهد. خط قرمز نشان دهنده‌ی مقدار میانه و خط آبی نشان دهنده‌ی میانگین می‌باشد. زمینه‌های سبز، نارنجی و قرمز به ترتیب کیفیت‌های خوب، مناسب و ضعیف را نشان می‌دهند. اگر چارک پایین‌تر، کمتر از ۱۰ باشد یا اگر میانه در هر کدام از موقعیت‌های بازی کمتر از ۲۵ باشد، FastQC یک هشدار صادر می‌نماید.

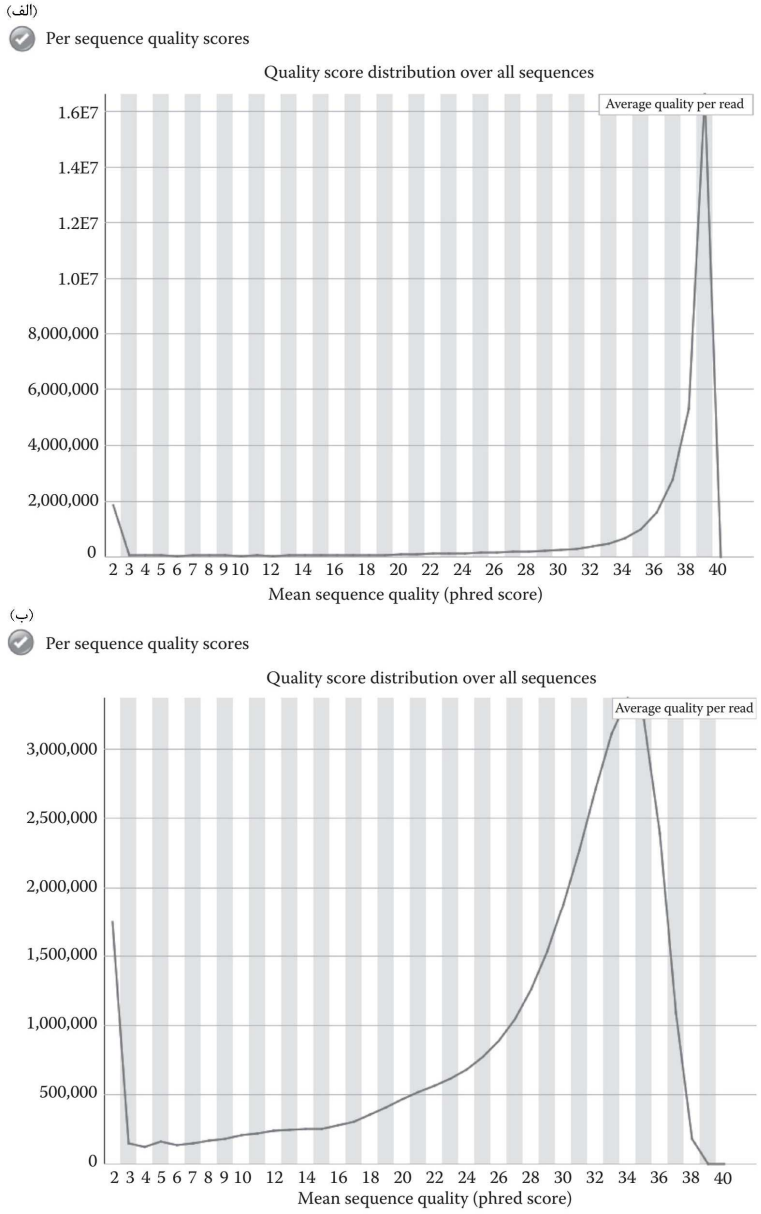
علاوه بر بررسی توزیع کیفیت‌های بازی به ازای هر موقعیت باز، بررسی چگونگی توزیع کیفیت میانگین خوانش‌ها نیز می‌تواند مفید واقع شود. با این کار می‌توان مجموعه‌ای از خوانش‌ها را مشاهده نمود که کیفیت کُلی بدی دارند. هر دو نرم‌افزار FastQC و PRINSEQ می‌توانند نمودار توزیع میانگین کیفیت‌های بازی خوانش‌ها را ارائه دهند. در حالت ایده‌آل بایستی میانگین کیفیت باز اکثریت خوانش‌ها ۲۵ یا بالاتر باشد. همان‌گونه که در نگاره‌ی ۳-۳ نشان داده شده است، هم خوانش‌های مستقیم و هم خوانش‌های معکوس حاوی حدود دو میلیون خوانش بوده که کیفیت باز آنها عموماً بد است.

خوانش‌های حاوی بازهای با کیفیت پایین را می‌توان یا پاکسازی نموده و یا پیرایش نمود. پاکسازی به معنای حذف کُل خوانش بوده ولی ویرایش تنها به معنای حذف انتهای با کیفیت پایین از خوانش‌ها است. اگر می‌خواهید خوانش‌های جفت انتهایی را پاکسازی یا پیرایش کیفی کنید، ابزاری را انتخاب نمایید که وقتی که یک خوانش (یا جفت آن) حذف می‌شود، قادر به حفظ ترتیب تطابق خوانش‌ها در فایل‌های خروجی باشد. این موضوع در زمانی که خوانش‌ها روی مرجع مکان‌یابی می‌گردند، اهمیت زیادی دارد. زیرا نرم‌افزار همردیف‌ساز انتظار دارد که خوانش‌های جفت شده را با ترتیب یکسانی در دو فایل بیابید. نکته‌ی مهم این است که علاوه بر پاکسازی، پیرایش نیز می‌تواند این ترتیب را بر هم بزند. زیرا برخی از خوانش‌ها به طور کامل پیرایش می‌شوند و سایر خوانش‌ها نیز می‌توانند آنقدر کوتاه شوند که به دلیل همین کوتاه شدن، حذف گردند.

۳-۳-۱-۱ پاکسازی

نرم‌افزارهای Trimmomatic ، FastX و PRINSEQ می‌توانند خوانش‌ها را بر مبنای کیفیت پاکسازی نمایند. ابزار پاکسازی کیفیت FastX اجازه می‌دهد که یک مقدار حداقل کیفیت و درصد بازهایی که بایستی این مقدار یا بالاتر از آنرا داشته باشند، تنظیم شوند. PRINSEQ و Trimmomatic می‌توانند پاکسازی را بر مبنای میانگین کیفیت بازی خوانش‌ها انجام داده و نکته‌ی مهم این است که این نرم‌افزارها می‌توانند با خوانش‌های جفت انتهایی کار کنند. دستور زیر در Trimmomatic خوانش‌های جفت انتهایی (PE) را که میانگین کیفیت بازشان کمتر از ۲۰ است (AVGQUAL:20) را پاکسازی می‌نماید. فایل‌های ورودی و خروجی می‌توانند فشرده شوند:

```
java -jar trimmomatic-0.32.jar PE -phred64 reads1.fastq.gz
reads2.fastq.gz paired1.fq.gz unpaired1.fq.gzpaired2.fq.gz
unpaired2.fq.gz AVGQUAL:20
```



نگاره‌ی ۳-۳: نمودار امتیازات کیفیت هر توالی از نرم‌افزار FastQC که توزیع میانگین کیفیت خوانش‌ها را نشان می‌دهد. هم خوانش‌های مستقیم (بالا) و هم خوانش‌های معکوس (پایین) در داده‌های مثال تقریباً شامل دو میلیون خوانش (۶ درصد) با میانگین کیفیت کمتر از ۲ هستند. معمولاً میانگین کیفیت خوانش‌های معکوس پایین‌تر است.

Trimmomatic بررسی می‌کند که جفت خوانش‌ها باقی مانده باشند و خوانش‌ها را به درستی در فایل‌های paired1.fq.gz و paired2.fq.gz گزارش می‌نماید. فایل‌های خروجی unpaired1.fq.gz و unpaired2.fq.gz حاوی خوانش‌هایی هستند که جفت‌های‌شان از دست رفته است. همان‌گونه که درخلاصه‌ی ارائه شده نیز نشان داده شده است، در اینجا تقریباً ۸۲ درصد از جفت‌ها باقی مانده‌اند:

```
TrimmomaticPE: Started with arguments: -phred64
reads1.fastq.gz reads2.fastq.gz paired1.fq.gz
unpaired1.fq.gz paired2.fq.gz unpaired2.fq.gz
AVGQUAL:20

Multiple cores found: Using 16 threads

Input Read Pairs: 34232081 Both Surviving:
27981021(81.74%)Forward Only Surviving:
3162984(9.24%)Reverse Only Surviving: 609823(1.78%)Dropped:
2478253(7.24%)

TrimmomaticPE: Completed successfully
```

با کمک دستور زیر می‌توان پاکسازی کیفی مشابهی را توسط PRINSEQ اجرا نمود:

```
prinseq-lite.pl -fastq reads1.fastq -fastq2 reads2.fastq -
phred64 -min_qual_mean 20 -out_good qual_filtered -out_bad
null -no_qual_header -log -verbose
```

جفت‌های باقیمانده در qual_filtered_1.fastq و qual_filtered_2.fastq گزارش گردیده و فایل‌های خروجی qual_filtered_1_singletons.fastq و qual_filtered_2_singletons.fastq نیز حاوی خوانش‌هایی هستند که جفت‌شان از دست رفته‌اند. توصیف کننده‌ی verbose- سبب می‌شود که بتوان مراحل و وقایع اجرا را دنبال کرده و آماره‌ها را مشاهده نمود (در فایل لاگ[1] نیز در دسترس است). no_qual_header- به PRINSEQ دستور می‌دهد که جهت کاهش اندازه‌ی فایل‌های نتایجِ FASTQ، به جای «read name +» به عنوان سرتیتر در هر خط کیفیت خوانش، تنها از «+» استفاده نماید.

۳-۳-۱-۲ *پیرایش*

اگر بازهای با کیفیت پایین در انتهای خوانش‌ها شناسایی شوند، ساده‌ترین راهِ حذفِ آنها، پیرایش کردن خوانش‌ها تا یک طول معین یا پیرایش کردن تعداد معینی از بازها از هر دو انتها

1- Log

است. ولی این روش توالی‌های با کیفیت خوب را نیز حذف می‌کند. برای کاهش از دست رفتن توالی‌ها می‌توان مقدار کیفی هر باز را در نظر گرفت. از انتهای '3 یا '5 خوانش کار شروع شده و اگر کیفیت یک باز پایین‌تر از حدی باشد که توسط کاربر تعریف شده است، حذف می‌گردد. پیرایشگر کیفی FastX بازها را از انتهای '3 پیرایش می‌کند. ولی PRINSEQ و Trimmomatic می‌توانند خوانش‌ها را از هر دو انتها پیرایش نمایند. چون ممکن است برخی از خوانش‌ها بسیار کوتاه شوند، معمولاً پیرایشگرها خوانش‌هایی را که از یک حداقل طول تعریف شده توسط کاربر کوتاه‌تر باشند، حذف می‌کنند. PRINSEQ ، Trimmomatic و Cutadapt از پیرایش جفت انتهایی پشتیبانی کرده و در نتیجه حتی اگر در طی فرآیند پیرایش، جفت یک خوانش نیز از دست برود، می‌توانند فایل‌های خوانش را به موازات هم حفظ نمایند.

فرمان زیر در Trimmomatic برای خوانش‌های جفت انتهایی (PE)، وقتی که کیفیت باز کمتر از ۲۰ باشد (TRAILING:20)، بازها را از انتهای '3 پیرایش کرده و خوانش‌های کوتاه‌تر از ۵۰ باز پس از پیرایش (MINLEN:50) را پاکسازی می‌نماید. ترتیب مراحل پیرایش و پاکسازی توسط این فرمان تعیین می‌شود و بنابراین فهرست کردن آنها با ترتیبی صحیح، از اهمیت زیادی برخوردار است:

```
java –jar trimmomatic–0.32.jar PE –phred64 reads1.fastq.gz
reads2.fastq.gz paired1.fq.gz unpaired1.fq.gzpaired2.fq.gz
unpaired2.fq.gz TRAILING:20 MINLEN:50
```

همان‌گونه که در خلاصه‌ی ارائه شده نیز می‌توان مشاهده کرد، در اینجا تقریباً ۸۲ درصد از جفت‌ها پس از پیرایش و سپس پاکسازی بر مبنای طول باقی‌مانده‌اند:

```
Input Read Pairs: 34232081 Both Surviving: 27992914(81.77%)
Forward Only Surviving: 3114023 (9.10%)Reverse Only
Surviving: 780195 (2.28%)Dropped: 2344949(6.85%)
```

به جای جستجوی کیفیت یک باز در یک زمان، پیرایش می‌تواند از یک روش پنجره‌ی لغزان[1] استفاده کند که در آن کیفیت باز در یک پنجره‌ی تعریف شده توسط کاربر با یک آستانه‌ی معین مقایسه می‌شود. Trimmomatic این پنجره را از ابتدای خوانش (انتهای '5) به انتهای آن می‌لغزاند. ولی PRINSEQ به این اجازه را می‌دهد که انتخاب کنید که پویش از کدام انتها شروع شود. باید توجه شود که لغزاندن این پنجره از انتهای '5 سبب می‌شود که آغاز خوانش تا زمانی که به زیر آستانه‌ی مورد نظر سقوط نکند، حفظ شود. در حالی‌که لغزاندن از انتهای '3 سبب می‌شود که با

1- Sliding window approach

رسیدن به یک پنجره‌ی با کیفیت مناسب، برش صورت گیرد. چون خوانش‌ها می‌توانند شیب‌هایی در کیفیت و در میانه داشته باشند، معمولاً لغزاندن پنجره از انتهای '5 سبب تولید خوانش‌های کوتاه‌تر می‌شود. اگر پیرایش با پاکسازی بر مبنای حداقل طول تعریف شده توسط کاربر تلفیق شود، اکثر خوانش‌ها می‌توانند حذف شوند. این مشکل با خوانش‌های جفت انتهایی تشدید می‌شود. زیرا از دست رفتن یک خوانش منجر به حذف جفتش از فایل‌های جفت شده نیز می‌شود. PRINSEQ علاوه بر جهت پویش، در سایر تنظیمات نیز انعطاف‌پذیرتر است. Trimmomatic اجازه می‌دهد که اندازه‌ی پنجره تنظیم شده و همواره از میانگین کیفیت در آن پنجره استفاده می‌کند. ولی PRINSEQ این امکان را می‌دهد که در مورد اندازه‌ی این مرحله برای حرکت پنجره تصمیم‌گیری شده و میانگین یا حداقل کیفیت بایستی با آستانه‌ی مورد نظر مقایسه شوند.

دستور زیر در Trimmomatic یک پنجره‌ی سه بازی را از انتهای '5 می‌لغزاند و وقتی که میانگین کیفیت به کمتر از ۲۰ برسد، خوانش‌ها را برش می‌دهد (SLIDINGWINDOW:3:20). همچنین خوانش‌هایی را که پس از پیرایش کوتاه‌تر از ۲۰ باز باشند (MINLEN:50)، پاکسازی می‌کند. همان‌گونه که قبلاً اشاره شد، جفت‌های باقیمانده در فایل‌های جداگانه گزارش می‌گردند.

```
java –jar trimmomatic–0.32.jar PE –phred64 reads1.fastq.gz
reads2.fastq.gz paired1.fq.gz unpaired1.fq.gzpaired2.fq.gz
unpaired2.fq.gz SLIDINGWINDOW:3:20MINLEN:50
```

همان‌گونه که خلاصه ارائه شده در زیر نیز نشان می‌دهد، ۶۴/۴ درصد از جفت‌ها پس از پیرایش و پاکسازی باقی می‌مانند. با افزایش اندازه‌ی پنجره می‌توان پیرایش را قدری ملایم‌تر نمود. به عنوان مثال، اگر پنجره هفت بازی باشد، ۷۳/۴ درصد از جفت خوانش‌ها حفظ می‌گردند.

```
Input Read Pairs: 34232081 Both Surviving: 22045360(64.40%)
Forward Only Surviving: 7811189 (22.82%)Reverse Only
Surviving: 607284 (1.77%) Dropped:3768248 (11.01%)
```

از دیدگاه مقایسه‌ای، فرمان زیر در PRINSEQ یک پنجره‌ی سه بازی را از انتهای '3 (جهت مخالف) می‌لغزاند و اگر میانگین کیفیت باز کمتر از ۲۰ باشد (1t)، خوانش‌ها را پیرایش می‌کند. همچنین خوانش‌هایی را که پس از پیرایش، کوتاه‌تر از ۵۰ باز هستند، پاکسازی می‌نماید. جفت‌های باقیمانده در فایل‌های window_1.fastq و window_2.fastq گزارش شده و فایل‌های خروجی window_1_singletons.fastq و window_2_singletons.fastq حاوی خوانش‌هایی هستند که جفت‌شان از دست رفته است.

```
prinseq-lite.pl -phred64 -trim_qual_window 3 - trim_qual_
type mean -trim_qual_right 20 -trim_qual_rule lt-fastq
reads1.fastq -fastq2 reads2.fastq -out_goodwindow -out_bad
null -verbose -min_len 50 -no_qual_header
```

بدین ترتیب همان‌گونه که در خلاصه نیز نشان داده‌می‌شود، ۲۸۱۳۳۷۸۹ (۸۱ درصد) از جفت‌ها پس از پیرایش و پاکسازی باقی می‌مانند:

```
Input and filter stats:
  Input sequences (file 1): 34,232,081
  Input bases (file 1): 2,567,406,075
  Input mean length(file 1): 75.00
  Input sequences (file 2): 34,232,081
  Input bases (file 2): 2,567,406,075
  Input mean length(file 2): 75.00
  Good sequences (pairs): 28,133,789
  Good bases (pairs): 4,220,068,350
  Good mean length(pairs): 150.00
  Good sequences (singletons file 1): 3,008,972(8.79%)
  Good bases (singletons file 1): 225,672,900
  Good mean length (singletons file 1): 75.00
  Good sequences (singletons file 2): 769,471(2.25%)
  Good bases (singletons file 2): 57,710,325
  Good mean length (singletons file 2): 75.00
  Bad sequences (file 1): 3,089,320(9.02%)
  Bad bases (file 1): 231,699,000
  Bad mean length(file 1): 75.00
  Bad sequences (file 2): 3,008,972(8.79%)
  Bad bases (file 2): 225,672,900
  Bad mean length (file 2): 75.00
  Sequences filtered by specified parameters:
  trim_qual_right: 3330145
  min_len: 50879967
```

یک جایگزین برای روش پنجره‌ی لغزان، روش مجموع اجرا[1] است که در همردیف‌ساز BWA اجرا می‌شود (۱۱) و به همین دلیل نیز اغلب پیرایش کیفیت BWA نیز نامیده می‌شود. این روش، خوانش‌ها را از سمت راست (انتهای '3) پوییده، کیفیت هر باز را با آستانه‌ای معین مقایسه کرده و همزمان با جلو رفتن، مجموع تفاوت‌ها را به دست می‌آورد. این خوانش در موقعیتی که بالاترین مقدار بدی[2] تجمعی به دست آید، ویرایش می‌شود. این روش در ابزار Cutadapt اجرا می‌گردد.

1- Running sum method
2- Badness

Trimmomatic یک روش تطبیقی برای پیرایش کیفیت ارائه می‌کند که اصطلاحاً MAXINFO نامیده می‌شود. هدف از اجرای این روش، برقراری تعادل بین نگه داشتن خوانش‌های تا حد ممکن بلند با حذف بازهای خطا است. در این روش، دو پارامتر طول خوانش هدف[1] و سخت‌گیری[2] دریافت شده و خوانش‌ها از انتهای '3 پیرایش شده و در هر باز یک امتیاز محاسبه می‌گردد. اگر یک خوانش کوتاه‌تر از طول هدف باشد، یک تاوان[3] داده می‌شود. برای خوانش‌های بلندتر، این پنالتی حاصل از احتمال خطا، افزایش یافته و در نهایت از پاداش[4] نگه داشتن بازهای اضافی فراتر می‌رود. این بالانس را می‌توان با استفاده از پارامتر سخت‌گیری کنترل نمود. مقدار پارامتر سخت‌گیری بین صفر و یک بوده و هر قدر مقدار آن بالاتر باشد، خوانش‌ها صحیح‌تر خواهند بود. فرمان پیرایش زیر در MAXINFO مقدار طول هدف را برابر با ۵۰ و مقدار سخت‌گیری را برابر با ۰/۷ در نظر می‌گیرد.

```
java –jar trimmomatic–0.32.jar PE –phred64 reads1.fastq.gz
reads2.fastq.gz paired1.fq.gz unpaired1.fq.gzpaired2.fq.gz
unpaired2.fq.gz MAXINFO:50:0.7 MINLEN:50
```

در این پیرایش و پاکسازی، تقریباً ۹۹ درصد از جفت خوانش‌ها باقی می‌مانند:

```
Input Read Pairs: 34232081 Both Surviving: 33724880(98.52%)
Forward Only Surviving: 63886 (0.19%)ReverseOnly Surviving:
4564(0.01%) Dropped: 438751 (1.28%)
```

اگر پارامتر سخت‌گیری به ۰/۸ افزایش یابد، صحت خوانش‌ها بر طول آنها برتری یافته و در نتیجه درصد جفت‌های باقیمانده به ۸۲ درصد کاهش می‌یابد:

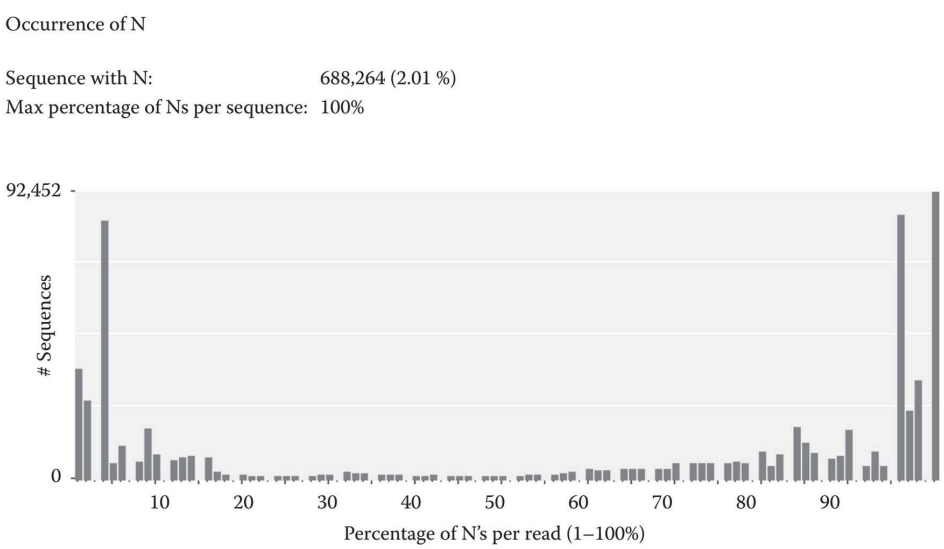
```
Input Read Pairs: 34232081 Both Surviving: 27993319
(81.78%)Forward Only Surviving: 3113077 (9.09%) ReverseOnly
Surviving: 780359(2.28%) Dropped: 2345326 (6.85%)
```

۳-۳-۲ بازهای مبهم

اگر یک باز در طی توالی‌یابی تشخیص داده نشود، در خوانش با N نشان داده می‌شود. اسمبل‌سازها و ردیف‌سازها راه‌های مختلفی برای برخورد با بازهای مبهم دارند. برخی از آنها N ها را با یک باز تصادفی جایگزین کرده و برخی دیگر آنها را با یک باز ثابت جایگزین می‌نمایند. چون

1- Target read length
2- Strictness
3- Penalty
4- Bonus

N ها می‌توانند منجر به اسمبل‌های اشتباهی یا مکان‌یابی غلط شوند، لذا خوانش‌های دارای تعداد زیادی N بایستی حذف شوند. گزارش کیفیت PRINSEQ شامل یک نمودار از وقوع N ها است (نگاره‌ی ۳-۴) که می‌توان با نگاه کردن به آن درصد N به ازای هر خوانش را مشاهده نمود.



نگاره‌ی ۳-۴: گزارش PRINSEQ از وقوع بازهای مبهم (N ها). دو درصد از بازها حاوی N ها بوده و تنها ۹۲۴۵۲ خوانش حاوی N هستند.

فرآیند پاکسازی درPRINSEQ این امکان را فراهم می‌آورد که بتوان حداکثر تعداد یا درصد N هایی را که یک خوانش مجاز است داشته باشد، تعیین نمود. فرمان زیر خوانش‌های جفت انتهایی را که بیش از دو N دارند، حذف و پاکسازی می‌نماید.

```
prinseq-lite.pl -fastq reads1.fastq -fastq2 reads2.fastq -
ns_max_n 2 -out_good nfiltered -out_bad null-no_qual_header
-log -verbose
```

جفت‌های باقیمانده در فایل‌های `nfiltered_1.fastq` و `nfiltered_2.fastq` گزارش شده و فایل‌های خروجی `nfiltered_1_singletons.fastq` و `nfiltered_2 _singletons.fastq` حاوی خوانش‌هایی هستند که جفت‌شان را از دست داده‌اند. همان‌گونه که در فایل لاگ نیز نشان داده می‌شود، ۳۳۵۴۶۹۰۶ جفت پس از پاکسازی باقی مانده‌اند:

```
Input sequences (file 1): 34,232,081
Input bases (file 1): 2,567,406,075
Input mean length (file 1): 75.00
Input sequences (file 2): 34,232,081
Input bases (file 2): 2,567,406,075
Input mean length(file 2): 75.00
Good sequences (pairs): 33,546,906
Good bases (pairs): 5,032,035,900
Good mean length (pairs): 150.00
Good sequences (singletons file 1): 58,095(0.17%)
Good bases(singletons file 1): 4,357,125
Good mean length (singletons file 1): 75.00
Good sequences (singletons file 2): 141,443(0.41%)
Good bases (singletons file 2): 10,608,225
Good mean length(singletons file 2): 75.00
Bad sequences (file 1): 627,080(1.83%)
Bad bases (file 1): 47,031,000
Bad mean length(file 1): 75.00
Bad sequences (file 2): 58,095(0.17%)
Bad bases (file 2): 4,357,125
Bad mean length(file 2): 75.00
Sequences filtered by specified parameters:
ns_max_n: 1170812
```

۳-۳-۳ آداپتورها

همان‌گونه که در فصل اول تشریح گردید، دستورالعمل‌های اِلومِنا و رُش ۴۵۴ از آداپتورهای توالی‌یابی استفاده می‌کنند که لازم است پیش از آنالیز داده‌ها پیرایش و حذف گردند. همچنین لازم است که سایر نشان‌ها[1] نظیر شناساگرهای چند عضوی و آغازگرها حذف گردند. علی‌رغم اینکه این کار ساده به نظر می‌رسد ولی با چالش‌هایی همراه است. نخست اینکه این نشان‌ها مانند هر بخش دیگری از خوانش‌ها می‌توانند خطاهایی در توالی‌یابی داشته و در نتیجه نرم‌افزارهای پیرایش بایستی قادر به مواجهه‌ی با عدم تطابق‌ها، ایندل‌ها[2] و بازهای مبهم باشند. دوم اینکه وقتی که RNA های کوچک توالی‌یابی می‌گردند، خوانش‌ها می‌توانند به آداپتور '3 افزوده شوند. این وضعیت را اصطلاحاً خوانش سراسری[3] می‌نامند. مشکل این خوانش سراسری آن است که آداپتور در انتهای '3 می‌تواند جزیی باشد و بنابراین تشخیص آن سخت است. سوم اینکه اگر داده‌ها از یک پایگاه داده‌ی عمومی گرفته شده باشند، ممکن است که اطلاعات توالی آداپتور در دسترس نباشد.

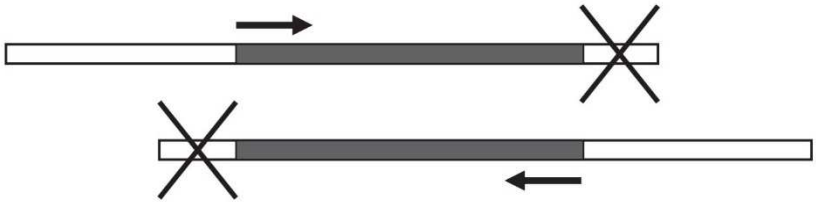
1- Tag
2- Indel
3- Read-through

اگر توالی آداپتور معلوم نباشد، TagCleaner می‌تواند آنرا پیش‌بینی نماید (۱۲). همچنین نمودار k-mer overrepresentation نرم‌افزار FastQC و نمودار TagSequence Check نرم‌افزار PRINSEQ نیز امکان تشخیص حضور آداپتورها را فراهم می‌آورند. ابزارهای موجود برای حذف آداپتورها شامل Trimmomatic ، FastX ، TagCleaner و Cutadapt هستند. در بین این نرم‌افزارها Trimmomatic، TagCleaner و Cutadapt می‌توانند با عدم تطابق‌ها مقابله کرده، آداپتورها را در هر دو انتها پیرایش کرده و به کاربر این امکان را بدهند که حداقل همپوشانی خوانش و توالی نشان را در نظر بگیرد. همچنین TagCleaner می‌تواند با ایندل‌ها و بازهای مبهم نیز مقابله نماید. Trimmomatic سریع است. زیرا نخست خوانش‌ها را با توالی‌های کوتاه سید[1] پویش کرده و همردیفی کامل را تنها برای خوانش‌هایی که به خوبی با این سیدها تطابق داشته باشند، انجام می‌دهد. این نرم‌افزار از جفت انتهایی‌ها پشتیبانی کرده و می‌تواند اطلاعات جفت انتهایی را نیز برای شناسایی آداپتورها استفاده کند. این روش که اصطلاحاً روش پالیندروم[2] نامیده می‌شود، بر مبنای این حقیقت استوار است که خوانش سراسری در هر دو جهت صورت گرفته و این قطعه کاملاً توالی‌یابی می‌گردد. بدین ترتیب خوانش‌ها می‌توانند همردیف شده و اجازه دهند که تشخیص آداپتورهای جزیی حتی با دقت یک باز انجام گیرد (نگاره‌ی ۳-۵). نکته‌ای که باید مورد توجه قرار گیرد این است که اگر پیرایش عمومی، کیفی و آداپتور در یک دستور Trimmomatic ترکیب شود، بایستی نخستین مرحله از پیرایش را به آداپتور تخصیص داد. زیرا شناسایی کُل آداپتورها آسان‌تر از آداپتورهای جزیی است.

داده‌های مثال قبلاً برای آداپتور پیرایش شده‌اند. ولی دستور زیر آداپتورهای TruSeq2 اِلومِنا را از خوانش‌های جفت انتهایی حذف می‌نماید. در این دستور، اجازه‌ی دو عدم تطابق در سید، آستانه‌ی گیرایی[3] پالیندروم برابر با ۳۰، آستانه‌ی گیرایی ساده برابر با ۱۰ و حداقل طول آداپتور شناسایی شده توسط حالت پالیندروم برابر با ۱ در نظر گرفته شده و خوانش معکوس که به صورت پیش‌فرض حذف می‌شود، حفظ گردیده است.

```
java -jar trimmomatic-0.32.jar PE -phred64 reads1.fastq.gz
reads2.fastq.gz paired1.fq.gz unpaired1.fq.gzpaired2.fq.gz
unpaired2.fq.gz ILLUMINACLIP:TruSeq2-PE.fa:2:30:10:1:true
```

1- Short seed sequence
2- Palindrome approach
3- Clip



نگاره‌ی ۳-۵: روش پالیندروم نرم‌افزار Trimmomatic می‌تواند حتی آداپتورهای جزیی بسیار کوتاه را نیز در خوانش‌های جفت انتهایی در یک وضعیت خوانش سراسری تشخیص دهد. دو خوانش جفت انتهایی همردیف می‌شوند (خوانش مستقیم در بالا و خوانش معکوس در پایین قرار دارد). آداپتورها با رنگ سفید مشخص شده و به بخش توالی‌یابی شده که با رنگ سیاه مشخص گردیده است، الحاق شده‌اند. وقتی که این بخش الحاقی کوتاه باشد، توالی‌یابی خوانش سراسری آن به انتهای '3 ، منجر به یک آداپتور جزیی (نه کامل) در آن انتها می‌گردد. Trimmomatic می‌تواند این آداپتورهای کوتاه را که روی نگاره‌ی فوق با علامت ضربدر مشخص شده‌اند، تشخیص داده و حذف نماید.

۳-۳-۴ طول خوانش‌ها

کنترل توزیع طول خوانش‌ها به عنوان بخشی از کنترل کیفیت، کار مفیدی است. این کنترل برای خوانش‌های اِلومِنا که اساساً طول‌های غیریکنواخت دارند، نیز به کار گرفته می‌شود. زیرا پیرایش بر مبنای کیفیت یا آداپتورها می‌تواند منجر به تولید قطعات بسیار کوتاه گردد. هر دو نرم‌افزار FastQC و PRINSEQ نمودارهای توزیع طول خوانش‌ها را ارائه می‌دهند. اکثر ابزارهای پیرایش شامل Trimmomatic ، PRINSEQ ، Cutadapt و FastX Quality Trimmer احتمال پاکسازی را بر مبنای طول خوانش ارائه می‌نمایند. حداقل طول مورد نیاز به کاربردهای پایین‌دستی بستگی دارد. مکان‌یابی بدون ابهام روی ژنوم برای خوانش‌های بسیار کوتاه سخت بوده و خوانش‌های بلندتر نیز برای اسمبل کردن و کمی‌سازی ایزوفرم‌های پیوندی مفیدتر هستند.

۳-۳-۵ اُریبی مختص توالی و عدم تطابق‌های ناشی از آغازگرهای شش نوکلئوتیدی تصادفی

در طی تهیه‌ی کتابخانه، RNA قطعه قطعه شده و آغازگرهای شش نوکلئوتیدی تصادفی جهت شروع رونویسی معکوس برای تولید cDNA به کار گرفته شده و سرانجام این انتهاها توالی‌یابی می‌شوند. بنابراین انتظار می‌رود که خوانش‌ها در نواحی تصادفی در طول رونوشت‌ها شروع شده و متعاقب آن نبایستی هیچ ترکیب بازی در طول خوانش‌ها اُریب باشند. ولی نشان داده شده است که

استفاده از آغازگرهای شش نوکلئوتیدی تصادفی منجر به اُریبی در ترکیب نوکلئوتیدها در ابتدای خوانش‌های توالی‌یابی RNA می‌گردد (۱۳). این اُریبیِ مختصِ توالی بر برآوردهای بیان ژن‌ها و ایزوفرم‌ها تاثیر گذاشته و در نتیجه‌ی آن پوشش در طول رونوشت‌ها یکنواخت نخواهد بود. اُریبی زیاد در موقعیت‌های بازی مختلف می‌تواند نشانه‌ای از آداپتورهای پیرایش نشده یا یک توالی فزون‌نمایی شده در کتابخانه باشد.

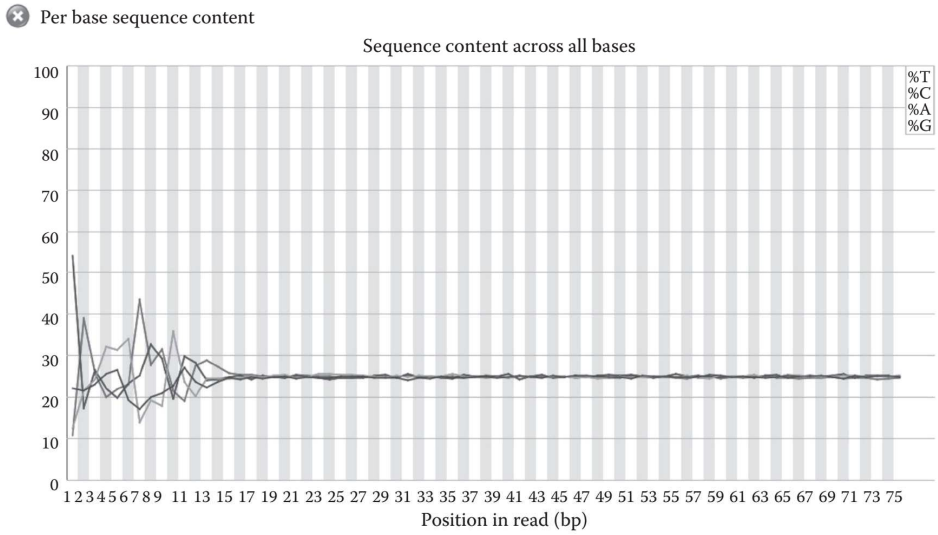
نرم‌افزار FastQC نمودار ترکیب بازی در طول خوانش‌ها (نگاره‌ی ۳-۶) را ترسیم می‌کند. اگر مقدار هر باز مشابه مقدار آن در جاندار مورد نظر باشد، این نمودار بایستی یک خط صاف باشد. اگر تفاوت بین A و T یا G و C بزرگ‌تر از یک درصد در هر موقعیت خوانش باشد، FastQC یک هشدار صادر می‌کند.

اُریبیِ مختصِ توالی نمی‌تواند با پاکسازی یا پیرایش حذف شود. ولی بسته‌های نرم‌افزاری Cufflinks و eXpress که در فصل ششم معرفی می‌شوند، تصحیحی را ارائه می‌نمایند که در آن توالی‌های منتخب حاصل از داده‌ها را آموخته و این اطلاعات را در برآورد فراوانی در نظر می‌گیرند (۱۴).

نکته‌ای که باید مورد توجه قرار گیرد این است که علاوه بر اُریبیِ مختصِ توالی، آغازگرهای شش نوکلئوتیدی تصادفی سبب عدم تطابق‌هایی در ابتدای خوانش‌های توالی‌یابی RNA اِلومِنا می‌شوند (۱۵). بالاترین نرخ عدم تطابق در نخستین نوکلئوتید دیده شده و بالغ بر هفت نوکلئوتید نیز می‌توانند تحت تاثیر واقع شوند. معمولاً بازهایی که عدم تطابق دارند، مقادیر کیفیت باز خوبی داشته و بنابراین نمی‌توانند از این طریق شناسایی گردند. جهت اجتناب از این مشکل، می‌توان توسط Trimmomatic یا PRINSEQ نخستین باز(ها) را پیرایش و حذف نمود.

۳-۳-۶ محتوای GC

محتوای GC خوانش‌ها بایستی از توزیع نرمال پیروی کرده و روی محتوای GC آن جاندار متمرکز باشد. یک توزیع با شکل نامعمول یا یک انحراف بزرگ نسبت به محتوای GC ژنوم جاندار مورد نظر، می‌تواند نشان دهنده‌ی آلودگی کتابخانه که در بخش‌های بعدی همین فصل مورد بحث و بررسی قرار می‌گیرد، باشد. هر دو نرم‌افزار FastQC و PRINSEQ توزیع میانگین GC خوانش‌ها را روی نمودار نشان می‌دهند. علاوه بر این، FastQC محتوای GC به ازای هر موقعیت بازی را روی نمودار نمایش می‌دهد. این نمودار بایستی یک خط راست در سطح محتوای GC ژنوم جاندار مورد نظر ایجاد کند. محتوای GC مختلف در یک موقعیت بازی معین نشان دهنده‌ی حضور یک توالی فزون‌نمایی شده در آن کتابخانه است. اُریبیِ مختصِ توالی که قبلاً مورد بحث و بررسی قرار گرفت، نیز در نمودار محتوای GC نمایش داده می‌شود.



نگاره‌ی ۳-۶: نمودار محتوای توالی به ازای هر باز که توسط FastQC تولید شده است. محور y نشان دهنده‌ی درصد هر نوکلئوتید است. موقعیت ۱۳ باز نخست نشان دهنده‌ی اُریبی مختص توالی برای خوانش‌های توالی‌یابی RNA اِلومِنا است.

نکته‌ای که باید مورد توجه واقع شود این است که روش‌های استاندارد تهیه‌ی کتابخانه که از تکثیر PCR بهره می‌گیرند، با مناطق غنی و فقیر از GC معارضه می‌نمایند. شاید این موضوع قدری برخلاف انتظار باشد ولی محتوای GC می‌تواند مختص نمونه باشد. این موضوع سبب پیچیدگی آنالیز افتراقی بیان می‌شود (۱۶). اُریبی GC را نمی‌توان با استفاده از پیش‌پردازش برطرف نمود. ولی روش‌های مختلف نرمال‌سازی و تصحیح برای مراحل مختلف آنالیز پیشنهاد شده است (۱۴ و ۱۶).

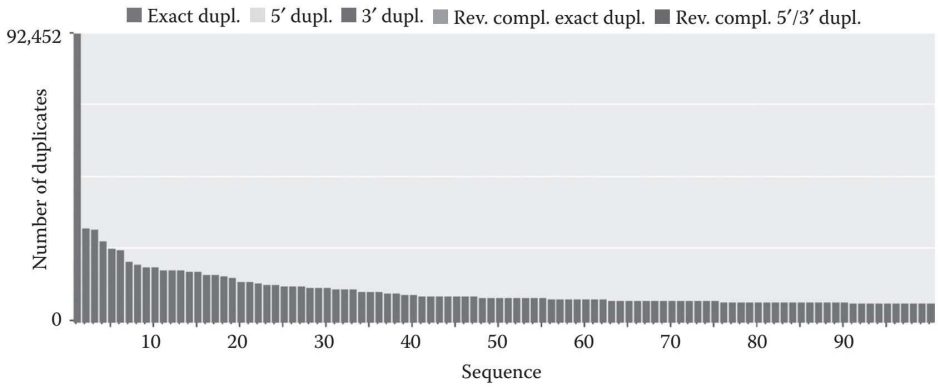
۳-۳-۷ مضاعف‌شدگی‌ها

همان‌گونه که قبلاً نیز بحث شد، خوانش‌های انتهای قطعات تصادفی بوده و در نتیجه بیشترین خوانش‌ها بایستی منحصر به فرد باشند. برای کاربردهای توالی‌یابی نسل جدید، بالا بودن سطح خوانش‌های یکسان می‌تواند نشان دهنده‌ی فزون تکثیری[1] PCR باشد. ولی در زمینه‌ی توالی‌یابی RNA مضاعف‌شدگی‌ها اغلب نتیجه‌ی طبیعی توالی‌یابی رونوشت‌های با بیان بالا هستند. برای آنالیز افتراقی بیان توصیه نمی‌شود که مضاعف‌شدگی‌ها حذف شوند. زیرا این مضاعف شدگی‌ها دامنه‌ی

1- Overamplification

پویایی را هموار کرده و در نتیجه شمارش خوانش‌ها دیگر متناسب با سطح بیان نخواهد بود. ولی اگر یک رونوشت در یک موقعیت، دارای یک برآمدگی شدید باشد، این موضوع احتمالاً نشان دهنده‌ی یک ورساختگی در PCR است.

هر دو نرم‌افزار FastQC و PRINSEQ آنالیز مضاعف‌شدگی را انجام می‌دهند. FsatQC می‌تواند تنها مضاف‌شدگی‌های دقیق را تشخیص دهد. ولی PRINSEQ می‌تواند مضاعف‌شدگی‌های انتهای 5' و 3' را نیز شناسایی کند. بر مبنای گزارش کیفی PRINSEQ ، ۵۱/۴ درصد از خوانش‌های مستقیم در داده‌های مثال، مضاعف‌شدگی دقیق داشتند. PRINSEQ نموداری را ارائه می‌دهد که تعداد مضاعف‌شدگی‌های ۱۰۰ خوانشی که بیشترین تعداد مضاعف‌شدگی را دارند، نمایش می‌دهد. این نمودار نشان می‌دهد که آیا خوانش‌های زیادی با سطح پایین مضاعف‌شدگی وجود دارند یا اینکه خوانش‌های زیادی مضاعف شده‌اند (نگاره‌ی ۳-۷).



نگاره‌ی ۳-۷: استخراج گزارش مضاعف‌شدگی PRINSEQ که تعداد مضاعف‌شدگی‌برای ۱۰۰ خوانش با بیشترین مضاعف‌شدگی را نشان می‌دهد. خوانشِ دارای بیشترین مضاعف‌شدگی،۹۲۴۵۲ کُپی دارد.

پاکسازی در PRINSEQ این امکان را به کاربر می‌دهد که تعداد مضاعف‌شدگی‌های مجاز یک خوانش را تعیین کند. نرم‌افزار FastX Collapser خوانش‌های یکسان را تلفیق کرده تا یک خوانش واحد به دست آمده و تعداد خوانش‌ها را حفظ نماید. این ابزارها روی خوانش‌های خام کار کرده و مضاعف‌شدگی‌ها را بر مبنای توالی شناسایی می‌کنند. باید توجه نمود که در واقع خوانش‌های حاصل از مضاعف‌شدگی‌های PCR قطعه‌ی همسان نمی‌تواند به واسطه‌ی خطاهای توالی‌یابی، توالی یکسانی داشته باشد. بنابراین روش‌های مبتنی بر توالی‌یابی می‌توانند زیربرآوردی[1] از حجم

1- Underestimate

مضاعف‌شدگی‌ها ارائه دهند. اگر خوانش‌ها قبلاً با یک مرجع همردیف شده باشند، مضاعف‌شدگی‌ها می‌توانند به جای محتوای توالی، بر مبنای موقعیت یکسان مکان‌یابی تشخیص داده شوند. جعبه ابزارهای مورد استفاده برای همردیف‌سازی خوانش‌ها در فصل بعد مورد بحث و بررسی واقع می‌شوند. این ابزارها از مختصات بیرونی مکان‌یابی ژنومی خوانش‌های جفت انتهایی به عنوان شاخص قطعات یکسان استفاده می‌کنند. البته این شاخص برای خوانش‌های توالی‌یابی RNA ایده‌آل نیست. زیرا خوانش‌های حاصل از رونوشت‌های جایگزین می‌توانند مختصات بیرونی ژنومی یکسانی داشته ولی به واسطه‌ی پرش اگزونی، محتوای متفاوتی داشته باشند.

اگر لازم است که مضاعف‌شدگی‌های خوانش‌های خام حذف شوند، می‌توان از بسته‌ی نرم‌افزاری PRINSEQ استفاده نمود. فرمان زیر خوانش‌های مضاعف دقیق (derep1-) را که بیش از ۱۰۰ بار تکرار شده باشند (derep _ min 101-)، حذف می‌نماید.

```
prinseq-lite.pl -fastq reads1.fastq -fastq2 reads2.fastq -
derep 1 -derep_min 101 -log -verbose -out_gooddupfiltered -
out_bad null -no_qual_header
```

جفت‌های باقیمانده در فایل‌های dupfiltered_1.fastq و dupfiltered_2.fastq گزارش گردیده و فایل‌های خروجی dupfiltered_1_singletons.fastq و dupfiltered_2_singletons.fastq حاوی خوانش‌هایی هستند که جفت‌هایشان را از دست داده‌اند. همان‌گونه که در لاگ نیز گزارش شده است، این فرمان ۸۰۸۲۹۵ از خوانش‌ها (۲/۴ درصد) را حذف کرده و ۳۳۴۲۳۷۸۶ از جفت‌ها (۹۶/۷ درصد) را حفظ می‌نماید:

```
Input sequences (file 1): 34,232,081
Input bases (file 1): 2,567,406,075
Input mean length (file 1): 75.00
Input sequences (file 2): 34,232,081
Input bases (file 2): 2,567,406,075
Input mean length (file 2): 75.00
Good sequences (pairs): 33,423,786
Good bases (pairs): 5,013,567,900
Good mean length(pairs): 150.00
Good sequences (singletons file 1): 0(0.00%)
Good sequences (singletons file 2): 0(0.00%)
Bad sequences (file 1): 808,295 (2.36%)
Bad bases (file 1): 60,622,125
Bad mean length(file 1): 75.00
Bad sequences (file 2): 0(0.00%)
Sequences filtered by specified parameters:
derep: 808295
```

۳-۳-۸ آلودگی توالی

اگر بدشانس باشید، ممکن است خوانش‌های شما حاوی توالی‌های مربوط به جانداران یا حاملین آلاینده باشد. همان‌گونه که قبلاً نیز اشاره شد، این موضوع در توزیع محتوای GC نشان داده می‌شود. نمودار فراوانی دو نوکلئوتیدی PRINSEQ می‌تواند سرنخ‌هایی از این آلودگی‌ها ارائه دهد. ولی احتمالاً مستقیم‌ترین راه همردیف‌سازی خوانش‌ها با توالی‌های حاصل از آلاینده‌های ممکن است. ابزار FastQ Screen به کمک همردیف‌ساز Bowtie ، خوانش‌ها را با موارد مشکوکی که توسط کاربر تعریف شده است، مکان‌یابی نموده و نتایج را در هر دو قالب متنی و گرافیکی تخلیص می‌نماید (۱۷). به جای این روش می‌توان یک زیرمجموعه‌ی تصادفی از خوانش‌ها را با یک پایگاه داده‌ی نوکلئوتیدی عمومی، BLAST نمود.

۳-۳-۹ توالی‌های با پیچیدگی پایین و دُم‌های پُلی A

توالی‌های با پیچیدگی پایین[1] محتوای اطلاعات محدودی داشته و لذا مکان‌یابی صحیح و قابل اعتماد آنها به مرجع، مشکل است. به عنوان مثال، این توالی‌ها می‌توانند شامل تکرارهای هموپلیمر (نظیر: AAAAAAAAAA)، دو نوکئوتیدی (نظیر: CACACACACA) یا سه نوکلئوتیدی (نظیر: CATCATCATCAT) باشند. PRINSEQ پیچیدگی توالی خوانش را گزارش کرده و این محاسبه را با دو روش DUST و انتروپی[2] انجام می‌دهد. امتیازات DUST از صفر تا ۱۰۰ تغییر کرده و بالاترین امتیازات نیز متعلق به هموپلیمرها است. یک خوانش با امتیاز DUST بالاتر از ۷ به عنوان یک خوانش با پیچیدگی پایین تلقی می‌گردد. امتیازات انتروپی برعکس بوده و در نتیجه مقدار انتروپی هموپلیمرها برابر با صفر است. هر توالی که امتیاز انتروپی‌اش کمتر از ۵۰ باشد، به عنوان توالی با پیچیدگی پایین محسوب می‌شود. می‌توان خوانش‌ها را با استفاده از PRINSEQ و با کمک گزینه‌ی `-lc-threshold` برای پایین بودن پیچیدگی پاکسازی نمود. همچنین باید روش مورد استفاده در این گزینه مشخص شود: `-lc-method (dust/entropy)`

دُم‌های پُلی A/T تکرارهای A یا T در انتهای خوانش‌ها هستند. PRINSEQ تعداد خوانش‌های حاوی دُم‌های پُلی A/T پنج بازی یا بیشتر و توزیع طول دُم‌ها را گزارش می‌نماید. این دُم‌ها را می‌توان با دادن حداقل طول دُم به گزینه‌ی `-trim_tail_right` (یا `-trim_tail_left`) پیرایش نمود.

1- Low-compexity sequence
2- Entropy

کنترل کیفیت و پیش‌پردازش در Chipster

- می‌توان گزارشات کیفی را با استفاده از FastQC ، PRINSEQ و FastX و با کمک ابزارهای موجود در دسته‌ی Quality control ایجاد نمود. به عنوان مثال، می‌توان فایل FASTQ را انتخاب کرده، سپس ابزار Quality control/read quality with FastQC را انتخاب نموده و نهایتاً روی Run کلیک نمود.
- ابزار PRINSEQ در دسته‌ی Preprocessing برای پاکسازی بر مبنای کیفیت، N ها، محتوای GC، پایین بودن پیچیدگی، طول و مضاعف‌شدگی‌ها در دسترس است. اگر خوانش‌های جفت انتهایی داشته باشید، هر دو فایل را به طور همزمان به نرم‌افزار داده و در پنل پارامتر، تخصیص صحیح خوانش‌های مستقیم و معکوس را کنترل کنید.
- در دسته‌ی Preprocecing ، ابزارهای پیرایش PRINSEQ ، Trimmomatic ، FastX و TagCleaner برای حذف بازهای با کیفیت پایین، آداپتورها و دُم‌های پُلی A پیشنهاد شده است. همچنین این دسته شامل ابزارهای Predict adapters و Sattistics for adapters که مبتنی بر TagCleaner می‌باشند، است. فایل FASTQ مورد نظر را انتخاب کرده، پارامترها را تنظیم کرده و روی Run کلیک کنید. اگر خوانش‌های جفت انتهایی دارید، هر دو فایل را به طور همزمان به نرم‌افزار بدهید و در پنل پارامتر، تخصیص صحیح خوانش‌های مستقیم و معکوس را کنترل نمایید.

۳ –۴ خلاصه

به طور کلی خوانش‌ها می‌توانند مسائل و مشکلات کیفی زیادی داشته باشند که اهمیت آنها نیز متفاوت است. در حالی که لازم است آداپتورها و آلودگی‌های توالی حذف شوند، مشکلات کیفیت باز می‌تواند نامحسوس‌تر بوده و مضاعف‌شدگی‌ها تنها نرمال‌سازی گردند. نیازهای کیفی نیز به استفاده‌ی بعدی از خوانش‌ها بستگی داشته و لذا هیچ قانون کلی برای کیفیت وجود ندارد که با همه‌ی وضعیت‌ها و موقعیت‌ها انطباق داشته باشد. به عنوان مثال، اُریبی مختص توالی و اُریبی GC ، برآورد فراوانی ایزوفرم‌ها و آنالیز افتراقی بیان را مختل کرده، طول خوانش برای اسمبل کردن از نو و شناسایی ایزوفرم بیش از آنالیز افتراقی بیان اهمیت داشته و توانایی همردیف‌سازها در مواجهه‌ی با بازهای خطا متفاوت است.

در حال حاضر هیچ نوع اجماعی بر سر این موضوع که آستانه‌ی بهینه‌ی کیفیت بازی برای پیرایش در زمینه‌ی توالی‌یابی RNA چقدر است، وجود ندارد. پیرایش بازهای با کیفیت پایین

اسمبل کردن از نو و همردیف‌سازی خوانش‌ها با یک توالی مرجع را بهبود بخشیده ولی پوشش را کاهش داده است. زیرا خوانش‌های پیرایش شده کوتاه‌تر و کمتر هستند. بنابراین انتخاب یک آستانه‌ی کیفی نوعی تبادل بینابینی است. هر چند که هنوز جای یک مطالعه‌ی همه جانبه روی اثرات مختلف پایین‌دستی پیرایش خالی است، ولی اخیراً یک گزارش نشان داده است که یک پیرایش ظریف با استفاده از آستانه‌ی ۲ تا ۵ برای اسمبل کردن ترانسکریپتوم از نو، بهینه است (۱۸). یک مطالعه‌ی عمومی‌تر نیز نشان داه است که خوانش‌های پیرایش شده با استفاده از آستانه‌ی کیفی بین ۲۰ و ۳۰، درصد همردیفی بالاتری داشته ولی تعداد کُل خوانش‌ها بسیار کاهش می‌یابد (۱۹).

منابع

1. FASTQ format description. Available from: http://en.wikipedia.org/wiki/FASTQ_format.
2. FastQC. Available from: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
3. Schmieder, R. and Edwards, R. Quality control and preprocessing of metagenomic datasets. Bioinformatics, 27(6):863–864, 2011.
4. Bolger, A.M., Lohse, M., and Usadel, B. Trimmomatic: A flexible trimmer for Illumina sequence data. Bioinformatics, 2014, doi: 10.1093/bioinformatics/btu170.
5. Martin, M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J, 17:10–12, 2011.
6. FASTX-toolkit. Available from: http://hannonlab.cshl.edu/fastx_toolkit /index .html.
7. Goecks, J., Nekrutenko, A., and Taylor, J. Galaxy: A comprehensive approachfor supporting accessible, reproducible, and transparent computationalresearch in the life sciences. Genome Biol, 11(8):R86, 2010.
8. Kallio, M.A., Tuimala, J.T., Hupponen, T. et al. Chipster: User-friendly analysis sofware for microarray and other high-throughput data. BMC Genomics, 12:507, 2011.
9. Li, H., Handsaker, B., Wysoker, A. et al. Te sequence alignment/Map format and SAM tools. Bioinformatics, 25(16):2078–2079, 2009.
10. PRINSEQ web application. Available from: http://edwards.sdsu.edu/cgi-bin/prinseq/prinseq.cgi.
11. Li, H. and Durbin, R. Fast and accurate long-read alignment with Burrows–Wheeler transform. Bioinformatics, 26(5):589–595, 2010.
12. Schmieder, R., Lim, Y.W., Rohwer, F., and Edwards, R. TagCleaner:Identifcation and removal of tag sequences from genomic and metagenomics datasets. BMC Bioinformatics, 11:341, 2010.
13. Hansen, K.D., Brenner, S.E., and Dudoit, S. Biases in Illumina transcriptome sequencing caused by random hexamer priming. Nucleic Acids Res,38(12):e131, 2010.

14. Roberts, A., Trapnell, C., Donaghey, J., Rinn, J.L., and Pachter, L. Improving RNA-seq expression estimates by correcting for fragment bias. Genome Biol, 12(3):R22, 2011.
15. van Gurp, T.P., McIntyre, L.M., and Verhoeven, K.J. Consistent errors in frst strand cDNA due to random hexamer mispriming. PLoS ONE, 8(12):e85583, 2013.
16. Benjamini, Y. and Speed, T.P. Summarizing and correcting the GC content bias in high-throughput sequencing. Nucleic Acids Res, 40(10):e72, 2012.
17. FastQ Screen. Available from: http://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/.
18. MacManes, M.D. On the optimal trimming of high-throughput mRNA sequence data. Front Genet, 5:13, 2014.
19. Del Fabbro, C., Scalabrin, S., Morgante, M., and Giorgi, F.M. An extensive evaluation of read trimming effects on illumina NGS data analysis. PLoS ONE, 8(12):e85024, 2013.

فصل چهارم

همردیف‌سازی خوانش‌ها با مرجع

۴–۱ مقدمه

همردیفی به معنای به خط کردن توالی‌ها برای پی بردن به این موضوع است که توالی‌های مزبور در چه مناطقی تشابه داشته و میزان این مشابهت چقدر است. همردیف‌سازی یا مکان‌یابی خوانش‌ها با یک ژنوم یا ترانسکریپتوم مرجع این امکان را فراهم می‌آورد که مشخص شود که خوانش مزبور از کجا منشا گرفته است. مکان‌یابی خوانش‌ها با ژنوم مرجع اطلاعات موقعیت ژنومی که می‌تواند در جستجو و شناسایی ژن‌ها و رونوشت‌های جدید به کار گرفته شود (ر.ک: فصل پنجم) را ارائه کرده و برای کمی‌سازی بیان (ر.ک: فصل ششم) به کار گرفته می‌شود. اگر یک ژنوم مرجع در دسترس نباشد، یا اگر هدف تنها کمی‌سازی رونوشت‌های شناخته شده باشد، می‌توان به جای آن خوانش‌ها را با یک ترانسکریپتوم مکان‌یابی نمود.

به دلایل متعددی، همردیف‌سازی خوانش‌ها با یک ژنوم مرجع یک فرآیند چالشی است. خوانش‌ها نسبتاً کوتاه بوده و میلیون‌ها خوانش وجود دارد. در حالی که ژنوم‌ها می‌توانند بزرگ بوده و حاوی توالی‌های غیر انحصاری نظیر تکرارها و ژن‌های کاذب باشند که قابلیت مکان‌یابی این مناطق را کاهش می‌دهند. علاوه بر این، همردیف‌سازها باید با عدم تطابق‌ها و ایندل‌هایی که ناشی از تنوع ژنومی و خطاهای توالی‌یابی هستند، مواجهه کنند. همچنین بیشتر جانداران اینترون‌هایی در ژن‌هایشان دارند و در نتیجه خوانش‌های توالی‌یابی RNA به صورت ناپیوسته با ژنوم همردیف می‌شوند. جاگیری خوانش‌های پیرایش شده در بین اینترون‌ها و تعیین صحیح حدود اگزون – اینترون کاری دشوار است. زیرا سیگنال‌های توالی در مناطق پیرایش محدود بوده و اینترون‌ها می‌توانند هزاران باز طول داشته باشند.

در این فصل انواع برنامه‌های همردیفی و مصورسازی خوانش‌های همردیف شده در زمینه‌ی ژنومی معرفی می‌شوند. ابزارهای همردیف‌سازی و دست‌ورزی نیز معرفی شده و سپس معیارهای کیفی مبتنی بر حاشیه‌نگاری در فصل ششم معرفی می‌گردند.

۴-۲ برنامه‌های همردیف‌سازی

ده‌ها برنامه‌ی همردیف‌سازی وجود داشته که روش‌های مختلفی را برای چیره‌شدن بر چالش‌های فوق پیشنهاد می‌کنند. Fonseca و همکاران (۱) یک مطالعه‌ی مقایسه‌ای بین همردیف‌سازها انجام داده و فهرست موجود در وب را به روز رسانی می‌نمایند (۲). به طور معمول همردیف‌سازها برخی از ابتکارها را به کار بسته و از طرح‌های نمایه‌گذاری مختلف برای سرعت‌بخشی به فرآیندها استفاده می‌کنند. ابزارهای متعددی می‌توانند در هنگام امتیازدهی به عدم تطابق‌ها، مقادیر کیفی بازها را بررسی کرده و از فاصله‌ی مورد انتظار و جهت نسبی خوانش‌های جفت انتهایی استفاده نمایند. همردیف‌سازها میزان اطمینان در موقعیت مکان‌یابی را به صورت کیفیت مکان‌یابی گزارش می‌نمایند ($Q = -10 \log P$. در اینجا P ، نشان دهنده‌ی احتمال اینکه خوانش مورد نظر متعلق به جای دیگر باشد، است). کیفیت مکان‌یابی به چند عامل بستگی داشته که مهمترین آنها یگانگی[1] می‌باشد. برخی از همردیف‌سازها می‌توانند خوانش‌هایی که در چند نقطه مکان‌یابی شده‌اند را متناسب با پوشش به صورت مساوی بین مکان‌های تطابق توزیع نمایند.

همردیف‌سازهای پیرایش شده[2] که مختص خوانش‌های توالی‌یابی RNA هستند، از روش‌های مختلفی برای همردیفی خوانش‌های پیرایش شده استفاده می‌کنند. این فرآیند می‌تواند شامل اجرای یک همردیفی مقدماتی برای شناسایی اتصالات[3] اگزونی بوده که سپس همردیفی نهایی را رهنمون می‌گردد. اگر حاشیه‌نگاری ژنومی در دسترس باشد، همردیف‌سازها می‌توانند از آن برای جادهی خوانش‌های پیرایش شده استفاده کنند. همان‌گونه که ارزیابی سیستماتیک انجام شده توسط Engstrom و همکاران نشان می‌دهد، همردیف‌سازهای پیرایش شده از نظر نتیجه‌ی همردیفی، عملکرد تشخیص پیرایش، صحت باز، تحمل عدم تطابق و تشخیص ایندل متفاوت هستند (۳).

اصلی‌ترین عاملی که در انتخاب یک همردیف‌ساز برای مطالعات توالی‌یابی RNA باید در نظر گرفته شود این است که آیا به همردیف‌های پیرایش شده نیاز است یا خیر؟ اگر جاندار مورد نظر اینترون نداشته یا microRNA ها توالی‌یابی شده باشند، استفاده از همردیف‌سازهای پیوسته نظیر Bowtie (۴) یا BWA (۵) که اساساً برای DNA طراحی شده‌اند، مطلوب است. اگر خوانش‌ها بیش از ژنوم با ترانسکریپتوم مکان‌یابی گردند، باز هم این همردیف‌سازها می‌توانند استفاده شوند. ولی اگر خوانش‌های توالی‌یابی RNA با ژنومی که حاوی اینترون‌ها بوده مکان‌یابی گردند، استفاده از یک همردیف‌ساز پیرایش شده نظیر TopHat (۶)، STAR (۷) و یا GSNAP (۸) ضروری است.

1- Uniqueness
2- Spliced aligner
3- Junction

Bowtie ۱-۲-۴

Bowtie یکی از متداول‌ترین نرم‌افزارهای همردیف‌ساز بوده که به واسطه‌ی سرعت بالا و نیاز کمتر به حافظه، مقبولیت یافته است. در اینجا روی ویرایش جدیدتر این نرم‌افزار که Bowtie2 بوده و به ویژه برای خوانش‌های بلند (از ۵۰ تا هزاران باز) مناسب است و می‌تواند همردیف‌سازی شکاف‌دار را برای ایندل‌ها انجام دهد، تمرکز می‌گردد. ویرایش قبلی این نرم‌افزار که Bowtie1 بود، می‌توانست حساسیت بیشتری برای خوانش‌های کوتاه‌تر داشته باشد. ولی امکان در نظر گرفتن شکاف در توالی‌ها را نداشت. علی‌رغم اینکه Bowtie2 نمی‌تواند همردیف‌سازی پیرایش شده را اجرا نماید، ولی می‌تواند به عنوان یک موتور همردیف‌سازی توسط همردیف‌ساز پیرایش شده‌ی TopHat2 به کارگرفته شود. همچنین همان‌گونه که در فصل ششم توضیح داده می‌شود، Bowtie2 می‌تواند همردیفی‌های ترانسکریپتوم را مستقیماً به ابزار کمی‌سازی eXpress ارسال نماید (۹).

Bowtie2 برای دستیابی به سرعت بالا و نیاز حافظه‌ی کم، ژنوم مرجع را با استفاده از یک نمایه‌ی FM که بر مبنای روش تبدیل باروز - ویلر عمل می‌کند، نمایه‌گذاری می‌نماید. این نرم‌افزار برای افزایش بیشتر سرعت فرآیند همردیف‌سازی، ابتدا با انجام یک همردیفی چندسیدی، فضای جستجو را محدود می‌کند. Bowtie2 در این مرحله‌ی ابتدایی، چند قطعه‌ی کوچک از خوانش‌ها (که اصطلاحاً سید نامیده می‌شوند) را بدون اینکه اجازه‌ی وجود شکاف در بین آنها یا مبهم بودن بازهای مرجع داده شود، همردیف‌سازی می‌نماید. کاربر می‌تواند طول سید را کنترل کرده و فاصله و تعداد عدم تطابق‌ها را مشخص کند.

Bowtie2 دارای دو وضعیت همردیف‌سازی است: انتها به انتها و موضعی. وضعیت انتها به انتها نیازمند آن است که همه‌ی بازها در خوانش همردیف شوند. ولی در وضعیت موضعی می‌توان برخی از بازهای موجود در یک یا هر دو انتها را جهت حداکثرسازی امتیازات همردیفی، پیرایش نمود. وضعیت انتها به انتها که اغلب زمانی به کار گرفته می‌شود که Bowtie2 توسط TopHat2 اجرا می‌گردد، وضعیت مشکل‌تر است. در این وضعیت، بهترین امتیاز همردیفی صفر بوده و تاوان‌ها برای عدم تطابق و شکاف از آن کسر می‌شوند. یک عدم تطابق در یک باز با کیفیت بالا تاوان بزرگ‌تری در مقایسه با یک عدم تطابق در یک باز با اطمینان پایین، که می‌تواند یک خطای نمونه‌گیری باشد، دریافت می‌کند. کاربر می‌تواند تاوان‌های مورد استفاده و اینکه آیا اطلاعات کیفیت باز بایستی در نظر گرفته شوند یا خیر، را انتخاب نماید. به جای تنظیم مقادیر پارامترهای جداگانه برای همردیفی چندسیدی و موارد فوق، می‌توان از یکی از ترکیبات مقادیر پارامترهای

آماده (خیلی سریع (very fast)، سریع (fast)، حساس (پیش فرض) ((sensitive (default) و خیلی حساس (very sensitive)) استفاده نمود.

به شکل پیش فرض، Bowtie2 برای چندین همردیفی جستجو می‌کند و این کار را تا آنجا ادامه می‌دهد که به محدودیت اعمال شده روی جستجو دست یافته و سپس بهترین حالت را گزارش می‌نماید. اگر چند همردیفی خوب با شرایط یکسان موجود باشد، نرم‌افزار یکی از آنها را به صورت تصادفی انتخاب کرده، تعداد جایگزین‌ها را گرارش کرده و مقدار کیفی مکان‌یابی را برای نشان دادن فقدان اطمینان در منشا خوانش، کاهش می‌دهد. مثال ارائه شده در زیر برای Bowtie2 نحوه‌ی همردیف‌سازی خوانش‌ها با ژنوم را نشان می‌دهد. باید توجه نمود که Bowtie2 می‌تواند به خوبی برای همردیف‌سازی خوانش‌ها با ترانسکریپتوم به کار گرفته شود. مثالی از این کاربرد نیز در فصل ششم در زمینه‌ی کمی‌سازی رونوشت با eXpress ارائه شده است.

ساخت یا دانلود یک نمایه‌ی مرجع

نمایه‌های ژنوم مرجع Bowtie2 برای تعداد زیادی از جانداران در وبسایت Bowtie2 (۱۰) و وبسایت Illumina iGenomes (۱۰) در دسترس است. در زمان دانلود کردن این نمایه باید اطمینان حاصل نمود که نمایه‌ی مربوط به Bowtie2 انتخاب شده است. زیرا نمایه‌های مزبور برای ویرایش‌های قبلی Bowtie ، تفاوت دارند. همچنین همان‌گونه که در زیر نشان داده شده است، می‌توان به آسانی و با کمک دستور `Bowtie2-build` یک نمایه‌ی شخصی ایجاد نمود. در هر کدام از مسیرهای فوق، ممکن است بخواهید بعداً از نمایه‌ی ژنومی / فایل‌های FASTA از همان عرضه کننده به عنوان فایل‌های GTF، به نحوی که با نام‌های کروموزومی مطابقت داشته باشد (مثلاً ۱ در برابر chr1)، استفاده کنید. اگر بعداً می‌خواهید بیان را با کمک HTSeq کمی نمایید، Ensembl می‌تواند یک انتخاب خوب باشد. زیرا GTF های Ensembl فرمت صحیحی برای این کار دارند. GTF های دانلود شده از iGenomes بخش‌های اضافی داشته که می‌توانند توسط برنامه‌ی Cuffdiff مورد استفاده قرار گیرند.

در داده‌های مثال، فایل‌های FASTA حاصل از Ensembl برای ساخت نمایه‌ی Bowtie2 استفاده می‌شود:

در آدرس http://www.ensembl.org/info/data/fp/index.html ، جاندار مورد نظر و گزینه‌ی DNA را انتخاب کنید. فایل dna,toplevel.fa.gz را که حاوی همه‌ی کروموزم‌ها در یک فایل است، نیاز دارید (از فایل‌های dna_rm و dna_sm که حاوی DNA تکراری هستند، اجتناب کنید). توجه کنید که این فایل FASTA علاوه بر کروموزوم‌ها، حاوی قطعات اسمبل و توالی‌های هاپلوتیپ است.

اگر بخواهید فقط از کروموزوم‌ها استفاده کنید، می‌توانید فایل‌های FASTA جداگانه را برای هر کروموزوم دانلود کرده و سپس آنها را با هم ادغام نمایید.

فایل زیر را دانلود کنید:

```
wget ftp://ftp.ensembl.org/pub/release-74/fasta/homo_sap
iens/dna/Homo_sapiens.GRCh37.74.dna.toplevel.fa.gz
```

و آنرا از حالت فشرده (زیپ) خارج نمایید:

```
gunzip Homo_sapiens.GRCh37.74.dna.toplevel.fa.gz
```

با کمک دستور bowtie2-build نمایه بسازید:

```
bowtie2-build  -f  Homo_sapiens.GRCh37.74.dna.toplevel.fa
GRCh37.74
```

بخش پایانی این دستور (GRCh37.74) نام پایه‌ی نمایه است که در شش فایل نمایه با فرمت bt2 استفاده می‌شود:

```
GRCh37.74.1.bt2
GRCh37.74.2.bt2
GRCh37.74.3.bt2
GRCh37.74.4.bt2
GRCh37.74.rev.1.bt2
GRCh37.74.rev.2.bt2
```

فایل FASTA ژنوم برای همردیف‌سازی توسط Bowtie2 مورد نیاز نیست. ولی اگر این نمایه را توسط TopHat2 استفاده می‌کنید، بایستی آنرا نگه دارید. برای این کار باید فایل مورد نظر را جهت تطابق با نام پایه‌ی شاخص، تغییر نام دهید:

```
mv Homo_sapiens.GRCh37.74.dna.toplevel.fa GRCh37.74.fa
```

همردیف‌سازی خوانش‌ها با ژنوم

Bowtie2 فایل‌های FASTQ و FASTA را به عنوان فایل‌های ورودی می‌پذیرد. این فایل‌ها می‌توانند فشرده نیز بشوند. یک دستور معمول همردیف‌سازی برای خوانش‌های تک‌انتهایی در زیر نشان داده شده است. اگر نمونه‌ی مورد نظر به چند فایل ورودی شکسته شده باشد، بایستی این‌فایل‌ها با علامت کاما (,) از هم جدا گردند:

```
bowtie2 -q --phred64 -p 8 --no-unal -x GRCh37.74 -
Ureads1.fastq.gz -S reads1aligned.sam
```

در اینجا نام پایه‌ی نمایه‌ی مرجع (-x) عبارت از GRCh37.74 بوده و فایل ورودی (-U) نیز در فرمت FASTA بوده (-q) و از رمزگذاری کیفی Phred+64 استفاده می‌نماید (--phred64). به این نکته باید توجه شود که اگر فایل‌های FASTQ جدیدتر بوده و رمزگذاری کیفی Phred+33 دارند، بایستی در اینجا از phred33- استفاده نمود. خروجی در قالب یک فایل با عنوان reads1aligned.sam نگارش یافته (-S) و نبایستی شامل خوانش‌های همردیف‌نشده باشد (--no-unal). هشت پردازنده به طور همزمان برای دستیابی به سرعت بالاتر در همردیف‌سازی مورد استفاده قرار می‌گیرند (-p 8) . گزینه‌های متعدد دیگری نیز می‌توانند تخصیص داده شوند که برای آگاهی از آنها و پارامترهای مختلف‌شان می‌توان به راهنمای نرم‌افزار مراجعه نمود. همان‌گونه که در خلاصه‌ی ارائه شده نیز دیده می‌شود، ۴۷/۴۲ درصد از خوانش‌ها به صورت یگانه همردیف شده و نرخ کُلی همردیف‌سازی نیز برابر با ۸۲/۶۸ درصد بوده است.

```
34232081 reads; of these:
  34232081(100.00%)were unpaired; of these:
    5928253(17.32%)aligned 0 times
    16232369(47.42%)aligned exactly 1 time
    12071459(35.26%)aligned > 1 times
82.68% overall alignment rate
```

این دستور مشابه دستور ارائه شده برای خوانش‌های جفت انتهایی بوده و تنها تفاوت آن در این است که دو فایل ورودی دارد (-1 و -2):

```
bowtie2 -q -phred64 -p 8 --no-unal -x GRCh37.74 -
1reads1.fastq.gz -2 reads2.fastq.gz -S paired.sam
```

باید توجه شود که خوانش‌ها بایستی در دو فایل مزبور دارای ترتیب یکسانی باشند تا Bowtie2 بتواند آنها را به عنوان یک جفت، مدیریت نماید. اگر فایل‌های بیشتری موجود باشد، بایستی آنها را به ترتیب تطابق مرتب کرده و با کمک کاما (,) از هم جدا نمود. چندین پارامتر، به ویژه برای خوانش‌های جفت انتهایی، نظیر حداکثر طول قطعه، مکان‌یابی خوانش‌ها به صورت منطبق[1] (با جهت و فاصله‌ی نسبی صحیح) و در نظر گرفتن یا نادیده گرفتن خوانش‌های جفت نشده در دسترس هستند. همان‌گونه که در خلاصه‌ی ارائه شده نیز می‌توان مشاهده نمود، ۳۹/۱۹ درصد از جفت خوانش‌ها دقیقاً یک بار به صورت منطبق و ۲۶/۸۳ درصد از آنها بیش از یکبار به صورت منطبق همردیف شده‌اند. نرخ کُلی همردیف‌سازی برابر با ۸۱/۶۷ درصد بوده است.

1- Concordant

```
34232081 reads; of these:
  34232081(100.00%)were paired; of these:
    11633330(33.98%)aligned concordantly 0 times
    13415597(39.19%)aligned concordantly exactly 1 time
    9183154(26.83%)aligned concordantly >1 times
    ----
    11633330 pairs aligned concordantly 0 times; of these:
      1999775(17.19%)aligned discordantly 1 time
    ----
    9633555 pairs aligned 0 times concordantly or
discordantly; of these:
      19267110 mates make up the pairs; of these:
        12546751(65.12%)aligned 0 times
        4349286(22.57%)aligned exactly 1 time
        2371073(12.31%)aligned >1 times
81.67% overall alignment rate
```

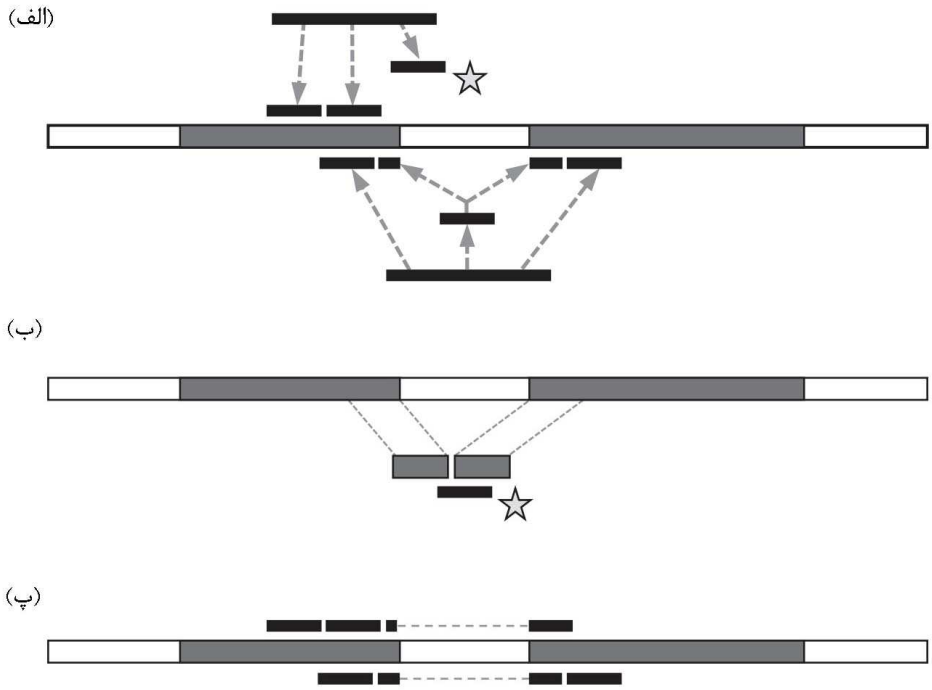
Bowtie2 نتایج همردیفی را در فرمت SAM (Sequence Alignment/Mapped) که استاندارد موجود برای همردیف‌های خوانش است، ارائه می‌نماید (۱۲). همان‌گونه که در بخش‌های بعدی این فصل توضیح داده می‌شود، برای صرفه‌جویی در فضا، می‌توان SAM را به ویرایش باینری BAM تبدیل نمود.

TopHat ۲-۲-۴

TopHat نسبتاً سریع و از نظر مصرف حافظه کارآمد بوده و یک برنامه‌ی متداول در همردیف‌سازی پیرایش شده برای خوانش‌های توالی‌یابی RNA است. در اینجا روی TopHat2 که از Bowtie2 به عنوان موتور همردیف‌سازی استفاده می‌کند (Bowtie2 را نیز پشتیبانی می‌نماید)، تمرکز می‌شود. این نرم‌افزار برای خوانش‌هایی که ۷۵ جفت باز یا بیشتر طول دارند، بهینه شده است. TopHat2 از یک فرآیند همردیف‌سازی چند مرحله‌ای پیروی می‌کند که در صورت موجود بودن حاشیه‌نگاری ژنومی، با همردیف کردن خوانش‌ها با ترانسکریپتوم شروع می‌شود. این کار صحت همردیف‌سازی را بهبود بخشیده، از جذب شدن خوانش‌ها به ژن‌های کاذب ممانعت کرده و سرعت کُلی فرآیند همردیفی را ارتقا می‌بخشد. اگر خوانش‌ها همردیف نگردند، TopHat2 انتهای آنها را برش نمی‌دهد. این بدان معناست که نسبت به عدم تطابق‌ها تحمل پایینی داشته بنابراین خوانش‌های با کیفیت پایین ممکن است همردیف نشوند. همچنین TopHat2 می‌تواند برای تشخیص جابجایی‌های ژنومی به کار گرفته شود. زیرا این نرم‌افزار می‌تواند خوانش‌ها را در میان نقاط انفصال تلفیقی[1]، همردیف نماید.

1- Fusion breakpoint

روش مکان‌یابی TopHat شامل سه بخش اصلی همردیفی ترانسکریپتوم اختیاری (مرحله‌ی ۱)، همردیفی ژنوم (مرحله‌ی ۲) و همردیفی پیرایش شده (مراحل ۳ تا ۶ (نگاره‌ی ۴-۱)) می‌باشد که جزییات آنها در زیر فهرست گردیده است. خوانش‌های جفت انتهایی نخست به صورت انفرادی همردیف شده و سپس با در نظر گرفتن طول و جهت قطعه، با همردیف‌های جفت انتهایی تلفیق می‌گردند.



نگاره‌ی ۴-۱: فرآیند همردیفی پیرایش شده در TopHat2 . الف) خوانش‌هایی که با ترانسکریپتوم یا ژنوم مکان‌یابی نشده‌اند، به قطعات کوچک خُرد شده و مجدداً با ژنوم مکان‌یابی می‌گردند. اگر TopHat2 خوانش‌هایی را بیابد که در محدوده‌ی اندازه‌ی اینترون تعریف شده توسط کاربر، در قطعه‌ی چپ و راست مکان‌یابی گردند، آنگاه نرم‌افزار کُل خوانش را در آن ناحیه‌ی ژنومی مکان‌یابی کرده تا از این طریق موقعیت‌های بالقوه‌ی پیرایش که حاوی سیگنال‌های پیرایشی معلوم است، را بیابد. ب) توالی‌های ژنومی مجاور موقعیت‌های بالقوه‌ی پیرایش به هم پیوند یافته و نمایه‌گذاری گردیده و قطعات خوانش مکان‌یابی نشده (در این نگاره با یک ستاره مشخص شده‌اند) توسط Bowtie2 با این اتصال که در مجاورت نمایه است، همردیف می‌شوند. پ) همردیف‌های قطعه‌ای به یکدیگر چسبانده شده و همردیف‌های کُل خوانش ایجاد می‌شوند.

۱- اگر اطلاعات حاشیه‌نگاری در دسترس باشد، نخست TopHat2 خوانش‌ها را با ترانسکریپتوم همردیف می‌سازد. این نرم‌افزار توالی‌های رونوشت را با استفاده از یک فایل GTF/GFF از نمایه‌ی ژنوم Bowtie2 استخراج می‌نماید. سپس Bowtie2 برای نمایه‌گذاری این ترانسکریپتوم مجازی و همردیف‌سازی خوانش‌ها با آن به کار گرفته می‌شود. در خروجی نهایی TopHat2 ، همردیف‌های ترانسکریپتوم به مکان‌یابی‌های ژنومی (پیرایش شده) تبدیل می‌گردند.

۲- خوانش‌هایی که کاملاً با ترانسکریپتوم همردیف نمی‌شوند، توسط Bowtie2 با ژنوم همردیف می‌گردند. در این مرحله، خوانش‌هایی که به طور پیوسته مکان‌یابی گردیدند (به یک اگزون)، مکان‌یابی خواهند شد. ولی خوانش‌های پیرایش شده‌ی چند اگزونی مکان‌یابی نمی‌گردند.

۳- خوانش‌هایی که مکان‌یابی نشده‌اند، به قطعات کوچک خُرد شده (به طور پیش فرض ۲۵ جفت باز) و مجدداً با ژنوم مکان‌یابی می‌گردند (نگاره‌ی ۱-۴). اگر TopHat2 خوانش‌هایی را بیابد که در محدوده‌ی اندازه‌ی اینترون تعریف شده توسط کاربر، در قطعه‌ی چپ و راست مکان‌یابی گردند، آنگاه نرم‌افزار کُل خوانش را در آن ناحیه‌ی ژنومی مکان‌یابی کرده تا از این طریق موقعیت‌های بالقوه‌ی پیرایش که حاوی سیگنال‌های پیرایشی معلوم است (GT-AG ، GC-AG یا AT-AC)، را بیابد. همچنین در این مرحله TopHat2 ایندل‌ها و نقاط انفصال تلفیقی را جستجو می‌نماید.

۴- توالی‌های ژنومی مجاور[1] موقعیت‌های بالقوه‌ی پیرایش به هم پیوند یافته و نمایه‌گذاری گردیده و قطعات خوانش مکان‌یابی نشده توسط Bowtie2 با این اتصال که در مجاورت نمایه است، همردیف می‌شوند.

۵- همردیف‌های قطعه‌ای حاصل از مراحل ۳ و ۴ به یکدیگر چسبانده شده و همردیف‌های کُل خوانش ایجاد می‌شوند.

۶- همردیف‌هایی که در مرحله‌ی ۲ به میزان چند باز به داخل یک اینترون امتداد یافته‌اند، مجدداً با استفاده از اطلاعات موقعیت پیرایش، با اگزون‌ها همردیف می‌شوند.

۷- جهت تصمیم‌گیری در مورد اینکه کدام همردیف‌ها برای خوانش‌های دارای چند مکان‌یابی گزارش شوند، TopHat2 با در نظر گرفتن تعداد خوانش‌هایی که از اتصالات، ایندل‌ها و غیره پشتیبانی می‌کنند، نسبت به محاسبه‌ی مجدد امتیاز همردیفی آنها اقدام می‌نماید.

1- Flanking

آماده‌سازی نمایه‌های مرجع

جهت استفاده از TopHat2 ، باید ژنوم مرجع با روشی که قبلاً در همین فصل برای Bowtie2 تشریح گردید، نمایه‌گذاری شود. همچنین TopHat2 نیازمند فایل FASTA ژنومیِ متناظر است تا از این طریق، وقتی که نمایه آماده می‌شود، آنرا حذف ننماید. اگر فایل FASTA در همان دایرکتوری که فایل‌های نمایه حضور دارند، موجود نباشد، TopHat2 در هر اجرا آنرا از فایل‌های نمایه خواهد ساخت. البته این فرآیند زمان‌بر است. اگر حاشیه‌نگاری‌های ژنومی در فرمت فایل GTF/GFF موجود باشند (۱۳)، نخست خوانش‌ها با ترانسکریپتوم همردیف می‌شوند. GTF های Ensembl در http://www.ensembl.org/info/data/fp/index.html موجود بوده و با انتخاب جاندار مورد نظر و گزینه‌ی GTF می‌توان به آنها دسترسی داشت. برای صرفه‌جویی در هر اجرای همردیف‌سازی که در ادامه راه‌اندازی می‌شود، می‌توان نمایه‌ی ترانسکریپتوم را از قبل آماده نمود.

```
tophat2 -G GRCh37.74.gtf --transcriptome-index=GRCh37.74.tr
GRCh37.74
```

در اینجا از فایل حاشیه‌نگاری GRCh37.74.gtf و GRCh37.74.tr نمایه‌ی ژنومی Bowtie برای ایجاد یک نمایه‌ی ترانسکریپتومی Bowtie2 که نام پایه‌ی آن GRCh37.74.tr می‌باشد، استفاده گردیده است. لازم به ذکر است که نام کروموزوم‌ها در فایل GTF و در نمایه‌ی ژنومی باید مطابقت داشته باشند. Bowtie2 باید روی این مسیر باشد. زیرا TopHat2 ازآن برای ساختن نمایه استفاده خواهد کرد. فایل‌های زیر ایجاد می‌شوند:

```
GRCh37.74.tr.1.bt2
GRCh37.74.tr.2.bt2
GRCh37.74.tr.3.bt2
GRCh37.74.tr.4.bt2
GRCh37.74.tr.fa
GRCh37.74.tr.fa.tlst
GRCh37.74.tr.gff
GRCh37.74.tr.rev.1.bt2
GRCh37.74.tr.rev.2.bt2
GRCh37.74.tr.ver
```

همردیف‌سازی خوانش‌ها

TopHta2 هر دو نوع فایل FASTQ و FASTA را به عنوان ورودی می‌پذیرد. فایل‌های خوانش‌ها می‌توانند فشرده شده (.gz) باشند. ولی تاربال‌ها (.tgz یا .tar.gz) لازم است که باز شده و

به صورت چند فایل تفکیک گردند. مثال‌های زیر دستورات لازم برای همردیف‌سازی جداگانه‌ی خوانش‌های تکی و جفت انتهایی است. ولی در صورت لزوم، TopHat2 می‌تواند خوانش‌های تکی را در یک همردیفی جفت انتهایی تلفیق کند.

دو دستور جایگزین بعدی خوانش‌های تکی را همردیف می‌نمایند. در هر دو مورد، خوانش‌ها با ژنوم مرجع انسان همردیف می‌شوند (نام پایه‌ی نمایه‌ی GRCh37.74). دستور نخست از یک نمایه‌ی ترانسکریپتوم از پیش آماده استفاده کرده ولی دستور دوم نمایه‌ی ترانسکریپتوم را در حین کار و با استفاده از فایل GTF می‌سازد. اگر چند فایل خوانش موجود باشد، بایستی آنها را با کمک کاما (,) از هم جدا نمود. توجه شود که هر دو نرم‌افزار Bowtie2 و SAMtools بایستی روی مسیر باشند. زیرا TopHat2 از این بسته‌های نرم‌افزاری به صورت داخلی استفاده می‌نماید.

```
tophat2 -o outputFolder --transcriptomeindex=GRCh37.74.tr -
p 8 --phred64-quals GRCh37.74reads1.fastq.gz
```

یا:

```
tophat2 -o outputFolder -G GRCh37.74.gtf -p 8--phred64-
quals GRCh37.74 reads1.fastq.gz
```

چون Sanger (phred+33) رمزگذاری کیفی باز در TopHat2 است، لذا باید توصیف کننده‌ی `-phred64-quals` نیز افزوده گردد تا این نوع رمزگذاری در داده‌های مثال که حاصل از ویرایش قبلی اِلومِنا هستند، نشان داده شود (از `--solexa1.3-qual` هم می‌توان استفاده نمود). در اینجا برای سرعت‌بخشی به این فرآیند، هشت پردازنده به طور همزمان مورد استفاده قرار گرفته است (`-p 8`). توجه شود که اگر داده‌ها با یک دستورالعمل مختص زنجیره تهیه شده باشند، بایستی پارامتر `--library-type` نیز تنظیم شود (پیش‌فرض بدون زنجیره است). TopHat2 گزینه‌های بیشاری برای همردیف‌سازی و گزارش‌دهی دارد. به عنوان مثال می‌توان خوانش‌ها را تنها با ترانسکریپتوم همردیف کرده (`-T`) یا حداکثر تعداد همردیف‌ها به ازای هر خوانش را گزارش نمود (`-g`) که این مقدار به صورت پیش‌فرض برابر با ۲۰ است.

align_summary.txt نشان می‌دهد که ۷۹/۳ درصد از خوانش‌ها مکان‌یابی شده‌اند:

```
Reads:
      Input : 34232081
       Mapped : 27140089(79.3% of input)
        of these: 1612317 (5.9%)have multiple
alignments (2771 have >20)
79.3% overall read mapping rate.
```

دستور همردیف‌سازی برای خوانش‌های جفت انتهایی در زیر نشان داده شده است. لازم به ذکر است که ترتیب این خوانش‌ها در دو فایل باید منطبق بر هم باشند تا TopHat2 بتواند آنها را به نحو صحیحی با هم جفت کند. اگر چند فایل خوانش موجود باشد، باید آنها را با استفاده از کاما (,) جدا کرده و به همان ترتیب آنها را وارد نموده و بین هر دو مجموعه نیز یک فاصله‌ی خالی در نظر گرفت.

```
tophat2 -o outputFolder --transcriptomeindex=GRCh37.74.tr
-p 8 --phred64-quals GRCh37.74reads1.fastq.gz reads2.fastq
.gz
```

یکی از پارامترهای اختصاصی TopHat2 برای همردیفی‌های جفت انتهایی، فاصله‌ی درونی مورد انتظار بین خوانش‌های جفتی (`-r`) است که بایستی بر اساس مجموعه‌ی داده‌ها تنظیم گردد. مقدار پیش‌فرض ۵۰ می‌باشد که در اینجا مناسب است. زیرا اندازه‌ی الحاق در داده‌های مثال مزبور برابر با ۲۰۰ بوده و طول خوانش‌ها نیز ۷۵ باز است (۵۰ = ۷۵ × ۲ - ۲۰۰). می‌توان تنظیمات را به گونه‌ای انجام دادکه یک جفت به صورت منطبق (با جهت و فاصله‌ی مورد انتظار) مکان‌یابی گردد (`--no-discordant`). اگر TopHat نتواند یک جفت را با هم مکان‌یابی کند، خوانش‌های مزبور را جداگانه مکان‌یابی می‌کند. ولی می‌توان این قابلیت که به صورت پیش‌فرض تعریف شده است را لغو کرد (`--no-mixed`).

پس از اجرا، خلاصه‌ی زیر ارائه می‌شود:

```
Left reads:
          Input : 34232081
           Mapped : 27143093(79.3% of input)
            of these: 1014796 (3.7%)have multiple
alignments (3621 have >20)
Right reads:
          Input : 34232081
           Mapped : 22600062 (66.0% of input)
            of these: 759539 (3.4%)have multiple
alignments (3193 have >20)
72.7% overall read mapping rate.

Aligned pairs: 21229613
     of these: 702920 (3.3%)have multiple alignments
               336032 (1.6%)are discordant alignments
61.0% concordant pair alignment rate.
```

TopHat چند فایل خروجی تولید می‌کند:

- فایل accepted_hits.bam حاوی همردیف‌ها در فرمت BAM است. همردیف‌ها بر مبنای مختصات کروموزومی مرتب می‌شوند.
- فایل junctions.bed حاوی اتصالات اگزونی یافت شده در فرمت BED است (۱۴). یک اتصال شامل دو بلوک می‌باشد که طول هر کدام از بلوک‌ها به اندازه‌ی طویل‌ترین پیش‌آمدگی[1] هر خوانشی که آن اتصال را می‌پوشاند، است. این امتیاز نشان دهنده‌ی تعداد همردیف‌هایی است که اتصال مزبور را می‌پوشاند.
- فایل insertions.bed حاوی الحاق‌های یافت شده است. chromLeft نشان دهنده‌ی آخرین باز ژنومی قبل از الحاق می‌باشد.
- فایل deletion.bed حاوی حذف‌های شناسایی شده است. chromLeft نشان دهنده‌ی اولین باز ژنومی حذف می‌باشد.
- فایل align_summary.txt نرخ همردیفی و تعداد خوانش‌ها و جفت‌های دارای چند همردیفی را گزارش می‌کند.

STAR ۳-۲-۴

STAR که مخفف شده‌ی Spliced Transcripts Alignment to a Refernce (همردیفی رونوشت‌های پیرایش شده با یک مرجع) است، یک برنامه‌ی نسبتاً جدید همردیف‌سازی پیرایش شده است که بسیار سریع اجرا می‌شود. در عوض، به عنوان مثال این برنامه در مقایسه با TopHat به حافظه‌ی بیشتری نیاز دارد. در راهنمای STAR (۱۱ فوریه ۲۰۱۳) اینطور ذکر شده است که ۳۱ گیگابایت RAM برای ژنوم انسان و موش کافی بوده ولی اگر نمایه‌ی مرجع به طور مناسبی تهیه شده باشد، می‌توان این نرم‌افزار را برای ژنوم انسان با ۱۶ گیگابایت RAM نیز اجرا نمود (به مطالب زیر توجه شود). علی‌رغم اینکه STAR به طور ویژه برای سرعتش شهرت یافته است، ولی مزایای زیاد دیگری نیز دارد. این نرم‌افزار می‌تواند یک جستجوی نااُریب برای اتصالات پیرایشی[2] انجام دهد. زیرا نیازی به هیچ‌گونه اطلاعات قبلی در مورد موقعیت آنها، سیگنال‌های توالی یا طول اینترون ندارد. STAR می‌تواند یک خوانش حاوی هر تعداد اتصالات پیرایشی، ایندل‌ها و عدم تطابق‌ها را همردیف کرده و با انتهاهای دارای کیفیت پایین، مقابله نماید. همچنین این نرم‌افزار می‌تواند خوانش‌های بلند و حتی mRNA با طول کامل را مکان‌یابی کرده و در نتیجه همگام با افزایش طول خوانش‌ها، نیاز به این نرم‌افزار هم افزایش می‌یابد.

1- Overhang
2- Splice junction

مزایای STAR تا حد زیادی بر مبنای رویکردی است که اصطلاحاً حداکثر طول قابل مکان‌یابی[1] نامیده می‌شود. STAR یک خوانش را به قطعاتی (که به شکل پیش‌فرض، ۵۰ باز طول دارند)، تقسیم کرده و بهترین بخشی را که برای هر قطعه می‌تواند مکان یابی کند، می‌یابد. سپس بخش باقیمانده را مکان‌یابی می‌کند که می‌تواند در مورد یک اتصال پیرایشی، خیلی دور باشد. این جستجوی متوالی حداکثر سید قابل مکان‌یابی تطابق‌های دقیق را یافته و از ژنوم در قالب آرایه‌های پسوندی غیرفشرده استفاده می‌نماید. در مرحله‌ی دوم، STAR سیدها را با یکدیگر در یک پنجره‌ی فرضی ژنومی پیوند زده و این امکان را برای عدم تطابق‌ها، ایندل‌ها و اتصالات پیرایشی نیز فراهم می‌آورد. در این مرحله این جفت سیدها به طور همزمان مدیریت می‌شوند تا حساسیت و دقت افزایش یابد.

STAR می‌تواند اتصالات پیرایشی را به صورت از نو بیابد. ولی می‌توان آنرا با حاشیه‌نگاری‌های اتصالات در هنگام ساخت نمایه‌ی مرجع نیز به کار گرفت. در این مورد تعدادی از بازهای اگزونی که توسط کاربر تعریف شده‌اند، از هر دو موقعیت دهنده و گیرنده‌ی پیرایش با هم تلفیق شده و این توالی‌ها به توالی ژنوم افزوده می‌شوند. در طی مکان‌یابی، خوانش‌ها هم با توالی ژنوم و هم با توالی‌های موقعیت پیرایش همردیف می‌شوند. اگر یک خوانش با توالی پیرایشی مکان‌یابی شده از این اتصال عبور کند، مختصات این مکان‌یابی با مختصات ژنومی تلفیق می‌گردد.

ساخت یا دانلود یک نمایه‌ی مرجع

پیش از اجرای STAR لازم است که یک نمایه‌ی مرجع برای ژنوم مورد نظر ساخته شده یا دریافت گردد. برای برخی از ژنوم‌ها (انسان، موش، گوسفند و مرغ) نمایه‌های مرجع STAR ساخته شده و برای دانلود در دسترس هستند (fp://fp2.cshl.edu/gingeraslab/tracks/STARrelease/ STARgenomes/). نمایه‌هایی با اندکی تفاوت برای ژنوم انسان با موارد استفاده‌ی مختلف ساخته شده است. به طور خاص، نمایه‌ای که نامش حاوی کلمه‌ی sparse (به معنای تُنُک) باشد، برای استفاده با حافظه‌ی کمتر ساخته شده است. اگر بخواهید نمایه‌ی شخصی خودتان را بسازید، باید دستور زیر را در STAR بنویسید:

```
STAR --runMode genomeGenerate --genomeDir /path/to/ Genome
Dir --genomeFastaFiles fasta1 fasta2--sjdbFileChrStartEnd
annotation.gtf.sjdb--sjdbOverhang 74 --runThreadN 8
```

1- Maximum mappable length

گزینه‌ی genomeDir-- نشان دهنده‌ی دایرکتوری است که نمایه‌ی مرجع (شامل توالی ژنوم باینری، فایل‌های آرایه‌ی پسوند و برخی از فایل‌های کمکی) در آن قرار خواهند گرفت. گزینه‌ی genomeFastaFiles-- نیز فایل‌های FASTA توالی مرجع را برای نمایه شدن فهرست می‌نماید. فرآیند نمایه شدن می‌تواند در یک روش چند بُعدی و با استفاده از گزینه‌ی runThreadN-- اجرا گردد. اگر بخواهید از یک حاشیه‌نگاری اتصال پیرایشی در مکان‌یابی استفاده کنید (که معمولاً ایده‌ی خوبی است)، لازم است که در هنگام ساخت نمایه‌ی مرجع، یک فایل مرجع اتصال پیرایشی فراهم گردد. دستوری که مثال زده شده است، برای ارائه‌ی فایل annotation.gtf.sjdb ، که حاوی مختصات اینترون‌ها در فرمت تعریف شده در راهنمای STAR است، از پارامتر sjdbFileChrStartEnd- استفاده می‌نماید. چنین فایلی برای ژنوم hg19 مرجع انسانی را می‌توان از لینک اختصاصی زیر دانلود نمود. به جای آن می‌توان از یک فایل GTF با یک پارامتر sjdbGTFfile-- استفاده کرد. در هر دو مورد، باید پارامتر sjdbOverhang- برای تعریف طول توالی حاصل از موقعیت‌های معلوم دهنده و گیرنده که بایستی در هنگام ساخت نمایه‌ی مرجع به کار گرفته شود، استفاده گردد. در حالت ایده‌آل این مقدار بایستی برای طول خوانش ۱- تنظیم شود. بدین ترتیب در مثال فوق فرض می‌شود که خوانش‌ها ۷۵ جفت بازی هستند. اگر خوانش‌هایی با طول متغیر دارید، استفاده از مقادیر بزرگ‌تر، صحیح‌تر است. اگر لازم است حجم حافظه‌ی مورد نیاز برای اجرای STAR کاهش یابد، می‌توان نمایه‌ی مرجع را با استفاده از مقادیر بالاتر برای گزینه‌ی genomeSAsparseD-- ساخت (مقدار پیش فرض، ۱ است). این گزینه از یک آرایه‌ی پسوندی تُنُک‌تر که نیازهای حافظه را در برآمد سرعت همردیفی کاهش می‌دهد، استفاده می‌نماید.

مکان‌یابی

دستور مکان‌یابی زیر در STAR از یک نمایه‌ی ژنوم انسانی حاشیه‌نگاری شده‌ی اتصال پیرایشی پیش ساخته که از وبسایت STAR دانلود شده است (ر.ک: لینک فوق)، استفاده می‌کند:

```
STAR --genomeDir hg19_Gencode14.overhang75
--readFilesIn reads1.fastq.gz reads2.fastq.gz
--readFilesCommand zcat--outSAMstrandField
intronMotif --runThreadN 8
```

گزینه‌ی genomeDir-- بایستی به دایرکتوری نمایه‌ی مرجع که بر مبنای دستورالعمل‌های فوق ساخته یا دانلود شده است، اشاره نماید. سپس بعد از readFilesIn-- ، باید فایل (های)

FASTQ تخصیص داده شوند. این فایل‌ها می‌توانند فشرده شوند. ولی در این شرایط لازم است که یک دستور برای باز کردن فرمت فشرده به صورت یک برهان[1] به گزینه‌ی --readFilesCommand تخصیص داده شود (در اینجا از zcat استفاده شده است). اگر چندین فایل خوانش موجود باشد، باید آنها را با کاما (,) جدا کرده و یک فاصله‌ی خالی قبل از فهرست نمودن فایل‌های جفت در ترتیبی منطبق با هم در نظر گرفت. پارامتر --outSAMstrandFieldintronMotif زنجیره‌ی SAM را با پسوند XS اضافه نموده که برای آنالیزهای پایین‌دستی در مواردی که بخواهید از برنامه‌ی Cufinks استفاده کنید، ضروری است. پارامترهای متعدد دیگر نیز برای کنترل جنبه‌های مختلف رفتار STAR طبق موارد تشریح شده در راهنمای این نرم‌افزار، وجود دارد. به عنوان مثال، ممکن است که بخواهید همردیف‌های حاوی بیش از یک تعداد عدم انطباق‌های ارائه شده یا همردیف‌های حاوی اتصالات پیرایشی پشتیبانی شده توسط خوانش‌های بسیار اندک را پاکسازی نمایید.

خروجی

از دسامبر ۲۰۱۳، فایل‌های زیر حداقل خروجی‌های STAR هستند:

- فایل Aligned.out.sam که شامل همردیف‌ها در فرمت SAM است (خوانش‌هایی که همردیف نشده‌اند را در بر نمی‌گیرد).
- فایل SJ.out.tab که یک فایل مرزبندی شده با تَب بوده و حاوی اطلاعات همردیف‌ها روی اتصالات پیرایشی است.
- فایل‌های Log.out ، Log.final.out و Log.progress.out که همان‌گونه که از نام‌شان پیداست، فایل‌های لاگ هستند که انواع اطلاعات راجع به نحوه‌ی عملکرد اجرا را ارائه می‌دهند. در بین این فایل‌ها، فایل Log.final.out از اهمیت بیشتری برخوردار است. زیرا آماره‌های مکان‌یابی مفیدی ارائه می‌دهد. باید توجه نمود که تعداد خوانش‌ها و طول خوانش، ترکیباتی از خوانش‌های جفتی هستند.

```
                                 Started job on | Feb 12 11:32:58
                              Started mapping on | Feb 12 11:46:52
                                     Finished on | Feb 12 11:51:09
Mapping speed, Million of reads per hour | 479.52
                          Number of input reads | 34232081
                    Average input read length | 150
```

1- Argument

```
                                UNIQUE READS:
            Uniquely mapped reads number | 27113906
                  Uniquely mapped reads% | 79.21%
                   Average mapped length | 147.51
                Number of splices: Total | 12176905
     Number of splices: Annotated (sjdb) | 12049801
                Number of splices: GT/AG | 12070507
                Number of splices: GC/AG | 78264
                Number of splices: AT/AC | 9359
       Number of splices: Non-canonical | 18775
                Mismatch rate per base,% | 1.04%
                  Deletion rate per base | 0.01%
                 Deletion average length | 2.20
                 Insertion rate per base | 0.02%
                Insertion average length | 1.85
                      MULTI-MAPPING READS:
 Number of reads mapped to multiple loci | 1376440
      % of reads mapped to multiple loci | 4.02%
 Number of reads mapped to too many loci | 7662
      % of reads mapped to too many loci | 0.02%
                           UNMAPPED READS:
% of reads unmapped: too many mismatches | 0.00%
          % of reads unmapped: too short | 15.70%
              % of reads unmapped: other | 1.05%
```

همردیف‌سازی خوانش‌ها با مرجع در Chipster

Chipster از Bowtie2 ، BWA و TopHat برای همردیف‌سازی خوانش‌ها با یک مرجع استفاده می‌کند. ابزارهای جداگانه‌ای نیز برای خوانش‌های تکی و جفت انتهایی در دسترس است.

- فایل‌های خوانش (FASTQ) و یکی از ابزارهای موجود در دسته‌ی Alignment را انتخاب نـمایید. در پنل پارامتر، مرجع صحیح و گزینه‌های همردیفی را انتخاب کـرده و

تخصیص صحیح خوانش‌های جفت انتهایی و یا GTF شخصی یا فایل FASTA مرجع را کنترل کنید.

- فایل‌های نتایج همواره فایل‌های BAM نمایه شده و به صورت مرتب شده بر مبنای مختصات هستند.

۳-۴ آماره‌های همردیف‌سازی و ابزارهایی برای دست‌ورزی فایل‌های همردیف‌سازی

فایل‌های SAM/BAM که معمولاً توسط همردیف‌سازها ایجاد می‌شوند، نیازمند پردازش‌هایی نظیر تبدیل SAM/BAM ، مرتب‌سازی، نمایه‌سازی یا ادغام هستند. دو بسته‌ی نرم‌افزاری اصلی تحت عنوان SAMtools (۱۲) و پیاده‌سازی جاوای آن که Picard نام دارد (۱۵)، برای این پردازش‌ها در دسترس است. Picard ابزارهای بیشتری داشته و در زمان اعتبارسنجی[1] فایل‌ها، سخت‌گیرانه‌تر از SAMtools رفتار می‌کند. در اینجا روی برخی از دستورات SAMtools که به طور معمول مورد استفاده واقع می‌شوند، تمرکز می‌گردد.

- تبدیل SAM به BAM : مرتب‌سازی همردیف‌ها در فرمت BAM منجر به صرفه‌جویی در فضا می‌گردد. همچنین اکثر ابزارهای پایین دستی از فرمت BAM بیشتر از فرمت SAM استفاده می‌کنند. در اینجا دستور به نحوی نوشته می‌شود که ورودی SAM (S-)، خروجی BAM (b-) و نام فایل خروجی alignments.bam باشد (o-).

```
samtools view -bS -o alignments.bam input.sam
```

- تبدیل SAM به BAM و در نظر گرفتن اطلاعات سرتیتر[2] (h-): خطوط سرتیتر با علامت @ شروع شده و حاوی اطلاعاتی راجع به نام و طول توالی‌های مرجع (@SQ)، برنامه‌ی ایجاد کننده‌ی این فایل (@PG) و مرتب‌سازی فایل مزبور و نحوه‌ی آن (@HD) است.

```
samtools view -h -o alignments.sam input.bam
```

- فقط بازیابی کردن سرتیتر (H-):

```
samtools view -H alignments.bam
```

1- Validating
2- Header

- مرتب‌سازی همردیف‌ها در BAM بر مبنای مختصات کروموزومی یا بر مبنای نام خوانش‌ها (n-): مرورگرهای ژنومی و برخی از ابزارهای آنالیز به مرتب‌سازی بر مبنای مختصات نیاز داشته و ابزارهای کمی‌سازی بیان نیز اغلب به مرتب‌سازی بر مبنای نام نیاز دارند.

```
samtools sort alignments.bam alignments.sorted

samtools sort –n alignments.bam alignments.
namesorted
```

- باید توجه نمود که SAMtools می‌تواند روی یک جریان[1] کار کند. بنابراین ترکیب کردن دستورات با مسیرهای یونیکس جهت پرهیز از فایل‌های واسطه‌ی بزرگ، امکان‌پذیر است. به عنوان مثال، دستور زیر خروجی SAM نرم افزار Bowtie2 را به BAM تبدیل کرده و آنرا بر مبنای مختصات کروموزومی مرتب نموده و یک فایل تحت عنوان alignments.sorted.bam ایجاد می‌کند:

```
bowtie2 –q --phred64 –p 4 –x GRCh37.74 –U reads1.fq |
samtools view –bS – | samtools sort –alignments.sorted
```

- نمایه‌گذاری فایل‌های BAM مرتب شده بر مبنای مختصات: نمایه‌گذاری، بازیابی سریع همردیف‌ها را امکان‌پذیر کرده و مرورگرهای ژنومی و برخی از ابزارهای پایین دستی به آن نیاز دارند. دستور زیر یک فایل نمایه‌گذاری تحت عنوان alignments.sorted.bam.bai ایجاد می‌کند:

```
samtools index alignments.sorted.bam
```

- ایجاد زیرمجموعه‌ای از همردیف‌ها با مشخص کردن یک کروموزوم معین یا ناحیه‌ی کروموزومی (در اینجا همردیف‌ها از کروموزوم شماره‌ی ۱۸ استخراج می‌شوند): در این دستور لازم است که یک فایل نمایه حضور داشته باشد.

```
samtools view –b –o alignments.18.bam alignments.bam 18
```

- فهرست کردن تعداد خوانش‌هایی که در هر کروموزوم مکان‌یابی می‌شوند: در این دستور لازم است که یک فایل نمایه حضور داشته باشد.

```
samtools idxstats alignments.sorted.bam
```

1- Stream

- فیلتر کردن همردیف‌ها بر مبنای کیفیت مکان‌یابی. دستور زیر همردیف‌هایی را که کیفیت مکان‌یابی بالاتر از ۳۰ دارند، نگه می‌دارد:

```
samtools view -b -q 30 -o alignments_MQmin30.bam alignments
.bam
```

- فیلتر کردن همردیف‌ها بر مبنای مقادیر زمینه‌ی نشانه‌نمای[1] SAM : گزینه‌ی F- خوانش‌هایی را که مقدار نشانه‌نمای در نظر گرفته شده را داشته باشند (در اینجا برابر با ۴ در نظر گرفته شده است که به معنای خوانش‌های مکان‌یابی نشده است) پاکسازی کرده و گزینه‌ی f- خوانش‌های دارای مقدار نشانه‌نمای داده شده (در اینجا برابر با ۲ در نظر گرفته شده است که بدان معناست که یک خوانش در یک جفت مناسب مکان‌یابی می‌شود) را نگه می‌دارد. برای اطلاع از جزییات مقادیر نشانه‌نما می‌توان خصوصیات SAM را مورد مطالعه قرار داد (۱۲).

```
samtools view -b -F 4 -o alignments.mapped_only.bam
alignments.bam
```

```
samtools view -b -f 2 -o properly_paired_reads.bam
alignments.bam
```

- به دست آوردن آماره‌های مکان‌یابی بر مبنای زمینه‌ی نشانه‌نما:

```
samtools flagstat alignment.bam
```

گزارش زیر حاوی اطلاعات پایه نظیر تعداد خوانش‌های مکان‌یابی شده و خوانش‌های جفت شده‌ی صحیح و تعداد جفت‌های مکان‌یابی شده به کروموزوم‌های مختلف است:

```
52841623 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 duplicates
52841623 + 0 mapped (100.00%:-nan%)
52841623 + 0 paired in sequencing
28919461 + 0 read1
23922162 + 0 read2
42664064 + 0 properly paired (80.74%:-nan%)
44904884 + 0 with itself and mate mapped
7936739 + 0 singletons (15.02%:-nan%)
999152 + 0 with mate mapped to a different chr
357082 + 0 with mate mapped to a different chr (mapQ >=5)
```

1- Flag

آماره‌های همردیفی می‌توانند با کمک بسته‌ی نرم‌افزاری RseQC که با جزییات بیشتر در فصل ششم و در مبحث معیارهای کیفی مبتنی بر حاشیه‌نگاری معرفی می‌شود، محاسبه گردند. RseQC شامل چند اسکریپت پایتون است که معیارهای همردیفی نظیر تعداد خوانش‌های همردیف شده، بخش‌هایی از خوانش‌ها که به صورت یگانه همردیف شده‌اند، توزیع فاصله‌ی داخلی و نسبتی از جفت‌ها که به طور دقیق در موقعیت یکسانی مکان‌یابی می‌شوند، را کنترل می‌کنند. نسبت جفت‌های مکان‌یابی شده به طور دقیق در موقعیت یکسان نشان دهنده‌ی خوانش‌هایی است که از قطعات یکسان منشاء گرفته و دلیل آن نیز احتمالاً ناشی از فزون‌تکثیری[1] PCR است. آماره‌های پایه‌ی همردیفی می‌توانند با ابزار bam_stat.py به دست آیند:

```
python bam_stat.py -i accepted_hits.bam
```

این ابزار جدول زیر را ایجاد می‌نماید که در آن اگر کیفیت مکان‌یابی خوانش‌ها بالاتر از ۳۰ باشد (می‌توان این آستانه را با افزودن پارامتر q- تغییر داد)، آنگاه خوانش‌های مزبور یگانه تلقی می‌گردند.

```
#==================================================
#All numbers are READ count
#==================================================
Totalrecords:                                 52841623

QCfailed:                                     0
Optical/PCR duplicate:                        0
Non primary hits                              3098468
Unmapped reads:                               0
mapq < mapq_cut(non-unique):                  1774335

mapq >= mapq_cut(unique):                     47968820
Read-1:                                       26128297
Read-2:                                       21840523
Reads map to'+':                              24085239
Reads map to'-':                              23883581
Non-splice reads:                             35970095
Splice reads:                                 11998725
Reads mapped in proper pairs:                 39702036
Proper-paired reads map to different chrom:0
```

1- Over-amplification

دست‌ورزی SAM/BAM و آماره‌های همردیفی در Chipster

- Chipster در دسته‌ی Utilities دارای ابزارهای متعدد مبتنی بر SAMtools است. این ابزارها SAM را به BAM و بالعکس تبدیل کرده، فایل‌های BAM را مرتب، نمایه‌گذاری، زیرمجموعه‌گیری و ادغام نموده، تعداد همردیفی‌ها به ازای هر کروموزوم و کُل را شمارش کرده و توالی توافقی همردیف‌ها را ایجاد می‌نمایند. برخی از ابزارها نیازمند فایل نمـایه‌ی BAM که قبلاً به آن اشاره شد، هستند. باید هر دو فـایل BAM و نمایه را انتخاب کرده و در پنل پارامتر، از تخصیص صحیح فایل‌ها اطمینان حاصل نمود.
- RseQC در دسته‌ی Quality control موجود است. این نرم‌افزار آماره‌های BAM ، توزیع فاصله‌ی داخلی، اطلاعات مربوط به زنجیره و معیارهای کیفی مبتنی بر حاشیه‌نگاری که در فصل ششم مورد بحث و بررسی واقع می‌شوند، را گزارش می‌نماید.

۴-۴ مصورسازی خوانش‌ها در زمینه‌ی ژنومی

مصورسازی خوانش‌های همردیف شده در یک زمینه‌ی ژنومی می‌تواند برای اهداف مختلف مفید واقع شده و به همین دلیل نیز بسیار توصیه می‌شود. شما می‌توانید ساختار رونوشت‌های جدید را مصورسازی کرده، تایید و پشتیبانی برای اتصالات جدید را قضاوت نموده، پوشش اگزون‌های مختلف را کنترل کرده، وجود یا عدم وجود برآمدگی‌های بلند از خوانش‌های مضاعف را بررسی نموده و ایندل‌ها و SNP ها را شناسایی کنید. از همه‌ی اینها مهم‌تر آن است که می‌توانید داده‌هایتان را با حاشیه‌نگاری‌های مرجع مقایسه نمایید.

چندین مرورگر ژنومی قادر به مصورسازی داده‌های توالی‌یابی کارآمد هستند که از جمله‌ی آنها می‌توان به Integrative Genomic Viewer (IGV) (۱۷) JBrowse (۱۸)، Tablet (۱۹) و مرورگرهای ژنومی UCSC (۲۰) و Chipster اشاره نمود. این مرورگرها عملکردهای متعددی داشته و تشریح همه‌ی آنها فراتر از اهداف این کتاب است. ولی توصیه می‌شود که شماره‌ی ویژه‌ی ژورنال Briffings in Bioinformatics در زمینه‌ی مصورسازی توالی‌یابی نسل جدید (۲۱) که مقالات آموزنده‌ای در زمینه‌ی چندین مرورگر ژنومی نیز دارد، مطالعه گردد. در فصل دوم نیز تصاویری از مرورگرهای ژنومی IGV و Chipster برای داده‌های توالی‌یابی RNA ارائه شده است.

چون نرم‌افزار Chipster در سرتاسر این کتاب مورد استفاده قرار می‌گیرد، لذا معرفی مختصری از مرورگر ژنومی آن نیز در اینجا ارائه می‌شود. Chipster داده‌ها را در زمینه‌ی حاشیه‌نگاری‌های

Ensembl مصورسازی کرده و از چندین فرمت فایل شامل BAM ، BED ، GTF ، VCF و tsv پشتیبانی می‌کند. کاربران می‌توانند در سطح نوکلئوتید بزرگ‌نمایی کرده، تفاوت‌های موجود با توالی مرجع را پُررنگ کرده و به صورت خودکار پوشش محاسبه شده (هم پوشش کُل و هم پوشش مختص زنجیره) را ملاحظه نمایند. برای فایل‌های BED ، مصورسازی امتیاز نیز امکان‌پذیر است. آنچه که اهمیت دارد این است که انواع مختلفی از داده‌ها را می‌توان با یکدیگر مصورسازی نمود. به عنوان مثال می‌توان داده‌های توالی‌یابی RNA و تعداد کُپی[1] اختلالات که توسط ریزآرایه‌ها سنجیده می‌شوند، را در کنار هم ملاحظه نمود. چون Chipster با یک محیط آنالیز جامع تلفیق شده است، لذا نیازی به خروج و ورود داده به یک نرم‌افزار خارجی نیست. البته اگر بخواهید از Chipster تنها برای اهداف مصورسازی استفاده کنید، باید بتوانید فایل‌های BAM را به آن وارد نمایید. در این مورد، فایل‌های داده‌ی شما به صورت خودکار و در حین ورود، ذخیره و نمایه‌گذاری می‌شود.

مصورسازی خوانش‌ها در زمینه‌ی ژنومی با Chipster

به عنوان مثال فایل‌های نتایج TopHat2 که شامل accepted_hits.bam و deletions.bed است، مصورسازی می‌گردد.

- می‌توانید از فایل BED به عنوان یک کمک منفی استفاده کنید. بنابراین نخست این فایل را از یک پنجره‌ی دیگر جدا نمایید. بدین منظور روی فایل مزبور کلیک کرده تا در یک نمای صفحه گسترده باز شده و سپس روی Detach کلیک کنید.
- فایل‌های BAM و BED را انتخاب نموده و در پنل Visualization ، روش مصورسازی Genome browser را انتخاب کرده و سپس اندازه‌ی را پنل برای نمایش بزرگ‌تر، حداکثر نمایید.
- از منوی پایین‌رو Genome ، گزینه‌ی hg19 را انتخاب کرده و روی Go کلیک نمایید. در صورت نیاز می‌توانید با حرکت قرقره‌ی ماوس و تغییر مقیاس پوشش، بزرگ‌نمایی یا کوچک‌نمایی کنید.
- از فایل BED جدا شده برای بررسی فهرست حذف‌ها به صورت موثر استفاده نمایید. روی مختصات شروع یک حذف (ستون ۱) کلیک نموده و مرورگر را به این مکان حرکت دهید. همچنین می‌توانید فایل BED را بر مبنای امتیاز (تعداد خوانش‌هایی که از حذف پشتیبانی می‌کنند) و از طریق کلیک کردن روی عنوان ستون ۴ ، مرتب نمایید.

1- Copy number

۴–۵ خلاصه

مکان‌یابی میلیون‌ها خوانش توالی‌یابی RNA با یک ژنوم مرجع یک کار محاسباتی سنگین بوده و معمولاً همردیف‌سازها از طرح‌های نمایه‌گذاری مختلف برای سرعت بخشیدن به این فرآیند استفاده می‌کنند. بسیاری از جانداران حاوی اینترون‌ها بوده و بنابراین برای مکان‌یابی غیرمستقیم خوانش‌ها با ژنوم به یک همردیف‌ساز پیرایشی نیاز است. همچنین به منظور مواجهه با واریانت‌های ژنومی، و خطاهای توالی‌یابلی و در نظر گرفتن کیفیت باز در هنگام امتیازبندی، همردیف‌سازها باید از از عدم انطباق‌ها و ایندل‌ها پشتیبانی نمایند. به جای مکان‌یابی خوانش‌ها روی یک ژنوم، می‌توان آنها را با یک ترانسکریپتوم نیز مکان‌یابی نمود. مکان‌یابی با ترانسکریپتوم برای جاندارانی که یک ژنوم مرجع برای آنها وجود ندارد، تنها مسیر تلقی می‌گردد.

انتخاب همردیف‌ساز به جاندار و هدف آزمایش بستگی دارد. به عنوان مثال اگرهمردیف‌های پیرایشی مورد نیاز نباشند و صحت همردیفی اهمیت داشته باشد، BWA می‌تواند یک انتخاب خوب باشد. اگر سرعت از اهمیت بیشتری برخوردار باشد، Bowtie2 توصیه می‌گردد. اگر جاندار اینترون داشته و حاشیه‌نگاری مرجع نسبتاً کاملی داشته باشد، TopHat2 می‌تواند همردیف‌های پیرایش شده‌ی خوبی ایجاد نماید. STAR با عدم تطابق‌ها بهتر مقابله کرده، اجراهایش سریع‌تر بوده، همردیف‌های بیشتری ایجاد نموده و می‌تواند اتصالات پیرایشی را با یک روش نااُریب شناسایی کند.

فایل‌های همردیفی می‌توانند با ابزارهای مختلفی نظیر SAMtools و Picard دست‌ورزی شوند. به عنوان مثال این ابزارها بازیابی کارآمد خوانش‌های مکان‌یابی شده با یک ناحیه‌ی معین یا خوانش‌های مکان‌یابی شده به صورت یگانه را امکان‌پذیر می‌سازند. ابزارهایی نظیر RseQC اطلاعات کیفی مهمی را در مورد خوانش‌های همردیف شده فراهم می‌آورند. چندین مرورگر ژنومی برای مصورسازی همردیف‌ها در زمینه‌ی ژنومی در دسترس هستند. مصورسازی به شدت توصیه شده است. زیرا هیچ چیزی نمی‌تواند در تشخیص الگوهای مورد نظر در داده‌ها از چشم انسان پیشی بگیرد.

منابع

1. Fonseca N.A., Rung J., Brazma A., and Marioni J.C. Tools for mapping highthroughput sequencing data. Bioinformatics 28(24):3169–3177, 2012.
2. Updated listing of mappers. Available from: http://wwwdev.ebi.ac.uk/fg/hts_mappers/.
3. Engström P.G., Steijger T., Sipos B. et al. Systematic evaluation of spliced alignment programs for RNA-seq data. Nat Methods 10(12):1185–1191, 2013.

4. Langmead B. and Salzberg S.L. Fast gapped-read alignment with Bowtie2. Nat Methods 9(4):357–359, 2012.
5. Li H. and Durbin R. Fast and accurate long-read alignment with Burrows–Wheeler transform. Bioinformatics 26(5):589–595, 2010.
6. Kim D., Pertea G., Trapnell C. et al. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol 14(4):R36, 2013.
7. Dobin A., Davis C.A., Schlesinger F., et al. STAR: Ultrafast universal RNAseq aligner. Bioinformatics 29(1):15–21, 2013.
8. Wu T.D. and Nacu S. Fast and SNP-tolerant detection of complex variants and splicing in short reads. Bioinformatics 26(7):873–881, 2010.
9. Roberts A. and Pachter L. Streaming fragment assignment for real-time analysis of sequencing experiments. Nat Methods 10(1):71–73, 2013.
10. Bowtie2. Available from: http://bowtie-bio.sourceforge.net/bowtie2/index. html.
11. iGenomes. Available from: http://support.illumina.com/sequencing/ sequencing_sofware/igenome.ilmn.
12. Li H., Handsaker B., Wysoker A. et al. The sequence alignment/map format and SAMtools. Bioinformatics 25(16):2078–2079, 2009.
13. GFF/GTF fle format description. Available from: http://genome.ucsc.edu/ FAQ/FAQformat.html#format3.
14. BED fle format description. Available from: http://genome.ucsc.edu/FAQ/ FAQformat.html#format1.
15. Picard. Available from: http://picard.sourceforge.net/.
16. Wang L., Wang S., and Li W. RSeQC: Quality control of RNA-seq experiments. Bioinformatics 28(16):2184–2185, 2012.
17. Torvaldsdottir H., Robinson J.T., and Mesirov J.P. Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. Brief Bioinform 14(2):178–192, 2013.
18. Westesson O., Skinner M., and Holmes I. Visualizing next-generation sequencing data with JBrowse. Brief Bioinform 14(2):172–177, 2013.
19. Milne I., Stephen G., Bayer M. et al. Using Tablet for visual exploration of second-generation sequencing data. Brief Bioinform 14(2):193–202, 2013.
20. Kuhn R.M., Haussler D., and Kent W.J. Te UCSC genome browser and associated tools. Brief Bioinform 14(2):144–161, 2013.
21. Special Issue: Next generation sequencing visualization. Brief Bioinform 14(12), 2013.

فصل پنجم

اسمبل کردن ترانسکریپتوم

۵–۱ مقدمه

هدف از اسمبل کردن توالی‌یابی RNA ، بازسازی کامل رونوشت‌ها بر مبنای خوانش‌های توالی است. به دلیل محدودیت‌های موجود در فناوری توالی‌یابی نسل دوم، تنها قطعات نسبتاً کوتاه می‌توانند به عنوان یک واحد منفرد توالی‌یابی شوند. علی‌رغم اینکه روش‌های امیدبخشی در توالی‌یابی نسل سوم نظیر PacBio از پاسیفیک بایوساینسز ارائه شده است که توالی‌یابی تک مولکول‌های با طول چند کیلو باز را امکان‌پذیر ساخته است، ولی در حال حاضر این فناوری‌ها به طور معمول در توالی‌یابی ترانسکریپتوم به کار گرفته نمی‌شوند. بنابراین در عمل برای دستیابی به توالی‌های کامل رونوشت‌ها، باید آنها را از قطعات کوچک همپوشان ساخت. اصولاً دو روش برای انجام این کار وجود دارد. اگر یک ژنوم مرجع در دسترس باشد، می‌توان از آن به عنوان راهنما برای اسمبل کردن استفاده نمود. خوانش‌های توالی‌یابی RNA ابتدا روی ژنوم مکان‌یابی شده و در اسمبل کردن نیز تعیین می‌شود که خوانش‌های مکان‌یابی شده متناظر با کدام رونوشت‌ها هستند. روش دیگر برای این کار، اسمبل کردن از نو است که از هیچ نوع اطلاعات خارجی استفاده نمی‌نماید. در فقدان یک ژنوم مرجع، اسمبل کردن بر مبنای استفاده از مشابهت توالی بین خوانش‌های توالی‌یابی RNA پایه‌ریزی می‌گردد. هر دو روش فوق می‌توانند به عنوان یک مساله‌ی محاسباتی که شامل یافتن یک مجموعه از مسیرها در یک نمودار است، فرمول‌بندی شوند. به دلیل طبیعت ترکیبی و پیچیده‌ی این مساله، حتی در یک اسمبل کردن نسبتاً کوچک نیز پاسخ‌های ممکن برای آن بی‌شمار است. برشمردن همه‌ی پاسخ‌های ممکن برای یافتن پاسخ بهینه‌ی کُلی به سادگی امکان‌پذیر نبوده و بنابراین روش‌های شناسایی و تقریب مختلف در طی فرآیند اسمبل کردن به کار گرفته می‌شود.

اسمبل کردن ترانسکریپتوم با اسمبل کردن ژنوم تفاوت دارد. معمولاً در اسمبل کردن ژنوم پوشش خوانش یکنواخت‌تر است (به غیر از اُریبی‌های مربوط به تهیه‌ی کتابخانه و فناوری توالی‌یابی). انحراف از عمق توالی یکنواخت در اسمبل کردن ژنوم نشان دهنده‌ی حضور تکرارها است. ولی در داده‌های توالی‌یابی RNA فراوانی بیان ژن می‌تواند چندین مقدار را بین ژن‌ها تغییر داده و ایزوفرم‌های مختلف همان ژن نیز می‌توانند در سطوح مختلف بیان شوند. علی‌رغم اینکه می‌توان از این موضوع برای تشخیص و ساخت ایزوفرم‌های مختلف در اسمبل کردن رونوشت

استفاده نمود ولی تفاوت زیاد فراوانی‌ها بین ژن‌ها نیز چالش‌برانگیز است. برای نشان دادن ژن‌های با فراوانی کمتر و وقایع نادر لازم است که عمق توالی‌یابی بیشتر باشد. برای متعادل کردن تفاوت‌های فراوانی بین ژن‌ها، روش‌های آزمایشگاهی جهت نرمال‌سازی کتابخانه وجود دارد. تشریح این روش‌ها فراتر از اهداف این کتاب است. ولی لازم است این موضوع به خاطر سپرده شود که کیفیت اسمبل کردن شامل ترکیبی از داده‌ها و روش‌های محاسباتی است. چون فناوری توالی‌یابی تنها محتوای یک کتابخانه‌ی توالی‌یابی RNA را به قالب عددی تبدیل می‌نماید، لذا آماده‌سازی کتابخانه یک عنصر کلیدی در به دست آوردن داده‌های با کیفیت خوب است. هم در توالی‌یابی و هم در اسمبل کردن، ورودی‌های نامطلوب منجر به خروجی‌های نامطلوب نیز می‌گردد. کنترل کیفیت داده‌ها بایستی قبل از هر نوع اسمبل کردن انجام شود.

برای این فصل دو بسته‌ی نرم‌افزاری اسمبل کردن مبتنی بر مکان‌یابی و دو بسته‌ی نرم‌افزاری اسمبل کردن از نو در نظر گرفته شده است. همه‌ی این نرم‌افزارها غیرتجاری بوده و به صورت عمومی در دسترس هستند. همانند استفاده از هر روش محاسباتی، در اینجا نیز باید توجه نمود که خروجی اسمبل کردن، بستگی به ترکیب داده‌ها و روش دارد. معمولاً هر روشی شامل پارامترهایی است که می‌توانند تغییر داده شوند و بنابراین بسته به روش و پارامترها، حتی در صورت استفاده از داده‌های یکسان، خروجی اسمبل کردن می‌تواند به نحو قابل ملاحظه‌ای تغییر نماید.

این فصل با تشریح مساله‌ی اسمبل کردن و روش حل آن شروع می‌شود. سپس هر کدام از چهار بسته‌ی نرم‌افزاری منتخب معرفی شده و کاربرد آنها با استفاده از مجموعه‌ی داده‌های یکسان نشان داده می‌شود. این مجموعه‌ی داده‌ها از پروژه ENCODE به دست آمده و شامل خوانش‌های جفت انتهایی یک فرد هستند. برای محدود کردن اندازه‌ی داده‌ها، در این مثال تنها از خوانش‌هایی که روی کروموزوم ۱۸ انسانی مکان‌یابی گردیده‌اند، استفاده شده است. خوانش‌های توالی جفت انتهایی از فایل wgEncodeCaltechRnaSeqH1hescR2x75Il200AlignsRep1V2.bam استخراج شده است که از آدرس http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCaltechRnaSeq/ قابل دسترس است. مجموعه‌ی داده‌های حاصل کوچک بوده و تنها حاوی ۳۴۴۰۰۰ جفت خوانش است. بنابراین زمان اجرای اسمبل‌سازها چندان طولانی نخواهد بود.

۵-۲ روش‌ها

ریشه‌های اسمبل کردن توالی‌یابی RNA را می‌توان تا نخستین روزهایی که توالی‌یابی EST (نشانه‌ی توالی‌های بیان شده) در ابتدای دهه‌ی ۱۹۹۰ آغاز گردید، دنبال نمود (۱). پردازش EST ها شامل خوشه‌بندی و اسمبل کردن بود (۲). خوشه‌بندی به معنای گروه‌بندی خوانش‌های

EST مشابه با یکدیگر از طریق محاسبه‌ی همپوشان‌های جفتی[1] است. به عنوان مثال اگر یک همپوشان طویل‌تر از ۴۰ جفت باز با توالی دیگر ۹۵ درصد مشابهت داشت، می‌توانستند آنها را به عنوان اعضای خوشه تعریف کنند. بعد از خوشه‌بندی خوانش‌ها، اسمبل کردن به صورت جداگانه در داخل هر خوشه اجرا می‌گردید. علی‌رغم تغییر در جزییات، این دو مرحله هنوز هم مراحل اصلی فرآیند اسمبل کردن رونوشت محسوب می‌گردند: ۱) یافتن خوانش‌هایی که متعلق به ژنگاه یکسانی هستند و ۲) ایجاد نمودارهایی که نشان دهنده‌ی رونوشت‌ها در داخل هر ژنگاه می‌باشند. یکی از تفاوت‌های اصلی بین EST ها و داده‌های فعلی توالی‌یابی RNA این است که EST ها معمولاً تنها نشان دهنده‌ی قطعات و رونوشت‌های جزیی هستند ولی طبیعت داده‌های کارآمد امروزی سبب می‌شود که این داده‌ها نشان دهنده‌ی طول کامل رونوشت‌ها باشند. علی‌رغم اینکه خوانش‌های تکی پلتفرم‌های نسل دوم توالی‌یابی کُل طول رونوشت را پوشش نمی‌دهند، ولی حجم بسیار بالای داده‌ها امکان بازسازی کُل طول رونوشت‌ها را فراهم می‌آورد.

۵-۲-۱ متفاوت بودن اسمبل کردن ترانسکریپتوم با اسمبل کردن ژنوم

در اوایل دوره‌ی EST ها، همان اسمبل‌سازهایی که برای ژنوم‌ها به کار گرفته می‌شدند، برای ترانسکریپتوم‌ها نیز استفاده می‌گردیدند. علی‌رغم اینکه هنوز این کار از نظر تکنیکی امکان‌پذیر است، ولی دیگر مورد استفاده قرار نمی‌گیرد. تفاوت‌های بنیادینی بین اسمبل کردن ژنوم و اسمبل کردن ترانسکریپتوم وجود دارد. علاوه بر تفاوت در یکنواختی عمق توالی‌یابی، تفاوت اصلی در این است که در اسمبل کردن ژنوم، خروجی ایده‌آل یک توالی خطی است که نشان دهنده‌ی هر ناحیه‌ی ژنومی می‌باشد. ولی در اسمبل کردن رونوشت، ممکن است چندین ایزوفرم از یک جایگاه ژنی یکسان وجود داشته باشد که این بدان معناست که اگزون یکسانی از یک ژن در زمینه‌های مختلف با اگزون‌های دیگر و بسته به رونوشت وجود دارد. بنابراین در اسمبل کردن رونوشت، ژن در طبیعی‌ترین حالت به صورت یک نمودار نشان داده شده که در آن گره‌ها[2] نشان دهنده‌ی اگزون‌ها و و قوس‌ها[3] نشان دهنده‌ی وقایع پیرایشی هستند. انشعاب‌ها در اتصالات گره متناظر با پیرایش جایگزین می‌باشد. یک رونوشت منفرد شامل یک مولکول تکی است که هنوز بایستی به عنوان یک توالی خطی نشان داده شده و مسیری در طول گره‌ها در یک نمودار تشکیل دهد. یک گره اگزونی تنها یکبار در یک ایزوفرم نمایش داده می‌شود. ولی همان اگزون می‌تواند در چندین ایزوفرم مختلف حضور داشته باشد. مجموعه‌ی همه‌ی مسیرهای ممکن در یک نمودار اگزونی شامل همه‌ی

1- Pairwise overlap
2- Node
3- Arc

ایزوفرم‌های ممکن است. تعداد مسیرهای ممکن می‌تواند زیاد باشد. ولی تنها تعداد بسیار کمی از آنها در ترانسکریپتوم واقعی حضور دارند. یک چالش در اسمبل کردن رونوشت پی بردن به این موضوع است که کدام ایزوفرم‌ها در بین همه‌ی کاندیداهای بالقوه، ایزوفرم‌های واقعی هستند. بار دیگر ملاحظه می‌شود که این مشکل نیز ناشی از کوتاه بودن خوانش توالی است. اگر توالی‌یابی کُل طول کامل یک رونوشت در قالب یک خوانش امکان‌پذیر بود، این مشکل حل می‌شد. پیچیدگی این مشکل زمانی آشکار می‌گردد که تلاش می‌شود توالی‌های بلند از قطعات کوتاه ساخته شوند.

۵-۲-۲ پیچیدگی بازسازی رونوشت

برای تشریح پیچیدگی بازسازی رونوشت، یک مثال ارائه می‌شود. فرض شود که سه اگزون در یک ژن وجود دارد. در اینصورت تعداد ایزوفرم‌های ممکن را می‌توان به صورت مجموع تعداد اگزون‌های تکی، تعداد جفت اگزون‌ها و تعداد اگزون‌های سه‌گانه شمارش نمود. این تعدادها به ترتیب برابر با ۳ ، ۳ و ۱ بوده که مجموع آنها برابر با ۷ می‌گردد. به طور کلی، در مورد N اگزون، تعداد ایزوفرم‌های ممکن عبارت است از:

$$\Sigma_{k=1}^{N} \binom{N}{k} = 2^{N} - 1$$

این بدان معناست که دو احتمال برای هر اگزون وجود دارد: یا اگزون مزبور در این ایزوفرم وجود داشته و یا وجود ندارد. عدد 2^{N} نیز نشان دهنده‌ی آن است که ایزوفرم خالی بوده و هیچ اگزونی در آن وجود نداشته و بنابراین یکی از مجموع کُل کم می‌شود (متناظر با $k = 0$ در معادله‌ی فوق). این مقدار فقط تعداد ایزوفرم‌های ممکن است. در ترانسکریپتوم، هر مجموعه‌ای از ایزوفرم‌ها می‌توانند حضور داشته باشند. علی‌رغم اینکه پیرایش جایگزین، ترکیبات متعددی را امکان‌پذیر می‌سازد، ولی همه‌ی آنها در یک ترانسکریپتوم واقعی حضور ندارند. مساله یافتن ترکیباتی است که صحیح هستند. تعداد ممکن مجموعه‌های ایزوفرم‌ها همانند خطوط فوق محاسبه می‌شود (یک ایزوفرم در ترانسکریپتوم حضور داشته یا حضور ندارد) و تعداد آن عبارت است از:

$$2^{2^{N}-1}$$

این تعداد به سرعت و همگام با رشد تعداد اگزون‌ها (N) رشد می‌نماید. برای مقادیر $N = 1$ ، ۲ ، ۳ ، ۴ و ۵ ، اندازه‌ی مجموع‌های ایزوفرم ممکن به ترتیب برابر با ۱ ، ۷ ، ۱۲۷ ،

۳۲۷۶۷ و ۲۱۴۷۴۸۳۶۴۷ است. این مثال نشان می‌دهد که اگر تعداد اگزون‌ها بیشتر از ۴ باشد، شمارش همه‌ی پاسخ‌های ممکن برای آزمودن اینکه کدام‌یک از اگزون‌ها بهترین انطباق را با داده‌ها دارند، عملی نخواهد بود.

۵-۲-۳ فرآیند اسمبل کردن

دو روش برای بازسازی رونوشت وجود دارد که عبارتند از: اسمبل کردن مبتنی بر مکان‌یابی[1] و اسمبل کردن از نو[2]. هر دو روش فوق شامل ساخت یک نمودار برای هر ژنگاه بر مبنای خوانش‌های توالی‌یابی RNA است. نمودار به عنوان یک نقطه‌ی شروع برای حل و فصل کردن ایزوفرم‌ها به کار می‌رود. در هر دو روش نیز مساله‌ی اصلی نحوه‌ی خُرد کردن داده‌ها، به نحوی که یک نمودار تنها نشان دهنده‌ی یک ژنگاه باشد، است.

مکان‌یابی در فصل چهارم این کتاب تشریح شده است. هر روشی که خُرد شدن خوانش‌ها را امکان‌پذیر سازد، می‌تواند برای همردیف‌سازی خوانش‌های توالی‌یابی RNA روی ژنوم به کار گرفته شود. اگر مدل‌های ژنی در دسترس باشند، اطلاعاتی راجع به اینکه کدام اگزون‌ها متعلق به کدام ژن‌ها هستند، ارائه خواهند داد. اگر هیچ مدل ژنی در دسترس نباشد، باید نخست خوانش‌های مکان‌یابی شده برای نشان دادن ژنگاه‌ها قطعه قطعه شوند. سپس یک نمودار اگزونی که نمودار پیرایشی نیز نامیده می‌شود، برای هر ژنگاه ساخته شده و جستجو برای یافتن مجموعه‌ی مسیرها در داخل هر نمودار انجام گیرد. هر مسیر نیز نشان دهنده‌ی یک ایزوفرم خواهد بود.

تعداد ایزوفرم‌های ممکن را می‌توان با محدود کردن ارتباطات در نمودار اگزونی کاهش داد. هر ارتباطی نشان دهنده‌ی یک اتصال اگزونی است. در یک نمودار کامل، همه‌ی ایزوفرم‌ها امکان‌پذیر هستند. زیرا یک قوس بین همه‌ی گره‌ها وجود دارد. بنابراین هدف، انتخاب یک توپولوژی برای نمودار که بهترین انطباق را با داده‌ها داشته باشد، است. وقایع پیرایشی که هیچ نوع دلیل و پشتیبانی در خوانش‌های توالی‌یابی RNA نداشته باشند، حذف شده و تنها ارتباطاتی در نمودار حفظ می‌شوند که مورد نیاز باشند. شواهد لازم برای حفظ یک قوس عبارتند از: اطلاعات خوانش‌های خُرد شده[3] و اطلاعات خوانش‌های جفت انتهایی. در مورد یک خوانش خُرد شده، اگر ابتدای خوانش با یک اگزون مکان‌یابی گردیده و انتهای آن نیز با خوانش دیگری مکان‌یابی شود، چنین وضعیتی دلیل و پشتیبانی لازم برای همسایه بودن دو اگزون در یک توالی رونوشت را ارائه می‌دهد. در مورد خوانش جفت انتهایی، این معیار برای هر دو انتهای جفت خوانش به کار رفته و

1- Mapping-based assembly
2- *de novo* assembly
3- Splite-read

در هر دو بایستی یک انتها با یک اگزون و انتهای دیگر با یک اگزون دیگر مکان‌یابی گردد. حضور یک خوانش خُرد شده در مقایسه با یک خوانش جفت انتهایی، شاهد محکم‌تری برای یک اتصال اگزونی محسوب می‌گردد. در مورد جفت خوانش‌های مکان‌یابی شده، اطلاعات اندازه‌ی الحاق باید برای حصول اطمینان از این موضوع که دو اگزون واقعاً یک اتصال را تشکیل می‌دهند، در مقایسه با این احتمال که دو اگزون تنها در یک رونوشت حضور داشته ولی چیزهای دیگری بین آنها وجود دارند، به کار گرفته شوند. توزیع اندازه‌ی الحاق به کتابخانه‌ی توالی‌یابی RNA بستگی دارد. معمولاً میانگین اندازه‌ی الحاق برای هر جفت خوانش مورد استفاده قرار گرفته و اگر واریانس داخل کتابخانه زیاد باشد، برآورد اندازه‌ی الحاق برای هر جفت خوانش معین نمی‌تواند صحیح باشد.

در اسمبل کردن از نو، اساساً دو روش وجود دارد: ۱) محاسبه‌ی همپوشان‌های جفتی بین خوانش‌هایی که توپولوژی نمودار اسمبل کردن را ارائه می‌دهند و ۲) ساختن یک نمودار دوبران[1] که همه‌ی داده‌های توالی را به صورت یک مجموعه از k-مِرها و ارتباطاتشان نشان می‌دهد. نمودار دوبران به عنوان یک موضوع ریاضی پیش از ابداع توالی‌یابی بسط داده شد (۳) و نخستین بار نیز توسط Pevzner و همکاران در زمینه‌ی اسمبل کردن ژنوم به کار گرفته شد (۴). هدف از اسمبل کردن از نو، استخراج قطعات پیوسته (کانتیگ‌ها[2]) با حداکثر طول ممکن از نمودار اسمبلی که نشان دهنده‌ی بخش‌های اصلی ژنوم یا ترانسکیرپتوم هستند، است. در طی پروژه‌ی ژنوم انسان (Human Genome Project) در دهه‌ی ۱۹۹۰، خوانش‌های توالی‌یابی نسبتاً بلند بوده (این خوانش‌ها از توالی‌یابی سَنگِر به دست می‌آمدند) و حجم‌شان کمتر از داده‌های امروزی بود. اسمبل‌سازهای ژنومی بر مبنای یک روش همپوشانی خوانش عمل می‌کردند و این راهبرد نیز اصطلاحاً همپوشانی - طرح‌بندی - اجماع[3] (OLC) که نشان دهنده‌ی سه مرحله‌ی این روش اسمبل کردن بود، نامیده می‌شد. علی‌رغم اینکه محاسبه‌ی همه‌ی همپوشان‌های جفتی بین خوانش‌ها زمان‌بر است، از نظر روش‌شناسی این راهبرد آسان‌ترین بخش از مشکل است. مشکل اصلی ناشی از این ترکیب است: چگونگی تعریف طرح نمودار که از آن توالی‌های اجماعی همردیف‌های چند خوانش به دست آید. می‌توان یک الگوریتم که پاسخ بهینه برای مشکل اسمبل کردن را بیابد، ایجاد نمود. ولی زمان اجرا برای هر مجموعه‌ی داده از مقادیر عملی بسیار طولانی خواهد بود و بنابراین یافته‌ها و تقریب‌های مختلف باید به کار گرفته شوند (۵). در مواردی که حجم داده‌های توالی افزایش یافته و همزمان با آن خوانش‌ها کوتاه‌تر می‌شوند، روش‌هایی که از نمودار دوبران استفاده می‌کنند نیز مطلوب‌تر می‌گردند. اکثر روش‌های فعلی برای اسمبل کردن

1- de Bruijn graph
2- Contig
3- Overlap-Layout-Consensus (OLC)

ترانسکریپتوم، بر مبنای نمودار دوبران بنا نهاده شده‌اند. البته استثناهایی نظیر اسمبل‌سازهای MIRA EST نیز وجود دارند که بر مبنای الگوی OLC کار می‌کنند.

۴-۲-۵ نمودار دوبران

هر گره از نمودار دوبران به یک (k - ۱)-مِر وابسته است. اگر یک k-مِر وجود داشته باشد که پسوند یا ابتدایش (k - ۱)-مِر گره A و پیشوند یا انتهایش (k - ۱)-مِر گره B باشد، آنگاه دو گره A و B مرتبط خواهند بود. بدین ترتیب، k-مِرها مرزهایی[1] را در نمودار دوبران ایجاد می‌نمایند (۷). توالی‌ها به صورت مسیرهایی در یک نمودار نشان داده می‌شوند و حتی یک خوانش منفرد توالی نیز به صورت چند گره متصل به هم پخش می‌شود (نخستین گره حاوی (k - ۱)-مِر بوده که از نخستین موقعیت یک توالی آغاز می‌گردد. دومین گره حاوی (k - ۱)-مِر بوده که از دومین موقعیت توالی آغاز می‌گردد و . . .). هر k-مِر حاضر در این نمودار تنها یک‌بار به عنوان یک مرز رابط دو گره عمل می‌کند. اگر دو خوانش توالی دارای یک k-مِر مشترک باشند، آنگاه آن دو خوانش در یک مرز اشتراک دارند. این وضعیت، اطلاعاتی را از همپوشان‌های بین خوانش‌ها ارائه کرده و در نتیجه نیازی به محاسبه‌ی هیچ مقایسه‌ی جفتی نیست. ایجاد یک نمودار دوبران کاری ساده بوده و در مقایسه با محاسبه‌ی همپوشان‌ها بین همه‌ی جفت خوانش‌ها بسیار سریع‌تر پیش می‌رود. این کار شامل استخراج ساده‌ی همه‌ی k-مِرها از خوانش‌ها و اتصال گره‌های حاضر در (k - ۱)-مِرها است. چالش اصلی در یافتن مسیرهای موجود در یک نمودار که رونوشت‌های صحیح را نشان می‌دهند، می‌باشد. خطاهای توالی‌یابی ناشی از نوک‌ها[2] که انتهاهای مرده در یک نمودار بوده و حباب‌ها[3] که ساختار نمودار را پیچیده می‌کنند، هستند. حباب‌ها از شاخه‌هایی که در بخش دیگری از نمودار به هم می‌پیوندند، تشکیل می‌گردند. برخی از حباب‌ها ناشی از خطای توالی‌یابی بوده ولی برخی دیگر ناشی از پیرایش جایگزین هستند (به عنوان مثال، یک اگزون در میانه‌ی یک مدل ژن که در یک ایزوفرم حضور داشته ولی در ایزوفرم دیگر نادیده گرفته می‌شود). این وضعیت سبب می‌شود که دو مسیر موجود در نمودار در ابتدا و انتها اشتراک داشته ولی یک شاخه نیز در بخش میانی داشته باشند. در زمان یافتن مسیرهای موجود در نمودارها، اطلاعات ترتیب k-مِر خوانش‌های تکی و جفت انتهایی به کار گرفته می‌شود. همچنین مرزها می‌توانند توسط فراوانی k-مِرها وزن‌دهی شوند که این کار مسیرهای خطا را کاهش خواهد داد. طول k-مِر بر پیچیدگی نمودار موثر است. اگر k-مِر خیلی کوچک باشد، طولش باید بسیار کوتاه‌تر از طول خوانش بوده و

1- Edge
2- Tip
3- Bubble

در نتیجه نمودار از نظر ارتباطات متراکم است. زیرا گره‌ها اختصاصی نیستند. ولی اگر k-مِر بزرگ باشد، باید اطلاعات کافی برای ایجاد نمودار موجود باشد. یک راه حل برای انتخاب یک مقدار مناسب برای k این است که چندین اسمبل با مقادیر مختلف k اجرا گردیده و بهترین اسمبل انتخاب شده و یا به جای آن کانتیگ‌های حاصل از چندین اسمبل با مقادیر مختلف k ترکیب گردند (۸، ۹ و ۱۰).

۵-۲-۵ استفاده از اطلاعات فراوانی

اگر مجموعه‌ی ایزوفرم‌های کاندیدا اندازه‌ی مناسبی داشته باشند، استفاده از اطلاعات فراوانی توالی‌یابی RNA برای حل و فصل کردن ایزوفرم‌ها امکان‌پذیر خواهد بود. زیرا فراوانی در همه‌ی اگزون‌های مربوط به یک رونوشت یکسان، بایستی مشابه باشد. یک رونوشت یک مولکول است و بنابراین اگر هیچ اُریبی در تهیه‌ی کتابخانه و توالی‌یابی (و مکان‌یابی) وجود نداشته باشد، خوانش‌های توالی بایستی یک رونوشت یکنواخت و کامل را پوشش داده و نمایش دهند. اگر انحرافاتی از این وضعیت وجود دارد، به عنوان مثال اگر برخی از اگزون‌ها عمق توالی بلندتری داشته باشند، نشان دهنده‌ی آن است که این اگزون‌ها در سایر ایزوفرم‌ها نیز حضور دارند.

برآورد فراوانی‌های نسبی برای یک مجموعه‌ی ثابت از ایزوفرم‌ها امکان‌پذیر است. در این مورد برای بهینه‌سازی باید فراوانی‌هایی را که به بهترین نحو داده‌ها را توصیف می‌کنند، مشخص نمود. این کار می‌تواند نخست با تنظیم مقادیر اولیه برای فراوانی‌ها (به عنوان مثال، با تفسیم فروانی‌ها به طور مساوی در بین همه‌ی ایزوفرم‌ها) و سپس تنظیم دقیق این پاسخ به صورت تکراری با استفاده از یک الگوریتم حداکثرسازی امیدریاضی[1] (EM) صورت گیرد. الگوریتم حداکثرسازی امید ریاضی در دهه‌ی ۱۹۷۰ معرفی شده (۱۱) و توسط Xing و همکاران در زمینه‌ی داده‌های ترانسکریپتومی مورد استفاده واقع گردید (۱۲). بهینه‌سازی شامل تکرار دو مرحله است: امید ریاضی (E) و حداکثرسازی (M). در مرحله‌ی E کلیه خوانش‌ها به تناسب و بر مبنای فراوانی به هر ایزوفرم تخصیص می‌یابند. در مرحله‌ی M فراوانی‌های نسبی ایزوفرم‌ها مجدداً محاسبه می‌شوند. این دو مرحله آنقدر تکرار می‌گردند تا مقادیر فراوانی براورد شده، دیگر تغییر ننمایند. در چنین شرایطی گفته می‌شود که الگوریتم به همگرایی[2] رسیده است. این پاسخ برای مجموعه‌ی ایزوفرم‌های ارائه شده بوده و اگر ایزوفرم‌های جدید به این مجموعه افزوده شوند، کلیه‌ی مقادیر می‌توانند تغییر کنند. به طور کلی چون الگوریتم EM یک مقدار بهینه‌ی موضعی را می‌یابد، در مواردی که چند

1- Expectation Maximum (EM)
2- Converge

مقدار بهینه‌ی موضعی وجود دارد، پاسخ حاصل به مقادیر اولیه بستگی خواهد داشت. ولی در مورد یک مدل خطی با پارامترهای غیرمنفی، تنها یک مقدار حداکثری وجود داشته و بنابراین مقدار بهینه‌ی موضعی همان مقدار بهینه‌ی کُلی نیز خواهد بود (۱۳). الگوریتم پایه‌ی EM را می‌توان به روش‌های متعددی تغییر داد. به عنوان مثال، در نرم‌افزار iReckon (۱۴)، برای کاهش تعداد بازسازی رونوشت‌های جعلی، الگوریتم تنظیم شده‌ی EM به کار برده می‌شود.

۵–۳ پیش‌پردازش داده‌ها

معمولاً کیفیت خوانش بازها به طرف انتهای خوانش کاهش می‌یابد. این موضوع از ویژگی‌های فناوری‌های توالی‌یابی نسل اول و دوم (سَنگِر، اِلومِنا، سولاید، ۴۵۴) بوده ولی الزاماً در فناوری‌های جدید توالی‌یابی (نظیر پاسیفیک بایوساینسز) دیده نمی‌شود. اگر کیفیت همردیفی در سرتاسر خوانش محاسبه شود، بخش کم کیفیت یک خوانش با خطاهای بیشتر، امتیاز همردیفی را کاهش می‌دهد. بنابراین با پیرایش بخش کم کیفیت خوانش، می‌توان تعداد خوانش‌های قابل مکان‌یابی را افزایش داد. همچنین اگر اسمبل کردن از نو بر مبنای همپوشان‌های خوانش جفتی بنا نهاده شود، پیرایش بخش‌های کم کیفیت خوانش‌ها مفید واقع می‌گردد. ولی در روش‌های مبتنی بر نمودار دوبران، دُم‌های خطای خوانش‌ها منجر به نوک‌ها و انتهاهای مرده در نمودار شده ولی چون نمودار بر مبنای k-مِرها پایه‌ریزی شده است، انتهای کم کیفیت خوانش بر k-مِرها در ابتدای خوانش تاثیر نمی‌گذارد. پیرایش خوانش‌ها نمودار را ساده کرده و تعداد خوانش‌های مرده را کاهش می‌دهد. اما در نظر گرفتن خطا و بخش‌های کم کیفیت خوانش‌ها به طور کامل مانع از اسمبل کردن نمی‌گردد. ولی حجم بسیار بالای داده‌های کم کیفیت می‌تواند بر اسمبل کردن تاثیر گذاشته و در هر حالت حجم بسیار بالای داده‌ها نیز منجر به کاهش سرعت محاسبه می‌گردد. خوانش‌های خطا تعداد گره‌ها در یک نمودار دوبران را افزایش داده و بدین ترتیب استفاده از حافظه را افزایش می‌دهند. در نتیجه بهتر است که ابتدا اسمبل کردن با کمک داده‌های اصلی صورت گیرد. با این کار راهی برای مقایسه‌ی نتایج و مشاهده‌ی اثر پیرایش باز می‌شود. همچنین نکته‌ی بسیار مهم این است که با این کار می‌توان خطاهای احتمالی در فرآیند پیرایش را نیز شناسایی نمود.

صرفنظر از روش اسمبل کردن، ورساخته‌های ناشی از ایجاد کتابخانه بایستی از خوانش‌ها حذف شوند. این ورساخته‌ها شامل توالی آداپتورها هستند که ممکن است در بخشی از خوانش‌های توالی باقی‌مانده باشند. همچنین اگر پُلی A در توالی‌یابی حضور داشته باشد، بایستی پیرایش گردد.

کاربر بایستی نحوه‌ی ساخت کتابخانه‌ی توالی‌یابی و نحوه‌ی جهت‌گیری خوانش‌ها را بداند. در خوانش‌های جفت انتهایی اِلومِنا، خوانش‌ها در مقابل یکدیگر قرار می‌گیرند. اطلاعات دیگر که لازم

است موجود باشد، خصوصیت زنجیره‌ی کتابخانه است. ساخت کتابخانه‌های توالی به نحوی که زنجیره‌ای که خوانش از آن حاصل شده است، معلوم باشد، امکان‌پذیر است. مزیت کتابخانه‌های مختص زنجیره این است که ژن‌های همپوشان که در زنجیره‌های مقابل حضور دارند را حل و فصل می‌نماید.

۵-۳-۱ تصحیح خطای خوانش

پاکسازی و پیرایش خوانش‌ها به معنای خلاص شدن از خطاهای توالی‌یابی از طریق حذف خوانش‌های کامل یا بخشی از آنها می‌باشد. این روش‌ها حجم داده‌های توالی‌یابی را کاهش می‌دهد. یک ایده‌ی کاملاً متفاوت این است که تلاش شود که خطاهای موجود در خوانش‌ها تصحیح گردد. اگر این کار با موفقیت صورت گیرد، داده‌های مفید بیشتری در دسترس خواهند بود.

یکی از کاربردهای اصلی تصحیح خوانش‌ها در اسمبل کردن از نو است. با استفاده از اسمبل‌سازهای مبتنی بر نمودار دوبران، هر k-مِر (در واقع (k-۱)-مِر) یک گره در نمودار به خود تخصیص می‌دهد. خطاهای توالی‌یابی منجر به ایجاد تعدادی k-مِر غلط و تولید گره‌های بدون استفاده می‌شوند که هر دو مورد فوق محاسبه را کُند کرده و مصرف حافظه را افزایش می‌دهند. ولی کلیه‌ی تنوع موجود در داده‌ها به دلیل وجود خطاهای توالی‌یابی از نوع تصادفی نیستند. در جانداران دیپلوئید و تری‌پلوئید ممکن است تنوع غیرتصادفی ناشی از تفاوت بین آلل‌ها دیده شود. در برخی از موارد، بهتر است که فوق تصحیح[1] انجام شده و این نوع از تنوع‌ها نیز حذف شوند. اگر SNP ها و ایندل‌ها از خوانش‌های توالی حذف شوند، داده‌ها همگن‌تر شده، نمودار دوبران ساده‌تر شده و کانتیگ‌های بلندتر می‌توانند ایجاد شوند. بعداً می‌توان با مکان‌یابی خوانش‌های تصحیح نشده اصلی در برابر کانتیگ‌ها، واریانت‌های توالی را نیز شناسایی کرد.

تصحیح خوانش بر مبنای استفاده از فراوانی در داده‌ها صورت می‌گیرد. برای انجام صحیح این کار، بایستی عمق کافی برای توالی‌یابی وجود داشته باشد. اگر خوانش‌ها به طور کامل و بدون خطا با ژنوم یا ترانسکریپتوم همردیف شوند، شناسایی خطاهای توالی‌یابی و تصحیح آنها بر مبنای رای اکثریت آسان خواهد بود. مشکل زمانی ایجاد می‌شود که هیچ مرجعی در دسترس نبوده و به دلیل وجود تکرارها یا سایر مناطق شبیه به هم، توالی‌های مشابه از بخش‌های مختلف ژنوم منشاء بگیرند.

SEECER ۲-۳-۵

نخستین نرم‌افزاری که برای تصحیح خطاهای موجود در داده‌های توالی‌یابی RNA پیشنهاد گردید، SEECER بوده است (۱۵). این نرم‌افزار خوانش‌ها را یک به یک تصحیح می‌کند. هر

1- Overcorrection

خوانشی که تصحیح می‌شود، سایر خوانش‌هایی هم که حداقل در یک k-مِر با آن اشتراک داشته باشند، انتخاب می‌شوند. برای جدا نمودن خوانش‌هایی که از رونوشت‌های مختلف می‌آیند، از خوشه‌بندی استفاده می‌شود. زیرمجموعه‌ای از خوانش‌ها برای ساخت یک مدل مارکوف مخفی[1] (HMM) که مدلی احتمالی برای نشان دادن گروه‌بندی توالی‌ها است، به کار گرفته می‌شوند. سپس خوانش‌ها با کمک الگوریتم ویتربی[2] با حالت‌های HMM همردیف شده و تصحیح خوانش بر مبنای اجماع HMM صورت می‌گیرد. همه‌ی توالی‌هایی که درستنمایی‌شان فراتر از یک آستانه در نظر گرفته شده باشند، نشان دهنده‌ی آن هستند که توالی‌های مزبور به خوبی با این مدل انطباق داشته و تصحیح می‌شوند. وقتی که یک خوانش تصحیح شد، از مجموعه‌ی توالی‌های در دسترس برای تصحیح حذف شده و این فرآیند برای داده‌های باقیمانده ادامه می یابد.

تصحیح خطا نیازمند حافظه بوده و نمی‌تواند با رایانه‌های خانگی معمولی اجرا شود. برای این کار و بسته به اندازه‌ی داده‌ها و طول خوانش، ده‌ها گیگا بایت رَم بایستی در دسترس باشد. SEECER را می‌توان از http://sb.cs.cmu.edu/seecer/ دانلود نمود. مراحل مورد نیاز برای تصحیح خطا در اسکریپت پوسته‌ی Bash ، run_seecer.sh ، که فایل‌های ورودی را در فرمت FASTA یا FASTQ دریافت می‌کند، پیاده‌سازی می‌شود. k-مِرها می‌توانند با استفاده از پیاده‌سازی داخلی یا نرم‌افزار خارجی Jellyfish محاسبه شوند. محاسبه k-مِرها به ویژه با مجموعه داده‌های بزرگ توصیه می‌گردد. طول پیش‌فرض برای k-مِر برابر با ۱۷ می‌باشد.

اجرای SEECER نیازمند GNU Scientific Library (GSL) است. برای نصب GSL در محل پیش‌فرض، به نرم افزار sudo نیاز است.

۱- Gsl-1.16.tar.gz را از آدرس http://fpmirror.gnu.org/gsl/ دریافت نمایید (در این مثال، نزدیک‌ترین آیینه‌ی fp عبارت است از: http://www.nic.funet.f/pub/gnu/fp.gnu.org/pub/gnu/gsl/).

```
$ tar xvfz gsl-1.16.tar.gz

$./configure

$ make

$ sudo make install
```

1- Hidden Markov Model (HMM)
2- Viterbi algorithm

۲- SEECER-0.1.2.tar.gz را از آدرس http://sb.cs.cmu.edu/seecer/install.html دریافت نمایید.

```
$./configure
$ make
```

۳- SEECER را اجرا کنید. گزینه‌ها را می‌توان با پارامتر h- فهرست نمایید.

```
$ bash bin/run _ seecer.sh -h
```

دایرکتوری موقت tmp را برای محاسبه و اجرای تصحیح خوانش‌ها ایجاد نمایید. فایل‌های reads1.fq و reads2.fq حاوی خوانش‌های جفت انتهایی هستند.

```
$ mkdir tmp
$ bash bin/run_seecer.sh -t tmp reads1.fq reads2.fq
```

خوانش‌های تصحیح شده در فرمت FASTA با پسوند corrected.fa- در همان دایرکتوری خوانش‌های اصلی حضور دارند.

۴-۵ اسمبل کردن مبتنی بر مکان‌یابی

در اینجا دو بسته‌ی نرم‌افزاری Cufflinks و Scripture که برای بازسازی کامل توالی‌های رونوشت بر مبنای مکان‌یابی خوانش‌های توالی‌یابی RNA به کار گرفته می‌شوند، معرفی می‌گردند. هر دو نرم‌افزار را می‌توان برای بازسازی از آغاز[1] رونوشت‌ها که به معنای عدم نیاز به در اختیار داشتن مدل‌های ژنی است، به کار گرفت. تفاوت اصلی بین این دو برنامه در روش حل و فصل کردن ایزوفرم‌ها است. Scripture کلیه‌ی ایزوفرم‌های ممکن را گزارش کرده ولی Cufflinks کوچک‌ترین مجموعه‌ی ممکن از ایزوفرم‌ها را که می‌توانند داده‌ها را توجیه نمایند، گزارش می‌کند. خروجی در فرمت BED یا GTF که حاوی مختصات رونوشت در یک توالی مرجع است، ارائه می‌شود. چون توالی مرجع معلوم است، لذا تبدیل مختصات توالی رونوشت به فایل FASTA با استفاده از هر زبان برنامه‌نویسی نظیر پایتون یا پِرل کاری ساده است.

مکان‌یابی می‌تواند توسط TopHat انجام شود که در اینجا از ویرایش 2.0 این نرم‌افزار استفاده شده است. داده‌های ورودی شامل یک فایل FASTA از کروموزوم شماره‌ی ۱۸ تحت عنوان chr18.fa و فایل‌های خوانش‌های جفت انتهایی chr18_1.fq و chr18_2.fq می‌باشد. فایل نمایه که با روش باروز – ویلر تبدیل شده است، تحت عنوان chr18 نام‌گذاری شده و خروجی مکان‌یابی در

1- *ab initio*

دایرکتوری top2 خواهد بود. چون خوانش‌ها ۷۵ × ۲ جفت باز بوده و اندازه‌ی الحاق قطعه نیز ۲۰۰ جفت باز می‌باشد، لذا فاصله‌ی داخلی بین خوانش‌ها ۵۰ جفت باز است که به صورت یک پارامتر با برهان r- در TopHat داده می‌شود. در اینجا مکان‌یابی با چهار جزء[1] انجام می‌گیرد. برای استفاده از TopHat باید هر دو نرم‌افزار SAMtools و Bowtie2 در دسترس باشند و مکان آنها نیز باید در متغیر PATH مشخص گردد.

```
$ bowtie2-build chr18.fa chr18
$ tophat2 -r 50 -p 4 -o top2 chr18 chr18_1.fq chr18_2.fq
```

Cufflinks ۱-۴-۵

Cufflinks در C++ نوشته شده است (۱۳). این نرم‌افزار به شکل فعالی به روز شده و آخرین ویرایش آنرا می‌توان از http://cuffinks.cbcb.umd.edu. دانلود نمود. وبسایت این نرم‌افزار حاوی راهنما و اطلاعات بیشتر راجع به آن است. Cufflinks ابتدا نموداری تشکیل می‌دهد که داده‌ها را به شیوه‌ای صرفه‌جویانه توجیه می‌کند. این بدان معناست که این نرم‌افزار کوچک‌ترین مجموعه از رونوشت‌ها را که می‌توانند خوانش‌های توالی‌یابی RNA را نشان دهند، می‌یابد. سپس فراوانی‌ها برای این مجموعه از رونوشت‌ها برآورد می‌شوند.

در زمان نگارش این کتاب، ویرایش 2.1.1 این نرم‌افزار ارائه شده بود و از فایل‌های BAM تولید شده توسط TopHat2 نیز پشتیبانی می‌کرد. برای استفاده از اطلاعات جفت انتهایی، نام خوانش‌ها در فایل‌های BAM نبایستی حاوی پسوندهای خوانش‌های جفتی باشند. علی‌رغم اینکه TopHat2 اطلاعات جفت انتهایی را در فایل BAM به نحو صحیح نشان می‌دهد، به این معنا که اگر هر دو انتها مکان‌یابی شده باشند، نشانه نمای « = » حضور دارد، ولی اگر آنها توسط جدا کننده‌ی[2] مورد نظر از هم تفکیک نشده باشند، پسوندهای خوانش‌های جفت انتهایی را حذف نمی‌کند. به عنوان مثال، پسوندهای /1 و /2 به صورت خودکار از نام خوانش‌ها حذف می‌گردند. ولی پسوندهای _1 و _2 حذف نمی‌گردند. برای اینکه Cufflinks بتواند از اطلاعات جفت انتهایی استفاده کند، هر دو خوانش در جفت خوانش بایستی تعریف کننده‌ی یکسانی در فایل BAM (نخستین ستون در فایل BAM) داشته باشند. این کار را می‌توان به آسانی با دستور view توسط SAMtools و با استفاده از فایل BAM به عنوان ورودی آن انجام داد. جهت استفاده از Cufflinks ، محل SAMtools باید در متغیر PATH موجود باشد.

1- Thread
2- Delimiter

چندین پارامتر را می‌توان تعریف کرد. برای ارتقای سرعت محاسبه، از چهار جزء در دستور زیر استفاده شده است. در غیر اینصورت پارامترهای پیش‌فرض به کار گرفته شده و دستور اجرای Cufflinks به صورت زیر خواهد بود:

```
$ cufflinks -p 4 -o outdir top2/accepted _ hits.bam
```

مدل‌های ژنی در یک فایل GTF در دایرکتوری خروجی ذخیره می‌شوند. چهار فایل خروجی وجود دارد:

```
-rw------- 1 somervuo 50K  Jul 15 10:43 genes.fpkm_tracking
-rw------- 1 somervuo 67K  Jul 15 10:43 isoforms.fpkm_tracking
-rw------- 1 somervuo 0    Jul 15 10:42 skipped.gtf
-rw------- 1 somervuo 898K Jul 15 10:43 transcripts.gtf
```

رونوشت‌های دارای اطلاعات اگزونی در فایل transcripts.gtf حضور دارند. در این مثال، ۷۵۰ رونوشت از ۶۳۴ ژن موجود است. این رونوشت‌ها و ژن‌ها به ترتیب در فایل‌های isoforms.fpkm_tracking و genes.fpkm_tracking فهرست شده‌اند.

اگر چند کتابخانه با اندازه‌های الحاق مختلف وجود داشته باشند، بهتر است که به جای اینکه ابتدا همه‌ی فایل‌های BAM به هم پیوند داده شده و سپس Cufflinks اجرا شود، Cufflinks برای هر کدام از آنها به صورت جداگانه اجرا شده و سپس نتایج‌شان با هم ادغام شوند. برنامه‌ی Cuffmerge می‌تواند برای ادغام چند اجرای Cufflinks به کار گرفته شود.

حجم ادغام قطعات جدا شده را می‌توان با برهان --overlap-radius کنترل نمود. مقدار پیش‌فرض ۵۰ جفت باز است. مقادیر بزرگ‌تر منجر به ادغام مدل‌های ژنی دورتر می‌گردد.

در مثال فوق هیچ‌گونه اطلاعی از مدل‌های ژنی مورد استفاده در دسترس نیست. اگر چنین اطلاعاتی موجود باشد، می‌توان آنها را با استفاده از برهان g- در قالب یک فایل GTF و به صورت یک راهنما به Cufflinks ارائه نمود.

برای مقایسه‌ی خروجی Cufflinks با مدل‌های ژنی موجود، می‌توان از برنامه‌ی Cuffcompare استفاده نمود. اگر مدل‌های ژنی مرجع در فایل ref.gtf موجود باشد، دستور لازم برای استفاده از این برنامه عبارت است از:

```
$ cuffcompare -r ref.gtf transcripts.gtf
```

فایل‌های خروجی حاوی خلاصه و اطلاعات ژنی برای شباهت بین مدل‌های ژنی در دو فایل است.

Scripture ۲-۴-۵

Scripture یک نرم‌افزار مبتنی بر جاوا است (۱۶). این نرم‌افزار را می‌توان از http://www.broadinstitute.org/sofware/scripture/ دانلود کرد. Scripture داده‌ها را بر مبنای اطلاعات خوانش خُرد می‌کند. نواحی از ژنوم با ارتباطات خوانش‌های خُرد شده، جزایری را تشکیل می‌دهند که می‌توانند با استفاده از اطلاعات خوانش‌های جفت انتهایی، بیشتر مرتبط گردند. ایزوفرم‌های موجود در این نواحی گزارش می‌شوند.

Scripture کار را با ساخت یک نمودار اتصالی شروع می‌کند. این نمودار حاوی کلیه‌ی بازهای ژنوم مرجع بوده و این بازها، گره‌های نمودار را تشکیل می‌دهند. اگر در یک ژنوم یا در یک رونوشت، دو باز متناظر با دو گره، همجوار باشند، آن دو گره متصل تلقی می‌گردند. خوانش‌های خُرد شده اطلاعات مرزهای اگزون – اینترون را ارائه کرده و هر ارتباطی باید توسط حداقل دو خوانش توالی‌یابی RNA پشتیبانی گردد. مناطق پیرایشی دهنده/گیرنده‌ی مجاز عبارت از GT/AG استاندارد و GC/AG و AT/AC غیراستاندارد هستند. این مسیرها در نمودار اتصالی از نظر مقداری که غنی‌سازی شده‌اند در مقایسه با توزیع مکان‌یابی خوانش زمینه، برای معنی‌داری آماری‌شان ارزیابی می‌گردند. این کار با پویش نمودار مزبور با استفاده از پنجره‌های با اندازه‌ی ثابت و تخصیص مقدار p به هر پنجره انجام می‌شود. پنجره‌های معنی‌دار برای ایجاد یک نمودار رونوشتی که با کمک خوانش‌های جفت انتهایی و با هدف پیوستن به قطعات جدا شده‌ی قبلی، تصفیه شده‌اند، ادغام می‌گردند.

داده‌های ورودی برای Scripture شامل یک فایل BAM مرتب شده و یک فایل FASTA کروموزوم مرجع است. ویرایش 2.0 نرم‌افزار Scripture ویرایش جدید این نرم‌افزار بوده و در زمان نگارش این کتاب هنوز به صورت عمومی عرضه نشده است. ولی نسخه‌ی اولیه‌ی آن به دست نویسندگان این کتاب رسیده است. دستور نگارش عبارت است از:

```
$ java –jar ScriptureVersion2.0.jar –task reconstruct
–alignment top2/accepted_hits.bam –genome chr18.fa–out out
–strand unstranded –chr 18
```

خروجی ویرایش قبلی Scripture شامل دو فایل است. یکی از فایل‌ها حاوی مدل‌های ژنی در فرمت BED و فایل دیگر حاوی نمودارهای رونوشتی در فرمت DOT است. در ویرایش 2.0 ، چهار فایل خروجی وجود دارد. علاوه بر این، ویرایش تازه یک فایل مختصات در همان دایرکتوری که فایل BAM واقع شده است، می‌سازد. این چهار فایل خروجی عبارتند از:

```
-rw------- 1 somervuo 80K  Jul 8 15:13 out.connected.bed
-rw------- 1 somervuo 250K Jul 8 14:09 out.pairedCounts.txt
-rw------- 1 somervuo 229K Jul 8 14:09 out.pairedGenes.bed
-rw------- 1 somervuo 104K Jul 8 14:09 out.scripture.paths.bed
```

فایل out.scripture.paths.bed رونوشت‌های اولیه‌ای را که تنها از اطلاعات خوانش‌های تکی و فایل out.connected.bed رونوشت‌هایی را که در آنها از اطلاعات جفت انتهایی استفاده شده است، گزارش می‌نمایند. در فایل out.connected.bed ، ۵۴۹ رونوشت از ۵۰۴ ژن وجود دارد.

۵–۵ اسمبل کردن از نو

در اینجا دو بسته‌ی نرم‌افزاری که برای بازسازی از نو توالی‌های کامل رونوشت، بدون استفاده از ژنوم مرجع، معرفی می‌گردند. هر دو نرم‌افزار مزبور از نمودار دوبران استفاده می‌کنند. نخستین بسته‌ی نرم‌افزاری شامل دو برنامه‌ی Velvet و Oases است. Velvet یک اسمبل‌ساز ژنومی است که یک نمودار اسمبل ایجاد می‌کند. این نمودار توسط برنامه‌ی Oases و برای یافتن مسیرهایی که نشان دهنده‌ی ایزوفرم‌ها هستند، به کار گرفته می‌شود. برنامه‌ی دیگر برای اسمبل کردن Trinity نام دارد که شامل سه ماژول است. ابتدا خوانش‌های توالی‌یابی RNA اسمبل شده و خوشه‌بندی می‌گردند. هر خوشه نشان دهنده‌ی یک ژنگاه در ژنوم است. برای هر خوشه یک نمودار دوبران ترسیم شده و توالی‌های رونوشت خطی استخراج می‌گردند. بدین ترتیب از یک ژنگاه می‌تواند چندین ایزوفرم وجود داشته باشد. هر دو ابزار فوق پیش از اسمبل کردن، داده‌های توالی‌یابی را در یک فایل کُپی کرده و در نتیجه اگر مجموعه‌ی داده‌های مورد استفاده بزرگ باشد، لازم است که فضای خالی دیسک، قبل از شروع اسمبل کردن بررسی شود.

Velvet + Oases ۱-۵-۵

Velvet در C نگارش یافته است. این نرم‌افزار به عنوان یک اسمبل‌ساز ژنومی معرفی شده است (۱۷). نرم‌افزار دیگر که Oases نامیده می‌شود برای اسمبل کردن رونوشت‌ها نگارش یافته و از خروجی Velvet استفاده می‌کند (۹). Velvet را می‌توان از http://www.ebi.ac.uk/~zerbino/velvet/ دانلود کرده و Oases را نیز از http://www.ebi.ac.uk/~zerbino/oases/ دانلود نمود. هر دو بسته‌ی نرم‌افزاری دارای راهنماهای خوبی هستند.

Velvet شامل دو برنامه‌ی velveth و velvetg است. velveth ، k-مِرهای داده‌ها را محاسبه کرده و velvetg ، کانتیگ‌ها را در یک نمودار دوبران یافته و استخراج می‌کند. Oases نمودار مزبور را قطعه‌بندی کرده و ایزوفرم‌ها را برای هر ژنگاه استخراج می‌نماید. معمولاً توالی‌های رونوشت

حاصل از Oases در مقایسه با کانتیگ‌های Velvet بسیار بلندتر هستند. برای استفاده از خوانش‌های جفت انتهایی در Velvet، این خوانش‌ها باید به نحوی تغییر داده شوند که هر دو خوانش موجود در یک جفت خوانش به صورت همجوار در یک فایل جا داده شوند. این فرآیند را اصطلاحاً میان جادهی[1] می‌نامند. در بسته‌ی نرم‌افزاری Velvet یک اسکریپت پِرل برای اجرای میان جادهی در نظر گرفته شده است. البته اجرای این فرآیند منوط به آن است که جفت خوانش‌ها از ابتدا در دو فایل جداگانه ذخیره شده باشند. دستور زیر یک فایل جدید با عنوان chr18_12.fq ایجاد می‌کند که در آن، خوانش‌ها میان جادهی شده‌اند.

```
$ shuffleSequences_fastq.pl chr18_1.fq chr18_2.fq chr18_12.fq
```

در گام نخست باید یک جدول هَش[2] تشکیل گردد. در اینجا طول k-مِر ۲۵ تعریف شده و دایرکتوری خروجی نیز vdir خواهد بود. فرمت داده‌ها نیز تعریف شده و خوانش‌های جفت انتهایی در فرمت FASTQ می‌باشند. گذرگاه نمودار و استخراج کانتیگ در دومین مرحله انجام خواهد شد. اندازه‌ی الحاق نیز ۲۰۰ جفت باز در نظر گرفته شده است. اندازه‌ی الحاق در Velvet، طول قطعه است. این بدان معناست که این اندازه شامل طول خوانش‌ها نیز می‌باشد. نکته‌ی مهم آن است که برای برهان read_trkg- پارامتر yes در نظر گرفته شود. زیرا Oases از اطلاعات ردیابی خوانش استفاده می‌نماید.

```
$ velveth vdir 25 –fastq –shortPaired chr18_12.fq
$ velvetg vdir –ins_length 200 –read_trkg yes
```

Oases برای ایجاد نمودار دوبران به کار برده می‌شود. ورودی Oases نام دایرکتوری است که حاوی خروجی Velvet می‌باشد. در مورد خوانش‌های جفت انتهایی، اندازه‌ی الحاق نیز باید تعریف شود. در اینجا حداقل اندازه‌ی رونوشت برابر با ۲۰۰ جفت باز تعریف شده است.

```
$ oases vdir –ins_length 200 –min_trans_lgth 200
```

دایرکتوری خروجی vdir حاوی فایل‌هایی است که در زیر نشان داده شده‌اند. توالی‌های رونوشت در یک فایل FASTA با عنوان transcripts.fa ذخیره شده‌اند. نام هر مدخل[3] FASTA نشان دهنده‌ی ژنگاه و ایزوفرم است. فایل دیگری که توسط Oases ایجاد می‌گردد، contig-ordering.txt می‌باشد.

1- Interleave
2- Hash table
3- Entry

```
-rw------- 1 somervuo 25M  Jul 16 11:56 Graph2
-rw------- 1 somervuo 11M  Jul 16 11:59 LastGraph
-rw------- 1 somervuo 1.2K Jul 16 11:59 Log
-rw------- 1 somervuo 5.5M Jul 16 11:56 PreGraph
-rw------- 1 somervuo 34M  Jul 16 11:55 Roadmaps
-rw------- 1 somervuo 84M  Jul 16 11:55 Sequences
-rw------- 1 somervuo 1.3M Jul 16 11:59 contig-ordering.txt
-rw------- 1 somervuo 2.6M Jul 16 11:56 contigs.fa
-rw------- 1 somervuo 253K Jul 16 11:59 stats.txt
-rw------- 1 somervuo 1.6M Jul 16 11:59 transcripts.fa
```

به عنوان مثال نام یکی از مدخل‌های موجود در FASTA عبارت از Locus_10_Transcript_1/3_Confidence_0.571_Length_3815 است. این نام نشان می‌دهد که سه رونوشت از ژنگاه ۱۰ وجود داشته و این رونوشت، نخستین رونوشت از آنها است. مقدار اطمینان (Confidence) عددی بین ۰ و ۱ بوده (هر قدر بالاتر باشد، بهتر است) و طول (Length) نیز نشان دهنده‌ی طول رونوشت بر حسب جفت باز می‌باشد. در این مثال، فایل transcripts.fa حاوی ۱۳۰۸ توالی رونوشت با حداقل طول ۲۰۰ جفت باز از ۸۶۲ ژنگاه است.

در ویرایش 2.0 نرم‌افزار Oases اجرای چندین اسمبل با طول‌های k-مِر متفاوت امکان‌پذیر بوده و این نرم‌افزار، اسمبل‌ها را با هم ادغام می‌نماید. بدین منظور در بسته‌ی نرم‌افزاری Oases ، یک اسکریپت پایتون وجود دارد. در اینجا به گونه‌ای تعریف می‌شود که کلیه‌ی k-مِرهای فرد از ۱۹ تا ۲۹ استفاده خواهند شد. پارامترهای اضافی برای Velvet و Oases با برهان‌های d- و p- ارائه می‌گردند. با به کار گرفتن این اسکریپت پایتون، نیازی به استفاده از برهان read_trkg- نیست.

```
$ python oases_pipeline.py -m 19 -M 29 -o odir -d " -fastq -
  shortPaired chr18_12.fq" -p " -ins_length 200 -min_trans_
  lgth 200"
```

این دستور خروجی جداگانه‌ای برای هر k-مِر ایجاد کرده و یک دایرکتوری تحت عنوان odirMerged که حاوی نتایج اسمبل ادغام شده است، نیز تولید می‌کند. در این مثال، فایل transcripts.fa در dirMerged حاوی ۴۴۶۸ توالی رونوشت از ۸۲۷ ژنگاه است. بعد از ایجاد دایرکتوری‌های خروجی برای هر k-مِر، ادغام نمودن تنها برخی از اسمبل‌ها بدون شروع از ابتدا امکان‌پذیر است. این کار با استفاده از برهان r- در اسکریپت پایتون صورت می‌گیرد. به عنوان مثال اگر تنها اسمبل‌های با k = ۲۵ و بزرگ‌تر ادغام شوند، از دستور زیر استفاده خواهد شد.

```
$ python oases_pipeline.py -m 25 -M 29 -r -o odir
```

این دستور ۲۱۵۹ رونوشت از ۷۸۳ ژنگاه ایجاد می‌نماید. بر اساس پیش‌فرض، حداکثر طول k-مِر در Velvet برابر با ۳۱ است. ولی می‌توان از مقادیر بزرگ‌تر نیز استفاده نمود. به عنوان مثال، اگر مقادیر k بیش از ۵۱ مورد نیاز باشند، Velvet می‌تواند با دستور زیر همگردانی[1] شود (و برای همگردانی Oases هم باید از همین دستور استفاده کرد).

```
$ make 'MAXKMERLENGTH=51'
```

وقتی که چندین اسمبل با مقادیر مختلف برای k اجرا شوند، استفاده از برهان‌های m- و M- بیش از روانه ساختن جداگانه‌ی هر اسمبل مفید خواهد بود. زیرا Velvet داده‌های خوانش را در فایل Sequences کُپی می‌کند. ولی با پارامترهای m- و M- این کُپی کردن تنها یک بار انجام شده و سایر دایرکتوری‌ها حاوی لینک‌های سمبلیک به فایل موجود در نخستین دایرکتوری خروجی خواهند بود.

Trinity ۲-۵-۵

بسته‌ی نرم‌افزاری Trinity (۱۸) را می‌توان از http://trinityrnaseq.sourceforge.net/ دانلود نمود. این وبسایت حاوی اطلاعات زیاد در مورد روش کار و موضوعات پیشرفته است. گردش کار مبتنی بر Trinity که شامل آنالیزهای پایین دستی نیز می‌باشد، توسط Haas و همکاران تشریح شده است (۱۹).

Trinity شامل سه برنامه‌ی جداگانه است: ۱) Inchworm که کانتیگ‌های اولیه را می‌سازد، ۲) Chrysalis که کانتیگ‌های تولید شده توسط Inchworm را خوشه‌بندی کرده و یک نمودار دوبران برای هر ژنگاه می‌سازد و ۳) Butterfly که ایزوفرم‌ها را در داخل هر نمودار دوبران استخراج می‌کند. در Trinity به جای کلمه‌ی «ژنگاه» (Locus) از کلمه‌ی «مولفه» (Component) استفاده می‌شود. اگر اینطور به نظر برسد که خوانش‌های توالی از بیش از یک ژنگاه آمده‌اند، ممکن است که در طی مرحله‌ی Butterfly یک مولفه‌ی تولید شده توسط Chrysalis به قطعات کوچک‌تر تقسیم شود. اگر این اتفاق بیفتد، این موضوع در نام‌های توالی‌های رونوشت خروجی گزارش می‌گردد.

هر سه برنامه می‌توانند با استفاده از اسکریپت پِرل Trinity.pl اجرا شوند. دستور زیر به گونه‌ای تعریف شده است که توالی‌ها در فرمت FASTQ بوده و تعداد پردازنده‌ها برای محاسبه برابر با ۴ می‌باشد. در ویرایش فعلی Trinity ، طول k-مِر ثابت و برابر با ۲۵ است. Jellyfish که نرم‌افزار مورد استفاده برای محاسبه‌ی k-مِرها می‌باشد، نیازمند تعریف حداکثر حافظه است که در این مثال برابر

1- Compile

با ۱۰ گیگا بایت تنظیم شده است. به طور پیش‌فرض، کوتاه‌ترین خوانش گزارش شده، برابر با ۲۰۰ جفت باز است. قبل از اجرایTrinity، اندازه‌ی برآمدگی[1] بایستی به صورت نامحدود تعریف شود. برای این کار بسته به توزیع لینوکس می‌توان از دستور پوسته‌ی `unlimit` یا `ulimit -s unlimited` استفاده نمود. تفاوت‌هایی بین ویرایش‌های مختلف Trinity وجود دارد. به عنوان مثال، در ویرایش‌های قدیمی‌تر لازم است که کاربر مشخص کند که کدام‌یک از روش‌های k-مِر را استفاده می‌نماید. همچنین تفاوت‌هایی در تعداد فایل‌های خروجی وجود دارد. در زمان نگارش این کتاب، آخرین ویرایش این نرم‌افزار، ویرایش r2013-02-25 بوده که در آن به صورت پیش‌فرض از روش Jellyfish استفاده می‌شود. دستور اجرای Trinity با پارامترهای پیش‌فرض و چهار CPU به شرح زیر است.

```
$ Trinity.pl --seqType fq --JM 10G --left chr18_1.fq--right
  chr18_2.fq --CPU 4
```

اگر هیچ نامی برای دایرکتوری خروجی تعریف نشده باشد، به صورت پیش‌فرض trinity_out_dir در نظر گرفته می‌شود. پس از اینکه فرآیند Butterfly به اتمام رسید، دایرکتوری خروجی حاوی یک فایل FASTA تحت عنوان Trinity.fasta خواهد بود که شامل همه‌ی ایزوفرم‌ها می‌باشد. ممکن است که یک یا چند نمودار در مرحله‌ی Butterfly هیچ توالی رونوشتی تولید نکنند. ولی کلیه‌ی اطلاعات در دایرکتوری خروجی و دایرکتوری فرعی chrysalis ذخیره می‌گردد. برای هر مولفه که حاوی توالی‌های رونوشتش است، یک فایل FASTA جداگانه وجود دارد. ساختار نمودار مولفه نیز ذخیره می‌گردد. اگر Butterfly در ایجاد هر توالی رونوشت برای مولفه مردود گردد، فایل FASTA متناظر وجود دارد. ولی اندازه‌اش صفر است. با استفاده از داده‌های مثال، دایرکتوری خروجی به صورت زیر خواهد بود.

```
-rw------- 1 somervuo 2.2M Dec 18 15:13 Trinity.fasta
-rw------- 1 somervuo  583 Dec 18 15:13 Trinity.timing
-rw------- 1 somervuo  78M Dec 18 14:56 both.fa
-rw------- 1 somervuo    7 Dec 18 14:56 both.fa.read_count
-rw------- 1 somervuo 159M Dec 18 14:59 bowtie.nameSorted.sam
-rw------- 1 somervuo    0 Dec 18 14:59 bowtie.nameSorted.sam.
  finished
-rw-------  1 somervuo    0  Dec 18 14:59 bowtie.out.finished
drwx------ 3 somervuo 4.0K  Dec 18 15:04 chrysalis
-rw-------  1 somervuo 3.6M  Dec 18 14:58 inchworm.K25.L25.DS.fa
-rw-------  1 somervuo    0  Dec 18 14:58 inchworm.K25.L25.DS.fa.
  finished
```

1- Stack size

```
-rw------- 1 somervuo     8 Dec 18 14:58 inchworm.kmer_count
-rw------- 1 somervuo 148 K Dec 18 14:59 iworm_scaffolds.txt
-rw------- 1 somervuo     0 Dec 18 14:59 iworm_scaffolds.txt.
  finished
-rw------- 1 somervuo    0 Dec 18 14:57 jellyfish.1.finished
-rw------- 1 somervuo 125M Dec 18 14:57 jellyfish.kmers.fa
-rw------- 1 somervuo  13M Dec 18 14:59 scaffolding_entries.sam
-rw------- 1 somervuo 6.3M Dec 18 14:59 target.1.ebwt
-rw------- 1 somervuo 279K Dec 18 14:59 target.2.ebwt
-rw------- 1 somervuo 170K Dec 18 14:58 target.3.ebwt
-rw------- 1 somervuo 557K Dec 18 14:58 target.4.ebwt
lrwxrwxrwx 1 somervuo   73 Dec 18 14:58 target.fa ->/.../
  inchworm.K25.L25.DS.fa
-rw------- 1 somervuo    0 Dec 18 14:59 target.fa.finished
-rw------- 1 somervuo 6.3M Dec 18 14:59 target.rev.1.ebwt
-rw------- 1 somervuo 279K Dec 18 14:59 target.rev.2.ebwt
```

در این مثال، فایل Trinity.fasta حاوی ۱۸۳۷ رونوشت از ۱۲۹۳ مولفه است. برای مقایسه، وقتی که با استفاده از برهان min_kmer_cov2- در خط دستور، حداقل پوشش k-مِر از ۱ (که پیش‌فرض است) به ۲ افزایش می‌یابد، اسمبل حاصل حاوی ۱۲۰۵ رونوشت در ۸۴۸ مولفه خواهد بود. در دایرکتوری خروجی، فایل both.fa حاوی کلیه‌ی داده‌های توالی ورودی است. در اینجا دو فایل خوانش جفت انتهایی بدون اطلاعات کیفیت پیوند داده می‌شوند. دایرکتوری chrysalis حاوی فایل‌های زیر است:

```
drwx------- 4 somervuo 4.0K Dec 18 15:02 Component_bins
-rw------- 1 somervuo     0 Dec 18 15:00 GraphFromIwormFasta.
  finished
-rw------- 1 somervuo  2.1M Dec 18 15:00 GraphFromIworm Fasta.out
-rw------- 1 somervuo  1.5M Dec 18 15:00 bundled_iworm_contigs.
  fasta
-rw------- 1 somervuo   57M Dec 18 15:02 bundled_iworm_contigs.
  fasta.deBruijn
-rw------- 1 somervuo     0 Dec 18 15:00 bundled_iworm_contigs.
  fasta.finished
-rw------- 1 somervuo  507K Dec 18 15:03 butterfly_commands
-rw------- 1 somervuo 507 K Dec 18 15:13 butterfly_commands.
  completed
-rw------- 1 somervuo     0 Dec 18 15:02 chrysalis.finished
-rw------- 1 somervuo  138K Dec 18 15:03 component_base_listing.
  txt
-rw------- 1 somervuo     0 Dec 18 15:03 file_partitioning.ok
-rw------- 1 somervuo 643 K Dec 18 15:03 quantifyGraph_commands
-rw------- 1 somervuo 643 K Dec 18 15:04 quantifyGraph_commands.
  completed
-rw------- 1 somervuo     0 Dec 18 15:04 quantifyGraph_commands.
  run.finished
-rw------- 1 somervuo     7 Dec 18 15:02 rcts.out
```

```
-rw------- 1 somervuo   0 Dec 18 15:02 readsToComponents.finished
-rw------- 1 somervuo 79M Dec 18 15:02 readsToComponents.out.sort
-rw------- 1 somervuo   0 Dec 18 15:02 readsToComponents.out.
   sort.finished
```

دایرکتوری Component_bins حاوی دایرکتوری‌های فرعی Cbin0 ، Cbin1 و . . . است. تعداد این دایرکتوری‌های فرعی به تعداد کُل مولفه‌های حاصل از اسمبل بستگی دارد. یک مثال از مولفه‌ای که بیش از یک رونوشت ایجاد می‌کند، در زیر ارائه شده است. فایل‌های با پیشوند dot. مربوط به مصورسازی نمودارها هستند. در ویرایش فعلی Trinity (ویرایش r2013-02-25)، این فایل‌ها به صورت پیش‌فرض تولید نشده و در زمان اجرای Trinity با استفاده از برهان ‘-V 5’ -bfly_opts در خط فرمان، می‌توان آنها را ایجاد نمود.

```
-rw------- 1 somervuo 5.9K Dec 18 16:49 c420.graph.allProb
   Paths.fasta
-rw------- 1 somervuo 134K Dec 18 16:23 c420.graph.out
-rw------- 1 somervuo 1.8M Dec 18 16:23 c420.graph.reads
-rw------- 1 somervuo  492 Dec 18 16:49 c420.graph_final
   CompsWOloops.L.dot
-rw------- 1 somervuo  492 Dec 18 16:49 c420.graph_withLoops.
   J.dot
```

سه توالی رونوشت در فایل c420.graph.allProbPaths.fasta وجود دارد که نام‌هایشان عبارتند از:

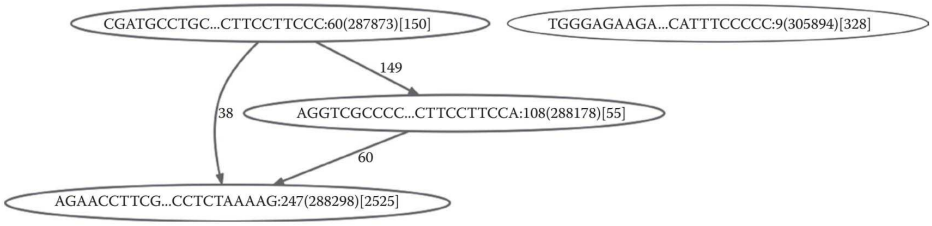
```
>c420.graph_c0_seq1 len= 328 path=[305894:0-327]
>c420.graph_c1_seq1 len= 2675 path=[287873:0-149 288298:150-2674]
>c420.graph_c1_seq2 len= 2730 path=[287873:0-149 288178:150-204
   288298:205-2729]
```

دو مولفه‌ی فرعی c0 و c1 وجود دارند که بدان معنا هستند که در طی فرآیند Butterfly مولفه‌ی اصلی c420 به دو مولفه‌ی فرعی تقسیم شده است. نام نیز شامل طول توالی رونوشت و مسیر گره در یک نمودار دوبران است. در نگاره‌ی ۵-۱ ، فایل نمودار c420.graph_withLoops.J.dot با استفاده از برنامه‌ی GraphViz نمایش داده شده است. رونوشت بلندتر (seq2) حاوی یک قطعه‌ی ۵۵ جفت بازی است که در رونوشت کوتاه‌تر (seq2) در مولفه‌ی c1 حضور ندارد. یک رونوشت ۳۲۸ جفت بازی متناظر با c0 در گوشه‌ی بالا سمت راست واقع شده است.

برای مقایسه‌ی بلندترین رونوشت c420 با رونوشت‌های معلوم، رونوشت مزبور با کمک مرورگر ژنومی UCSC (http://genome.ucsc.edu/) با ژنوم انسان مکان‌یابی شد. بهترین توفیق[1] BLAT

1- Hit

متعلق به کروموزوم ۱۸ بود. در نگاره‌ی ۵-۲، توالی رونوشت Trinity روی برچسب بالایی تحت عنوان YourSeq نمایش داده شده است.

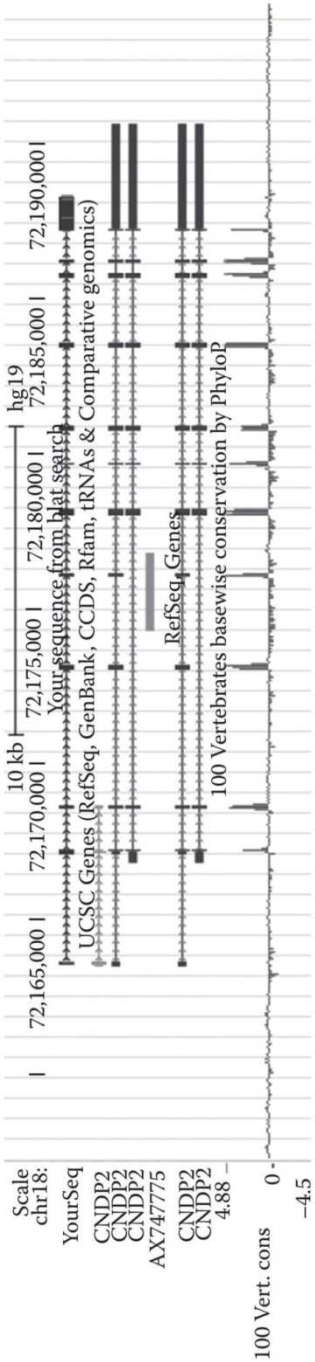


نگاره‌ی ۵-۱: نمودار رونوشت مثال، حاصل از اسمبل کردن توسط Trinity

اسمبل کردن رونوشت‌ها در Chipster

دسته‌ی ابزارهای RNA-seq ، بسته‌ی نرم‌افزاری Cufflinks را برای اسمبل کردن ترانسکریپتوم ارائه می‌دهد.

- فایل BAM ایجاد شده توسط TopHat2 در فصل چهارم و ابزار Assemble reads into transcripts using Cufflinks را انتخاب نمایید. توجه کنید که اگر می‌خواهید از حاشـیه‌نگاری موجود به عنوان راهنما بهـره ببرید، می‌توانید از یک فایل GTF به عنوان ورودی استفاده کنید. با حرکت از ابتدا به انتهای پنل پارامتر، از تخصیص صحیح فایل‌ها اطمینان حاصل کرده و سپس روی Run کلیک نمایید. باید توجه شود که این پارامترها امکان تصحیح برآورد فراوانـی برای خوانش‌های چندگانه و اُریبی مختص توالی را نیز فراهم می‌آورند.
- مدل‌های ژنی را می‌توان با گشودن فایل خروجی transcripts.gtf در مرورگر ژنومی Chipster و بر مبنای روش تشریح شده در فصل چهارم مصورسازی نمود. برای هدایت و حرکت موثر از یک رونوشت به رونوشت دیگر، فایل GTF مزبور را در یک پنجره‌ی جداگانه باز کرده و روی مختصات شروع کلیک کنید.
- می‌توان اسمبل‌ها را از چندین نمونه و با استفاده از ابزار Cuffmerge ادغام کرده و آنها را با کمک ابزار Cuffcompare با هم مقایسه نمود.



نگاره‌ی ۵-۲: رونوشت اسمبل شده‌ی YourSeq مکان‌یابی شده روی ژنوم انسان و نمایش داده شده در مرورگر ژنومی UCSC

۵–۶ خلاصه

در این فصل روش پایه و چهار نرم‌افزار برای بازسازی توالی‌های رونوشت بر مبنای خوانش‌های کوتاه توالی‌یابی RNA تشریح شده است. تعداد ژنگاه‌ها و رونوشت‌های حاصل از هر اسمبل با داده‌های نمونه با هدف تکرارپذیر بودن گزارش گردیده است. البته از موارد فوق برای مقایسه‌ی روش‌ها استفاده نشده است. مقایسه‌ی چندین ابزار برای بازسازی رونوشت، شامل Cufflinks و Oases توسط Steijger و همکاران ارائه گردیده است (۲۰). اسمبل‌های از نو مبتنی بر Trinity و Oases توسط francis و همکاران آنالیز شده‌اند (۲۱). همچنین Schulz و همکاران Cufflinks را نیز در این مقایسه‌ها وارد کرده‌اند.

علاوه بر داده‌ها و پیش‌پردازش، خروجی اسمبل کردن به تنظیم پارامترهای اختصاصی هر نرم‌افزار بستگی دارد. تعداد رونوشت‌ها را می‌توان به آسانی با تغییر در پارامترهای مرتبط با حداقل طول کانتیگ و پوشش تغییر داد. ولی تعداد و طول رونوشت‌ها هیچ‌گونه اطلاعی از صحت و خطاهای اسمبل کردن ارائه نمی‌دهد. در حقیقت سنجش کیفیت یک اسمبل کار ساده‌ای نیست. مشکل کار به ویژه از جایی شروع می‌شود که هیچ‌گونه مدل ژنی مرجع یا معلومی از گذشته در دسترس نباشد. در عمل، یک تبادل و تعادل بین اختصاصی بودن و حساسیت وجود دارد. سخت‌گیری در اسمبل کردن بایستی خطاها را کاهش دهد. ولی اگر پوشش کافی وجود نداشته باشد، کانتیگ‌ها کوتاه شده و رونوشت‌های بلند، قطعه قطعه می‌شوند. علی‌رغم اینکه هر دو روش مکان‌یابی و اسمبل کردن از نو قادرند که رونوشت‌های کامل را از خوانش‌های مرجع کوتاه بازسازی نمایند، ولی لازم به ذکر است که این کار به کیفیت داده‌ها و پوشش بستگی دارد. با پیشرفت فناوری و امکان‌پذیر شدن توالی‌یابی رونوشت کامل در یک خوانش، چالش‌های متعددی در اسمبل کردن رونوشت ایجاد می‌شود. در حال حاضر، چنین خوانش‌هایی از توالی‌یاب‌های نسل سوم پاسیفیک بایوساینسز به دست می‌آیند. علی‌رغم اینکه این فناوری خوانش‌های بلند ایجاد می‌نماید، ولی نقطه‌ی ضعفش این است که عمق توالی‌یابی به ازای هر اجرا پایین بوده و در نتیجه در حال حاضر پلتفرم‌های دیگر در مقایسه با این پلتفرم مقرون به صرفه‌تر هستند. به همین دلیل، توالی‌یابی نسل دوم و روش‌های تشریح شده در این فصل را می‌توان در حال حاضر و نیز احتمالاً همزمان با پیشرفت ابزارهای جدید به کار گرفت.

منابع

1. Adams M.D., Kelley J.M., Gocayne J.D., et al. Complementary DNA sequencing: Expressed sequence tags and human genome project. Science 252(5013):1651–1656, 1991.

2. Quackenbush J., Liang F., Holt I., Pertea G., and Upton J. Te TIGR gene indices: Reconstruction and representation of expressed gene sequences. Nucleic Acids Research 28(1):141–145, 2000.
3. de Bruijn N.G. A combinatorial problem. Koninklijke Nederlandse Akademie v. Wetenschappen 49:758–764, 1946.
4. Pevzner P., Tang H., and Waterman M. An Eulerian path approach to DNA fragment assembly. Proceedings of the National Academy of Sciences of UnitedStates of the America 98(17):9748–9753, 2001.
5. Kececioglu J. and Myers E. Combinatorial algorithms for DNA sequence assembly. Algorithmica 13:7–51, 1995.
6. Chevreux B., Pfsterer T., Drescher B., et al. Using the miraEST assembler for reliable and automated mRNA transcript assembly and SNP detection in sequenced ESTs. Genome Research 14:1147–1159, 2004.
7. Compeau P., Pevzner P., and Tesler G. How to apply de Bruijn graphs to genome assembly. Nature Biotechnology 29(11):987–991, 2011.
8. Robertson G., Schein J., Chiu R., et al. De novo assembly and analysis of RNA sequence data. Nature Methods 7(11):909–912, 2010.
9. Schulz M.H., Zerbino D.R., Vingron M., and Birney E. Oases: Robust de novo RNA-seq assembly across the dynamic range of expression levels. Bioinformatics 28(8):1086–1092, 2012.
10. Surget-Groba Y. and Montoya-Burgos J. Optimization of de novo transcriptome assembly from next-generation sequencing data. Genome Research 20(10):1432–1440, 2010.
11. Dempster A.P., Laird N.M., and Rubin D.B. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1):1–38, 1977.
12. Xing Y., Yu T., Wu Y.N., Roy M., Kim J., and Lee, C. An expectation maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs. Nucleic Acids Research 34(10):3150–3160,2006.
13. Trapnell C., Williams B.A., Pertea G., et al. Transcript assembly and quantifcation by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nature Biotechnology 28(5):511–515, 2010.
14. Mezlini A., Smith E., Fiume M., et al. iReckon: Simultaneous isoform discovery and abundance estimation from RNA-seq data. Genome Research 23: 519–529, 2013.
15. Le H., Schulz M., McCauley B., Hinman V., and Bar-Joseph Z. Probabilistic error correction for RNA sequencing. Nucleic Acids Research 41(10):e109,2013.
16. Guttman M., Garber M., Levin J.Z., et al. Ab initio reconstruction of cell type-specifc transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. Nature Biotechnology 28(5):503–510, 2010.
17. Zerbino D.R. and Birney E. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. Genome Research 18(5):821–829, 2008.
18. Grabherr M.G., Haas B.J., Yassour M., et al. Full-length transcriptome assembly from RNA-seq data without a reference genome. Nature Biotechnology 29(7):644–652, 2011.

19. Haas B.J., Papanicolaou A., Yassour M., et al. De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nature Protocols 8(8):1494–1512, 2013.
20. Steijger T., Abril J.F., Engström P.G., et al. Assessment of transcript reconstruction methods for RNA-seq. Nature Methods 10(12):1177–1184, 2013.
21. Francis W.R., Christianson L.M., Kiko R., Powers M.L., Shaner N.C., and Haddock S.H. A comparison across non-model animals suggests an optimal sequencing depth for de novo transcriptome assembly. BMC Genomics 14:167, 2013.

فصل ششم

کمی‌سازی و کنترل کیفیت مبتنی بر حاشیه‌نگاری

۶–۱ مقدمه

موقعیت خوانش‌هایی که قبلاً با یک ژنوم مرجع مکان‌یابی شده‌اند، را می‌توان با حاشیه‌نگاری ژنومی انطباق داد. با این کار می‌توان بیان ژن را با استفاده از شمارش خوانش‌ها به ازای هر ژن، رونوشت و اگزون کمی‌سازی نموده و نیز امکانات جدیدی را برای کنترل کیفیت فراهم نمود. جنبه‌های کیفی که تنها با خوانش‌های مکان‌یابی شده سنجیده می‌شوند، شامل اشباع عمق توالی‌یابی، توزیع خوانش بین انواع مختلف ترکیبات[1] ژنومی و یکنواختی پوشش در طول رونوشت‌ها هستند. نیمه‌ی ابتدایی فصل حاضر این معیارهای کیفی مبتنی بر حاشیه‌نگاری را بررسی نموده و برخی از نرم‌افزارهای ویژه برای کنترل آنها را معرفی می‌نماید.

نیمه‌ی دوم این فصل، کمی‌سازی بیان ژن که یکی از بخش‌های اصلی اکثر مطالعات توالی‌یابی RNA است، را مورد بحث قرار می‌دهد. در اصول کار، محاسبه‌ی تعداد خوانش‌های مکان‌یابی شده یک مسیر مستقیم برای برآورد فراوانی رونوشت بوده ولی در عمل چندین مولفه‌ی جانبی دیگر نیز بایستی در نظر گرفته شود. معمولاً ژن‌های یوکاریوتی چندین ایزوفرم رونوشت از طریق پیرایش جایگزین و استفاده از پروموتور ایجاد می‌کنند. با این حال کمی‌سازی در سطح رونوشت با خوانش‌های کوتاه، بی‌اهمیت نیست. زیرا اغلب مواقع ایزوفرم‌های رونوشت دارای اگزون‌های مشترک یا همپوشان هستند. علاوه بر این، به دلیل قابلیت مکان‌یابی رونوشت‌ها و اُریبی ایجاد شده در تهیه کتابخانه، پوشش در طول رونوشت‌ها یکنواخت نیست. به دلیل همین مسائل جانبی، در اغلب مواقع بیان در سطح ژن به جای بیان در سطح اگزون برآورد می‌گردد. با این حال شمارش‌ها در سطح ژن برای آنالیز افتراقی بیان در این ژن‌ها که دستخوش تغییر ایزوفرم می‌گردند، بهینه نیست. زیرا تعداد شمارش‌ها به اندازه‌ی رونوشت بستگی دارد. این موضع به طور مفصل در فصل هشتم و در قالب آنالیز افتراقی بیان مورد بحث و بررسی قرار می‌گیرد.

۶–۲ معیارهای کیفیت مبتنی بر حاشیه‌نگاری

همان‌گونه که در فصل سوم بحث شد، دستورالعمل‌هایِ آزمایشگاهیِ تولید داده‌های توالی‌یابی

1- Feature

RNA هنوز کامل نبوده ولی خوشبختانه تعداد زیادی از مشکلات کیفیت خوانش نظیر بازهای با اطمینان پایین و اُریبی در ترکیب نوکلئوتیدها را می‌توان در سطح خوانش‌های خام تشخیص داد. با این حال برخی از جنبه‌های کیفی مهم را تنها زمانی می‌توان اندازه‌گیری نمود که خوانش‌ها با یک ژنوم مرجع مکان‌یابی شده و موقعیت‌شان با حاشیه‌نگاری انطباق داده شود. این جنبه‌ها عبارتند از:

- *اشباع عمق توالی‌یابی:* قابلیت اعتماد پروفایل‌بندی بیان، آنالیز پیرایش و ساخت رونوشت به عمق توالی‌یابی بستگی دارد. چون توالی‌یابی پُرهزینه است، لازم است که میزان قرابت آن با اشباع‌شدگی داده‌ها کنترل شود. این بدان معناست که ژن‌های جدید و اتصالات پیرایشی با توالی‌یابی اضافی کشف می‌گردند. در حالت ایده‌آل، عمق صحیح بایستی پیش از شروع کار تعیین گردد. ولی این امر مستلزم دسترسی به یک مجموعه‌ی داده از همان گونه و بافت است. زیرا اشباع‌شدگی به ترکیب و پیچیدگی ترانسکریپتوم بستگی دارد.
- *توزیع خوانش بین انواع مختلف ترکیبات ژنومی:* این کار می‌تواند در چند سطح انجام شود. به عنوان مثال، خوانش‌ها می‌توانند در نواحی اگزونی، اینترونی و بین ژنی شمارش شده و خوانش‌های اگزونی نیز می‌توانند در بین اگزون‌های رمزگر، 5'UTR و 3'UTR باشند. اگر نسبت بالایی از خوانش‌ها با نواحی اینترونی و بین ژنی مکان‌یابی شوند، جستجوی ایزوفرم‌های جدید و ژن‌ها می‌تواند ارزشمند باشد. ولی این موضوع می‌تواند نشانه‌ای از آلودگی با DNA ژنومی نیز باشد. خوانش‌های مکان‌یابی شده با ژن‌ها می‌توانند با بیوتیپ‌هایی[1] نظیر ژن‌های رمزگر پروتئین‌ها، RNA های ریبوزومی (rRNA)، miRNA ها و غیره توزیع شوند. در این بین محتوای rRNA اهمیت ویژه‌ای دارد. زیرا دستورالعمل‌های آزمایشگاهی برای حذف rRNA غیرقابل اعتماد بوده و در بین نمونه‌ها متناقض و ناسازگار هستند. اگر یک بخش بزرگ از خوانش‌ها با rRNA مکان‌یابی گردند (به عنوان مثال، با مکان‌یابی نمودن آنها در برابر توالی‌های rRNA با Bowtie2 (همان‌گونه که در فصل چهارم تشریح گردید) و نگه داشتن خوانش‌های همردیف نشده)، می‌توان آنها را حذف نمود.
- *یکنواختی پوشش در طول رونوشت‌ها:* دستورالعمل‌های آزمایشگاهی مختلف می‌توانند اُریبی‌های مکانی مختلفی را نشان دهند. به عنوان مثال، دستورالعمل‌هایی که شامل یک مرحله برای در نظر گرفتن پُلیA هستند، می‌توانند خوانش‌هایی را ایجاد نمایند که عمدتاً از انتهاهای '3 رونوشت‌ها ایجاد شده‌اند. این اُریبی '3 می‌تواند بین نمونه‌های مختلف متفاوت باشد. بنابراین برآورد درجه‌ی این اُریبی مهم است.

1- Biotype

۶-۲-۱ ابزارهای کنترل کیفیت مبتنی بر حاشیه‌نگاری

چندین ابزار کنترل کیفیت برای داده‌های توالی‌یابی RNA همردیف شده در دسترس است. این ابزارها عبارتند از: RSeQC (۱)، RNA-SeQC (۲)، Qualimap (۳) و Picard's CollectRNASeqMetrics tool (۴). تعداد زیادی از معیارهایی که توسط این ابزارها گزارش می‌شوند، با هم همپوشانی دارند. ولی هر کدام از این ابزارها نقاط قوت اختصاصی نیز دارند. همه‌ی این ابزارها از رابط کاربری خط فرمان بهره برده ولی RNAseQC و Qualimap دارای GUI اختصاصی خود بوده و RseQC نیز در نرم‌افزار Chipster قابل دسترسی است. معمولاً اطلاعات حاشیه‌نگاری در قالب فایل‌های GTF (۵) یا BED (۶) ارائه می‌شود که لازم است نام‌گذاری کروموزوم در آنها مشابه فایل‌های BAM باشد.

RNA-SeQC در جاوا اجرا شده و حاشیه‌نگاری‌ها را با فرمت GTF دریافت می‌دارد. همچنین این نرم‌افزار نیازمند یک فایل FASTA مرجع با یک نمایه (fai.) و یک فایل واژه‌نامه‌ی توالی (dict.) است. RNA-SeQC به طور خاص یک گزارش مفصل از معیارهای پوشش ارائه کرده و می‌تواند نمونه‌های مختلف را نیز مقایسه نماید. گزارش معیارهای پوشش شامل پوشش میانگین، پوشش نواحی انتهایی رونوشت، اُریبی برای انتهاهای '3 و '5 و تعداد، طول تجمعی و درصد شکاف‌ها است. کلیه‌ی مقادیر برای ژن‌های با بیان کم، متوسط و بالا به صورت جداگانه محاسبه می‌شوند. همچنین مقادیر پوشش برای سه سطح محتوای GC نیز گزارش می‌گردد. علاوه بر نمودار یکنواختی پوشش، RNA-SeQC نمودار پوشش را در برابر فاصله از انتهای '3 (بر حسب جفت باز) نیز ترسیم می‌کند. خروجی شامل گزارش‌های HTML و فایل‌های متنی با جدا کننده‌ی تَب است.

Qualimap یک برنامه‌ی جاوا بوده و در درون خود از بسته‌های R و Bioconductor استفاده می‌کند. این نرم‌افزار حاشیه‌نگاری‌ها را در فرمت GTF/BED دریافت کرده و نیازمند یک فایل بیوتیپ جداگانه نیز می‌باشد. Qualimap نمودارهای زیبایی برای اشباع و توزیع بیوتیپ ارائه می‌دهد. نمودار اشباع تعداد ترکیبات شناسایی شده در عمق‌های مختلف توالی‌یابی را نشان داده و به طور معمول تعداد ترکیبات جدید شناسایی شده بر اثر افزایش هر یک میلیون عمق توالی‌یابی را نیز گزارش می‌کند. نمودار توزیع بیوتیپ نشان دهنده‌ی نحوه‌ی توزیع خوانش‌ها بین ژن‌های رمزگر پروتئین، شبه‌ژن‌ها، rRNA ، miRNA ها و غیره بوده و نشان می‌دهد که چند درصد از این ترکیبات در یک ژنوم دیده می‌شوند.

RseQC شامل چندین برنامه‌ی پایتون بوده و حاشیه‌نگاری‌های ژنومی را در فرمت BED دریافت می‌دارد. لازم است که R روی مسیر حضور داشته باشد. زیرا R به صورت داخلی برای ترسیم نتایج استفاده می‌شود. RseQC دارای ویژگی‌های جالبی است که در سایر برنامه‌ها یافت

نمی‌شود: الف) وقتی که توزیع خوانش بین ترکیبات ژنومی مختلف را محاسبه می‌کند، چند قطعه در پایین دست و بالا دست رونوشت‌ها را نیز گزارش می‌نماید، ب) علاوه بر ژن‌ها، وضعیت اشباع‌شدگی برای اتصالات پیرایشی را نیز محاسبه می‌نماید و پ) اتصالات پیرایشی را به صورت شناخته شده (Known)، جدید (Novel) و نسبتاً جدید (Partially novel) حاشیه‌نگاری می‌نماید.

فایل‌های BED دارای سه ستون اجباری و بسته به مشخصاتشان دارای نُه ستون اختیاری هستند (۶). RseQC نیازمند فایل‌های BED با ۱۲ ستون کامل است. زیرا اطلاعات اگزون برای هر ژن در سه ستون آخر (blockCount ، blockSizes و blockStarts) وجود دارد. با استفاده از UCSC Table Browser می‌توان فایل‌های BED را برای جانداران مختلف دریافت نمود (۷). بدین منظور در منوی Group گزینه‌ی Genes and gene predictions را انتخاب نمایید. منوی Track به شما اجازه می‌دهد که یک مجموعه‌ی ژنی نظیر ژن‌های RefSeq و Ensembl را انتخاب کنید. ناحیه را روی genome و فرمت خروجی را روی BED تنظیم نمایید. توجه داشته باشید که نام کروموزوم‌ها در فایل‌های BED حاصل از UCSC دارای پیشوند chr بوده ولی همردیف‌های ایجاد شده توسط Ensembl چنین نیستند. به آسانی می‌توانید پیشوند chr را با استفاده از دستور sed در یونیکس حذف نمایید:

```
sed 's/^chr//' hg19_Ensembl_chr.bed > hg19_Ensembl.bed
```

دستورات مثال زده شده در زیر برای RseQC از فایل همردیفی جفتی TopHat2 با عنوان accepted_hits.bam استفاده می‌نمایند (ر.ک: فصل چهارم).

ابزار read_distribution.py توزیع خوانش‌ها برای انواع مختلف ترکیبات ژنومی را محاسبه می‌نماید:

```
python read_distribution.py -r hg19_Ensembl.bed-i accepted
_hits.bam
```

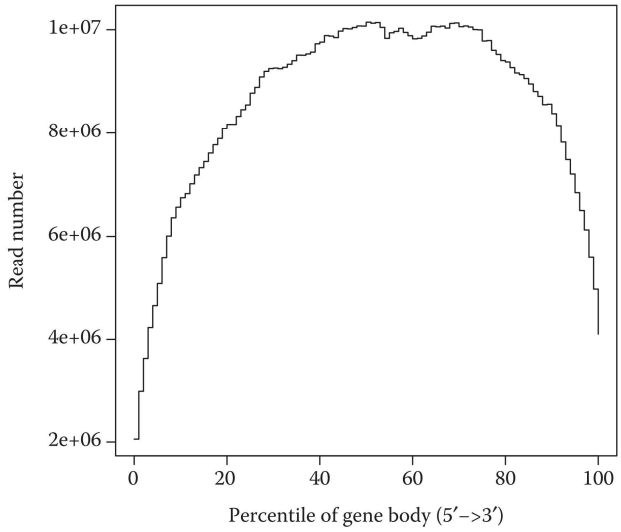
جدول نتایج تعداد کُل خوانش‌ها (غیر از توفیقات فرعی) و تگ‌ها (قطعات پیرایشی جداگانه‌ی یک خوانش) را گزارش می‌نماید. تعداد کُل تگ‌های تخصیص یافته نشان می‌دهد که چه تعداد از تگ‌ها می‌توانند به طور قطعی و بدون تردید به ده دسته‌ی زیر تخصیص داده شوند.

```
Total Reads 49743155
Total Tags 63012643
Total Assigned Tags 57529077
====== == ======== ===== ===== ============ == == ======
Group            Total_bases       Tag_count   Tags/Kb
CDS_Exons        36821030          34763281    944.11
```

```
5'UTR_Exons          34901580            2856644     81.85
3'UTR_Exons          54908278            9772738     177.98
Introns              1450606807          8468986     5.84
TSS_up_1kb           31234456            94103       3.01
TSS_up_5kb           139129272           161914      1.16
TSS_up_10kb          249300845           217980      0.87
TES_down_1kb         32868738            789703      24.03
TES_down_5kb         142432117           1368378     9.61
TES_down_10kb        251276738           1449448     5.77
====== == ======== ===== ===== ============ == == ======
```

ابزار geneBody_coverage.py یک نمودار پوشش (نگاره‌ی ۶-۱) ایجاد کرده و این امکان را فراهم می‌آورد که بتوانید یکنواختی پوشش در طول رونوشت‌ها و وجود اُریبی در انتهای '3 یا '5 را کنترل نمایید. پارامتر o- به شما اجازه می‌دهد که یک پیشوند به نام فایل‌های نتایج بدهید.

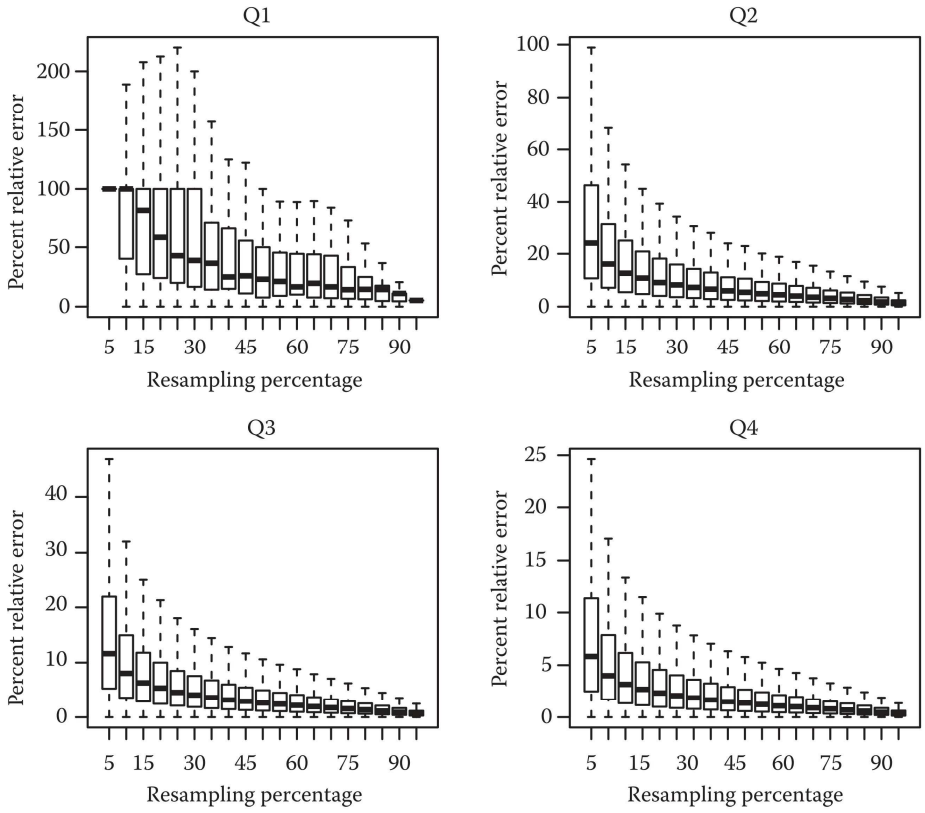
```
python geneBody_coverage.py -r hg19_Ensembl.bed-i accepted_
hits.bam -o file
```



نگاره‌ی ۶-۱: نمودار RseQC برای یکنواختی پوشش در سرتاسر رونوشت‌ها. طول همه‌ی رونوشت‌ها ۱۰۰ نوکلئوتید مقیاس‌بندی شده است.

دقت برآوردهای فراوانی بیان ژن در عمق فعلی توالی‌یابی توسط ابزار `RPKM_saturation.py` محاسبه می‌شود. این ابزار زیرمجموعه‌هایی از خوانش‌ها را مجدداً نمونه‌گیری کرده، فراوانی را بر حسب واحد RPKM (این واحد در ادامه‌ی همین فصل تشریح می‌شود) برای هر زیر مجموعه محاسبه نموده و پایداری یا عدم پایداری آنها را کنترل می‌نماید. این کار همان‌گونه که در نگاره‌ی ۶-۲ نشان داده شده است، به صورت جداگانه برای چهار دسته سطوح بیان مختلف انجام می‌شود.

```
python RPKM_saturation.py -r hg19_Ensembl.bed-i accepted
_hits.bam -o file
```



نگاره‌ی ۶-۲: نمودار اشباع توالی‌یابی ترسیم شده توسط RseQC. زیرمجموعه‌هایی از خوانش‌ها مجدداً نمونه‌گیری شده و RPKM ها برای هر زیرمجموعه محاسبه گردیده و با RPKM های حاصل از کُل خوانش‌ها مقایسه می‌شوند. این کار به صورت جداگانه برای چهار دسته سطوح بیان مختلف انجام می‌شود.

ابزار junction_annotation.py اتصالات پیرایشی را به جدید (Novel)، نسبتاً جدید (Partially novel) (یک مکان پیرایش جدید است) و حاشیه‌نگاری شده (Annotated) (هر دو مکان پیرایش در مدل‌های ژنی مرجع حضور دارند) تقسیم کرده و نتایج را به صورت یک نمودار دایره‌ای گزارش می‌دهد (نگاره‌ی۶-۳-الف).

```
python junction_annotation.py -r hg19_Ensembl.bed-i accepted_
hits.bam -o file
```

وضعیت اشباع توالی برای اتصالات پیرایشی را می‌توان با استفاده از ابزار junction_saturation.py کنترل نمود. این ابزار زیرمجموعه‌هایی از خوانش‌ها را مجدداً نمونه‌گیری کرده، اتصالات در هر زیرمجموعه را شناسایی نموده و آنها را با حاشیه‌نگاری مرجع مقایسه می‌کند. همان‌گونه که در نگاره‌ی ۶-۳-ب نشان داده شده است، نتایج برای اتصالات جدید و شناخته شده به صورت جداگانه گزارش می‌شود.

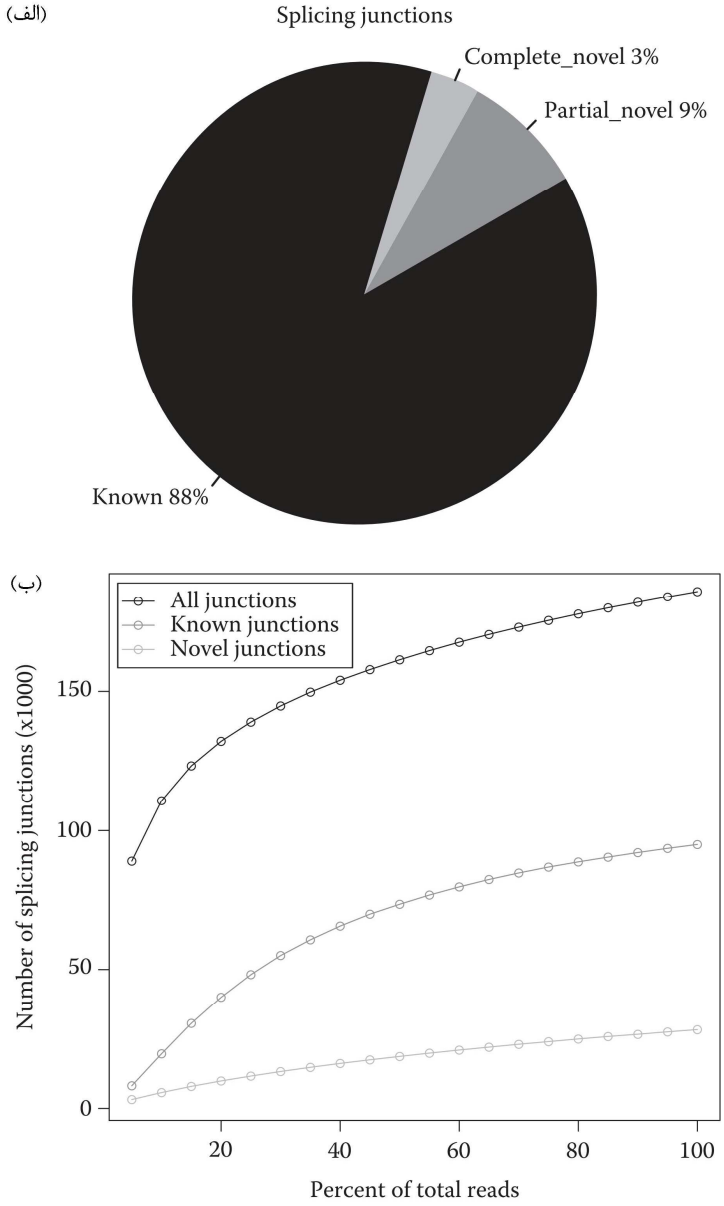
```
python junction_saturation.py -r hg19_Ensembl.bed-i accepted_
hits.bam -o file
```

کنترل کیفیت مبتنی بر حاشیه‌نگاری در Chipster

- فایل همردیفی(BAM)، یک فایل BED حاوی حاشیه‌نگاری‌ها و ابزار Quality control/RNA-seq quality metrics with RseQC را انتخاب کنید. اطمینان حاصل کنید که در پنل پارامتر، فایل‌ها به درستی تخصیص داده شده‌اند.

۶-۳ کمی‌سازی بیان ژن

وقتی که یک ژنوم مرجع حاشیه‌نگاری شده در دسترس باشد، می‌توان خوانش‌های مکان‌یابی شده به ازای هر ترکیب ژنومی را بر مبنای اطلاعات مکانی شمارش نمود. استفاده از یک فایل حاشیه‌نگاری که توسط اسمبل‌سازهای از آغاز نظیر Cufflinks (۸) (ر.ک: فصل پنجم) ایجاد شده است، امکان کمی‌سازی ژن‌ها و رونوشت‌های جدید را فراهم می‌آورد. به جای آن و به ویژه اگر هیچ ژنوم مرجعی در دسترس نباشد، خوانش‌ها می‌توانند با ترانسکریپتوم مکان‌یابی شده و شمارش گردند. اگر هیچ ترانسکریپتوم مرجعی وجود نداشته باشد، می‌توان با استفاده از اسمبل‌سازهای از نو (ر.ک: فصل پنجم) یک ترانسکریپتوم اسمبل نموده و سپس خوانش‌ها را با این ترانسکریپتوم مکان‌یابی کرده و شمارش نمود.



نگاره‌ی ۶-۳: نرم‌افزار RseQC اتصالات پیرایشی شناسایی شده را الف) به صورت جدید (Novel)، نسبتاً جدید (Partially novel) و شناخته شده (Known) حاشیه‌نگاری کرده و ب) وضعیت اشباع‌شدگی آنها را با استفاده از نمونه‌گیری مجدد آنالیز می‌نماید.

تعداد خوانش‌های ایجاد شده به ازای هر رونوشت به چندین عامل بستگی دارد. برخی از این عوامل نظیر عمق توالی‌یابی و طول خوانش (وقتی که در طی تهیه‌ی کتابخانه قطعه قطعه می‌شوند، رونوشت‌های بلندتر، قطعات بیشتری تولید کرده و در نتیجه خوانش‌های بیشتری نیز ایجاد خواهند نمود) واضح و بدیهی هستند. ولی برخی از عوامل موثر بر تعداد خوانش‌ها نظیر ترکیب ترانسکریپتوم، اُریبی GC و اُریبی مختص توالی که توسط هگزامرهای تصادفی ایجاد می‌شوند، ممکن است به سختی مورد توجه و دقت قرار گیرند. اگر بخواهید شمارش خوانش‌ها بین ژن‌ها یا نمونه‌های مختلف را با هم مقایسه کنید، لازم است که این عوامل را در نظر بگیرید. روش‌های متعدد نرمال‌سازی در دسترس بوده و انتخاب از بین آنها به نوع مقایسات بیانی مورد نظر بستگی دارد. معمولاً نرم‌افزار کمی‌سازی در خروجی خود فراوانی‌ها را یا در قالب شمارش‌های خام و یا در قالب FPKM (تعداد قطعات به ازای هر کیلوباز در هر یک میلیون خوانش مکان‌یابی شده) ارائه می‌دهند. خوانش‌های خام برای آنالیز افتراقی بیان لازم هستند (ر.ک: فصل هشتم). ولی FPKM ها می‌توانند برای اهداف گزارش‌دهی بیان مورد استفاده قرار گیرند. RPKM (تعداد خوانش‌ها به ازای هر کیلوباز در هر یک میلیون خوانش مکان‌یابی شده) که سَلَف FPKM محسوب می‌گردد، توسط Mortazavi و همکاران و با هدف تصحیح شمارش‌های اندازه‌ی کتابخانه و طول رونوشت ابداع و معرفی گردید (۹). RPKM از تقسیم شمارش‌ها بر طول رونوشت (بر حسب کیلوباز) و سپس بر تعداد کُل خوانش‌ها به دست می‌آید. به عنوان مثال اگر یک رونوشت ۲ کیلوبازی دارای ۱۰۰۰ خوانش بوده و تعداد کُل خوانش‌ها نیز ۲۵ میلیون باشد، آنگاه $\text{RPKM} = \frac{\frac{۱۰۰۰}{۲}}{۲۵}$ خواهد بود. FPKM معادل آزمایشات جفت انتهایی است که در آنها قطعات از هر دو انتها توالی‌یابی شده و دو خوانش به ازای هر قطعه به دست می‌آید. روش جایگزینی برای FPKM وجود دارد که اصطلاحاً TPM (تعداد رونوشت‌ها در هر یک میلیون) نامیده شده و در آن توزیع طول رونوشت‌ها در نمونه در نظر گرفته شده و بنابراین بایستی فراوانی‌هایی که بین نمونه‌ها پایدارتر هستند، ایجاد شوند (۱). در TPM به جای تقسیم بر تعداد کُل خوانش‌ها، بر مجموع خوانش‌های نرمال شده برای طول رونوشت تقسیم صورت می‌گیرد.

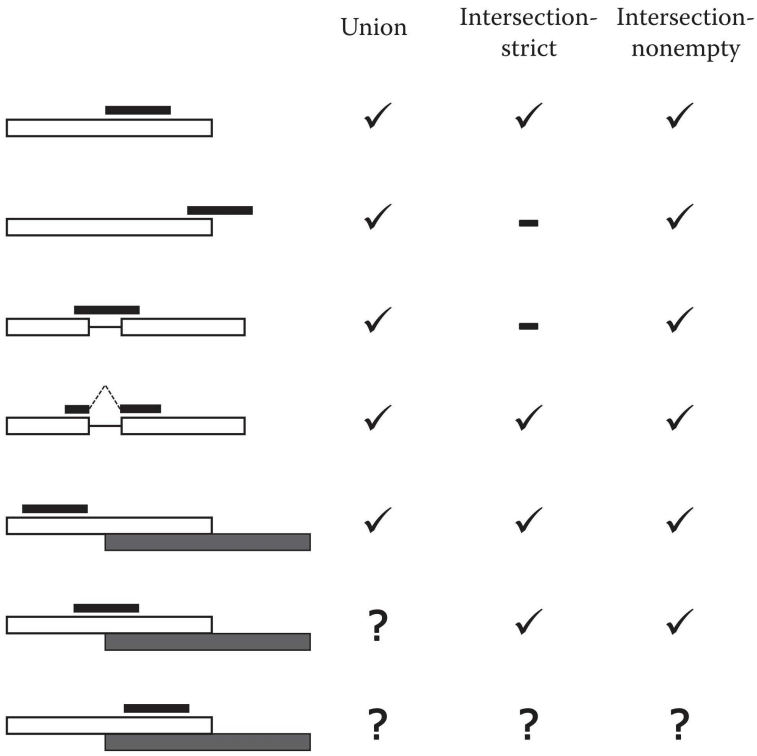
۶-۳-۱ شمارش خوانش‌ها به ازای هر ژن

ساده‌ترین راه برآورد بیان، شمارش خوانش‌ها به ازای هر ژن است. چندین ابزار نظیر HTSeq (۱۱)، BEDTools (۱۲) و Qualimap برای این کار در دسترس است. همچنین برخی از بسته‌های Bioconductor نظیر Rsubread و GenomicRanges می‌توانند کار شمارش را انجام دهند (مثالی

از کُدنویسی با کمک GenomicRanges در فصل هفتم ارائه شده است). همچنین بسته‌ی نرم‌افزاری Cufflinks نیز زمانی که رونوشت‌ها را اسمبل می‌کند (ر.ک: فصل پنجم) و آنالیز افتراقی بیان را انجام می‌دهد (ر.ک: فصل هشتم)، برآوردهای سطوح بیان ژن را علاوه بر رونوشت‌ها ارائه می‌دهد. کلیه‌ی این ابزارها همردیف‌های خوانش ژنومی را در فرمت SAM/BAM و حاشیه‌نگاری ژنومی در فرمت GFF/GTF یا BED را به عنوان ورودی دریافت می‌کنند. این ابزارها از نظر روش مدیریت خوانش‌های دارای چند مکان‌یابی (خوانش‌هایی که به دلیل همولوژی یا تکرارهای توالی، با چندین موقعیت ژنومی مکان‌یابی می‌شوند) با یکدیگر تفاوت دارند. HTSeq این خوانش‌های چندتایی را نادیده گرفته ولی Qualimap شمارش‌ها را به صورت مساوی بین موقعیت‌های مختلف تقسیم کرده و Cufflinks نیز یک گزینه برای تقسیم هر خوانش دارای چند مکان‌یابی به صورت احتمالی و بر مبنای فراوانی ژن‌هایی که با آن مکان‌یابی شده‌اند، دارد. همچنین ابزارهای شمارش گزینه‌های مختلفی برای مدیریت خوانش‌هایی که با بیش از یک ژن همپوشانی داشته یا خوانش‌هایی که بخشی از آنها در نواحی اینترونی واقع می‌گردند، دارند. نگاره‌ی ۶-۴ سه وضعیت ارائه شده از سوی HTSeq که در مثال‌های این کتاب نیز استفاده می‌شوند را نمایش می‌دهد. همه‌ی این ابزارها برای استفاده روی خط فرمان در دسترس بوده و Cufflinks ، HTSeq و BEDTools در رابط کاربری گرافیکی Chipster نیز در اختیار هستند.

۶-۳-۱-۱ *HTSeq*

Htseq-count بخشی از بسته‌ی نرم‌افزاری HTSeq از اسکریپت‌های پایتون برای آنالیز داده‌های NGS است. ولی استفاده از این نرم‌افزار نیازمند هیچ‌گونه اطلاعاتی از پایتون نیست. Htseq-count خوانش‌های همردیف شده را در فرمت SAM/BAM و حاشیه‌نگاری ژنوم را به صورت فایل GFF/GTF دریافت می‌نماید. باید توجه نمود که برای انطباق موقعیت مکان‌یابی خوانش با ترکیبات ژنومی، نام‌های کروموزومی در فایل همردیفی و فایل حاشیه‌نگاری باید یکسان باشند. Htseq-count اگزون‌هایی که خوانش با آنها همپوشانی دارد، پیدا کرده و سپس شمارش‌های سطح اگزونی را بر مبنای ID ژنی اگزون‌ها در فایل GTF گروه‌بندی می‌کند. برای این کار لازم است که کلیه‌ی اگزون‌های یک ژن، ID ژنی یکسانی داشته باشند. فایل‌های GTF حاصل از Ensembl از این قاعده پیروی می‌کنند. ولی فایل‌های GTF که از مرورگر UCSC Table به دست می‌آیند، یک ID رونوشت تکرار شده به عنوان ID ژن دارند. این موضوع برای Htseq-count مشکل‌ساز است. زیرا Htseq-count نمی‌تواند حدس بزند که رونوشت‌ها متعلق به همان ژن بوده و بنابراین خوانش‌ها را به صورت جداگانه شمارش خواهد نمود. GTF های Ensembl با انتخاب جاندار مورد نظر و



نگاره‌ی ۶-۴: سه وضعیت ارائه شده از سوی HTSeq برای شمارش خوانش‌ها به ازای هر ترکیب ژنومی. میله‌ی سیاه نشان دهنده‌ی یک خوانش، میله‌ی سفید نشان دهنده‌ی یک ژن خوانش مزبور که با آن مکان‌یابی شده است و میله‌ی خاکستری نشان دهنده‌ی ژن دیگری که به طور جزیی با ژن سفید رنگ همپوشانی دارد، می‌باشد. علامت تیک (✓) بدان معناست که خوانش مزبور برای ژن سفید شمارش شده و علامت سوال (?) نیز بدان معناست که خوانش مزبور به دلیل مبهم بودن، برای ژن سفید شمارش نشده است. اگر با نواحی اینترونی و بین ژنی همپوشانی وجود داشته باشد، وضعیت intersection_strict خوانش مزبور را شمارش نمی‌کند (در اینجا علامت خط تیره (▬) نشان دهنده‌ی no_feature است). تنظیمات پیش‌فرض روی وضعیت union است.

گزینه‌ی GTF در http://www.ensembl.org/info/data/fp/index.html قابل دسترس است. در مثال زیر از فایل همردیفی جفت شده‌ی TopHat2 که در فصل چهارم به دست آمده است، استفاده شده و هدف، دانلود GTF انسانی است:

```
wget  ftp://ftp.ensembl.org/pub/release74/gtf/homo_sapiens/
Homo_sapiens.GRCh37.74.gtf.gz
```

دستور زیر نیز فایل را از حالت فشرده خارج می‌کند:

```
gunzip Homo_sapiens.GRCh37.74.gtf.gz
```

به صورت پیش‌فرض Htseq-count داده‌های جفت انتهایی را بر مبنای نام خوانش‌ها مرتب کرده و بدین ترتیب خوانش‌های جفت شده به صورت پشت سر هم در فایل مزبور قرار می‌گیرند. همردیف‌ها نیز بر مبنای موقعیت ژنومی مرتب می‌شوند (با استفاده از گزینه‌ی-order=pos). ولی این کار نیازمند حافظه‌ی بیشتری است. دستور زیر BAM را بر مبنای نام خوانش‌ها مرتب کرده و یک فایل hits_namesorted.bam ایجاد می‌نماید.

```
samtools sort -n accepted_hits.bam hits_namesorted
```

دستور Htseq-count به صورت زیر است (باید اطمینان حاصل نمود که فایل htseqqa روی مسیر قرار دارد):

```
htseq-count -f bam --stranded = no hits_namesorted.bamHomo_
sapiens.GRCh37.74.gtf > counts.txt
```

در اینجا -f bam نشان دهنده‌ی این است که فرمت ورودی BAM می‌باشد. به صورت پیش‌فرض خوانش‌هایی که با موقعیت‌های اگزونی در فایل GTF انطباق دارند (--type=exon)، شمارش گردیده و شمارش‌های اگزون‌ها که متعلق به ژن یکسانی هستند، تلفیق می‌گردند (--idattr=gene_id). Htseq-count فرض می‌کند که داده‌ها با دستورالعمل مختص زنجیره ایجاد شده و تنها اگر خوانش‌ها با زنجیره‌ی یکسان همانند ژن مکان‌یابی گردند، آنها را شمارش می‌نماید. چون داده‌های مثال زنجیره‌بندی نشده‌اند، باید --stranded=no نیز افزوده شود تا بدین ترتیب زمانی که یک خوانش با زنجیره‌ی روبرو مکان‌یابی می‌گردد، شمارش شود. وضعیت پیش‌فرض برای شمارش، union است. ولی می‌توان آنرا با استفاده از گزینه‌ی --mode تغییر داد. همچنین می‌توان یک مقدار حداقل برای کیفیت مکان‌یابی خوانشی که شمارش می‌گردد، تنظیم نمود (به عنوان مثال: -a30). این مقدار به صورت پیش‌فرض، ۱۰ در نظر گرفته شده است.

فایل خروجی counts.txt شامل یک جدول از شمارش‌ها برای هر ژن است. در انتهای این فایل، پنج ردیف دیده می‌شود که تعداد خوانش‌هایی را که به دلایل زیر برای هیچ ژنی شمارش نشده‌اند، فهرست می‌نمایند:

الف- بر مبنای تگ NH در فایل BAM ، این خوانش‌ها با بیش از یک مکان در ژنوم مرجع همردیف شده‌اند (alignment_not_unique).

ب- این خوانش‌ها به هیچ‌وجه همردیف نشده‌اند (not_aligned).

پ- کیفیت همردیفی این خوانش‌ها کمتر از آستانه‌ی تخصیص یافته توسط کاربر است (too_low_aQual).

ت- همردیفی این خوانش‌ها با بیش از یک ژن همپوشانی داشته است (ambiguous).

ث- همردیفی این خوانش‌ها با هیچ ژنی همپوشانی نداشته است (no_feature).

```
...
ENSG00000273490   0
ENSG00000273491   0
ENSG00000273492   0
ENSG00000273493   0
_no_feature       6125428
_ambiguous        1808462
_too_low_aQual    0
_not_aligned      0
_alignment_not_unique   2947054
```

می‌توان فایل‌های شمارش حاصل از نمونه‌های مختلف را با کمک دستور join در یونیکس به صورت یک جدول تلفیق نمود:

```
join counts1.txt counts2.txt > count_table.txt
```

در صورت تمایل پیش از آزمون آماری برای آنالیز افتراقی بیان، می‌توان پنج ردیف انتهایی را حذف نمود. دستور head در یونیکس همه‌ی خطوط به غیر از پنج خط انتهایی (-n -5) را نگه می‌دارد:

```
head -n -5 count_table.txt > genecounts.txt
```

شمارش خوانش‌ها به ازای ژن‌ها در Chipster

- فایل همردیفی (BAM)، و ابزار RNA-seq/Count aligned reads per genes withHTSeq را انتخاب کنید. در پارامترها، جاندار مورد نظر را انتخاب نموده و مشخص نمایید که آیا داده‌های شما با یک دستورالعمل مختص زنجیره ایجاد شده‌اند یا خیر. همچنین می‌توانید انتخاب را به نحوی انجام دهید که ژن‌های مختصات کروموزومی در فایل شمارش در نظر گرفته شوند (این کار بعداً به مصورسازی آنالیز افتراقی بیان

در یک مرورگر ژنومی کمک می‌کند). روی Run کلیک کنید.

- توجه کنید که اگر جاندار نمونه شما در Chipster موجود نباشد، می‌توانید از ابزار RNA-seq/Count aligned reads per genes with HTSeq using own GTF استفاده نمایید. فایل GTF را به Chipster وارد کرده و آنرا همراه با فایل BAM به عنوان ورودی انتخاب نمایید. در پنجره‌ی پارامتر، از تخصیص صحیح فایل‌ها اطمینان حاصل کنید.
- فایل‌های شمارش را برای کلیه‌ی نمونه‌ها انتخاب کرده و آنها را با استفاده از ابزار Utilities/Define NGS experiment در قالب یک جدول شمارش تلفیق کنید. در پارامترها ستون‌های حاوی شمارش‌ها را تعیین کرده و نیز مشخص کنید که آیا داده‌های شما حاوی مختصات کروموزومی است یا خیر.

۶-۳-۲ شمارش خوانش‌ها به ازای هر رونوشت

شمارش خوانش‌ها در سطح رونوشت با در نظر گرفتن این حقیقت که معمولاً ایزوفرم‌های رونوشت دارای بخش‌های همپوشان هستند، کاری پیچیده است. برای تخصیص مبهم مکان‌یابی خوانش‌های دارای ایزوفرم‌های مختلف، از یک روش حداکثرسازی امیدریاضی (EM) استفاده می‌شود. در این روش دو مرحله به تناوب اجرا می‌گردد: مرحله‌ی پیش‌بینی که در آن خوانش‌ها با احتمالی بر مبنای فراوانی رونوشت‌هایشان (که در ابتدا مساوی فرض می‌شوند) به رونوشت‌ها تخصیص می‌یابند و مرحله‌ی حداکثرسازی که در آن فراوانی‌ها بر مبنای احتمال تخصیص به‌روزرسانی می‌گردند. برنامه‌هایی که فراوانی رونوشت‌ها را در ژن‌های دارای چند ایزوفرم با این روش برآورد می‌کنند، عبارتند از: Cufflinks و eXpress (۱۳). Cufflinks از یک روش EM دسته‌ای[1] بهره گرفته ولی eXpress از یک الگوریتم EM آنلاین استفاده کرده و بنابراین سریع‌تر بوده و از نظر حافظه کارآمدتر است.

علی‌رغم اینکه Cufflinks از همردیف‌های ژنومی به عنوان ورودی استفاده می‌کند، ولی eXpress همردیف‌های ترانسکرپتومی را به کار گرفته و بنابراین برای گونه‌هایی که هنوز فاقد ژنوم مرجع هستند، مناسب است. اگر ترانسکریپتوم مرجع نیز در دسترس نباشد، می‌توان آنرا با یک اسمبل‌ساز از نو نظیر Trinity یا Oases (ر.ک: فصل پنجم) ساخت. برآوردهای فراوانی ایجاد شده توسط eXpress می‌توانند در هنگام تغییر حاشیه‌نگاری‌های رونوشت، توسط ابزار ReXpress به طور کارآمدی به‌روزرسانی شوند (۱۴). اجتناب از آنالیز مجدد کُل مجموعه‌ی داده‌ها که کاری

1- Batch

وقت‌گیر است، به ویژه برای جاندارانی که به تازگی توالی‌یابی شده و حاشیه‌نگاری رونوشت‌های‌شان اغلب تغییر می‌کند، از اهمیت ویژه‌ای برخوردار است.

هر دو نرم‌افزار Cufflinks و eXpress می‌توانند مکان‌یابی چندتایی خوانش‌ها در بین خانواده‌های ژنی را برطرف کرده وتوزیع طول قطعات را از داده‌ها استخراج نموده و برای اُریبی مختص توالی مجاور انتهاهای قطعات که ناشی از آغازگرهای مورد استفاده در تهیه‌ی کتابخانه است، تصحیح نمایند. علاوه بر این، eXpress شامل یک مدل برای خطاهای توالی‌یابی نظیر ایندل‌ها نیز بوده و می‌تواند بیان مختص آلل را نیز برآورد نماید. Cufflinks و eXpress علاوه بر خط فرمان، می‌توانند در رابط کاربری گرافیکی Chipster نیز به کار گرفته شوند.

۶-۳-۲-۱ *Cufflinks*

Cufflinks همردیف‌های ژنومی را در فرمت BAM و حاشیه‌نگاری‌ها را در قالب فایل GTF دریافت می‌کند. فایل GTF اختیاری است. زیرا Cufflinks می‌تواند برآورد فراوانی ایزوفرم را با اسمبل تلفیق نماید. استفاده از تصحیح اُریبی قطعه توصیه شده است. وقتی که این تصحیح انجام شود، Cufflinks از داده‌ها می‌آموزد که چه توالی‌هایی برای آن انتخاب شده و فراوانی‌ها را با یک تابع درستنمایی جدید که اُریبی مختص توالی را در نظر می‌گیرد، مجدداً برآورد می‌کند (Cufflinks از اطلاعات فراوانی اصلی جهت ایجاد یک تفاوت بین توالی‌هایی که بیش از اُریبی به دلیل بیان بالا متداول هستند، استفاده می‌کند).

دستور مثال زده شده برای Cufflinks ، همردیف ژنومی جفت انتهایی ایجاد شده توسط TopHat2 را به عنوان ورودی دریافت می‌کند. این دستور بیان رونوشت‌های معلوم را برآورد کرده و برای آنهایی که جدید هستند، این کار را انجام نمی‌دهد (`-G`). این دستور برای اُریبی قطعه تصحیح انجام داده (`-b GRCh37.74.fa`) و خوانش‌هایی که با چند موقعیت مکان‌یابی می‌گردند، را وزن‌دهی می‌کند (`-u`). هشت ریزپردازنده برای سرعت‌دهی به فرآیندپردازش مورد استفاده قرار می‌گیرند (`-p 8`). لازم است که SAMtools روی مسیر قرار گیرد. زیرا Cufflinks از این نرم‌افزار به صورت داخلی بهره می‌گیرد.

```
cufflinks -G Homo_sapiens.GRCh37.74.gtf -b GRCh37.74.fa -u
-p 8 accepted_hits.bam -o outputFolder
```

خروجی شامل رونوشت و فایل‌های ردیابی FPKM در سطح ژن است که حاوی مقادیر FPKM و فاصله اطمینان آنها می‌باشند. همچنین فایل‌های ردیابی FPKM وقتی ایجاد می‌شوند که یک

مجموعه از نمونه‌ها برای آنالیز افتراقی بیان با استفاده از Cufflinks مورد آزمون واقع شوند. این موضوع به طور مفصل در فصل هشتم مورد بحث و بررسی واقع می‌گردد.

۲-۲-۳-۶ *eXpress*

eXpress توالی‌های رونوشت در فرمت FASTA چندتایی و همردیف‌های خوانش‌ها که با کمک این مجموعه از رونوشت‌ها ایجاد شده‌اند، را به عنوان ورودی دریافت می‌کند. همردیف‌ها می‌توانند در یک فایل BAM بوده یا مستقیماً از همردیف‌سازهایی نظیر Bowtie2 به eXpress جریان یابند (نیازی به همردیف‌ساز پیرایشی نیست. زیرا خوانش‌ها به جای ژنوم با ترانسکریپتوم مکان‌یابی می‌شوند). لازم است که تا حد ممکن به مکان‌یابی‌های چندتایی بیشتری اجازه‌ی ورود داده شود. همچنین می‌توان ورود عدم انطباق‌های بیشتری را مجاز دانست. زیرا eXpress یک مدل خطا برای تخصیص مبتنی بر احتمال خوانش‌ها می‌سازد. فایل‌های BAM/SAM حاوی داده‌های جفت انتهایی مورد نیاز ذخیره شده بر مبنای نام خوانش‌ها (ر.ک: بخش HTSeq در همین فصل) هستند.

در مثال زیر، توالی‌های رونوشت از پایگاه داده‌ی RefSeq دانلود شده (۱۵)، یک نمایه‌ی Bowtie2 برای این مجموعه ایجاد گردیده، خوانش‌ها با کمک Bowtie2 همردیف شده و فراوانی‌های رونوشت‌ها با استفاده از eXpress محاسبه گردیده‌اند.

رونوشت‌ها از RefSeq دانلود می‌شوند:

```
wget   ftp://ftp.ncbi.nlm.nih.gov/refseq/H_sapiens/mRNA_
Prot /human.rna.fna.gz
```

فایل مزبور از حالت فشرده خارج می‌گردد:

```
gunzip human.rna.fna.gz
```

برای در خاطر نگه داشتن ویرایش RefSeq مورد استفاده، فایل مزبور تغییر نام داده می‌شود:

```
mv human.rna.fna refseq63.fasta
```

نمایه‌ی Bowtie2 برای رونوشت‌ها ایجاد می‌شود (ر.ک: فصل چهارم). پارامتر offrate تعداد ردیف‌هایی که در نمایه‌ی مرجع علامت‌گذاری می‌شوند را کنترل می‌کند. مقدار پیش‌فرض برای این پارامتر برابر با ۵ است که بدان معناست که ۳۲ ردیف ($۳۲ = ۲^۵$) علامت‌گذاری می‌گردد. برای جستجوی سریع‌تر موقعیت مرجع در طی همردیفی، این مقدار در اینجا به ۱ تغییر داده شده است و بدین‌ترتیب دو ردیف علامت‌گذاری می‌شود. این کار ضرورت دارد. زیرا در اینجا هدف آن است که

مکان‌یابی‌های چندتایی بیشتری در طی همردیفی مجاز شمرده شوند. این موضوع سبب می‌شود که Bowtie2 بسیار کُند گردد.

```
bowtie2-build -offrate=1 -f refseq63.fasta refseq63
```

دستور زیر با استفاده از تنظیمات پارامتر Bowtie2 که توسط نویسندگان eXpress توصیه شده است (http://bio.math.berkeley.edu/ReXpress/rexpress_manual.html) خوانش‌ها را با رونوشت‌ها همردیف می‌کند. از گزینه‌ی `-k` برای فرمان دادن به Bowtie2 جهت گزارش ۱۰۰۰ همردیف به ازای هر خوانش (به جای تنها یک همردیف به ازای هر خوانش) استفاده می‌شود. در حالت ایده‌آل این تمایل وجود دارد که همه‌ی همردیف‌ها موجود باشند (`-a`). ولی این کار سبب کُند شدن روند آنالیز می‌گردد. زیرا Bowtie2 برای چنین کاربری‌هایی طراحی نشده است. خروجی SAM حاصل از Bowtie2 به SAMtools انتقال می‌یابد تا جهت صرفه‌جویی در فضا، به BAM تبدیل شود (فایل SAM تولید شده توسط Bowtie2 به صورت خودکار بر مبنای نام مرتب شده است. بنابراین در اینجا مرحله‌ی مرتب‌سازی حذف می‌شود).

```
bowtie2 -q -k 1000 -p 8 --phred64 --no-discordant--no-mixed
--rdg 6,5 --rfg 6,5 --score-min L,-.6,-.4-x refseq63 -1
reads1.fq.gz -2 reads2.fq.gz | samtoolsview -Sb - >
transcriptome_aligned.bam
```

جستجو به همردیف‌های منطبق و جفت شده محدود می‌گردد (`--nodiscordant --no-mixed`). با این کار، در مقایسه با مقادیر پیش‌فرض، تاوان‌های[1] افزایش شکاف خوانش و مرجع (`--rdg 6,5 --rfg 6,5`) و حداقل امتیاز پذیرفته شده‌ی همردیفی (`--score-min L,-.6,-.4`) سخت‌تر می‌شود. همان‌گونه که در خلاصه نیز نشان داده شده است، نرخ همردیفی کُلی برابر با ۶۹/۱۷ درصد است:

```
34232081 reads; of these:
 34232081 (100.00%) were paired; of these:
  10553741 (30.83%) aligned concordantly 0 times
  4166418 (12.17%) aligned concordantly exactly 1 time
  19511922 (57.00%) aligned concordantly >1 times
69.17% overall alignment rate
```

1- Penalty

فراوانی‌های خوانش‌ها با کمک eXpress و با استفاده از تصحیح اُریبی و تصحیح خطا محاسبه می‌گردد:

```
express refseq63.fasta transcriptome_aligned.bam –o
outputFolder
```

به جای این کار و برای اجتناب از ایجاد یک فایل بزرگ BAM حد واسط، می‌توان خروجی Bowtie2 را مستقیماً به eXpress انتقال داد:

```
bowtie2 –k 1000 –p 8 --phred64 --no-discordant --no-mixed--
rdg 6,5 --rfg 6,5 --score-min L,-.6,-.4 -x refseq63 -
1reads_1.fq.gz –2 reads_2.fq.gz | express refseq63.fasta-o
outputFolder
```

فایل نتایج results.xprs حاوی برآوردهای فراوانی می‌باشد. رونوشت‌ها بر مبنای دسته[1] (bundle_id) مرتب می‌شوند. دسته عبارت از گروهی از رونوشت‌ها که خوانش‌های مکان‌یابی چندتایی مشترک دارند، است. این فایل چندین ستون دارد که مهم‌ترین آنها عبارتند از: شمارش‌های برآورد شده (est_counts)، شمارش‌های موثر (eff_counts)، FPKM و TPM. شمارش‌های موثر برای اُریبی‌های قطعه و طول تصحیح شده‌اند و نویسندگان eXpress توصیه نموده‌اند که از آنها به صورت گرد شده برای ابزارهای آنالیز افتراقی بیان مبتنی بر شمارش نظیر edgeR استفاده شود. دستور awk مشخصه‌ی رونوشت و ستون شمارش‌های موثر را استخراج می‌کند:

```
awk '{print$2"\t"$8}'results.xprs > eff_counts.txt
```

بخش ابتدایی فایل نتایج به صورت زیر است:

```
target_id                                   eff_counts
gi|530366287|ref|XM_005273173.1|            0.000000
gi|223555918|ref|NM_152415.2|               463.539280
gi|530387564|ref|XM_005273400.1|            0.481096
gi|530387566|ref|XM_005273401.1|            25.786556
gi|223555920|ref|NM_001145152.1|            9.204109
gi|225543473|ref|NM_004686.4|               28.171057
```

می‌توان تنها مشخصه‌های RefSeq را حفظ کرده و اعشارها را با استفاده از دستور زیر پیرایش نمود. در این دستور، گزینه‌ی F- علامت جدا کننده (در اینجا |) را تخصیص می‌دهد. نخستین خط

1- Bundle

به همان صورتی که هست، کُپی می‌شود (NR==1{print;next}). برای خطوط بعدی، تنها موارد چهارم و پنجم نگه داشته شده و تعداد در مورد پنجم گرد می‌شود.

```
awk -F'|''NR==1 {print;next}
{print$4"\t"int($5 + 0.5)}'eff_counts.txt > eff_counts_
rounded.txt
```

بخش ابتدایی فایل نتایج به صورت زیر است:

```
target_id          eff_counts
XM_005273173.1     0
NM_152415.2        464
XM_005273400.1     0
XM_005273401.1     26
NM_001145152.1     9
NM_004686.4        28
```

با استفاده از دستور زیر می‌توان تعداد رونوشت‌هایی که شمارش موثر گرد شده دارند، را بررسی نمود. در اینجا از دستور awk برای گردآوری خطوطی که مقدار موجود در ستون دوم آنها صفر نیست، استفاده شده و نتایج به دستور یونیکس wc -l که تعداد خطوط موجود را شمارش می‌کند، ارسال می‌گردد.

```
awk '$2!=0{print}'eff_counts_rounded.txt | wc -l
```

بر این اساس، ۵۲۲۵۹ رونوشت (از ۹۱۹۵۰ رونوشت اندازه‌گیری شده) دارای شمارش موثر هستند. برای تلفیق فایل‌های شمارش حاصل از نمونه‌های مختلف و تشکیل یک جدول شمارش، لازم است که داده‌ها بر مبنای ستون مشخصه‌ها مرتب شوند. دستورات زیر ردیف عنوان را استخراج کرده و داده‌های مرتب شده را به آن اضافه می‌کند:

```
head -n 1 eff_counts_rounded.txt > eff_counts_rounded_
sorted.txt

tail -n +2 eff_counts_rounded.txt | sort -k 1,1 >>eff_
counts_rounded_sorted.txt
```

شمارش خوانش‌ها به ازای رونوشت‌ها در Chipster

همان‌گونه که در فصل پنجم نیز تشریح گردید، می‌توان از ابزار RNA-seq/Assemble reads to transcripts with Cufflinks استفاده نمود. همچنین می‌توان eXpress را به کار گرفت.

- فایل(های) FASTQ ، فایل FASTA چندتایی حاوی توالی‌های رونوشت و ابزار RNA-seq / Count reads per transcripts usingeXpress را انتخاب کنید. در پنجره‌ی پارامتر، از تخصیص صحیح فایل‌ها اطمینان حاصل کنید.
- فایل‌های شمارش برای کلیه‌ی نمونه‌ها را انتخاب کرده و با استفاده از ابزار Utilities/Defne NGS experiment آنها را در قالب یک جدول شمارش تلفیق کنید. در پارامترها، ستون حاوی شمارش‌ها را انتخاب کرده و مشخص کنید که داده‌ها حاوی مختصات کروموزومی نیستند.

۶-۳-۳ شمارش خوانش‌ها به ازای هر اگزون

آنالیز افتراقی بیان می‌تواند در سطح اگزونی و با استفاده از بسته‌ی نرم‌افزاری DEXSeq در Bioconductor صورت گیرد (۱۶) (ر.ک: فصل نهم). بدین منظور لازم است که خوانش‌ها به ازای هر اگزون شمارش شوند. ایزوفرم‌های رونوشت تمایل دارند که برخی از اگزون‌ها را به صورت مشترک داشته باشند. بنابراین یک اگزون می‌تواند چندین بار در یک فایل GTF ظاهر شود. همچنین اگر مختصات شروع/پایان اگزون‌ها متفاوت باشند، می‌توانند با یکدیگر همپوشانی داشته باشند. برای اهداف شمارش؛ لازم است که یک مجموعه از نواحی اگزونی غیرهمپوشان ساخته شود. بسته‌ی نرم‌افزاری DEXSeq حاوی یک اسکریپت پایتون dexseq_prepare_annotation.py برای این کار است. این بسته یک فایل GTF را به فهرستی از اگزون‌ها که حاوی بِن‌ها[1] (متناظر با یک اگزون یا بخشی از یک اگزون (در مورد همپوشانی)) است، مسطح[2] می‌سازد. همان‌گونه که قبلاً در مورد HTSeq توضیح داده شد، استفاده از یک فایل GTF که در آن کلیه‌ی اگزون‌های یک ژن دارای ID یکسانی باشند، اهمیت دارد. فایل‌های GTF حاصل از Ensembl توصیه می‌شوند. زیرا این فایل‌ها از این قاعده پیروی می‌کنند. مثال زیر از فایل همردیفی جفت شده‌ی TopHat2 که در فصل چهارم به دست آمد، استفاده می‌نماید.

طبق روشی که در بخش HTSeq در همین فصل تشریح گردید، یک فایل GTF انسانی از Ensembl دانلود کرده و با کمک دستور زیر آنرا مسطح کنید:

```
python dexseq_prepare_annotation.py Homo_sapiens.GRCh37.74.
gtf GRCh37.74_DEX.gtf
```

1- Bin
2- Flatten

اسکریپت پایتون dexseq_count.py در بسته‌ی نرم‌افزاری DEXSeq برای شمارش خوانش‌ها به ازای بخش‌های اگزونی غیرهمپوشان استفاده می‌شود. این بسته‌ی نرم‌افزاری فایل GTF مسطح شده و خوانش‌های همردیف شده در فرمت SAM را به عنوان ورودی دریافت می‌کند. BAM نیز می‌تواند مورد استفاده قرار گیرد. ولی برای این کار باید بسته‌ی Pysam در پایتون را نصب نمود (۱۷). دستور زیر نشان می‌دهد که داده‌ها جفت انتهایی بوده (-p yes) و بر مبنای نام خوانش مرتب شده‌اند (-r name). همچنین این اسکریپت داده‌های مرتب شده بر مبنای مختصات کروموزومی را نیز می‌پذیرد (-r pos). لازم است که مشخص شود که داده‌ها زنجیره‌بندی نشده‌اند (-s no). زیرا این اسکریپت فرض می‌کند که داده‌ها با یک دستورالعمل مختص زنجیره ایجادگردیده‌اند. علاوه بر این می‌توان یک آستانه‌ی کیفی مکان‌یابی برای خوانش‌هایی که شمارش می‌شوند، تنظیم نمود (نظیر: -a 30). مقدار پیش‌فرض این آستانه‌ی کیفی برابر با ۱۰ است.

```
python dexseq_count.py -p yes -s no -r name GRCh37.74_DEX.
gtf hits_namesorted.sam exon_counts.txt
```

فایل شمارش، تعداد خوانش‌ها برای هر بِن شمارش شده‌ی اگزون را فهرست می‌نماید. مشخصه‌ی بِن شامل مشخصه‌ی ژن بوده و به دنبال آن یک شماره‌ی بِن اگزون نیز آورده شده است. همان‌گونه که در زیر نیز مشاهده می‌شود، برخی از مشخصه‌های بِن دارای دو مشخصه‌ی ژن جداگانه هستند که یک علامتِ بعلاوه در بین‌شان وجود دارد. این بدان معناست که دو ژن روی یک زنجیره واقع شده و اگزون‌های‌شان همپوشانی دارند.

```
ENSG00000001036:001     210
ENSG00000001036:002     12
ENSG00000001036:003     6
ENSG00000001036:004     135
ENSG00000001036:005     82
ENSG00000001036:006     205
ENSG00000001036:007     138
ENSG00000001036:008     2
ENSG00000001036:009     21
ENSG00000001036:010     76
ENSG00000001036:011     25
ENSG00000001084 + ENSG00000231683:001     57
ENSG00000001084 + ENSG00000231683:002     57
ENSG00000001084 + ENSG00000231683:003     50
ENSG00000001084 + ENSG00000231683:004     34
```

چهار ردیف انتهایی این فایل تعداد خوانش‌هایی را که به دلایل زیر شمارش نشده‌اند، فهرست می‌نمایند:

الف- این خوانش‌ها به هیچ‌وجه همردیف نشده‌اند (__notaligned).

ب- کیفیت همردیفی این خوانش‌ها کمتر از آستانه‌ی تخصیص یافته توسط کاربر است (__lowaqual).

پ- همردیفی این خوانش‌ها با بیش از یک بِن شمارش شده‌ی اگزونی همپوشانی داشته است (__ambiguous).

ت- همردیفی این خوانش‌ها با هیچ بِن شمارش شده‌ی اگزونی همپوشانی نداشته است (__empty).

همان‌گونه که در بخش HTSeq توضیح داده شد، می‌توان این خطوط را حذف نمود (head -n -4). همچنین همان‌گونه که در بخش HTSeq نشان داده شد، شمارش‌های حاصل از نمونه‌های مختلف را می‌توان با استفاده از دستور join یونیکس در قالب یک جدول شمارش تلفیق نمود. البته این کار برای DEXSeq ضرورتی ندارد.

شمارش خوانش‌ها به ازای اگزون‌ها در Chipster

- فایل همردیفی (BAM)، و ابزار RNA-seq/Count aligned reads per exonsforDEXSeq را انتخاب کنید. در پارامترها، جاندار مورد نظر را انتخاب نموده و مشخص نمایید که داده‌های شما با کدام‌یک از دستورالعمل‌های جفت انتهایی یا مختص زنجیره ایجاد شده‌اند. روی Run کلیک کنید.
- فایل‌های شمارش را برای کلیه‌ی نمونه‌ها انتخاب کرده و آنها را با استفاده از ابزار Utilities/Define NGS experiment در قالب یک جدول شمارش تلفیق کنید. در پارامترها ستون‌های حاوی شمارش‌ها را تعیین کرده و نیز مشخص کنید که آیا داده‌های شما حاوی مختصات کروموزومی است یا خیر.

۶-۴ خلاصه

انطباق موقعیت‌های ژنومی خوانش‌های همردیف شده با حاشیه‌نگاری مرجع این امکان را فراهم می‌آورد که بتوان جنبه‌های کیفی مهم نظیر اشباع عمق توالی‌یابی، یکنواختی پوشش در طول

رونوشت‌ها و توزیع خوانش بین انواع ترکیبات ژنومی مختلف را مورد بررسی قرار داد. چندین ابزار برای کنترل کیفیت مبتنی بر حاشیه‌نگاری در دسترس بوده و همگی آنها مزایای اختصاصی خود را دارند.

وقتی که خوانش‌ها با یک مرجع مکان‌یابی می‌شوند، می‌توان با استفاده از شمارش‌خوانش‌ها به ازای ژن‌ها، رونوشت‌ها و اگزون‌ها، بیان ژن را نیز کمی‌سازی نمود. کمی‌سازی و آنالیز افتراقی بیان ذاتاً به هم مرتبط بوده و بهترین شیوه‌ها هنوز مورد بحث و مناقشه است. خوانش‌ها به ازای ژن‌ها را می‌توان با کمک ابزارهایی نظیر HTSeq شمارش نمود. ولی شمارش‌های در سطح ژن برای آنالیز افتراقی بیان ژن‌هایی که دچار تغییر ایزوفرم شده‌اند، بهینه نیست (زیرا رونوشت‌های بلندتر، شمارش‌های بیشتری نیز ارائه می‌دهند). چالش اصلی در کمی‌سازی بیان در سطح رونوشت، چگونگی تخصیص مبهم خوانش‌های مکان‌یابی شده به ایزوفرم‌های مختلف است. Cufflinks و eXpress از یک روش EM برای این منظور استفاده می‌کنند. Cufflinks نیازمند یک ژنوم مرجع است. ولی eXpress از همردیف‌های ترانسکریپتومی استفاده کرده و بنابراین برای جاندارانی که ژنوم مرجع‌شان در دسترس نیست نیز به کار برده می‌شود. کمی‌سازی در سطح ایزوفرم نیز چالش‌برانگیز است. زیرا معمولاً به دلیل مسائل مربوط به قابلیت مکان‌یابی و اُریبی‌های ایجاد شده در مرحله‌ی آماده‌سازی کتابخانه و توالی‌یابی، پوشش رونوشت یکنواخت نیست. برای اهداف گزارش فراوانی، شمارش‌ها را می‌توان برای اندازه‌ی کتابخانه و طول رونوشت، با استفاده از واحدهایی نظیر FPKM و TPM نرمال‌سازی نمود. معمولاً آنالیز افتراقی بیان از شمارش‌های خام استفاده کرده و یک روش نرمال‌سازی داخلی را برای در نظر گرفتن تفاوت‌ها در ترکیب ترانسکریپتوم به کار می‌گیرد.

منابع

1. Wang L., Wang S., and Li W. RSeQC: Quality control of RNA-seq experiments. Bioinformatics 28(16):2184–2185, 2012.
2. DeLuca D.S., Levin J.Z., Sivachenko A. et al. RNA-SeQC: RNA-seq metrics for quality control and process optimization. Bioinformatics 28(11):1530–1532, 2012.
3. Garcia-Alcalde F., Okonechnikov K., Carbonell J. et al. Qualimap: Evaluating next-generation sequencing alignment data. Bioinformatics 28(20):2678–2679, 2012.
4. Picard. Available from: http://picard.sourceforge.net/.
5. GFF/GTF fle format description. Available from: http://genome.ucsc.edu/FAQ/FAQformat.html#format3.
6. BED fle format description. Available from: http://genome.ucsc.edu/FAQ/FAQformat.html#format1.

7. UCSC Table Browser. Available from: http://genome.ucsc.edu/cgi-bin/hgTables.
8. Trapnell C., Williams B.A., Pertea G. et al. Transcript assembly and quantifcation by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 28(5):511–515, 2010.
9. Mortazavi A., Williams B.A., McCue K., Schaeffer L., and Wold B. Mapping and quantifying mammalian transcriptomes by RNA-seq. Nat Methods 5(7):621–628, 2008.
10. Wagner G.P., Kin K., and Lynch V.J. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. Teory Biosci 131(4):281–285, 2012.
11. Anders S., Pyl P.T., and Huber, W. HTSeq – A Python framework to work with high-throughput sequencing data. bioRxiv doi: 10.1101/002824, 2014.
12. Quinlan A.R. and Hall I.M. BEDTools: A flexible suite of utilities for comparing genomic features. Bioinformatics 26(6):841–842, 2010.
13. Roberts A. and Pachter L. Streaming fragment assignment for real-time analysis of sequencing experiments. Nat Methods 10(1):71–73, 2013.
14. Roberts A., Schaeffer L., and Pachter L. Updating RNA-seq analyses afer reannotation. Bioinformatics 29(13):1631–1637, 2013.
15. Pruitt K.D., Tatusova T., Brown G.R., and Maglott D.R. NCBI Reference Sequences (RefSeq): Current status, new features and genome annotation policy. Nucleic Acids Res 40(Database issue):D130–D135, 2012.
16. Anders S., Reyes A., and Huber W. Detecting differential usage of exons from RNA-seq data. Genome Res 22(10):2008–2017, 2012.
17. Pysam. Available from: https://code.google.com/p/pysam/.

فصل هفتم

چارچوب آنالیز توالی‌یابی RNA در R و Bioconduictor

۷–۱ مقدمه

R یک نرم‌افزار آماری متن باز برای برنامه‌نویسی آماری، یک محیط آنالیز و یک اجتماع تشکیل شده از کاربران و توسعه‌دهندگان این نرم‌افزار است (تیم هسته R (۱) http://www.r-project.org).
نرم‌افزار R شامل یک هسته و هزاران بسته‌ی افزودنی اختیاری است که عملکرد هسته را گسترش می‌دهند. هسته‌ی R توسط تیم توسعه‌ی هسته‌ی R بسط و توسعه یافته ولی اکثر بسته‌های افزودنی با مشارکت افراد دیگر نظیر محققین دانشگاهی از دانشگاه‌های مختلف در گوشه و کنار جهان گسترش می‌یابند. Bioconductor یک پروژه‌ی بزرگ توسعه‌ی نرم‌افزاری است که ابزارهایی برای آنالیز داده‌های کارآمد و پُربروندا فراهم می‌آورد (۲) (http://www.bioconductor.org).
نرم‌افزارهای توسعه یافته در پروژه‌ی Bioconductor به صورت بسته‌های افزودنی R ارائه می‌گردند.

R یک زبان برنامه‌نویسی برای آمار، داده‌کاوی و نیز بیوانفورماتیک است. این زبان از جهت تاکید زیادی که بر عملکرد آماری دارد، با اکثر زبان‌های برنامه‌نویسی متفاوت است. چند زبان دیگر نظیر پایتون نیز هستند که عملکردهای جامع آماری و محاسباتی دارند. ولی R یک نفش ویژه در این بین دارد. زیرا این زبان اکثر مرزهای داغ توسعه‌ی دانش را پیش از سایر زبان‌ها درک می‌کند. در زمینه‌ی آماری، R می‌تواند تا حدودی با نرم‌افزارهایی نظیر SAS و Stata که هر دو نیز دارای زبان برنامه‌نویسی برای اجرای آنالیزها هستند، مقایسه شود. برای کارهای آماری یا بیوانفورماتیکی پایه، آگاهی از همه‌ی ظرافت‌های برنامه‌نویسی R ضرورتی نداشته و فرد می‌تواند تنها با دانستن متداول‌ترین توابع و عملکردهای مورد استفاده، آنالیزها را تقریباً با موفقیت انجام دهد. ولی جستجوی عمیق‌تر در این زبان به آنالیزهای سخت‌تر یا مراحل مختلف دست‌ورزی داده‌ها که گاهی اوقات می‌توانند پیچیده‌تر نیز شوند، کمک می‌کند.

این فصل دیدگاهی کُلی از عملکرد R و Bioconductor برای آنالیزهای توالی‌یابی کارآمد و پُربروندا ارائه می‌دهد. اگر می‌خواهید با عملکرد R آشنا شوید، راهنماهای موجود در http://cran.r-project.org/manuals.html نیز می‌تواند مفید واقع شود. در زمان نصب R این راهنماها همراه با آن ارائه می‌گردند. علاوه بر این راهنماهای پایه، کتاب‌های مفیدی نیز در این زمینه نگارش یافته است. در بین این کتاب‌ها، کتاب R in action نوشته‌ی Kabacoff از انتشارات

Manning ، کتاب Statistical computing with R نوشته Rizzo از انتشارات CRC Press و کتاب R in a nutshell نوشته Adler از انتشارات O'Reilly بیشتر توصیه می‌گردند.

۷-۱-۱ نصب R و افزودن بسته‌های نرم‌افزاری به آن

R را می‌توان از شبکه‌ی آرشیو جامع R[1] (CRAN) (http://cran.at.r-project.org/) یا از هر کدام از نشانی‌های آیینه‌ای آن در سرتاسر دنیا نصب نمود. لینک نشانی‌های آیینه‌ای به CRAN در صفحه‌ی اصلی پروژه‌ی R و در ذیل عنوان Download, Packages یافت می‌شود. لازم است که پایه‌ی R از یکی از سرورهای آیینه‌ای CRAN دانلود و نصب شود. لینک مستقیم به پایه‌ی R برای صفحه‌ی دانلود ویندوز روی نشانی آیینه‌ای CRAN اصلی در استرالیا عبارت است از: http://cran.at.r-project.org/bin/windows/base/. نصب کننده را دانلود کرده، اجرا نموده و دستورات و راهنمایی‌های ارائه شده از سوی آنرا دنبال کنید. اگر موسسه‌ای که در آن کار می‌کنید، اجازه‌ی نصب نرم‌افزار روی رایانه‌ی شخصی در محیط کار را نمی‌دهد، با واحد پشتیبانی IT مشورت کرده و آنها را به صفحات فوق‌الذکر رهنمون سازید.

بعد از نصب R پایه، معمولاً می‌توانید بسته‌های افزودنی را مستقیماً از R نصب نمایید. برای شناختن بسته‌های مورد نیاز، لازم است که قدری تحقیق نمایید. یک فهرست قابل جستجو از بسته‌های CRAN در http://cran.at.r-project.org/web/packages/available_packages_by_name.html قابل دسترس است. این فهرست حاوی توضیحاتی از عملکردهای هر کدام از این بسته‌ها است. علاوه بر این، یک گروه‌بندی جامع‌تر از بسته‌های مزبور بر مبنای عملکرد آنها در http://cran.at.r-project.org/web/views/ ارائه شده است. توضیحی از بسته‌های Bioconductor نیز در http://www.bioconductor.org/packages/release/BiocViews.html موجود است.

وقتی که بسته‌های مورد نیاز را شناسایی نمودید، می‌توانید آنها را به روش زیر نصب کنید:

۱- برای بسته‌های CRAN، به منوی Packages در برنامه‌ی R رفته و Install Packages(s) را انتخاب نمایید. در اینجا نیز لازم است که نشانی آیینه‌ای CRAN که می‌خواهید نصب را از آنجا انجام دهید و نیز بسته(هایی) که می‌خواهید نصب کنید، را انتخاب نمایید. سپس R به صورت خودکار، بسته‌ها را دانلود کرده و نصب می‌کند.

۲- بسته‌های Bioconductor را می‌توان مشابه بسته‌های CRAN نصب نمود. ولی روش پیشنهادی آن است که ابتدا تابع کمکی biocLite() از سایت Bioconductor لود شود. سپس `source("http://www.bioconductor.org/biocLite.R")` را

1- Comprehensive R Archive Network (CRAN)

روی خط فرمان R تایپ کرده و دکمه‌ی Enter را فشار دهید تا فرمان مزبور اجرا شود. اگر قبلاً تابع کمکی لود شده باشد، می‌توانید Bioconductor را برای نخستین بار با دادن فرمان `biocLite()` نصب نمایید. هر بسته را می‌توان با دادن برهان مربوط به آن بسته به تابع کمکی نصب نمود. به عنوان مثال، بسته‌ی Gviz برای مصورسازی ژنومی را می‌توان با دستور `biocLite("Gviz")` نصب کرد.

در برخی از موارد، بسته‌ها نمی‌توانند مستقیماً نصب شوند. زیرا دیواره‌ی آتشین شبکه ارتباط با نشانی‌های آیینه‌ای CRAN را مسدود می‌نماید. در اغلب موارد این وضعیت روی ویندوز با دادن دستور `setInternet2()` برطرف می‌شود. این دستور به R این امکان را می‌دهد که از عملکردهای اینترنت اکسپلورر نظیر تخصیص پروکسی‌ها استفاده نماید.

۷-۱-۲ استفاده از R

R یک ابزار مبتنی بر خط فرمان است. سیستم‌های عامل ویندوز و مَک یک رابط کاربری گرافیکی ساده برای R ارائه می‌دهند. ولی در لینوکس (و یونیکس) خط فرمان تنها رابط کاربری برای R است. چند رابط کاربری گرافیکی برای R نظیر R Commander و محیط‌های توسعه یافته‌تر و ویرایشگرهای کُد نظیر R Studio و Tinn R نیز وجود دارند. حتی محیط‌های برنامه‌نویسی گرافیکی برای R نظیر آنچه که توسط Alteryx پیشنهاد شد، نیز موجود است. ولی اگر R در قالب خط فرمان مورد استفاده قرار گیرد، دسترسی به همه‌ی عملکردها به جامع‌ترین شکل ممکن امکان‌پذیر است.

هر خط جدید در ویرایشگر R با پرومپت[1] که یک علامت ساده‌ی > است، آغاز می‌شود. دستورات و توابع در جلوی پرومپت نوشته شده و سپس با فشردن دکمه‌ی Enter روی صفحه کلید اجرا می‌شوند. نکته‌ی کلیدی برای استفاده‌ی موفقیت‌آمیز از R این است که باید بدانید که چه چیزی در جلوی پرومپت بنویسید. هدف فصل‌های بعدی این کتاب نیز آن است که به شما این ایده را بدهد که چگونه برخی از انواع آنالیزها را می‌توان در R اجرا نمود. ولی کتاب حاضر یک راهنمای پایه برای R نبوده و حداقل نیازمند قدری آگاهی قبلی از R است تا بتوان با موفقیت ایده‌ی اصلی این کتاب را اجرایی نمود.

در این کتاب وقتی که با خطوط کُد مواجه می‌شوید، آنها را در یک خط و به صورت همزمان اجرا کرده و سپس آنچه از اجرای خط مزبور حاصل می‌شود را ملاحظه نمایید. همچنین، استفاده از راهنمای برنامه برای توابع جدیدی که قبلاً با آنها آشنایی ندارید، نیز مفید واقع می‌گردد. صفحه‌ی

1- Prompt

راهنمایی و کمک (help) برای یک تابع را می‌توان با استفاده از ? از تابع help() فراخوان نمود. به عنوان مثال، صفحه‌ی راهنمای تابع lm() را می‌توان با نوشتن دستور ?lm فراخوانی کرد.

۷-۲ نگاه کلی به بسته‌های نرم‌افزار Bioconductor

بسته‌های افزودنی تولید شده توسط پروژه‌ی Bioconductor را می‌توان به بسته‌های نرم‌افزاری، حاشیه‌نگاری و آزمایشی تقسیم نمود. بسته‌های نرم‌افزاری دارای عملکرد آنالیزی، بسته‌های حاشیه‌نگاری حاوی انواع مختلف حاشیه‌نگاری‌ها و بسته‌های آزمایشی دارای مجموعه‌ی داده‌هایی که اغلب به عنوان مثال‌هایی برای عملکردهای بسته‌ها استفاده می‌شوند، هستند. در زیر جزییات بیشتری از هر کدام از این دسته‌ها ارائه می‌شود.

۷-۲-۱ بسته‌های نرم‌افزاری

عموماً بسته‌های نرم‌افزاری Bioconductor دارای قابلیت ورود، دست‌ورزی (پیش‌پردازش و کنترل کیفیت)، آنالیز، رسم نمودار و گزارش‌دهی نتایج حاصل از آزمایشات کارآمد و پُربروندادهستند. مهم‌ترین بسته‌ها برای آزمایشات توالی‌یابی RNA عبارتند از: ۱) Short Read و Rsamtools برای خواندن و نگارش فایل‌های توالی، ۲) IRanges ، GenomicRanges و Biostrings برای دست‌ورزی داده‌ها، ۳) edgeR ، DESeq و DEXSeq برای آنالیزهای آماری و ۴) rtracklayer ، BSgenome و biomaRt برای حاشیه‌نگاری نتایج.

۷-۲-۲ بسته‌های حاشیه‌نگاری

پروژه‌ی Bioconductor بسته‌های حاشیه‌نگاری پایه برای تعداد زیادی از جانداران تولید کرده است. این بسته‌های حاشیه‌نگاری را می‌توان به توالی‌های ژنومی (بسته‌ی BSgenome)، حاشیه‌نگاری گستره‌ی ژنومی (بسته‌ی org)، رونوشت (بسته‌ی TxDB)، همولوژی (بسته‌ی hom)، هدف microRNA (بسته‌های RmiR و targetscan)، بسته‌های حاشیه‌نگاری عملکردی (DO ، GO، KEGG و reactome)، واریانت‌ها (بسته‌ی SNPlocs) و پیش‌بینی توابع واریانت (بسته‌ی SIFT و بسته‌ی PolyPhen) تقسیم نمود. معمولاً این بسته‌ها حاشیه‌نگاری‌هایی از منابع آمریکایی نظیر Genbank و UCSC ارائه کرده و شماره‌های دسترسی[1] برای ژن‌ها را از Entrez Gene اخذ می‌کنند. با این حال این بسته‌ها از طریق org-package مختص جاندار، قابلیت ترجمه‌ی ID های Entrez Gene به سایر ID ها نظیرID های Ensembl را دارند.

1- Accession number

علاوه بر بسته‌های حاشیه‌نگاری آماده، حاشیه‌نگاری‌ها می‌توانند مستقیماً از منابع آنلاین نیز جستجو گردند. بسته‌ی biomaRt در Bioconductor این امکان را به کاربر می‌دهد که به مخزن داده‌های ژنومی BioMart دسترسی پیدا کند. همچنین بسته‌ی rtracklayer به کاربر اجازه‌ی جستجو در مسیرهای حاشیه‌نگاری مرورگر ژنومی UCSC را می‌دهد. علاوه بر این، بسته‌های arrayexpress و GEOquery می‌توانند R را با پایگاه داده‌های ArrayExpress و GEO مرتبط گردانند.

۷-۲-۳ بسته‌های آزمایشی

بسته‌های آزمایشی حاوی مجموعه‌ی داده‌های آماده شده و با دسترسی رایگان هستند. در این کتاب، مجموعه‌ی داده‌های parathyroid از بسته‌ای به همین نام و برای نشان دادن آنالیزهای آماری با استفاده از بسته‌های نرم‌افزاری DESeq و DEXSeq به کار گرفته شده است.

۷-۳ خصوصیات توصیفی بسته‌های Bioconductor

بسته‌های Bioconductor به طور وسیعی از الگوی برنامه‌نویسی شیءگرا[1] (OOP) بهره می‌برند. OOP در R با روش‌هایی که روی کلاس‌های شیء S3 و S4 کار می‌کنند، تفسیر می‌گردد. S3 تنها جنبه‌های خاصی از OOP را شبیه‌سازی کرده ولی S4 یک سامانه‌ی رسمی OOP بوده که ویرایش چهارم زبان برنامه‌نویسی S نامیده می‌شود. از این زبان برای پیاده‌سازی متن باز R استفاده شده است. هر کلاس یک یا چند کلاس دیگر را بسط داده و در مقایسه با کلاس‌های جاوا، کلاس‌های S4 روش‌های اختصاصی خود را ندارند. معمولاً یک تابع عمومی وجود دارد که یک تابع اختصاصی برای مجموعه‌ی خاصی از توابع را انتخاب می‌نماید. توابع اختصاصی را اصطلاحاً روش[2] می‌نامند. سامانه‌ی OOP که در R پیاده‌سازی شده است، توسط Chambers تشریح گردیده است (۳ و ۴).

۷-۳-۱ خصوصیات OOP در R

جایی‌که یک تابع در R پایه وجود دارد، اغلب یک روش نیز در OOP وجود دارد. به همین ترتیب وقتی که یک جدول (ماتریس یا چارچوب داده[3]) یا یک فهرست در R پایه استفاده می‌شود، یک شیء S3/S4 در OOP مورد استفاده واقع می‌گردد. کلاس‌های S3 و S4 اشیاء حاوی شیارهایی هستند که انواع مختلف داده‌ها را ذخیره می‌نمایند. یک ستون از چارچوب داده‌ها می‌تواند توسط

1- Object-Oriented Programming (OOP)
2- Method
3- Data frame

عملگر[1] $ در دسترس قرار گیرد. ولی برای کلاس‌های S3/S4 اشیاء، شیارهای منفرد با استفاده از عملگر @ در دسترس قرار می‌گیرند. برای استخراج یک شیار از اشیاء S4 بهتر است که به جای عملگر @ از یک تابع دسترسی[2] استفاده شود. زیرا استفاده از تابع دسترسی، مستقل از ارائه‌ی کلاس است. اگر نام شیار تغییر کند، فعالیت عملگر @ متوقف می‌شود. ولی اگر تابع دسترسی توسط طراح بسته به خوبی به‌روزرسانی شده باشد، بدون مشکل به فعالیت خود ادامه می‌دهد.

برای ملموس‌تر کردن موضوع، یک ژن را به صورت یک شیء دامنه‌ی توالی در نظر بگیرید. برای این منظور می‌توان بسته‌ی GenomicRanges را مورد استفاده قرار داد. با استفاده از تابع GRanges() ، یک شیء دامنه‌ی توالی جدید ایجاد می‌گردد. کُد زیر نمایشی از ژن XRCC1 است که روی زنجیره‌ی مستقیم کروموزوم شماره‌ی ۱۹ بین موقعیت‌های ۴۴۰۴۷۴۶۴ و ۴۴۰۴۷۴۹۹ واقع شده است:

```
library(GenomicRanges)
read<-GRanges(seqnames=c("19"),
                ranges=IRanges(start=c(44047464),
                end=c(44047499)), strand=c("+"),
                seqlenghts=c("19"=591289983))
names(read)<-c("XRCC1")
```

اگر محتوای خوانش شیء توسط تابع str() کنترل شود، بایستی خروجی به صورت زیر نمایش داده شود. آیا می‌توانید اطلاعات وارد شده برای ژن XRCC1 را در این خروجی پیدا کنید؟

```
str(read)

Formal class:'GRanges ' [package "GenomicRanges"] with 6
Slots
  .. @ seqnames: Formal class 'Rle'[package "IRanges"]
  with 4 slots
  ......@ values             : Factor w/1 level "19":1
  ......@ lengths            : int 1
  ......@ elementMetadata    : NULL
  ......@ metadata           : list()
  ..@ ranges: Formal class 'IRanges' [package
  "IRanges"]with 6 slots
  ......@ start              : int 44047464
  ......@ width              : int 36
  ......@ NAMES              : chr "XRCC1"
```

1- Operator
2- Accessor function

```
  ......@ elementType         : chr "integer"
  ......@ elementMetadata     : NULL
  ......@ metadata            : list()
  ..@ strand: Formal class 'Rle' [package "IRanges"]
  with 4 slots
  ......@ values              : Factor w/3 levels " + ",
  "-","*":1
  ......@ lengths             : int 1
  ......@ elementMetadata     : NULL
  ......@ metadata            : list()
  ..@ elementMetadata: Formal class 'DataFrame'
  [package "IRanges"] with 6 slots
  ......@ rownames            : NULL
  ......@ nrows               : int 1
  ......@ listData            : List of 1
  ........$ seqlenghts:Named num 59128983
  ..........- attr(*,"names") = chr "19"
  ......@ elementType         : chr "ANY"
  ......@ elementMetadata     : NULL
  ......@ metadata            : list()
  .. @ seqinfo: Formal class 'Seqinfo' [package
  "GenomicRanges"] with 4 slots
  ......@ seqnames            : chr "19"
  ......@ seqlengths          : int NA
  ......@ is_circular         : logi NA
  ......@ genome              : chr NA
  .. @ metadata               : list()
```

همه‌ی شیارهای اشیاء قبل از علامت @ نوشته شده و می‌توانند با همین عملگر نیز در دسترس قرار گیرند. به عنوان مثال، شیار NAMES می‌تواند با دستور زیر استخراج شود:

```
read@ranges@NAMES
[1]"XRCC1"
```

با این حال، تابع دسترسی برای مهم‌ترین شیارهای اشیاء نیز وجود دارد. به عنوان مثال، نام توالی با استفاده از تابع `names()` نیز در دسترس قرار می‌گیرد:

```
names(read)
[1]"XRCC1"
```

گاهی اوقات اشیاء S3/S4 می‌توانند مستقیماً به انواع دیگری از اشیاء تبدیل شوند. ولی معمولاً این کار همیشه امکان‌پذیر نیست. به عنوان مثال، برای نوشتن اشیاء جی‌رنجز[1] نظیر یک جدول روی یک دیسک، می‌توان ابتدا آنرا به یک چارچوب داده تبدیل نمود:

```
read.df <-as.data.frame(read)
read.df
     seqnames start end width strand seqlenghts
XRCC1 19 44047464 44047499 36 + 59128983
```

بدین ترتیب، نگارش یک چارچوب داده روی یک دیسک می‌تواند از طریق مسیر معمولی `write.table()` انجام شود. برای دسترسی به هر ستون چارچوب داده‌ها می‌توان از عملگر @ استفاده نمود. به عنوان مثال، پهنای یک خوانش که در چارچوب داده‌های read.df ذخیره شده است، را می‌توان به صورت زیر در صفحه‌ی نمایش مشاهده کرد:

```
read.df$width
[1]36
```

به طور مشابه می‌توان با تابع زیر، به نام توالی‌های ذخیره شده روی خطوط دسترسی پیدا نمود:

```
rownames():
rownames(read.df)
[1]"XRCC1"
```

۴-۷ تعریف ژن‌ها و رونوشت‌ها در R

معمولاً ژن‌ها و رونوشت‌ها به صورت دامنه‌های توالی[2] (نظیر اشیاء جی‌رنجز) در R تعریف می‌گردند. زیرا ژن‌ها و رونوشت‌ها ویژگی‌های محاسباتی خوبی دارند: فضای کمی را اشغال کرده و می‌توان آنها را با الگوریتم‌های سریع مدیریت نمود. بسته‌هایی که بدین منظور مورد استفاده قرار می‌گیرند، IRanges و GenomicRanges هستند.

Rsamtools بسته‌ای است که عملکرد samtools را در R فراهم می‌سازد. به عنوان مثال، این بسته می‌تواند برای خواندن فایل‌های BAM در R به کار گرفته شود. فایل BAM به یک شیء کلاس جی‌رنجز تبدیل می‌شود که حاوی کلیه‌ی خوانش‌های منفرد حاصل از فایل BAM خواهد

1- GRanges
2- Sequence range

بود. برای خواندن یک فایل BAM در R ، می‌توان از تابع readGappedAlignments() استفاده نمود. به عنوان مثال:

```
library(Rsamtools)
h1b <-readBamGappedAlignments("hESC1_chr18.bam")
```

شیء h1b حاوی توالی همردیف شده‌ی حاصل از یک فایل BAM است. اطلاعات کلیدی برای هر خوانش شامل کروموزوم، زنجیره و موقعیت جفت باز خوانش مکان‌یابی شده در ژنوم است. با تایپ کردن نام شیء در پرومپت می‌توان یک بسط کوتاه از ابتدا و انتهای شیء مزبور را در R روی صفحه‌ی نمایش ملاحظه نمود:

```
h1b
```

```
GappedAlignments with 836162 alignments and 0
metadatacolumns:
```

	seq names <Rle>	strand <Rle>	cigar <char-acter>	qwidth <integer>	start <integer>	end <integer>	width <integer>
[1]	chr18	+	75M	75	28842	28916	75
[2]	chr18	+	75M	75	35847	35921	75
[3]	chr18	-	75M	75	46570	46644	75
[4]	chr18	-	75M	75	46570	46644	75
[5]	chr18	+	75M	75	47246	47320	75
...	...	...	...	...	...	...	...
[836158]	chr18	+	75M	75	78005301	78005375	75
[836159]	chr18-	-	75M	75	78005301	78005375	75
[836160]	chr18	+	75M	75	78005307	78005381	75
[836161]	chr18-	-	75M	75	78005309	78005383	75
[836162]	chr18-	-	75M	75	78005366	78005440	75

	ngap <integer>
[1]	0
[2]	0
[3]	0
[4]	0
[5]	0
...	...
[836158]	0
[836159]	0
[836160]	0
[836161]	0
[836162]	0

```
--
seqlengths:
     chr1      chr2      chr3      chr4 ...     chr22      chrX  chrM
249250621 243199373 198022430 191154276 ... 51304566 155270560 16571
```

در صورت لزوم، اشیاء دامنه‌ای نظیر آنچه که در کلاس جی‌رنجز دیده می‌شود، می‌توانند با استفاده از مسیر معمول زیرمجموعه‌بندی اشیاء توسط کروشه‌ها، زیرمجموعه‌بندی شوند. به عنوان مثال، ۱۰ ژن نخست را می‌توان با دستور زیر در صفحه‌ی نمایش مشاهده نمود:

```
h1b[1:10,]
```

به طور مشابه، می‌توان ژن‌هایی را که فقط روی زنجیره‌ی مستقیم قرار دارند، استحراج کرد:

```
h1b[strand(h1b) = =" + ",]
```

اگر قبلاً توالی‌های همردیف شده‌ی حاصل از فایل BAM در یک شیء مبتنی بر دامنه لود شده‌اند، می‌توان به آسانی تعداد خوانش‌ها برای هر ژن را شمارش نمود. رونوشت‌ها برای جانداران مدل به صورت آماده و در قالب بسته‌های R در دسترس هستند. به عنوان مثال، رونوشت حاصل از اسمبل hg19 ژنوم انسانی که از سرور Genome پایگاه UCSC گرفته شده است، در بسته‌ی TxDb.Hsapiens.UCSC.hg19.knownGene از Bioconductor در دسترس است. ژن‌ها را می‌توان به صورت دامنه‌هایی از این بسته و در قالب یک شیء جدید txdb استخراج نمود:

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb<-transcriptsBy(TxDb.Hsapiens.UCSC.hg19.knownGene,
                    "gene")
```

به جای ژن‌ها می‌توان اگزون‌ها، توالی‌های رمزگر یا رونوشت‌های با عملکرد یکسان را نیز استخراج کرد. اگر قبلاً ترکیبات ژنومی مورد نظر استخراج شده باشند، می‌توان تعداد خوانش‌ها (یا دامنه‌های‌شان) که با ترکیبات ژنومی (یا دامنه‌های‌شان) همپوشانی دارند، را شمارش نمود. تابع countOverlaps() جزییات این شمارش را نشان می‌دهد:

```
hits <- countOverlaps(h1b, txdb)
ol<-countOverlaps(txdb, h1b[hits==1])
```

نخستین اجرای countOverlaps() خوانش‌های تکی را با شماره‌ی اجرا علامت‌گذاری می‌نماید. خوانش‌های با علامت یک تنها با یک ژن تکی مکان‌یابی شده و خوانش‌هایی که با این ژن‌ها مکان‌یابی شده‌اند، در طی دومین اجرای تابع countOverlaps() مکان‌یابی می‌گردند. شیء ol حاصل یک بردار عددی نامگذاری شده است که در آن نام‌ها همان مشخصه‌های Entrez Gene بوده و شماره‌ها نیز شمارش‌های خوانش‌هایی که با ژنِ نام‌گذاری شده همپوشانی دارند، می‌باشند.

با استفاده از توابع معرفی شده می‌توان تابعی را ایجاد نمود که تعداد فایل‌های BAM را بخواند، تعداد خوانش‌های مکان‌یابی شده با هر ژن را شمارش نماید و یک جدول شمارش ارائه کند. طبیعتاً این تابع تنها زمانی به درستی عمل می‌نماید که مکان‌یابی با همان نسخه از اسمبل ژنوم که از پروژه‌ی Bioconductor در دسترس است، انجام گیرد. کُدنویسی برای این تابع به شرح زیر است:

```
generateCountTable <- function(
    files,
    transcripts="TxDb.Hsapiens.UCSC.hg19.knownGene",
    overlapto="gene") {
        require(transcripts, character.only=TRUE)
        require(GenomicRanges)
        require(Rsamtools)
        txdb<-transcriptsBy(get(transcripts,
                                envir=.GlobalEnv),
                                overlapto)
        l<- vector("list", length(files))
        for(i in 1:length(files)) {
            alns <- readGappedAlignments(files[i])
            strand(alns) <- "*"
            hits <- countOverlaps(alns,txdb)
            l[[i]] <- countOverlaps(txdb, alns[hits==1])
            names(l) <- gsub("\\.bam", "", files)
       }
        ct<-as.data.frame(l)
        ct
}
```

تابع `generateCountTable()` به صورت زیر کار می‌کند. نخست دایرکتوری فعال را به همان دایرکتوری که حاوی فایل‌های BAM بوده تغییر دهید و سپس تابع را با استفاده از دستور زیر اجرا نمایید:

```
counttable<-generateCountTable(dir(pattern=".bam"))
```

نتیجه‌ی کار بایستی یک جدول شمارش مشابه جدول زیر باشد:

```
head(counttable)
```

	Gm12892_1_chr18	Gm12892_2_chr18	Gm12892_3_chr18	hESC1_chr18	hESC2_chr18	hESC3_chr18	hESC4_chr18
1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0
1000	27	72	12	4446	3300	3605	3498
10000	0	0	0	0	0	0	0
100008586	0	0	0	0	0	0	0

سپس جدول شمارش حاصل را می‌توان برای آنالیزهای بیشتر که در فصل بعد تشریح می‌گردند، مورد استفاده قرار داد.

۷-۵ تعریف ژنوم‌ها در R

ژنوم‌ها توسط اشیاء بایواسترینگ[1] تعریف می‌شوند. اسمبل ژنوم انسانی ویرایش‌های ۱۷ تا ۱۹ به صورت آماده از پروژه‌ی Bioconductor در دسترس است. ویرایش فعلی این ژنوم در بسته‌ی BSgenome.Hsapiens.UCSC.hg19 ارائه شده است. این ژنوم حاوی چندین شیء بایواسترینگ (یک شیء به ازای هر کروموزوم) بوده و در صورت لزوم می‌تواند به صورت اشیاء جداگانه استخراج شود.

```
library(BSgenome.Hsapiens.UCSC.hg19)
chr18 <-(BSgenome.Hsapiens.UCSC.hg19[["chr18"]])
```

یکی از کاربردهای جالب داده‌های ژنومی، مکان‌یابی مجدد خوانش‌ها است. مکان‌یابی در R نمی‌تواند به طور کامل توسط توابع مکان‌یاب‌های خارجی صورت گیرد. ولی این فرآیند در خود R قابل اجرا است. مثال زیر نحوه‌ی عملکرد مکان‌یابی در R را نشان می‌دهد. نخست لازم است که خوانش‌ها به صورت توالی‌هایی در R خوانده شوند. این کار با تخصیص پارامترهایی به تابع readBamGappedAlignments() صورت می‌گیرد:

```
p2 <-ScanBamParam(what=c("rname", "strand", "pos",
                         "qwidth", "seq"))
h1b<-readBamGappedAlignments("hESC1_chr18.bam",
                             param=p2)
```

سپس لازم است که توالی‌ها به یک شیء DNAStringSet تبدیل شوند. این کار به آسانی و با استخراج ستون‌های فراداده[2] از شیء h1b و سپس استخراج توالی‌های DNA از آن صورت می‌گیرد. به صورت پیش‌فرض، توالی‌های DNA در یک فرمت مناسب ذخیره می‌شوند. فرض شود که تنها چند هزار خوانش برای این مثال مورد استفاده قرار می‌گیرند:

```
seqs<-mcols(h1b)$seq
seqs2<-seqs[100:1100,]
```

1- Biostring
2- Metadata

سپس می‌توان مکان‌یابی را با استفاده از تابع matchPDict() اجرا نمود:

```
mpd<-matchPDict(seqs2, chr18)
```

به طور پیش‌فرض، مکان‌یابی به گونه‌ای انجام می‌شود که اجازه‌ی وجود هیچ عدم تطابق یا ایندلی در همردیف‌های جفت شده داده نشود. اگر قبلاً خوانش‌ها مکان‌یابی شده‌اند، می‌توان تعداد خوانش‌های مکان‌یابی شده که در ژن‌های معلوم واقع می‌شوند، را شمارش نمود. اصول شمارش در فوق مورد بررسی قرار گرفته است. ولی دریافت خوانش‌های مکان‌یابی شده در یک فرمت مناسب، مستلزم چند نکته است. ابتدا شیء حاوی خوانش‌های مکان‌یابی شده به یک CompressedIRangesList تبدیل شده و سپس به یک شیء جی‌رنجز تبدیل می‌گردد. هر دو تبدیل با استفاده از تابع as() صورت می‌گیرد. شیء جی‌رنجز قابل آنالیز نهایی از این شیء جی‌رنجز موقت ساخته می‌شود. چرخه‌ی کامل تبدیل در زیر تشریح شده است:

```
mpd2<-as(mpd, "CompressedIRangesList")
mpd3<-as(RangedData(mpd2), "GRanges")
gr<-GRanges(seqnames=Rle(rep("chr18", length(mpd3))),
            ranges=mpd3@ranges,
            strand=strand(mpd3),
            seqinfo = Seqinfo("chr18", 78077248)
            )
```

اگر از قبل اشیاء در فرمت مناسب باشند، می‌توان تعداد خوانش‌های مکان‌یابی شده در ژن‌های معلوم را شمارش نمود:

```
txdb_chr18<-keepSeqlevels(txdb, "chr18")
hits <- countOverlaps(gr, txdb_chr18)
ol<-countOverlaps(txdb, gr[hits==1])
```

در نتیجه‌ی نهایی، دو ژن دارای خوانش‌های مکان‌یابی شده هستند (یکی با ۱۷۸ خوانش و دیگری با ۲۶۲ خوانش). مجموعاً ۲۲۹۳۰ ژن نیز وجود دارند که هیچ ژنی با آنها مکان‌یابی نشده است:

```
table(ol)
ol

0178262
22930 1 1
```

۷–۶ تعریف SNP ها در R

Bioconductor بسته‌ی SNPlocs را برای ژنوم انسان ارائه داده است. بسته‌ی فعلی موقعیت SNP در زمان نگارش این کتاب، مبتنی بر ویرایش 137 پایگاه داده‌ی dbSNP بوده و تحت عنوان SNPlocs.Hsapiens.dbSNP.20120608 نامیده می‌شود. در صورت لزوم با استفاده از تابع `injectSNPs()` می‌توان موقعیت‌های SNP را در توالی ژنوم تزریق نمود. بعد از تزریق، می‌توان اطلاعات SNP را در طی مکان‌یابی کاوشگرها و سایر عملکردهای مشابه در نظر گرفت. تزریق SNP ها در ژنوم به سادگی صورت می‌گیرد:

```
library(SNPlocs.Hsapiens.dbSNP.20120608)
library(BSgenome.Hsapiens.UCSC.hg19)
genome <- injectSNPs(BSgenome.Hsapiens.UCSC.hg19,
                     "SNPlocs.Hsapiens.dbSNP.20120608")
```

پس از تزریق SNP ها، ژنوم شیء می‌تواند به عنوان توالی ژنومی برای هر آنالیز پایین‌دستی مورد استفاده واقع شود.

۷–۷ ساختن بسته‌های حاشیه‌نگاری جدید

پروژه‌ی Bioconductor بسته‌های حاشیه‌نگاری برای تعداد زیادی از جانداران مدل ارائه می‌کند. ولی اگر جاندارانی که شما با آنها کار می‌کنید، به صورت آماده در پروژه‌ی Bioconductor در دسترس نبوده ولی در عین حال در برخی از مرورگرهای ژنومی (معمولاً UCSC یا Ensembl) موجود باشند، می‌توان بسته‌های حاشیه‌نگاری را برای آنها ایجاد نمود. اگر بخواهید حاشیه‌نگاری‌ها را برای جانداران مورد نظر خود به‌روزرسانی کنید، ولی Bioconductor هنوز آنرا انجام نداده باشد (چرخه‌ی به‌روزرسانی برای Bioconductor شش ماهه است)، فرآیند مشابهی مورد نیاز است.

بسته‌های حاشیه‌نگاری گستره ژنومی[1] را می‌توان با استفاده از بسته‌ی AnnotationForge ایجاد نمود. فرض کنید که می‌خواهید یک بسته‌ی حاشیه‌نگاری گستره‌ی ژنومی جدید برای آلپاکا که یک حیوان جذاب کُرکی است، ایجاد نمایید. بدین منظور نیازمند دانستن نام علمی این گونه (*Vicugna pacos*) و ID آن در پایگاه اطلاعات ژنومی NCBI هستید. برای انجام کار نیز از تابع `makeOrgPackageFromNCBI()` استفاده می‌شود:

```
library(AnnotationForge)
makeOrgPackageFromNCBI(version = "0.1",
      author = "JarnoTuimala < name@server > ",
      maintainer = "JarnoTuimala < name@server > ",
```

1- Genome-wide

```
      outputDir = "C:/Users/JarnoTuimala/Desktop/
                  alpaca",
      tax_id = "30538",
      genus = "Vicugna",
      species = "pacos")
```

این تابع یک بسته‌ی حاشیه‌نگاری در مسیر تخصیص یافته توسط برهان outputDir (در اینجا در مسیر C:/users/JarnoTuimala/Desktop/alpaca) ایجاد خواهد نمود. پس از اتمامِ ساختن بسته، می‌توان آنرا با کمک تابع `install.packages()` روی R نصب نمود:

```
install.packages(pkgs="C:\\Users\\Jarno Tuimala\\
                  Desktop\\alpaca\\org.Vpacos.eg.db",
                  lib="C:\\Users\\ Jarno Tuimala\\
                  Documents\\R\\win-library\\3.0",
                  type="source", repos=NULL)
```

اگر بسته‌ی مختص جاندار مناسب باشد، می‌توان یک بسته‌ی رونوشت جداگانه برای آلپاکا ایجاد کرد. بدین منظور لازم است که جدولی که اطلاعات مورد نیاز در پایگاه داده‌ی Genome از NCBI در آن ذخیره می‌شود، در دسترس باشد. جداول موجود را می‌توان در R با کمک تابع `supportedUCSCtables()` فهرست نموده و نام دقیق جدول را در سایت UCSC و با کمک Table Browser یافت. برای آلپاکا، نام این جدول xenoRefGene است. به طور مشابه و با کمک Table Browser باید نام ژنوم آلپاکا را در پایگاه داده‌ی UCSC یافت. نام ژنوم مزبور نیز vicPac2 است.

بعد از یافتن نام‌های صحیح جدول و ژنوم، می‌توان با استفاده از تابع `makeTranscriptDbFromUCSC()` از بسته‌ی GenomicFeatures ، می‌توان یک شیء transcriptDb ایجاد نمود:

```
txdb <- makeTranscriptDbFromUCSC(genome="vicPac2",
                                 tablename="xenoRefGene")
```

با کمک شیء transcriptDb و با استفاده از دستور زیر، می‌توان یک بسته را اسمبل کرد:

```
makeTxDbPackage(txdb,
version="0.1",
maintainer="JarnoTuimala < name@server > ",
author="JarnoTuimala < name@server > ",
destDir="C:/Users/Jarno Tuimala/Desktop/alpaca",
license="Artistic-2.0")
```

توجه شود که مسیر دایرکتوری تخصیص داده شده توسط برهان destDir باید موجود باشد. بنابراین لازم است که ابتدا این دایرکتوری در مسیر مزبور ایجاد شود.

به جای این کار و در صورتی که بخواهید از Ensembl برای ساخت بسته‌ی حاشیه‌نگاری رونوشت استفاده کنید (بدون اجرای یک مرحله‌ی اضافی برای ساخت شیء txdb)، می‌توانید از دستور زیر بهره بگیرید:

```
makeTxDbPackageFromBiomart(
    version="0.1",
    maintainer="Jarno Tuimala < name@server > ",
    author="Jarno Tuimala < name@server > ",
    destDir="C:/Users/Jarno Tuimala/Desktop/alpaca2",
    license="Artistic-2.0",
    biomart="ensembl",
    dataset="vpacos_gene_ensembl",
    transcript_ids=NULL,
    circ_seqs=DEFAULT_CIRC_SEQS,
    miRBaseBuild=NA)
```

بسته‌ی حاصل را می‌توان مشابه آنچه که در فوق تشریح گردید، به بسته‌ی مختص جاندار مورد نظر نصب نمود.

علاوه بر بسته‌ی مختص جاندار و بسته‌ی رونوشت، با استفاده از این توابع می‌توان یک بسته‌ی جدید ژنومی نیز از بسته‌ی نرم‌افزاری BSgenome ایجاد کرد. ساخت بسته‌های SNP اندکی سخت‌تر از سایر بسته‌ها بوده ولی فولدر ابزارها در ذیل بسته‌ی SNPlocs حاوی اسکریپت‌های لینوکس است که می‌توان آنها را جهت ایجاد بسته‌ی SNP برای انسان و سایر جانداران تغییر داد. این بسته‌ها با جزییات در اینجا تشریح نشده‌اند. زیرا در مقایسه با سایر بسته‌های حاشیه‌نگاری، کمتر نیازمند به‌روزرسانی می‌شوند. همچنین این بسته‌ها در مقایسه با معمول‌ترین جانداران مدل، نیازمند تغییرات اضافی برای کار کردن هستند.

۷-۸ خلاصه

R حاوی مجموعه‌ی خوبی از مستندات است که می‌توانند در داخل R یا به صورت آنلاین و بر حسب صلاحدید کاربر در دسترس قرار گیرند. Bioconductor از سامانه‌ی OOP مربوط به S4 به صورت نسبتاً گسترده استفاده می‌کند که این امر در تضاد با اکثر بایگانی‌های CRAN است. Bioconductor علاوه بر پیاده‌سازی‌های توابع، مجموعه وسیعی از انواع مختلف حاشیه‌نگاری‌ها

برای تعداد زیادی از جانداران مدل را نیز ارائه می‌دهد. اگر بسته‌های حاشیه‌نگاری برای یک جاندار از پروژه‌ی Bioconductor در دسترس نباشد، می‌توان چنین بسته‌ای را ایجاد نمود.

منابع

1. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing, 2013. http://www.Rproject.org/.
2. Gentleman R.C., Carey V.J., Bates D.M., Bolstad B., Dettling M., Dudoit S., Ellis B. et al. Bioconductor: Open sofware development for computational biology and bioinformatics. Genome Biology, 5:R80, 2004.
3. Chambers J.M. Programming with Data: A Guide to the S Language. Berlin: Springer, 1998. ISBN 0-387-98503-4.
4. Chambers J.M. Sofware for Data Analysis Programming with R. Berlin: Springer, 2008. ISBN 0-387-75935-2.

فصل هشتم

آنالیز افتراقی بیان

۸–۱ مقدمه

آنالیز افتراقی بیان[1] (DE) شامل شناسایی ژن‌ها (یا سایر انواع ترکیبات ژنومی نظیر رونوشت‌ها یا اگزون‌ها) است که در کمیت‌های متفاوت معنی‌داری در گروه‌های متمایز نمونه‌ها (از نظر شرایط زیستی نظیر درمان شده با دارو در برابر شاهد، افراد بیمار در برابر افراد سالم، بافت‌های مختلف و مراحل مختلف تکامل) بیان می‌شوند. علی‌رغم اینکه ژن‌ها مستقل از هم بیان نمی‌شوند، ولی معمولاً آنالیز افتراقی بیان روی یک ژن در یک زمان انجام می‌شود (البته همان‌گونه که در بخش‌های بعدی ملاحظه می‌گردد، گاهی اوقات اطلاعات ژن‌های مختلف تبادل می‌شود). این بدان معناست که این آنالیز یک روش تک متغیری[2] است. زیرا علی‌رغم اینکه ممکن است بیان ده‌ها هزار ژن اندازه‌گیری شوند، ولی معمولاً تعداد نمونه‌های زیستی بسیار کمتر است. به عبارت دیگر، تعداد مثال‌ها بسیار کمتر از تعداد ترکیبات است و این موضوع سبب می‌شود که برازش یک مدل آماری که شامل کلیه‌ی ژن‌ها به صورت یک مجموعه‌ی کُلی باشد، مشکل‌تر گردد. روش‌های کاهش ابعاد چند متغیری نظیر آنالیز مولفه‌های اصلی[3] (PCA) (۱) یا تجزیه‌ی ماتریس نامنفی[4] (NMF) (۲) می‌توانند برای ایجاد نمایش‌های کم‌ابعاد پروفایل‌های بیان به کار گرفته شوند. این روش‌ها برخی از خصوصیات مجموعه‌ی داده‌های کامل را حفظ کرده و بنابراین اغلب برای مصورسازی مفید بوده و گاهی اوقات نیز به عنوان مرحله‌ی پیش‌پردازش در آنالیز به کار گرفته می‌شوند.

آنالیز DE داده‌های توالی‌یابی RNA که در آن داده‌های مشاهده شده در قالب شمارش‌های گسسته از یک فرآیند نمونه‌گیری به دست می‌آیند، با آنالیز DE داده‌های ریزآرایه که در آن داده‌ها شامل اندازه‌گیری‌های پیوسته از یک سیگنال فلورسنت هستند، متفاوت است. یک جنبه از این موضوع آن است که چون توالی‌یابی RNA یک فرآیند نمونه‌گیری است، لذا حجم مشخصی از نمونه (تعداد کُل همه‌ی خوانش‌های حاصل از دستگاه توالی‌یابی) وجود دارد که رونوشت‌های واقعی در کتابخانه توالی‌یابی باید در آن سهم داشته باشند. این بدان معناست که رونوشت‌های با بیان بالا، اغلب حجم بزرگی از کتابخانه‌ی توالی‌یابی را تشکیل داده و در آزمایشات توالی‌یابی کم عمق،

1- Differential Expression Analysis (DE)
2- Univariate
3- Principal Component Analysis (PCA)
4- Nonnegative Matrix Factorization (NMF)

ژن‌های با بیان کمتر، حتی اگر در نمونه حضور داشته باشند، نمی‌توانند در داده‌های نهایی نمایش داده شوند. ولی در مقابل، ریزآرایه‌ها علی‌رغم داشتن محدودیت‌های دیگر، با چنین محدودیتی مواجه نیستند. از طرف دیگر، یک ویژگی جالب توالی‌یابی RNA این است که می‌توان یک کتابخانه را برای بهبود بالقوه‌ی رونوشت‌های با بیان بیشتر، مجدداً توالی‌یابی نمود.

۸–۲ تکرارهای تکنیکی در برابر تکرارهای زیستی

لحظه‌ای به موضوع «تکرار» و چگونگی کمک کردن آن به آنالیز افتراقی بیان فکر کنید. تکرار از نظر لغوی می‌تواند چندین معنی داشته باشد. ولی معنای مورد نظر در اینجا عبارت است از: بیش از یک بار اندازه‌گیری کردن کمیت مورد نظر. به عنوان مثال، اگر بخواهید بیان مختص بافت را در مگس سرکه اندازه‌گیری کرده و تنها یک نمونه از mRNA استخراج شده از یک غده‌ی بزاقی و یک نمونه از mRNA استخراج شده از نخاع را با هم مقایسه کنید، این آزمایش فاقد تکرار محسوب می‌گردد.

تکرار یکی از سه رکن اصلی یک طرح آزمایشی مناسب که توسط Fisher (۱۹۳۵) تعریف گردیدند، است. این ارکان عبارتند از: تصادفی کردن، تکرار و بلوک‌بندی. یک تعریف دقیق از این مفاهیم در توالی‌یابی RNA را می‌توان در مقاله‌ی Auer و Doerge یافت (۳). مطالعه‌ی این مقاله پیش از اقدام به طراحی آزمایش عمیقاً توصیه می‌گردد.

هدف از تکرار آن است که برآورد تنوع بین و داخل گروه‌ها که به عنوان مثال در آزمون‌های فرضیه اهمیت دارند، امکان‌پذیر گردد. تکرار تکنیکی برای برآورد تنوع تکنیک سنجش و اندازه‌گیری (نظیر توالی‌یابی RNA) به کار گرفته می‌شود. از تکرار زیستی برای یافتن تنوع در داخل یک گروه زیستی بهره گرفته می‌شود. به بیان ساده‌تر، یک تغییر مشاهده شده در بیان ژن بین دو گروه را تنها زمانی می‌توان معنی‌دار تلقی نمود که تفاوت بین گروه‌ها، در مقایسه با تنوع داخل گروه، با در نظر گرفتن اندازه‌ی نمونه، بزرگ باشد.

انواع مختلفی از تکرارهای فنی، نظیر توالی‌یابی یک کتابخانه در دو چاهک مختلف یک دستگاه توالی‌یاب یا روش‌های آماده‌سازی مختلف کتابخانه روی یک نمونه‌ی RNA استخراج شده، می‌تواند وجود داشته باشد. معمولاً استخراج RNA در تکرارهای تکنیکی مشابه بوده ولی در تکرارهای زیستی متفاوت است. همچنین مواردی نیز وجود دارند که حتی اگر تکرارها از منابعی که بتوان آنها را متفاوت تلقی نمود، گرفته شده باشند، باز هم به سختی می‌توان آنها را تکرار زیستی نامید. به عنوان مثال می‌توان به کشت‌های مختلف رده‌های سلولی همگن[1] که از نظر ژنتیکی یکسان

1- Homogeneous

هستند، اشاره کرد. در این موارد آنچه که اهمیت دارد این است که به جای تاکید بر واژه‌ها و کلمات، به اینکه یک مقایسه‌ی افتراقی بیان به صورت واقعی به چه پرسش‌هایی پاسخ می‌دهد، اندیشیده شود.

چه تعداد تکرار بایستی در نظر گرفته شود؟ این امر به ویژگی‌های آزمایش بستگی دارد. همگنی زیستی نمونه‌های مختلف، هدف آزمایش و سطح مطلوب توان آماری[1] بر تعداد تکرارهای مورد نیاز تاثیر می‌گذارند. شما می‌توانید از یک ابزار محاسبه توان برای توالی‌یابی RNA ، نظیر Scotty (http://euler.bc.edu/marthlab/scotty/scotty.php) جهت تعیین تعداد تکرارها استفاده نمایید.

اکثر ابزارها و امکانات توالی‌یابی نیاز دارند یا توصیه می‌کنند که حداقل از سه یا چهار تکرار به ازای هر گروهی که در مقایسه وارد می‌شوند، استفاده شود. تقریباً در تمامی موارد، دو تکرار بسیار کم محسوب می‌گردد. در استفاده از سه تکرار این خطر وجود دارد که حداقل یک نمونه در تهیه‌ی کتابخانه یا توالی‌یابی مردود گردیده و در نتیجه در یکی از گروه‌ها تنها دو تکرار موجود باشد.

زمانی که از خون و برخی از نمونه‌های بافتی انسان برای مطالعات ترانسکریپتومی مورد–شاهد[2] استفاده می‌شود، به نظر می‌رسد که تنوع قابل توجهی بین افراد دیده شود. به طور ویژه در بافت‌های پیچیده، برای مشاهده‌ی تفاوت در بیان بین موردها و شاهدها، لازم است که تعداد بسیار زیادی تکرار (شاید صدها یا هزاران تکرار) به کار گرفته شوند. برای رده‌های سلولی یا نمونه‌های حاصل از بافت‌های متمایز، تعداد اندکی تکرار لازم است.

۸–۳ توزیع‌های آماری در داده‌های توالی‌یابی RNA

همان‌گونه که در PCR کمی نیز دیده می‌شود، سطوح بیانی یک ژن در بین سلول‌های مختلف از توزیع نرمال لگاریتمی پیروی می‌کند (توجه شود که این مورد با توزیع بیان ژن‌های مختلف در یک سلول متفاوت است) (۴). با این حال، اکثر آزمایشات بیان ژن در یک جمعیت از سلول‌ها انجام شده و توزیع‌شان اندکی متفاوت است.

در آزمایشات توالی‌یابی RNA که فرض می‌شود که توالی‌ها به صورت تصادفی از کتابخانه‌ی توالی‌یابی نمونه‌گیری شده‌اند، انتظار می‌رود که شمارش خوانش‌های خام از توزیع پواسون[3] پیروی کند. ولی با قدری دقت انتظار می‌رود که حتی اگر یک کتابخانه در شرایط یکسان دوبار توالی‌یابی گردد، شمارش‌ها اندکی متفاوت به دست آیند. این تداخل که از فرآیند نمونه‌گیری حاصل شده و

1- Statistical power
2- Case-control study
3- Poisson distribution

اصطلاحاً تداخل تلاش[1] نامیده می‌شود، اجتناب‌ناپذیر بوده و اغلب تنوع بین تکرارهای تکنیکی در توالی‌یابی RNA را می‌توان به خوبی با این نوع تداخل پواسونی توجیه نمود. با این حال وقتی که نمونه‌ها از منابع زیستی متمایز (نظیر افراد مختلف) اخذ می‌گردند، تنوع بین آنها اغلب با یک توزیع دو جمله‌ای منفی[2] (گاهی اوقات توزیع گاما-پواسون[3] نیز نامیده می‌شود) مدل‌بندی می‌گردد. در واقع این توزیع یک توزیع پواسون فوق پراکنده[4] (نوعی از توزیع پواسون که واریانس بالاتری دارد) است. چون در توزیع پواسون واریانس و میانگین با هم برابر هستند، لذا واریانس توزیع دو جمله‌ای منفی را می‌توان به این صورت نوشت: $\sigma^2 = \mu + (\frac{1}{r})\mu^2$. در این رابطه، r ، یک عدد صحیح مثبت بوده که بدان معناست که واریانس همواره بزرگ‌تر از میانگین است. تعدادی از بسته‌های معمول نظیر DESeq (۵) و edgeR (۶) از توزیع دو جمله‌ای منفی به عنوان پایه‌ی مدل‌سازی شمارش‌های توالی‌یابی RNA استفاده می‌کنند.

با این حال داده‌های شمارش توالی‌یابی RNA نیز خصوصیاتی نظیر تورم صفر[5] (بخش بزرگی از مقادیر با شمارش‌های صفر) دارند که برازش آنها با یک توزیع دوجمله‌ای منفی را سخت‌تر می‌کند (۷). یک مقاله که اخیراً منتشر شده است (۷) چنین استدلال کرده است که پروفایل‌های شمارش توالی‌یابی RNA با استفاده از یک خانواده‌ی عمومی‌تر از توزیع‌ها که موسوم به خانواده‌ی پواسون–توییدی[6] است، می‌توانند مدل‌بندی شوند. نویسندگان این مقاله یک بسته‌ی نرم‌افزاری R با عنوان tweeDESeq نیز برای پیاده‌سازی این روش ارائه کرده‌اند.

بسته‌ی نرم‌افزاری limma (۸) که از قبل برای آنالیز ریزآرایه‌ها به کار گرفته می‌شده است، از روش دیگری استفاده کرده و نخست، داده‌های شمارش خام (با استفاده از تابع voom) را به مقادیر پیوسته با وزن‌های اطمینان مرتبط تبدیل کرده و سپس برای استفاده از چارچوب آماری بسط داده شده برای ریزآرایه‌ها روی این مقادیر اقدام می‌نماید. بسته‌ی نرم‌افزاری DESeq2 (۹) یک ویرایش به‌روزرسانی شده از DESeq بوده که می‌تواند تبدیل‌های مشابهی را نیز انجام دهد.

روش‌های ناپارامتری نظیر SAMSeq (۱۰) و NOISeq (۱۱) هیچ فرضی در مورد قالب توزیع در نظر نگرفته ولی ترجیحاً ژن‌ها را بر مبنای بیان رتبه‌بندی کرده و از آمار و آزمون‌های مبتنی بر این فهرست‌های رتبه‌بندی شده استفاده کرده و از جایگشت‌های تصادفی لیست‌های مزبور برای شناسایی ژن‌هایی که بیان متفاوتی دارند، بهره می‌گیرد (جدول ۸-۱).

1- Shot noise
2- Nagative binomial distribution
3- Gamma-Poisson distribution
4- Overdispersed
5- Zero inflation
6- Poison-Tweedie

جدول ۸-۱: فهرست برخی از ابزارهای نرم‌افزاری برای آنالیز افتراقی بیان

ابزار نرم‌افزاری	نوع نرم‌افزار	روش آنالیز	توضیحات
DESeq (۵)	بسته‌ی R/Bioconductor	مبتنی بر شمارش (دوجمله‌ای منفی)	محافظه‌کار تلقی می‌شود (نرخ مثبت غلط پایین)
DESeq2 (۹)	بسته‌ی R/Bioconductor	مبتنی بر شمارش (دوجمله‌ای منفی)	نویسندگان آنرا بیش از DESeq توصیه کرده‌اند. محافظه‌کاری کمتری نسبت به DESeq دارد.
edgeR (۶)	بسته‌ی R/Bioconductor	مبتنی بر شمارش (دوجمله‌ای منفی)	از اصولی مشابه DESeq استفاده می‌کند.
tweeDESeq (۷)	بسته‌ی R/Bioconductor	مبتنی بر شمارش (خانواده‌ی توزیع توییدی)	از DESeq/edgeR عمومی‌تر بوده ولی تازه است و به صورت گسترده مورد آزمون واقع نشده است.
Limma (۸)	بسته‌ی R/Bioconductor	مدل‌های خطی روی داده‌های پیوسته	اساساً برای آنالیز ریزآرایه‌ها بسط داده شده و به صورت وسیعی آزمون شده است. نیازمند پیش‌پردازش خوانش‌ها برای مقادیر پیوسته است.
SAMSeq (samr) (۱۰)	بسته‌ی R	آزمون ناپارامتری	از روش آنالیز افتراقی بیان ریزآرایه‌ی SAM تطبیق یافته است. با تکرارهای بیشتر، بهتر کار می‌کند.
NOISeq (۱۱)	بسته‌ی R/Bioconductor	آزمون ناپارامتری	—
Cuffdiff (۱۸)	ابزار خط فرمان لینوکس	کاهش همتابی ایزوفرم‌ها و آزمون‌های مبتنی بر شمارش	می‌تواند ایزوفرم‌ها و ژن‌های با بیان متفاوت را بگیرد (همچنین استفاده‌ی افتراقی از TSS و نقاط پیرایش).
BitSeq (۲۱)	ابزار خط فرمان لینوکس و بسته‌ی R	کاهش همتابی ایزوفرم‌ها در یک چارچوب بیزی	می‌تواند ایزوفرم‌های با بیان متفاوت را بگیرد. همچنین برآوردهای بیان (ژن و ایزوفرم) را محاسبه می‌نماید.
ebSeq (۲۲)	بسته‌ی R/Bioconductor	کاهش همتابی ایزوفرم‌ها در یک چارچوب بیزی	می‌تواند ایزوفرم‌های با بیان متفاوت را بگیرد. می‌تواند در یک مسیر که توسط برآورد بیان RSEM شروع شده است، به کار گرفته شود.

۸-۳-۱ تکرار زیستی، توزیع‌های شمارش و انتخاب نرم‌افزار

تعداد تکرارهای زیستی در دسترس می‌تواند بر انتخاب نرم‌افزار آنالیز افتراقی بیان تاثیر بگذارد. اگر تعداد تکرارهای زیستی کم باشد (۵ تا ۱۰ تکرار زیستی به ازای هر گروه، بسته به ویژگی مجموعه‌ی داده‌ها)، بهتر است که از روش‌های ناپارامتری که هیچ فرضی در مورد قالب توزیع آماری داده‌های مشاهده شده در نظر نمی‌گیرند، استفاده نمود. معمولاً روش‌های ناپارامتری برای موارد متداول‌تر با تکرارهای زیستی خیلی کم، از توان کمی برخوردارند. در چنین مواردی استفاده از روش‌های پارامتری که قالب توزیع معینی را بر مبنای داده‌های تجربی در نظر می‌گیرند (همان‌گونه که در فوق نیز اشاره شد، نظیر بسته‌های DESeq و edgeR که از توزیع دو جمله‌ای منفی استفاده می‌کنند یا tweeDESeq که از خانواده‌ی توزیع پواسون-توییدی بهره می‌گیرد)، بهتر به نظر می‌رسد. گزارش‌های اخیر نیز نشان می‌دهند که در چنین مواردی limma به خوبی کار می‌کند.

۸-۴ نرمال‌سازی

معمولاً داده‌های توالی‌یابی RNA قبل یا در حین آنالیز افتراقی بیان، نرمال‌سازی می‌گردند. اکثر بسته‌های نرم‌افزاری تنها خواستار شمارش‌های خام بوده و نرمال‌سازی را به صورت داخلی انجام می‌دهند. دلایل نرمال‌سازی عبارتند از:

- امکان‌پذیر ساختن مقایسات بین نمونه‌ها
- امکان‌پذیر ساختن مقایسات بین ژن‌ها
- ایجاد توزیع سطح بیان مطابق با فرض‌های به کار گرفته شده در روش‌های آماری

سنجه‌ی استاندارد RPKM (یا معادل جفت انتهایی‌اش: FPKM) در سال ۲۰۰۸ و در قالب یک مقاله معرفی گردید (۱۲) و مقایسات سطوح بیان یک ژن در بین نمونه‌های مختلف یا سطوح متفاوت بیان ژن در یک نمونه را امکان‌پذیر می‌سازد. RPKM (یا FPKM) مخفف عبارت Reads (Fragmennts) Per Kilobase per Milion (mapped) reads (تعداد خوانش (قطعه) به ازای هر کیلوباز در هر میلیون خوانش (مکان‌یابی شده)) بوده و شمارش‌های خام متناظر با طول ژن یا رونوشت و عمق توالی‌یابی را تصحیح می‌کند. در نحوه‌ی کاربرد این سنجه قدری تنوع دیده می‌شود. به عنوان مثال، مطالعات مختلف از مقادیر مختلف در مخرج کسر (نظیر: تعداد کُل خوانش‌های حاصل ازدستگاه توالی‌یاب، تعداد خوانش‌های مکان‌یابی شده با ژنوم یا ترانسکریپتوم، تعداد خوانش‌های مکان‌یابی شده با اگزون‌های معلوم) استفاده کرده‌اند. به همین ترتیب برای طول

ژن یا رونوشت برخی از محققین کُل طول رونوشت را انتخاب نموده و برخی دیگر نیز از طول موثر[1] (۱۳) یا طول قابل مکان‌یابی[2] (۱۴) استفاده کرده‌اند.

علی‌رغم اینکه R/FPKM هنوز هم متداول‌ترین سنجه‌ی مورد استفاده برای بیان حاصل از توالی‌یابی RNA است، ولی سنجه‌های دیگری نیز برای تصحیح اُریبی‌های ممکن در برخی موارد پیشنهاد شده‌اند. TPM (تعداد رونوشت در هر میلیون) (۱۵) شباهت زیادی به R/FPKM داشته ولی توزیع طول رونوشت‌ها را نیز در جمعیت RNA در نظر می‌گیرد. نویسندگان مقاله‌ای که این سنجه را پیشنهاد داده‌اند استدلال کرده‌اند که بدون این تصحیح (مثلاً در R/FPKM) و در زمان مقایسه‌ی دو مجموعه RNA با توزیع طول رونوشت متفاوت، اُریبی رُخ خواهد داد. TMM (میانگین‌های پیرایش شده‌ی مقادیر M) نیز برای تصحیح ترکیبات مختلف مجموعه‌های RNA به کار گرفته می‌شود. آزمایش فرضی زیر که از مقاله‌ی Robinson و Oshlack نقل می‌شود، این ایده را تشریح می‌کند:

> فرض کنید که می‌خواهید در یک آزمایش توالی‌یابی دو جمعیت RNA تحت عنوان A و B را با هم مقایسه نمایید. در این مثال فرض می‌شود که هر ژن که در B بیان می‌شود، در A نیز با تعداد رونوشت یکسان بیان می‌شود. ولی فرض می‌شود که نمونه‌یA حاوی مجموعه‌ای از ژن‌ها است که از نظر تعداد و بیان با هم برابر بوده اما این ژن‌ها در مجموعه‌ی B بیان نمی‌شوند. بنابراین کُل ژن‌های بیان شده در نمونه‌ی A دو برابر کُل ژن‌های بیان شده در نمونه‌ی B بوده و این بدان معناست که تولید RNA در نمونه‌ی A دو برابر تولید RNA در نمونه‌ی B است. سپس فرض شود که هر نمونه با عمق یکسان توالی‌یابی شده است. بدون هرگونه تصحیح اضافی، یک ژن که در هر دو نمونه بیان شده است، به طور متوسط نصفِ تعداد خوانش‌ها از نمونه‌ی A را خواهد داشت. زیرا در این نمونه خوانش‌ها روی اکثر ژن‌ها دوبار کشیده می‌شوند. بنابراین نرمال‌سازی صحیح، نمونه‌ی A را با یک فاکتورِ (ضریبِ) ۲ تصحیح می‌نماید.

یک تفاوت مهم بین TMM از یک طرف و TPM و R/FPKM از طرف دیگر در این است که TMM یک روش نرمال‌سازی دسته‌ای است. این بدان معناست که این روش برای استفاده روی تک نمونه طراحی نشده ولی روی گروهی از نمونه‌ها قابل استفاده است. بنابراین در هر بار تغییر در مجموعه‌ی نمونه‌ها، فاکتورهای تصحیح حاصل از نرمال‌سازی TMM بایستی مجدداً محاسبه گردند. ولی نرمال‌سازی‌های TPM و R/FPKM برای هر نمونه موضعی بوده و تحت تاثیر سایر نمونه‌ها

1- Effective length
2- Mappable length

واقع نمی‌شوند. تفاوت دیگر این است که TMM طول رونوشت را در نظر نمی‌گیرد. با این حال، اگر آنالیز افتراقی بیان استاندارد انجام می‌دهید، این موضوع مشکل‌ساز نخواهد بود. زیرا رونوشت‌های مختلف را با یکدیگر مقایسه کرده ولی یک رونوشت در شرایط مختلف مقایسه گردیده و در نتیجه طول رونوشت همواره یکسان خواهد بود.

روش‌های مختلف نرمال‌سازی توالی‌یابی RNA توسط Dillies و همکاران مرور شده‌اند (۱۷).

نحوه‌ی انتخاب یک بسته‌ی نرم‌افزاری برای آنالیز افتراقی بیان

در اینجا یک درخت تصمیم‌گیری ساده ارائه شده است که می‌توانید بسته به نیازتان، از آن برای انتخاب یک بسته‌ی نرم‌افزاری استفاده کنید.

نوع ترکیبی را که می‌خواهید در آنالیز افتراقی بیان مورد آزمون قرار دهید، انتخاب کنید.

اگزون‌ها به صورت متفاوت بیان شده‌اند ⇐ DEXSeq

ایزوفرم‌ها به صورت متفاوت بیان شده‌اند ⇐ BitSeq ، Cuffdiff یا ebSeq

ژن‌ها به صورت متفاوت بیان شده‌اند ⇐ **نوع طرح آزمایشی را انتخاب نمایید**

طرح مرکب (بیش از یک عامل متغیر) ⇐ DESeq ، edgeR ، limma

طرح ساده (مقایسه‌ی گروه‌ها) ⇐ **چند تکرار زیستی وجود دارد؟**

بیش از ۵ تکرار زیستی در هر گروه ⇐ SAMSeq

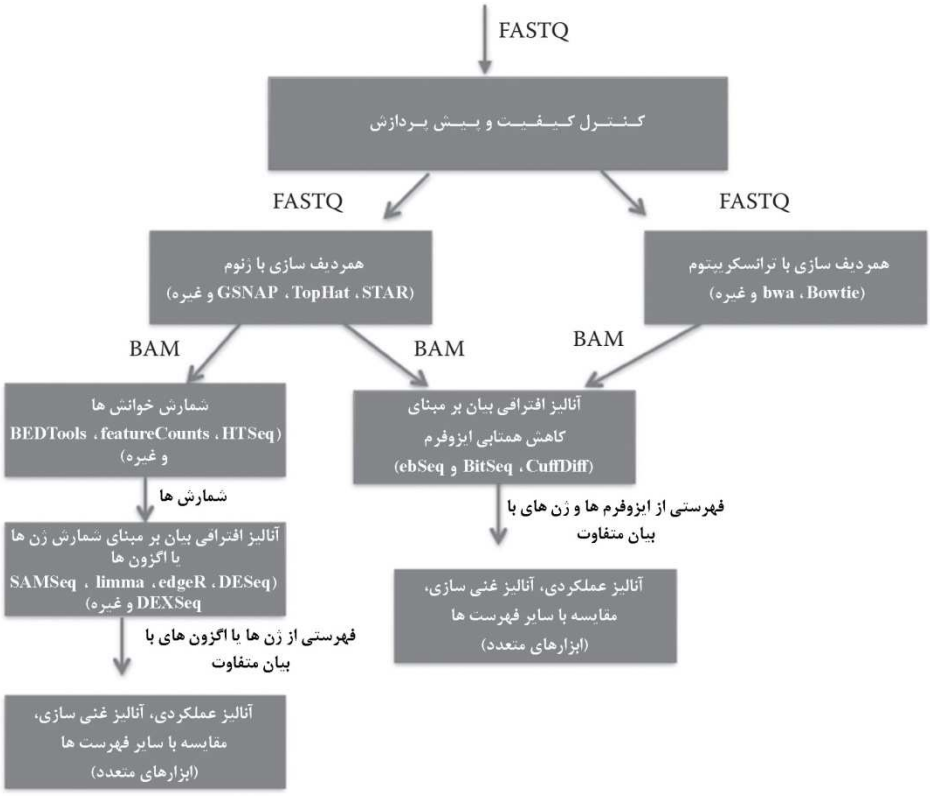
کمتر از ۵ تکرار زیستی در هر گروه ⇐ DESeq ، edgeR ، limma

۸-۵ مثال‌هایی از کاربرد نرم‌افزارها

در اینجا مثال‌هایی از نحوه‌ی استفاده از دو برنامه‌ی متداول برای آنالیز افتراقی بیان (Cuffdiff و DESeq) ارائه می‌گردد. هر کدام از این برنامه‌ها یکی از گردش کارهای معمول در آنالیز افتراقی بیان را نمایندگی می‌کند (نگاره‌ی ۸-۱).

۸-۵-۱ استفاده از Cuffdiff

برنامه‌ی Cuffdiff بخشی از بسته‌ی نرم‌افزاری متداول Cufflinks برای اسمبل نمودن، کمی‌سازی و آنالیز افتراقی بیان توالی‌یابی RNA است. این برنامه می‌تواند به طور همزمان تفاوت بیان روی سطوح ژن و رونوشت را ارزیابی کند. نویسندگان این برنامه این‌گونه استدلال کرده‌اند که

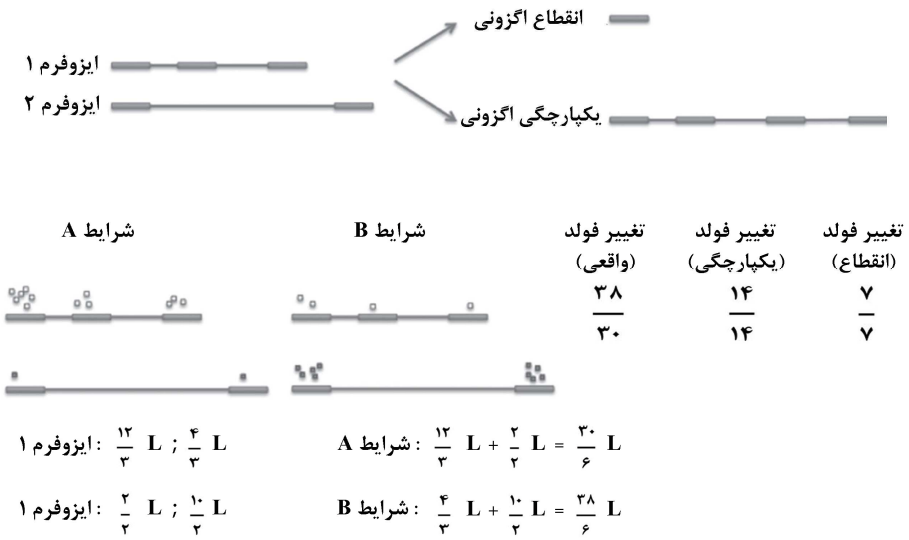


نگاره‌ی ۸-۱: گردش کارهای مورد استفاده در آنالیز افتراقی بیان توالی‌یابی RNA

Cuffdiff در مقایسه با برنامه‌های معمول دیگر نظیر DESeq و edgeR بهتر عمل می‌کند. زیرا این نرم‌افزار کاهش همتایی[1] داده‌های بیان در ایزوفرم‌های دارای بیان متفاوت را تلفیق می‌نماید (۱۸). در مقابل، معمولاً DESeq و edgeR ژنی را که از یک بسته‌ی نرم‌افزاری دیگر نظیر HTSeq ، BEDTools یا featureCounts حاصل آمده است، در نظر می‌گیرند و در نتیجه ایزوفرم‌ها بررسی نمی‌شوند که این امر در مواردی که بیش از یک ایزوفرم بیان شده باشد، منجر به اُریبی در محاسبات افتراقی بیان روی سطح کُل ژن می‌گردد (نگاره‌ی ۸-۲).

با این حال یک مطالعه‌ی مقایسه‌ای راجع به نرم‌افزارهای DE در توالی‌یابی RNA (۱۹) نشان داده است که در عمل این مزیت Cuffdiff الزاماً منجر به نتایج بهتر نمی‌شود. مزیت دیگر Cuffdiff

1- Deconvolution



نگاره‌ی ۸-۲: لازم است که ایزوفرم‌ها جهت به دست آوردن برآوردهای ناأریب بیان در سطح ژن مورد بررسی واقع شوند. یک ژن با دو ایزوفرم مختلف در نظر گرفته شود (گوشه‌ی چپ بالا). برای ساده‌سازی فرض می‌شود که همه‌ی اگزون‌ها طول یکسانی (L) دارند. دو روش معمول برای محاسبه‌ی شمارش‌ها در سطح ژن عبارتند از: مدل انقطاع اگزونی (در این روش تنها خوانش‌های مکان‌یابی شده با اگزون‌ها که بخشی از کلیه‌ی ایزوفرم‌های ژنی هستند، در نظر گرفته می‌شوند) و مدل یکپارچگی اگزونی (در این روش کلیه‌ی خوانش‌های مکان‌یابی شده با هر اگزون در نظر گرفته می‌شوند) (گوشه‌ی راست بالا). در مورد فرضی که در بخش پایینی نشان داده شده است، خوانش‌های مربوط به هر ایزوفرم در دو شرایط مختلف A و B نشان داده شده است. تغییر فولد واقعی کُل ژن با در نظر گرفتن ایزوفرم‌ها به صورت $\frac{۳۸}{۳۰}$ برآورد شده ولی این مقدار در هر دو مدل انقطاع اگزونی و یکپارچگی اگزونی برابر با ۱ (که به معنای هیچ‌گونه تغییر می‌باشد) برآورد شده است.

این است که خروجی‌اش می‌تواند به عنوان ورودی برای یکی از مفیدترین بسته‌های مصورسازی و آنالیز R تحت عنوان CummeRbund مورد استفاده قرار گیرد (۲۰). از طرف دیگر، برخلاف DESeq ، edgeR و limma ، برنامه‌ی Cuffdiff از طرح‌های آزمایشی پیچیده‌تر پشتیبانی نمی‌کند. در این زمینه لازم به ذکر است که BitSeq (۲۱) و ebSeq (۲۲) در مزایا و معایب با Cuffdiff مشترک هستند.

Pros و Cons در Cuffdiff

Pros

- سطوح بیان ایزوفرم‌ها را محاسبه می‌کند.
- تفاوت بیان ژن‌ها، ایزوفرم‌ها، مناطق پیرایش و محل‌های آغاز رونویسی را آزمون می‌کند.
- از پشتیبانی مصورسازی خوبی برخوردار است.

Cons

- از طرح‌های فاکتوریل پشتیبانی نمی‌کند (تنها می‌تواند دو گروه را با هم مقایسه نماید).

اگر بخواهید Cuffdiff را روی داده‌های مثال اجرا کنید، باید بدانید که این نرم‌افزار فایل‌های همردیفی SAM/BAM را به عنوان ورودی قبول کرده و نیازی به فایل حاشیه‌نگاری رونوشت در فرمت GTF برای تعریف مجموعه‌ای از ترکیبات ژنومی که مورد بررسی قرار خواهند گرفت، ندارد.

در دستور زیر، تنها از برخی از گزینه‌های متعدد Cuffdiff استفاده شده است (۲). برای مشاهده‌ی کلیه‌ی دستورات مزبور می‌توانید برنامه را بدون وارد کردن برهان‌ها اجرا کنید. گزینه‌ی `-o` دایرکتوری خروجی را تخصیص می‌دهد که معمولاً حاوی فایل‌های زیادی بوده و در صورت تمایل کاربر این ساختار به صورت یک مجموعه‌ی کُلی و برای مصورسازی و آنالیز در cummeRbund وارد می‌گردد. گزینه‌ی `-p 4` مشخص می‌کند که می‌خواهید از چهار پرداشگر استفاده کنید. گزینه‌ی `-L` فهرستی از برچسب‌های مورد استفاده برای شرایط (در مثال مورد نظر، انواع سلول (hESC و GM12892) مبنا قرار گرفته است) را ارائه می‌کند. گزینه‌ی `--FDR` نرخ یافته‌های غلط در آنالیز DE را ارائه کرده و گزینه‌ی `-u` نیز مشخص می‌کند که می‌خواهید از تصحیح چندخوانشی استفاده کنید. برهان‌های ورودی اجباری شامل فایل‌های حاشیه‌نگاری GTF (توجه شود که در این فایل باید از نام‌های کروموزوم یا کانتیگ مشابه آنچه که در فایل‌های BAM/SAM استفاده شد، بهره گرفته شود) و فهرستی از نام فایل‌های BAM برای همان گروه که توسط کاما از یکدیگر جداشده‌اند، هستند. برای نشان دادن فایل‌های BAM مربوط به هر گروه از یک کاراکتر فاصله (space) استفاده می‌شود. وجود کاما بین نام فایل‌ها در هر گروه نیز نشان دهنده آن است که فایل‌های مزبور تکرار محسوب می‌شوند. فرض شود که $CUFFPATH یک متغیر محیطی حاوی مسیر دایرکتوری است که فایل اجرایی Cuffdiff در آن واقع شده و $GTFPATH نیز دایرکتوری دیگری است که حاوی یک فایل GTF می‌باشد. بدین ترتیب، دستور مورد نظر به صورت زیر خواهد بود:

```
$CUFFPATH/cuffdiff -o chr18_hESC_vs_GM12892 -p 4 -L
hESC,GM12892 --FDR 0.01 -u $GTFPATH/Homo_sapiens.
GRCh37.59.chr-added.gtf hESC1_chr18.bam,hESC2_chr18.
bam,hESC3_chr18.bam,hESC4_chr18.bam Gm12892_1_chr18.
bam,Gm12892_2_chr18.bam,Gm12892_3_chr18.bam
```

ممکن است چند هشدار در مورد طول قطعات دریافت کنید که نادیده گرفته خواهند شد. پس از اتمام این دستور، می‌توانید فایل‌های حاوی اطلاعاتِ تفاوتِ بیان را در فولدری که برای فایل‌های خروجی تخصیص یافته است، بیابید. فایل gene _ exp.diff حاوی اطلاعات سطح ژن است. سه لینک ابتدایی آن به صورت زیر می‌باشد:

test_id	gene_id	gene	locus	sample_1	sample_2	status	value_1	value_2	log2(fold_change)	test_stat	p_value	q_value	significant
ENSG00000000003	ENSG00000000003	TSPAN6	chrX:99883666-99894988	hESC	GM12892	NOTEST	0	0	0	0	1	1	no
ENSG00000000005	ENSG00000000005	TNMD	chrX:99839798-99854882	hESC	GM12892	NOTEST	0	0	0	0	1	1	no

آنچه که در این خروجی دیده می‌شود، چندان جالب نیست. زیرا این چند ژن ابتدایی (ژن‌ها همواره بر مبنای ستون gene_id مرتب می‌شوند) روی کروموزوم ۱۸ واقع نشده و از طرف دیگر چون هیچ‌گونه داده‌ای خارج از کروموزوم ۱۸ در فایل‌های BAM وجود نداشته است، لذا در اینجا Cuffdiff حتی تفاوت در بیان را نیز بررسی نکرده و در نتیجه عبارت NOTEST را در ستون هفتم (status) درج نموده است.

با ملاحظه‌ی ردیف‌هایی که برای آنها در ستون آخر عبارت yes درج گردیده است می‌توان دریافت که ۲۳۷ ژن روی کروموزوم ۱۸ به عنوان ژن‌های دارای بیان متفاوت خوانده شده‌اند (این بدان معناست که این ژن‌ها در بین دو گروه مزبور به صورت معنی‌داری متفاوت بیان شده‌اند). یک راه برای یافتن این ژن‌ها استفاده از دستور awk در لینوکس است:

```
awk'$14=="yes"' gene_exp.diff|wc-l
```

تعدادی از خطوط متناظر با چنین ژن‌هایی که در ستون آخرشان عبارت yes درج شده است، در زیر نمایش داده شده‌اند:

ENSG00000017797	ENSG00000017797	RALBP1	chr18:9475006-9538112	hESC	GM12892	OK	0	5453.82	inf	nan	5e-05	6.07477e-05	yes
ENSG00000039139	ENSG00000039139	DNAH5	chr5:13690439-13944652	hESC	GM12892	OK	207.984	0	-inf	nan	5e-05	6.07477e-05	yes
ENSG00000040731	ENSG00000040731	CDH10	chr5:24487208-24645087	hESC	GM12892	OK	238.193	0	-inf	nan	5e-05	6.07477e-05	yes

شش ستون ابتدایی نشان دهنده‌ی ID های ژن، نام‌های متداول، مختصات کروموزومی و برچسب‌های مربوط به گروه‌بندی نمونه‌ها است. در اینجا ستون هفتم (status) حاوی عبارت OK است که بدان معناست که Cuffdiff داده‌های کافی برای اجرای یک آزمون معنی‌داری برای ژن مورد نظر داشته است. ستون‌های بعدی (هشتم و نهم) به ترتیب حاوی میانگین FPKM در هر گروه و hESC یا GM12892 است. ستون دهم مقدار لگاریتم در مبنای دو برای تغییر فولد بین میانگین FPKM های گروه‌ها بوده و چون در این مثال میانگین مزبور در یکی از این گروه‌ها برابر با صفر بوده است، لذا لگاریتم تغییر فولد نیز در اینجا بی نهایت (inf) شده است. ستون یازدهم حاوی مقادیر یک تست آماری است. در این ستون در مثال مزبور عبارت nan (مخفف شده‌ی

not a number) درج شده است که با بی‌نهایت شدن مقدار لگاریتم تغییر فولد مرتبط است. ستون دوازدهم مقدار p برآورد شده را نشان داده و ستون سیزدهم نیز مقدار q برآورد شده (مقدار p تصحیح شده برای مقایسات چندگانه) را نشان می‌دهد. معمولاً ستون‌های مقدار q و لگاریتم در مبنای دو برای تغییر فولد، مهم‌ترین و جالب‌ترین ستون‌ها هستند.

همچنین فایل `isoform_exp.diff` حاوی اطلاعات مشابه در مورد سطح ایزوفرم است. در اینجا تعداد ایزوفرم‌های دارای بیان متفاوت برابر با ۲۳۸ است که به طور نامعمولی به مقادیر سطح ژن نزدیک می‌باشد.

۸-۵-۲ استفاده از بسته‌های Bioconductor : DESeq ، edgeR ، limma

در پروژه‌های R مربوط به Bioconductor ، بسته‌های نرم‌افزاری متعددی برای آنالیز افتراقی بیان وجود دارد. در اینجا عمدتاً روی DESeq2 تمرکز می‌شود. ولی بسته‌های edgeR و limma نیز مورد بحث واقع می‌شوند. زیرا این بسته‌ها نیز عمومیت داشته و در حال حاضر نیز بهترین پشتیبانی را از طرح‌های فاکتوریل می‌کنند. سایر بسته‌های نرم‌افزاری مفید عبارت از بسته‌ی ناپارامتری NOISeq و بسته‌های مبتنی بر روش‌های بیزی شامل BitSeq ، baySeq و ebSeq و سرانجام بسته‌ی tweeDESq هستند.

۸-۵-۳ مدل‌های خطی، ماتریس طرح و ماتریس مقایسه

بسته‌های نرم‌افزاری DESeq2 ، edgeR و limma همگی مبتنی بر مفهوم مدل خطی تعمیم یافته هستند (در واقع limma مخفف شده‌ی linear models for microarray data است). آشنایی و معرفی مناسب مفهوم مدل‌های خطی فراتر از موضوع این کتاب است. لذا از خوانندگان درخواست می‌گردد که یک کتاب مرجع آمار را مطالعه نمایند. ایده‌ی مبنایی عبارت از مدل‌بندی بیان هر ژن به صورت یک ترکیب خطی از چند متغیر تشریحی مختلف (یا عامل) است. به عنوان مثال، اگر یک آزمایش شامل بیماران، تیمارها و نقاط زمانی مختلف باشد، مدل خطی برای هر ژن می‌تواند به صورت زیر تعریف گردد:

$$y = a + b \times \text{treatment} + c \times \text{time} + d \times \text{patient} + e$$

در این رابطه، y ، بیان ژن اندازه‌گیری شده بر حسب واحد، e ، خطا و a ، b ، c و d پارامترهای برآورد شده از داده‌ها هستند. a ، عرض از مبداء بوده و نشان دهنده‌ی میانگین سطح بیان ژن در

زمانی است که کلیه‌ی فاکتورهای دیگر (تیمار، زمان و بیمار) در وضعیت مبنایی باشند (می‌توان ترکیبی از این عوامل را به عنوان وضعیت مبنایی در نظر گرفت. این ترکیب، اختیاری است).

مدل خطی تعمیم یافته[1] (GLM) نسخه‌ی انعطاف‌پذیرتری از مدل خطی استاندارد است که در مقایسه با سایر مدل‌های خطی، این امکان را فراهم می‌آورد که توزیع متغیر پاسخ، متفاوت از توزیع نرمال به کار گرفته شده در رگرسیون خطی استاندارد باشد. GLM ها در edgeR و DESeq با این فرض مورد استفاده قرار گرفته‌اند که توزیع شمارش خوانش‌ها بر مبنای توزیع دو جمله‌ای منفی بوده است.

ویژگی دیگر DESeq ، edgeR و limma این است که این بسته‌ها می‌توانند از اطلاعات سایر ژن‌ها نیز برای افزایش توان آماری استفاده کنند. این بسته‌ها از طرح‌های مختلف برای دستیابی به برآوردهای واریانس تعدیل شده که در آن هر واریانس ژنی به صورت یک ترکیب وزنی از واریانس خود ژن که از داده‌های اختصاصی ژن و میانگین واریانس کلیه‌ی ژن‌ها یا مجموعه‌ای از ژن‌ها مدل‌بندی می‌شود، بهره می‌برند.

مدل خطی را به صورت عمومی‌تر در قالب نمایش ماتریسی می‌توان به صورت زیر نوشت:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

در این رابطه، $\mathbf{y}$ ، سطح بیان و $\boldsymbol{\varepsilon}$ ،خطا است. $\boldsymbol{\beta}$ ، برداری از پارامترهای برآورد شده از داده‌ها بوده و $\mathbf{X}$ ، که ماتریس طرح[2] نامیده می‌شود، عوامل آزمایشی را تشریح می‌نماید.

ماتریس طرح همراه با ماتریس مقایسه[3] دو مفهومی هستند که بایستی قبل از کار با DESeq ، edgeR یا limma با آنها آشنا شوید (علی‌رغم اینکه ممکن است که مجبور نباشید مستقیماً با آنها کار کنید. به عنوان مثال، در DESeq به صورت ضمنی و غیرمستقیم با این ماتریس‌ها سر و کار دارید). معمولاً این مفاهیم در طرح‌های آزمایشی به کار گرفته شده و توصیه می‌شود که آنها را از یک کتاب مرجع آمار مطالعه نمایید. راهنمای کاربری limma (۲۳) نیز حاوی مثال‌های مفیدی از چگونگی تنظیم ماتریس‌های آزمایشی و طرح است.

۸-۵-۳-۱ ماتریس طرح

در مباحث زیر راجع به ماتریس طرح، به یک شیء R که طرح آزمایش را توصیف می‌کند، ارجاع داده می‌شود (البته همان‌گونه که در فوق نیز اشاره شد، بحث در مورد این ماتریس بسیار

1- Generalized Linear Model (GLM)
2- Design matrix
3- Contrast matrix

عمومی‌تر است). به عنوان مثال، اگر داده‌های یک آزمایش که در آن نمونه‌های بافت تومور و سالم از بیماران گرفته شده باشد، را آنالیز نمایید، می‌توانید جدولی داشته باشید (expTable) که این آزمایش را توصیف نماید:

```
expTable <- data.frame(Individual=c("Patient1","Pati
  ent1", "Patient2","Patient2"),Status=c("Tumor","Heal
  thy", "Tumor","Healthy"),row.
  names=paste0("sample",1:4))

expTable
          Individual     Status
sample1     Patient1      Tumor
sample2     Patient1    Healthy
sample3     Patient2      Tumor
sample4     Patient2    Healthy
```

با کمک تابع model.matrix() می‌توان این جدول را به یک ماتریس طرح تبدیل نمود:

```
design.matrix <- model.matrix (~Individual+Status,
  data = expTable)
```

ماتریس طرح ممکن است که قدری پیچیده به نظر برسد. ولی پس از چند بار تمرین، تفسیر آن سخت نخواهد بود.

```
>design.matrix
        (Intercept) IndividualPatient2      StatusTumor
sample1           1                   0                1
sample2           1                   0                0
sample3           1                   1                1
sample4           1                   1                0
attr(,"assign")
[1]0 1 2
attr(,"contrasts")
attr(,"contrasts")$Individual
[1]"contr.treatment"

attr(,"contrasts")$Status
[1]"contr.treatment"
```

در اینجا سه ستون دیده می‌شود که عبارتند از: (Intercept) ، IndividualPatient2 و StatusTumor . این ستون‌ها حاوی متغیرهای شاخص[1] هستند. اگر یک عامل در یک نمونه دارای

1- Indicator variable

یک مقدار معین باشد، این متغیر عدد ۱ دریافت کرده و در غیر اینصورت عدد صفر دریافت خواهد نمود. ستون intercept حاوی مقدار یکسان ۱ برای همه‌ی نمونه‌ها بوده و به سادگی نشان می‌دهد که مدل خطی که برای هر ژن تشکیل خواهد شد، یک عرض از مبداء خواهد داشت. این عرض از مبداء متناظر با میانگین سطح بیان در شرایطی است که کلیه‌ی عوامل آزمایشی در وضعیت مبنایی‌شان باشند (که در این مورد عبارتند از: (Individual = Patient1) و (Status = Normal)). ستون دوم در ماتریس طرح که با IndividualPatient2 مشخص شده است، نشان دهنده‌ی نمونه‌هایی است که از بیمار دوم (Patient 2) گرفته شده‌اند (یعنی مقدار عامل Individual برابر با Patient2 قرار داده شده است). روی ردیف‌های مربوط به این بیمار، عدد ۱ درج شده و در سایر ردیف‌ها عدد صفر قرار داده شده است. در ستون سوم که با StatusTumor مشخص شده است، برای نمونه‌هایی که عامل Status در آنها دارای مقدار Tumor هستند، عدد ۱ درج گردیده است.

۸-۵-۳-۲ ماتریس مقایسه

وقتی که می‌خواهید تفاوت در بیان را مورد آزمون قرار دهید، ممکن است که لازم باشد که یک ماتریس مقایسه تشکیل دهید تا از این طریق نشان دهید که می‌خواهید کدام مقایسه‌ها را انجام دهید (این کار در DESeq ضرورتی ندارد). این ماتریس اغلب فقط یک عنصر غیرصفر دارد (البته اگر بخواهید فقط یک مقایسه انجام دهید). به عنوان مثال با ماتریس طرح ارائه شده در فوق، می‌توانید با استفاده از تابع makeContrasts در بسته‌ی نرم‌افزاری limma یک مقایسه بین بافت تومور و بافت سالم انجام دهید:

```
contrast.matrix <- makeContrasts(StatusTumor,
  levels=design.matrix)
```

اگر بخواهید بیمار ۱ (Patient1) را با بیمار ۲ (Patient2) مقایسه کنید، می‌توانید مقایسه‌ای به شرح زیر تعریف نمایید:

```
contrast.matrix <- makeContrasts(IndividualPatient2,
levels=design.matrix)
```

چون مدل دارای یک عرض از مبداء است، هیچ ستونی برای (Individual = Patient 1) وجود ندارد. این بدان معناست که این وضعیت به عنوان مقدار مبنایی عامل Individual در نظر گرفته می‌شود. بنابراین IndividualPatient2 به طور ضمنی تفاوت بین بیمار ۲ (Patient 2) و بیمار ۱ (Patient 1) را نشان می‌دهد.

به جای این کار می‌توان با استفاده از 0~ در تابع model.matrix ، عرض از مبداء را از مدل حذف نمود. با این کار ماتریس طرح ستون (Intercept) را نخواهد داشت:

```
design.matrix <- model.matrix (~0+Individual+Status,
  data = expTable)
design.matrix
 IndividualPatient1     IndividualPatient2     StatusTumor
sample1            1                      0               1
sample2            1                      0               0
sample3            0                      1               1
sample4            0                      1               0
attr(,"assign")
[1]1 1 2
attr(,"contrasts")
attr(,"contrasts")$Individual
[1]"contr.treatment"

attr(,"contrasts")$Status
[1]"contr.treatment"
```

همان‌گونه که در فوق نیز تشریح گردید، در اینجا هیچ وضعیت مبنایی وجود نداشته و برای انجام مقایسه بین بیمار ۲ (Patient 2) و بیمار ۱ (Patient 1) لازم است که دستور زیر اجرا گردد:

```
contrast.matrix <- makeContrasts(IndividualPatient2-
  IndividualPatient1,levels = design.matrix)
```

برای کسب اطلاعات بیشتر راجع به ماتریس‌های طرح و مقایسه، به راهنمای نرم‌افزارهای limma یا edgeR یا یک کتاب مرجع آمار مراجعه نمایید.

۸-۵-۴ آماده‌سازی‌های پیش از آنالیز افتراقی بیان

معمولاً آنالیز با لود کردن یک جدول شمارش آغاز می‌شود. برای انجام این کار با کمک داده‌های مثال، در صورتی که یک جدول شمارش در دسترس نباشد، ابتدا نحوه‌ی ایجاد جدول مزبور از فایل‌های BAM یا فایل‌های شمارش جداگانه توضیح داده می‌شود.

۸-۵-۴-۱ شروع از فایل‌های BAM

اگر کار را با فایل‌های همردیف شده شروع می‌کنید، لازم است که شمارش‌ها را با استفاده از یک برنامه نظیر HTSeq ، BEDTools یا featureCounts (که می‌تواند از طریق بسته‌ی Rsubread

در R نیز استفاده شود) به دست آورید. برای استفاده از HTSeq و FeatureCounts لازم است که ابتدا فایل‌های باینری BAM به فایل‌های SAM تبدیل شود (BEDTools می‌تواند فایل‌های BAM را مستقیماً مدیریت نماید). این کار می‌تواند با استفاده از مجموعه‌ی samtools از ابزارهای commandline صورت گیرد.

```
samtools sort –no Gm12892_1_chr18.bam Gm12892_1_chr18_
sorted |samtools view – > Gm12892_1_chr18.sam
```

از گزینه‌ی n- مرتب‌سازی samtools استفاده می‌شود تا از این طرق اطمینان حاصل شود که فایل SAM بر اساس مشخصه‌ی خوانش مرتب می‌گردد. زیرا HTSeq برای خوانش‌های جفت انتهایی نیازمند این مرتب‌سازی است. البته این مشکل در آخرین ویرایش HTSeq رفع شده است.

دستورات مشابه برای سایر نمونه‌ها نیز به کار گرفته می‌شوند. سپس می‌توان HTSeq را به صورت یک ماژول پایتون اجرا نمود:

```
python –m HTSeq.scripts.count –s no Gm12892_1_chr18.
sam Homo_sapiens.GRCh37.70.chr18.chr.gtf > gm1.txt
```

(و دستورات مشابه برای سایر نمونه‌ها به کار گرفته می‌شود). با این کار، فایل‌های شمارش برای هر نمونه ایجاد می‌گردد.

۸-۵-۴-۲ شروع از فایل‌های شمارش جداگانه

این نوع فایل‌ها دراین کتاب شامل فایل‌های شمارش حاصل از HTSeq هستند که تحت عنوان gm1.txt ، gm2.txt ، ... ، h1.txt ، ... ، h2.txt ، نامیده می‌شوند. راه‌های متعددی برای تلفیق این فایل‌ها در قالب یک جدول وجود دارد که از آن جمله می‌توان به دستور join در یونیکس و اسکریپت‌های سفارشی مختلف نظیر دستور DESeqDataSetFromHTSeqCount در DESeq2 اشاره نمود. یکی از راه‌ها نیز استفاده از دستورات زیر در R است:

```
samples <- c(paste0("gm",1:3),paste0("h",1:4))
first.sample <- read.delim(paste0(samples[1],".
   txt"),header=F,row.names=1)
count.table <- data.frame(first.sample)
for(s in samples[2:length(samples)]){
      fname <- paste0(s,".txt")
      column <- read.delim(fname,header=F,row.names=1)
      count.table <- cbind(count.table,s = column)
}
```

```
colnames(count.table) <- samples
write.table(count.table,file = "count_table_chr18.
  txt",sep="\t",quote=F)
```

۸-۵-۴-۳ *شروع از یک جدول شمارش موجود*

اگر قبلاً یک جدول شمارش تهیه کرده‌اید، می‌توانید آنرا به سادگی و با استفاده از دستور زیر لود کنید:

```
d.raw <- read.delim("count_table_chr18.txt",sep = "\t")
```

این دستور یک فایل با جدا کننده‌ی تَب[1] لود خواهد نمود. اگر شما یک فایل با جدا کننده‌ی فاصله[2] در اختیار دارید، به جای `sep = "\t"` از `sep = ""` استفاده نمایید. برای سایر جدا کننده‌ها نیز می‌توانید به همین ترتیب تغییر لازم را اعمال کنید.

۸-۵-۴-۴ *پاکسازی مستقل*

معمولاً توصیه می‌شود که پیش از آنالیز افتراقی بیان مبتنی بر شمارش، رونوشت‌های با بیان پایین پاکسازی شوند (۲۴). لازم است که تعداد شمارش‌ها در بیش از دو نمونه، بالاتر از ۳ باشد. البته این آستانه‌های قطع کم و بیش اختیاری هستند. در DESeq2 این مرحله به صورت خودکار و با آستانه‌ی قطع محاسبه شده توسط نرم‌افزار انجام می‌شود.

```
d <- d.raw[rowSums(d.raw>3) > 2,]
```

با اجرای این دستور، ۳۶۵ ژن (از مجموع ۷۷۴ ژن) برای آنالیزهای بیشتر جدا می‌شود.

۸-۵-۵ مثالی از کُدنویسی برای DESeq2

در اینجا نحوه‌ی کُدنویسی برای DESeq ارائه می‌شود. علاوه بر این مثال‌هایی از خلاصه کُدهای مورد نیاز برای سایر بسته‌های نرم‌افزاری متداول نیز ارائه می‌گردد. برای توضیح بیشتر در مورد ویژگی‌های هر بسته می‌توان به راهنمای مربوط به آن در R مراجعه کرد. این راهنماها اغلب حاوی اطلاعات مفیدی از کاربرد و طراحی نرم‌افزار هستند.

1- Tab separated
2- Space separated

بسته‌ی DESeq را لود کرده، گروه‌های مورد نظر را تعریف کرده و یک چارچوب داده برای اطلاعات گروه تهیه نمایید.

```
library(DESeq2)
grp <- c(rep("GM",3),rep("hES",4))
cData <- data.frame(celltype = as.factor(grp))
rownames(cData) <- colnames(d)
```

سپس از تابع `DESeqDataSetFromMatrix` برای ایجاد شیء DESeqDataSet استفاده کنید. برای آگاهی از سایر روش‌های ایجاد این شیء به راهنمای مرجع DESeq2 مراجعه نمایید.

```
d.deseq <- DESeqDataSetFromMatrix(countData = d,
  colData = cData,design = ~celltype)
```

تابع بعدی، چندین مرحله‌ی متمایز آنالیز را اجرا می‌کند. در بسته‌ی DESeq این مراحل در قالب دستورات جداگانه تعریف شده بودند. ولی در DESeq2 جهت راحتی کار در قالب یک دستور واحد تجمیع شده‌اند.

```
d.deseq <- DESeq(d.deseq)
```

با کمک دستور `results()` می‌توان نتایج را استخراج نمود. ستون نتایج را می‌توان تحت عنوان celltype_hES_vs_GM نامگذاری کرده و فهرست نتایج موجود را نیز با کمک تابع `resultsNames()` فراخوان نمود.

```
res <- results(d.deseq,"celltype_hES_vs_GM")
```

اگر بخواهید روی ژن‌هایی که مقدار p تصحیح شده‌ی آنها کمتر از ۰/۰۱ است تمرکز کنید، می‌توانید از دستور زیر استفاده نمایید:

```
sig <- res[which(res$padj < 0.01),]
```

تابع زیر نیز تنها فهرستی از نام‌های ژن‌های مزبور را ارائه می‌دهد:

```
sig.deseq <- rownames(sig)
```

۸-۵-۶ مصورسازی

می‌توان با روش‌های مختلف داده‌ها را به صورت تصویری نمایش داد. یک راه برای کنترل داده‌های پرت و نمونه‌های مخلوط شده، ترسیم یک نقشه‌ی حرارتی همبستگی یا نمودار مختصات

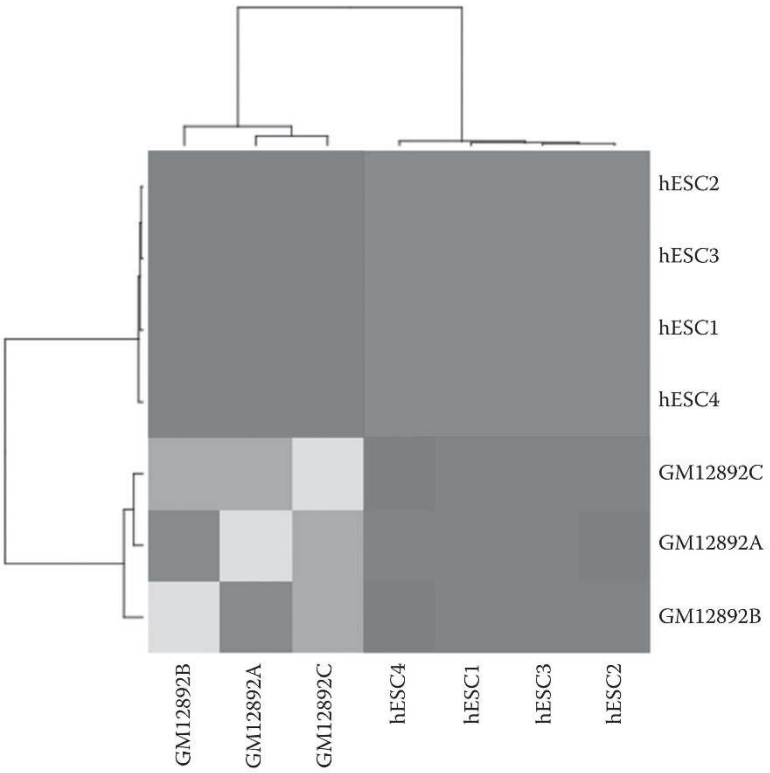
اصلی (PCA) نمونه‌ها است. البته این کار را می‌توان پیش از آنالیز افتراقی بیان نیز انجام داد. DESeq یک تابع مناسب برای تبدیل داده‌ها در قالبی که برای مصورسازی نقشه‌ی حرارتی و PCA مناسب‌تر است، ارائه می‌دهد:

```
vsd <- getVarianceStabilizedData(d.deseq)
```

حال می‌توان همبستگی بین نمونه‌ها را نیز به دست آورد:

```
heatmap(cor(vsd),cexCol=0.75,cexRow=0.75)
```

از برهان‌های `cexCol` و `cexRow` برای ایجاد برچسب‌های نمونه که به اندازه‌ی کافی کوچک بوده و در نمودار جای می‌گیرند، استفاده می‌شود (نگاره‌ی ۸-۳).



نگاره‌ی ۸-۳: نقشه‌ی حرارتی همبستگی نمونه‌های رده‌های سلولی. یک گروه‌بندی مشخص در بین نمونه‌ها وجود داشته و آنها را به دو گروه متمایز تقسیم نموده است.

با کمک دستور prcomp() در R می‌توان یک نمودار PCA نیز ترسیم کرد (نگاره‌ی ۸-۴). برای این کار لازم است که ترانهاده‌ی ماتریس بیان به دست آید تا امتیازات مختصات اصلی به جای ژن‌ها برای نمونه‌ها در نظر گرفته شود:

```
pr <- prcomp(t(vsd))
plot(pr$x,col="white",main="PC plot",
xlim=c(-22,15))
text(pr$x[,1],pr$x[,2],labels=colnames(vsd),
cex=0.7)
```

همچنین می‌توان از تابع biplot برای به دست آوردن یک نمودار دوتایی[1] PCA مشابه که شامل اطلاعاتی در مورد نحوه‌ی مشارکت هر ژن در مولفه‌های اصلی است، استفاده نمود:

```
biplot(pr,cex=c(1,0.5),main="Biplot",
col=c("black","grey"))
```

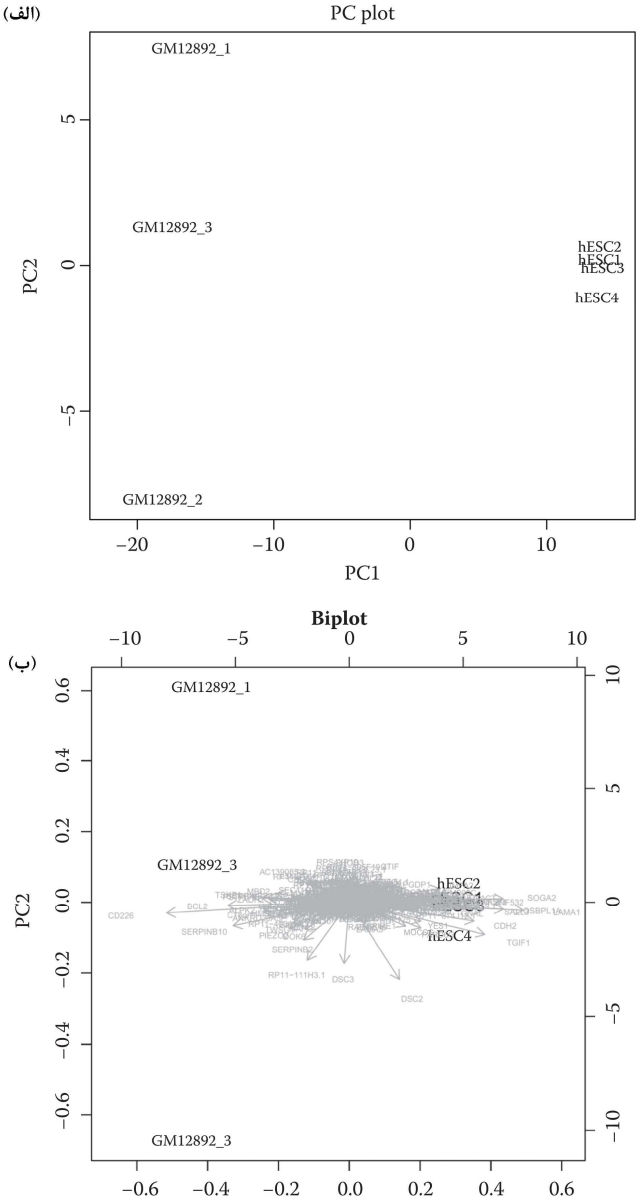
پس از تکمیل آنالیز افتراقی بیان، نمودارهای مختلفی می‌توانند مفید واقع شوند. در یک نمودار آتشفشانی[2] (لگاریتم منفیِ) مقدار p در برابر تغییر فولد هر ژن ترسیم شده و اغلب نیز منجر به ایجاد شکلی مشابه یک آتشفشان در حال فوران می‌شود. ژن‌هایی که مقدار p تصحیح شده‌ی آنها کمتر از ۰/۰۱ باشد، با رنگ قرمز مشخص می‌گردد.

```
plot(res$log2FoldChange,-log(res$padj),pch=15)
points(sig$log2FoldChange,-log(sig$padj),
  col="grey",pch=15)
library("calibrate")# if not installed,run 'install.
  packages("calibrate")'
textxy(sig$log2FoldChange,-log(sig$padj),rownames(sig),
  cex=0.9)
```

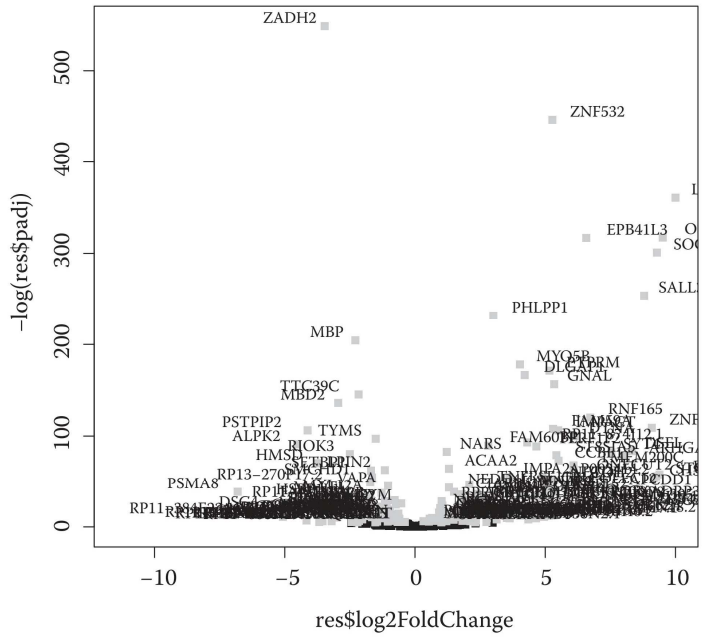
در مثال حاضر به نظر می‌رسد که تقریباً کلیه‌ی ژن‌ها بیان متفاوت دارند (در واقع تقریباً نیمی از ژن‌ها چنین هستند. ولی اکثر ژن‌هایی که به صورت متفاوت بیان نشده‌اند، در کنار یکدیگر در اطراف مبداء فشرده گردیده‌اند). این نتیجه چندان جالب نیست. زیرا رده‌های سلولی مزبور، بسیار متفاوت هستند (نگاره‌ی ۸-۵).

بررسی تصویری تک ژن‌های جداگانه نیز می‌تواند مفید واقع گردد. در اغلب مواقع نمودارهای میـله‌ای[3] یا ستونی[4] برای این کار مناسـب هستند. ۱۰ ژن دارای بیشتـرین تفاوت در بیان را بـرای

1- Biplot
2- Volcano plot
3- Bar plot
4- Box plot



نگاره‌ی ۸-۴: الف) نمودار مختصات اصلی نمونه‌های رده‌های سلولی. در طول مولفه‌ی اول (محور X) گروه‌بندی نمونه‌ها در قالب دو گروه متمایز مشهود است. ب) یک نمودار دوتایی موقعیت نسبی نمونه‌ها در فضای PC1-PC2 و میزان مشارکت ژن‌های مختلف در مختصات اصلی را نشان می‌دهد.



نگاره‌ی ۸-۵: یک نمودار آتشفشانی که منفیِ لگاریتم مقدار p در برابر لگاریتم تغییر فولد برای هر ژن را نشان می‌دهد. ژن‌های با مقدار p کمتر از ۰/۰۱ با رنگ قرمز مشخص شده‌اند.

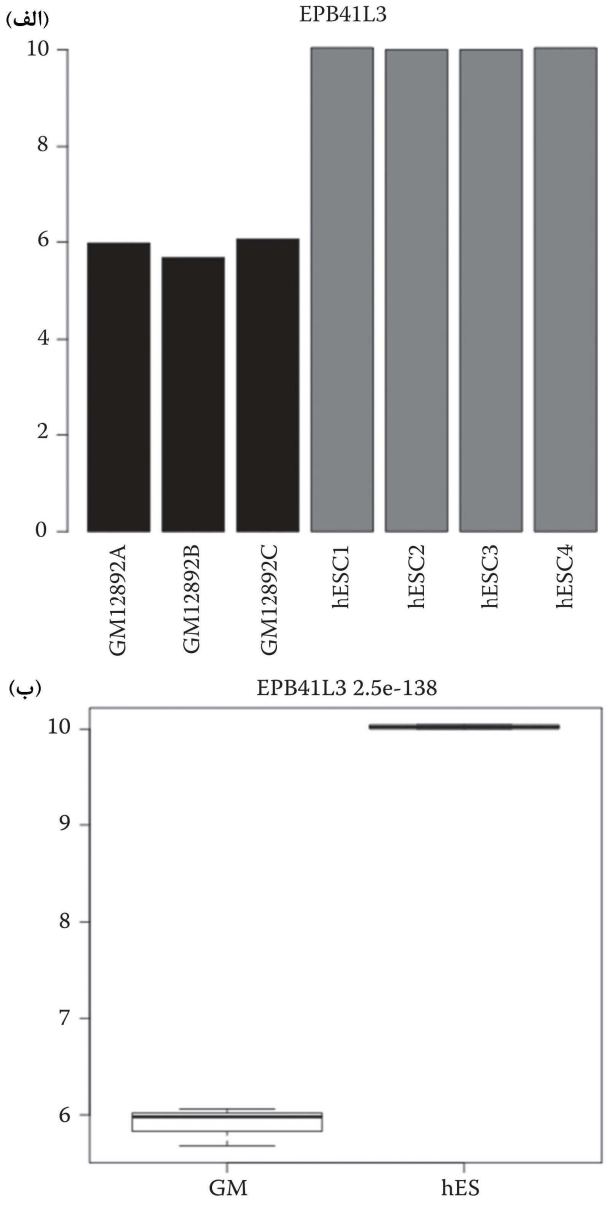
ترسیم نمودار ستونی در نظر بگیرد. نخست، ژن‌ها را بر مبنای مقدار p تصحیح شده مرتب نمایید:

```
sig.ordered <- sig[order(sig$padj),]
for(gene in head(rownames(sig.ordered))) {
boxplot(vsd[gene,which(grp=="GM")],vsd[gene,
which(grp=="hES")],main=paste (gene,signif(sig
[gene,"padj"],2)),names=c("GM","hES"))
readline()
}
```

یا برای ترسیم نمودار میله‌ای:

```
for(gene in head(rownames(sig.ordered))) {
barplot(vsd[gene,],las=2,col=as.numeric(as.
factor(grp)),main=gene,cex.names=0.9)
readline()
}
```

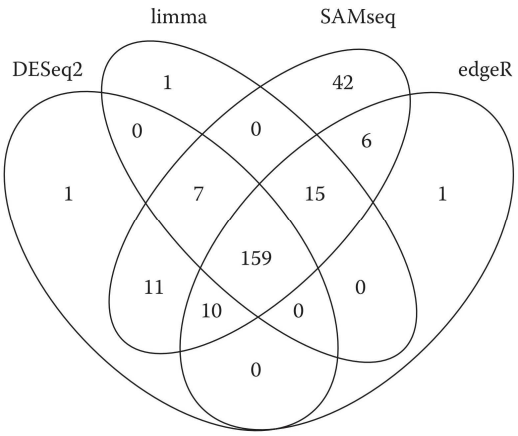
نمودارهای ترسیم شده برای یک ژن منفرد در نگاره‌ی ۸-۶ نمایش داده شده است.



نگاره‌ی ۸-۶: الف) نمودار میله‌ای که سطح بیانِ (بر حسب واحد شمارش نرمال شده) یک ژن خاص در نمونه‌های GM و hES را نشان می‌دهد. ب) نمودار ستونی که توزیع بیان همان ژن در داخل هر گروه (GM یا hES) را نمایش می‌دهد. خط سیاه پُررنگ نشان دهنده‌ی میانه است.

۸-۵-۷ مثال‌هایی از کُدنویسی برای سایر بسته‌های Bioconductor

در زیر مثال‌هایی از کُدنویسی جهت انجام آنالیز افتراقی بیان با استفاده از بسته‌های limma ، edgeR و sam rBioConductor ارائه شده است. میزان هماهنگی و توافق بین روش‌های مختلف در مثال ساده‌ی ارائه شده (سلول‌های GM در برابر سلول‌های hES) در نگاره‌ی ۸-۷ خلاصه گردیده است. با اطمینان می‌توان گفت که هر چهار برنامه تشخیص داده‌اند که ۱۵۹ ژن بیان متفاوتی دارند (توجه شود که این امر به نسخه‌ی نرم‌افزار مورد استفاده بستگی دارد). هر کدام از نرم‌افزارهای DESeq ، limma و edgeR تنها یک ژن را شناسایی کرده‌اند که توسط سایر نرم‌افزارها شناسایی نشده‌اند. ولی SAMSeq تعداد ۴۲ عدد از چنین ژن‌هایی را شناسایی نموده است. بدون داشتن اطلاعات اضافی نمی‌توان قضاوت نمود که آیا این موضوع ناشی از حساسیت بالاتر SAMSeq می‌باشد یا از نرخ بالاتر شناسایی غلط در این نرم‌افزار ناشی می‌شود. تنها می‌توان گفت که در این مثال، به نظر می‌رسد که SAMSeq از معیارهای انعطاف‌پذیرتری برای آزمون افتراقی بیان استفاده می‌کند. سطح هماهنگی و توافق به دست آمده در اینجا بین روش‌های مختلف، معمولاً برای کلیه‌ی آزمایشات صدق نکرده و این موضوع احتمالاً ناشی از آن است که رده‌های سلولی مقایسه شده، از نظر زیستی تفاوت فاحشی دارند. برای گروه‌های نمونه‌ی مشابه‌تر (نظیر مطالعات مورد – شاهد بیماران)، مشاهده‌ی عدم توافق و هماهنگی بین روش‌های مختلف آنالیز افتراقی بیان امری متداول و مرسوم است.



نگاره‌ی ۸-۷: نمودار وِن چهار مسیری برای نشان دادن هماهنگی و توافق بین آنالیز افتراقی بیان اجرا شده توسط DESeq2 ، limma ، SAMSeq و edgeR. هر چهار روش تشخیص داده‌اند که تعداد ۱۵۹ ژن بیان متفاوتی دارند.

limma ۸-۵-۸

```
library(limma)
grp <- c("GM","GM","GM","hES","hES","hES","hES")
des <- model.matrix(~0+grp)
colnames(des) <- c("GM","hES")
contrast.matrix <- makeContrasts(GM-hES,levels=des)

d.norm <- voom(d,design=des)

fit <- lmFit(d.norm,des)
fit2 <- contrasts.fit(fit,contrast.matrix)
fit2 <- eBayes(fit2)

topTable(fit2,adjust="BH")

all <- topTable(fit2,adjust.method="BH",number=10000)
sig <- all[all$adj.P.Val < 1e-2,]

sig.limma <- sig$ID # If this doesn't work, try sig.
limma <- rownames(sig)
# If you want to compare the consistency of the limma
and DESeq results:
intersect(sig.limma,sig.deseq)
```

SAMSeq (بسته‌ی smar) ۹-۵-۸

SAMSeq با سایر بسته‌ها تفاوت دارد. این بدان معناست که برای یک مقایسه‌ی مشابه، نتایج متفاوتی ارائه می‌دهد. زیرا این نرم‌افزار بر مبنای آزمون‌های جایگشت که در آنها زیرنمونه‌هایی از مجموعه‌ی داده‌ها بیرون کشیده می‌شوند، عمل می‌کند. این کار می‌تواند برای تخصیص واضح یک سید تصادفی (آنچنان‌که در نمونه‌ی کُد نیز انجام شده است) جهت بازتولید نتایج مفید واقع گردد.

```
library(samr)

# Note that samr is (as of Jan 2014)not in
BioConductor, so it needs to be installed using
install.packages(). However,it depends on the impute
package which is a BioConductor package.

num.grp <- c(rep(1,3),rep(2,4))
samfit <- SAMseq(d,num.grp,resp.type="Twoclass
  unpaired", genenames=rownames(d),random.
  seed=101010,fdr.output=0.01,nperms=1000)

sig.sam <- c(samfit$siggenes.table$genes.up[,1],
samfit$siggenes.table$genes.lo[,1])
```

۸-۵-۱۰ edgeR

```
library(edgeR)
edgeR.dgelist = DGEList(counts = d,group = factor(grp))
edgeR.dgelist = calcNormFactors(edgeR.
  dgelist,method = "TMM")
edgeR.dgelist = estimateCommonDisp(edgeR.dgelist)
edgeR.dgelist = estimateTagwiseDisp(edgeR.
  dgelist,trend = "movingave")
edgeR.test = exactTest(edgeR.dgelist)
edgeR.pvalues = edgeR.test$table$PValue
edgeR.adjpvalues = p.adjust(edgeR.pvalues,method = "BH")

sig.table <- edgeR.test$table[which(edgeR.
adjpvalues < 0.01),]
sig.edgeR <- rownames(sig.table)
```

۸-۵-۱۱ مثالی از کُدنویسی DESeq2 برای یک آزمایش چند عاملی

برای مثال ارائه شده از یک آزمایش چند عاملی، روی آنالیز افتراقی بیان تمرکز گردیده و توضیحی در مورد آماده‌سازی داده‌ها ارائه نگردیده و وقتی برای این کار صرف نمی‌شود. در عوض بسته‌ی parathyroid که حاوی شمارش خوانش‌ها روی سطوح ژن و اگزون برای یک آزمایش که در آن چهار آدنوکارسینومای پاراتیروئید کشت داده شده و با دو داروی مختلف تیمار شده‌اند (و یک کنترل تیمار نشده) و در دو نقطه‌ی زمانی مختلف نمونه‌گیری گردیده‌اند، نصب و بارگذاری می‌شود. از ۲۴ نمونه (۴ بیمار × ۳ تیمار × ۲ زمان) توالی‌یابی RNA صورت گرفته است. یکی از نمونه‌ها به دلیل مشکلات موجود در تهیه‌ی کتابخانه، حذف شده است.

```
source("http://bioconductor.org/biocLite.R")
biocLite("parathyroid")
library(parathyroid)
# You may need to install the bitops and/or DEXSeq
packages for this to work.
```

سامانه‌ی راهنما (عبارت ??parathyroid را تایپ کنید) به شما می‌گوید که می‌توانید یک شیء حاوی اطلاعات سطح ژن را با کمک دستور زیر لود کنید:

```
data("parathyroidGenes")
```

می‌توان با استفاده از pData() اطلاعاتی در مورد نمونه‌ها کسب کرد:

```
meta <- pData(parathyroidGenes)
head(meta)
```

	size	Factor	experiment	patient	treat-ment	time	submis-sion	study	sample	run
SRR479052		NA	SRX140503	1	Control	24h	SRA051611	SRP012167	SRS308865	SRR479052
SRR479053		NA	SRX140504	1	Control	48h	SRA051611	SRP012167	SRS308866	SRR479053
SRR479054		NA	SRX140505	1	DPN	24h	SRA051611	SRP012167	SRS308867	SRR479054
SRR479055		NA	SRX140506	1	DPN	48h	SRA051611	SRP012167	SRS308868	SRR479055
SRR479056		NA	SRX140507	1	OHT	24h	SRA051611	SRP012167	SRS308869	SRR479056
SRR479057		NA	SRX140508	1	OHT	48h	SRA051611	SRP012167	SRS308870	SRR479057

دستور `counts` جدول شمارش را بازمی‌گرداند:

```
dim(counts(parathyroidGenes))
```

اگر با توجه به اطلاعات نمونه‌ها و به دقت در این جدول نگریسته شود، می‌توان دریافت که برای برخی از نمونه‌ها دو ورودی وجود دارد (نظیر ردیف‌های ۹ و ۱۰). این ردیف‌ها متناظر با دورهای مختلف توالی‌یابی برای یک نمونه‌ی واحد هستند. این تداخل را می‌توان از چند طریق مدیریت نمود. یک راه این است که به سادگی کلیه‌ی شمارش‌های متعلق به آن نمونه، افزوده گردند. ولی این کار سبب می‌شود که اُریبی بالقوه‌ی حاصل از دسته‌های مختلف توالی‌یابی پوشانیده شود. به جای آن می‌توان ورودی‌های جداگانه را نگه داشت. ولی در عوض یک پارامتر برای اجرای توالی‌یابی در مدل خطی گنجاند. بدین ترتیب در این روش می‌توان حداقل در برخی از زمینه‌ها نسبت به تصحیح اثر دسته اقدام نمود.

```
meta$run <- c(rep(1,9),2,1,1,2,rep(1,10),2,1,1,2)
```

فرض کنید که می‌خواهید ژن‌های دارای بیان متفاوت بین تومورهای تیمار شده با DPN و تومورهای تیمار نشده را پیدا کنید. در اینصورت چه باید کرد؟ توجه کنید که چهار چیزِ متفاوت دارید: بیمار، زمان، تیمار و اجرای توالی‌یابی. ابتدا جدولی که آزمایش را توصیف کرده و نیز جدول خوانش‌ها را تهیه کنید:

```
countData <- counts(parathyroidGenes)
```

همان‌گونه که قبلاً نیز تشریح شد، ژن‌های با بیان پایین پاکسازی می‌شوند. در اینجا لازم است که تعداد کُل شمارش‌ها حداقل با تعداد نمونه‌ها برابر باشد (بنابراین به طور متوسط، هر نمونه باید حداقل یک خوانش داشته باشد).

```
countData <- countData[rowSums (countData) >= ncol
(countData),]
```

با در نظر گرفتن بیمار، اجرا، زمان و تیمار به عنوان کوواریت، شیء DESeqDataSet را بسازید:

```
dds <- DESeqDataSetFromMatrix(countData = countData,
              colData = meta,
              design = ~patient + run + time + treatment)
```

حال می‌توان در شیء dds و با کمک counts(dds) به جدول شمارش و با استفاده از design(dds) به فرمول طرح دسترسی یافت. پیش از اجرای دستور DESeq ، فاکتورهای اندازه برای نرمال‌سازی را برآورد نمایید (اجرای این مرحله در همه‌ی نسخه‌های DESeq الزامی نیست):

```
dds <- estimateSizeFactors(dds)
```

دستور DESeq را همانند قبل اجرا کنید. با این کار، یک GLM ایجاد شده، مدل‌ها برازش یافته و آزمون‌ها برای مقایسات مختلف اجرا می‌گردند. اگر در تابع results() هیچ چیزی تخصیص نداده باشید، مقایسه‌ی پیش‌فرض برای تفاوت‌های تیمارها خواهد بود (چون تیمار به عنوان آخرین فاکتور در فرمول طرح درج شده است):

```
dds <- DESeq(dds)
```

برای یافتن مقایساتی که نتایج‌شان در دسترس است، از resultsNames() استفاده می‌شود:

```
resultsNames(dds)
```

برای به دست آوردن نتایج و انتخاب ژن‌های با بیان متفاوت برای DNP در مقایسه با شاهد، از دستورات زیر استفاده می‌شود:

```
res <- results(dds, contrast=c("treatment","DPN",
"Control")) # In older versions of DESeq2, you may
instead need to write:
res <- results(dds,"treatment_DPN_vs_Control")
sig.deseq <- res[which(res$padj<0.01),]
sig.deseq.names <- rownames(sig.deseq)
```

اگر بخواهید یک مقایسه‌ی کُلی بین زمان ۲۴ ساعت و ۴۸ ساعت داشته باشید (شامل کلیه‌ی تیمارها و بیماران)، می‌توانید از `results(dds,"time_48h_vs_24 h")` استفاده نمایید. برای سایر عوامل نیز می‌توان این کار را انجام داد. ولی اگر بخواهید مقایسات پیچیده‌تر یا تخصصی‌تر انجام دهید، چه باید بکنید؟ به عنوان مثال فرض شود که می‌خواهید تفاوت بین DPN و شاهد از یک طرف و تفاوت بین OHT و شاهد از طرف دیگر را مقایسه نمایید. این بدان معناست که یکی از این تفاوت‌ها اطلاعاتی در مورد اثرات تیمار DPN در اختیار گذاشته و چنین اطلاعاتی در تیمار OHT دیده نمی‌شود. برای انجام این کار، می‌توان (DPN در برابر کنترل) – (OHT در برابر کنترل) را با استفاده از برهان `contrast` در تابع `results` کُدنویسی کرد (توجه شود که این قابلیت تنها در نسخه‌های 1.1.24 و بالاتر نرم‌افزار DESeq2 وجود دارد). چون ورودی هفتم و هشتم این بردار که توسط `resultsNames()` بازگردادنده می‌شود، به ترتیب treatment_DPN_vs_Control و treatment_OHT_vs_Control است، لذا می‌توان نوشت:

```
res <- results(dds,contrast=c(0,0,0,0,0,0,1,-1))
```

۸-۵-۱۲ مثالی از کُدنویسی edgeR

```
library(edgeR)
d <- DGEList(counts=counts(parathyroidGenes))
d <- d[rowSums(d$counts)>=ncol(counts(parathyroidGe
  nes)),]
d <- calcNormFactors(d,method="TMM")

# For checking if the normalization worked.This plots
the "count-per-million"distributions for each sample
boxplot(cpm(d,normalized.lib.sizes=T),outline=F,las=2)

meta <- pData(parathyroidGenes)
# Replace "run" by original run(1)or rerun (2)
eta$run <- c(rep(1,9),2,1,1,2,rep(1,10),2,1,1,1)
design <- model.matrix(~treatment+patient+run+time,
  data=meta)
```

```
d <- estimateGLMCommonDisp(d,design)
d <- estimateGLMTrendedDisp(d,design)
d <- estimateGLMTagwiseDisp(d,design)
fit <- glmFit(d,design)
lrt <- glmLRT(fit,coef=2)
# The value of coef above depends on the comparison
you are interested in.You can check available
comparisons with colnames(coef(fit)).
temp <- topTags(lrt,n=100000)$table
sig.edger <- temp[temp$FDR < 0.01,]
sig.edger.names <- rownames(sig.edger)

# To compare(DPN vs control)vs(OHT vs control)instead:
lrt <- glmLRT(fit,contrast=c(0,1,-1,0,0,0,0,0)) #
Corresponding to columns 2 and 3 in fit$design
```

۸-۵-۱۳ مثالی از کُدنویسی limma

```
library(limma)
countData <- counts(parathyroidGenes)
meta <- pData(parathyroidGenes)
# Replace "run" byoriginal run(1)or rerun(2)
meta$run <- c(rep(1,9),2,1,1,2,rep(1,10),2,1,1,1)
nf = calcNormFactors(countData,method="TMM")
voom.data <- voom(countData,design=model.matrix(~patie
  nt+run+time+treatment,data=meta),lib.
  size=colSums(countData)*nf)
voom.data$genes<- rownames(countData)
design = model.matrix(~patient+run+treatment+time,
  data=meta)
voom.fitlimma <- lmFit(voom.data,design)
voom.fitbayes <- ebayes(voom.fitlimma)
voom.pvalues <- voom.fitbayes$p.value[,"treatmentDPN"]
voom.adjpvalues <- p.adjust(voom.pvalues,method="BH")
sig.limma.names <- names(which(voom.adjpvalues<0.01))
# To compare (DPN vs control) vs (OHT vs control)
instead:
contrast.matrix <- makeContrasts(treatmentDPN
  treatmentOHT,levels=design)
voom.fitlimma2 <- contrasts.fit(voom.fitlimma,
  contrast.matrix)
voom.fitbayes <- ebayes(voom.fitlimma2)
voom.pvalues <- voom.fitbayes$p.value
voom.adjpvalues <- p.adjust(voom.pvalues,method="BH")
sig.limma.names <- rownames(voom.pvalues)[(which(voom.
  adjpvalues<0.01))]
```

آنالیز افتراقی بیان در Chipster

آنالیز افتراقی بیان را می‌توانید در Chipster و با کمک Cuffdiff ، DESeq2 و edgeR انجام دهید. Cuffdiff فایل‌های BAM را به عنوان ورودی دریافت کرده و می‌توان فایل GTF را نیز به صورت اختیاری به آن عرضه نمود. در اینجا روی ابزارهای DESeq و edgeR که به یک جدول شمارش به عنوان ورودی نیاز دارند، تمرکز می‌شود:

- خوانش‌های هر ژن یا رونوشت به ترتیب با کمک HTSeq یا eXpress شمارش شده و همان‌گونه که در فصل ششم تشریح گردید، فایل‌های شمارش برای کلیه‌ی نمونه‌ها با استفاده از ابزار Utilities/Defne NGS experiment در قالب یک جدول تلفیق می‌شوند. علاوه بر جدول شمارش، این ابزار یک فایل فنودیتا نیز تولید می‌کند که این امکان را برای شما فراهم می‌سازد که نحوه‌ی اجرا و تنظیمات آزمایش را تشریح کنید. با استفاده از ویرایشگر فنودیتا، کلیه‌ی نمونه‌های متعلق به هر گروه را که با شماره‌ی یکسان در ستون گروه مشخص شده‌اند، انتخاب کنید. اگر آزمایش شما پیچیده‌تر است، به ازای هر عامل یک ستون جدید اضافه نمایید.
- جدول شمارش و یکی از ابزارهای DESeq یا edgeR را انتخاب نمایید. آستانه‌ی معنی‌داری مورد نظر را تنظیم کرده و روی Run کلیک کنید. فهرستی از ژن‌های دارای بیان متفاوت هم به صورت یک جدول (که با کلیک کردن روی سرستون می‌توان آنرا مرتب نمود) و هم به صورت یک فایل BED ارائه می‌گردد. فایل BED این امکان را فراهم می‌آورد که طبق روش تشریح شده در فصل چهارم، در مرورگر ژنومی Chipster جستجو نمایید. فایل‌های نتایج شامل یک نمودار پراکندگی و یک نمودار MDS که شباهت‌های نسبی بین نمونه‌ها را نشان می‌دهد، است.
- اگر از ابزار Differential expression with edgeR for multivariate experiments استفاده می‌کنید، برای تنظیم صحیح پارامترها از صفحات راهنما کمک بگیرید. توجه کنید که جدول نتایج شامل کلیه‌ی ژن‌ها بوده و شما می‌توانید آنها را بر مبنای هر ستون که مورد نظرتان باشد و با استفاده از ابزار Filtering/Filter table a column value پاکسازی نمایید.

۸-۶ خلاصه

روش‌های آنالیز افتراقی بیان برای توالی‌یابی RNA هنوز در حال بسط و گسترش بوده و هنوز

مجموعه‌ای از بهترین‌ها گردآوری نشده‌اند. پایگاه‌های داده‌ی فعالی درباره‌ی بهترین راه‌های نرمال‌سازی، مزایای آنالیز در سطح ژن در مقایسه با آنالیز در سطح ایزوفرم و واحد اندازه‌گیری برای استفاده در گزارش‌دهی سطوح بیان ژن وجود دارند.

بنابراین کدام روش آنالیز DE بایستی انتخاب شود؟ اگر شما بخواهید آنالیز افتراقی بیان را برای ایزوفرم‌ها انجام دهید و از جنبه‌ی نظری نیز به این نتیجه برسید که این روش بر آنالیز افتراقی بیان در سطح ژن برتری دارد (چون تغییرات در سطح ژن به طور دقیقی با تغییرات در سطح ایزوفرم مرتبط هستند)، آنگاه روش‌های مبتنی بر ایزوفرم (نظیر BitSeq ، Cuffdiff و ebSeq) انتخاب مناسبی خواهند بود. ولی مطالعات موجود نشان داده‌اند که تمایل به سوی استفاده از بسته‌های نرم‌افزاری متداول R/Bioconductor نظیر DESeq ، edgeR و limma که از داده‌های ساده‌ی شمارش خوانش‌های مکان‌یابی شده با ژن‌ها استفاده می‌کنند، است. این روش‌های خاص نیز این مزیت را دارند که می‌توانند با استفاده از یک چارچوب مدل‌بندی خطی تعمیم یافته، بیش از یک عامل آزمایشی متغیر (کوواریت) را در نظر بگیرند. در آزمایش‌های دارای تکرارهای زیستی متعدد به ازای هر گروه، استفاده از روش‌های ناپارامتری نظیر SAMSeq و NOISeq نیز می‌تواند مفید و ارزشمند باشد. این روش‌ها از مشکلات مدل‌بندی توزیع شمارش خوانش‌ها به دور بوده ولی در عوض نیازمند تکرارهای بیشتر برای دستیابی به توان آماری هستند.

از جنبه‌ی آسانی کار، بسته‌های R/Bioconductor همگی مشابه هم بوده ونیازمند سطح پایه‌ای از مهارت و تخصص برای اجرا شدن در زبان R هستند. راهنمای مرجع و عیب‌یابی در قالب فایل‌های PDF برای DESeq ، limma و edgeR وجود داشته و حاوی اطلاعات ارزشمندی برای کاربران مبتدی هستند. لازم به ذکر است که اجرای یک آنالیز چند عاملی اندکی پیچیده‌تر بوده ولی نتایج آن ارزشمندتر خواهد بود. Cuffdiff یک نرم‌افزار تحت لینوکس یا مَک بوده و بنابراین لازم است که کاربر با نحوه‌ی کار کردن با خط فرمان آشنایی داشته باشد. اجرای این آنالیز یک فرآیند تک مرحله‌ای است. البته اجرای یک آنالیز می‌تواند ساعت‌ها یا حتی روزها زمان بگیرد و به طور بالقوه نیازمند حجم بسیار بالایی از حافظه (RAM) است. معمولاً روش‌های مبتنی بر R سریع‌تر بوده و با limma این سرعت در حداکثر مقدار بوده و کمترین حجم حافظه نیز مصرف می‌شود. بنابراین این روش‌ها را حتی می‌توان روی یک لپ‌تاپ متوسط نیز اجرا نمود.

منابع

1. Ringnér M. What is principal component analysis? Nat Biotechnol 26:303–304, 2008.

2. Müller F.-J., Schuldt B.M., Williams R., Mason D., Altun G., Papapetrou E., Danner S. et al. A bioinformatics assay for pluripotency in human cells. Nat Methods 8:315–317, 2011.
3. Auer P.L. and Doerge R.W. Statistical design and analysis of RNA sequencing data. Genetics 185:405–416, 2010.
4. Bengtsson M., Ståhlberg A., Rorsman P., and Kubista M. Gene expression profling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels. Genome Res 15(10):1388–1392, 2005.
5. Anders S. and Huber W. Differential expression analysis for sequence count data. Genome Biology 11:R106, 2010.
6. Robinson M.D., McCarthy D.J., and Smyth G.K. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26(1):139–140, 2010.
7. Esnaola M., Puig P., Gonzalez D., Castelo R., and Gonzalez J.R. A flexible count data model to ft the wide diversity of expression profles arising from extensively replicated RNA-seq experiments. BMC Bioinformatics 14(1):254, 2013.
8. Law C.W., Chen Y., Shi W., Smyth G.K. Voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. Genet Mol 15: R29, 2014.
9. Love MI, Huber W, and Anders S. Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. bioRxiv, doi:10.1101/002832, 2014.
10. Li J. and Tibshirani R. Finding consistent patterns: A nonparametric approach for identifying differential expression in RNA-Seq data. Stat Methods Med Res 22(5):519–36, 2013. doi: 10.1177/0962280211428386.
11. Tarazona S., García-Alcalde F., Dopazo J., Ferrer A., and Conesa A. Differential expression in RNA-seq: A matter of depth. Genome Res 21(12):2213–2223, 2011. doi: 10.1101/gr.124321.111.
12. Mortazavi A., Williams B.A., McCue K., Schaeffer L., and Wold B. Mapping and quantifying mammalian transcriptomes by RNA-seq. Nat Methods 5(7):621–628, 2008. doi: 10.1038/nmeth.1226.
13. Trapnell C., Williams B.A., Pertea G., Mortazavi A., Kwan G., van Baren M.J., Salzberg S.L., Wold B.J., and Pachter L. Transcript assembly and quantifcation by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 28(5):511–515, 2010. doi: 10.1038/nbt.1621.
14. Wang E.T., Sandberg R., Luo S., Khrebtukova I., Zhang L., Mayr C., Kingsmore S.F., Schroth G.P., and Burge C.B. Alternative isoform regulation in human tissue transcriptomes. Nature 456(7221):470–476, 2008.
15. Wagner G.P., Kin K., and Lynch V.J. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. Teory Biosci 131(4):281–285, 2012.
16. Robinson M.D. and Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biol 11(3):R25, 2010.doi: 10.1186/gb-2010-11-3-r25.
17. Dillies M.A., Rau A., Aubert J., Hennequet-Antier C., Jeanmougin M., Servant N., Keime C. et al. on behalf of the FrenchStatOmique Consortium.

A comprehensive evaluation of normalization methods for Illumina highthroughput RNA sequencing data analysis. Brief Bioinform 14(6):671–683,2013.
18. Trapnell C., Hendrickson D.G., Sauvageau M., Goff L., Rinn J.L., and PachterL. Differential analysis of gene regulation at transcript resolution with RNAseq. Nat Biotechnol 31(1):46–53, 2013. doi: 10.1038/nbt.2450.
19. Soneson C. and Delorenzi M. A comparison of methods for differential expression analysis of RNA-seq data. BMC Bioinformatics 14:91, 2013.
20. Trapnell C., Roberts A., Goff L., Pertea G., Kim D., Kelley D.R., Pimentel H., Salzberg S.L., Rinn J.L., and Pachter L. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufinks. Nat Protoc 7(3):562–578, 2012. doi: 10.1038/nprot.2012.016.
21. Glaus P., Honkela A., and Rattray M. Identifying differentially expressed transcripts from RNA-seq data with biological variation. Bioinformatics 28(13):1721–1728, 2012.
22. Leng N., Dawson J.A., Tomson J.A., Ruotti V., Rissman A.I., Smits B.M.G., Haag J.D., Gould M.N., Stewart R.M., and Kendziorski C. EBSeq: An empirical Bayes hierarchical model for inference in RNA-seq experiments. Bioinformatics 29(8):1035–1043, 2013.
23. Limma user guide. http://www.bioconductor.org/packages/2.12/bioc/vignettes/limma/inst/doc/usersguide.pdf (Accessed 24 October 2013).
24. Bourgon R., Gentleman R., and Huber W. Independent fltering increases detection power for high-throughput experiments. Proc Natl Acad Sci USA 107(21):9546–9551, 2010. doi: 10.1073/pnas.0914005107.

فصل نهم

آنالیز افتراقی استفاده از اگزون

۹–۱ مقدمه

در انسان و بسیاری از یوکاریوت‌های دیگر یک ژن می‌تواند در قالب‌های مختلفی بیان شود. معمول‌ترین سازوکارهای ایجاد این ایزوفرم‌ها عبارتند از: استفاده‌ی مجدد از پروموتور و پیرایش مجدد. محل‌های آغاز مجدد رونویسی منجر به تفاوت در شروع mRNA می‌شوند. ولی پیرایش مجدد سبب می‌شود که برخی از اگزون‌ها نادیده گرفته شده و در نتیجه ترجمه نگردند. توالی‌یابی RNA امکانات جالبی را برای مطالعه‌ی بیان و تنظیم ایزوفرم‌ها در سطح کُل ژنوم فراهم می‌آورد.

اکثر روش‌های فعلی توالی‌یابی RNA خوانش‌های کوتاهی تولید می‌کنند که کُل رونوشت‌ها را پوشش نمی‌دهند. در عوض لازم است که رونوشت‌ها از قطعات توالی‌یابی اسمبل شوند. این اسمبل کردن و سپس برآورد فراوانی می‌تواند چالش‌برانگیز باشد. زیرا معمولاً ایزوفرم‌ها اگزون‌های مشترک یا همپوشان دارند. علاوه بر این به دلیل اُریبی ناشی از توالی‌یابی و تهیه‌ی کتابخانه، پوشش در طول رونوشت‌ها یکنواخت نیست. برای اجتناب از عدم قطعیت در اسمبل کردن، یک روش برای مطالعه‌ی تنظیم ایزوفرم جایگزین آن عبارت از جستجوی تفاوت‌ها در استفاده از اگزون‌های منفرد است. در فصل قبل، آنالیز افتراقی بیان در سطح ژن مورد بحث و بررسی واقع شد. ولی خوانش‌های توالی‌یابی RNA می‌توانند با اگزون‌ها نیز مکان‌یابی شده و بدین ترتیب تفاوت‌ها در شمارش‌های مختص اگزون‌ها نیز می‌تواند بین شرایط، گروه‌ها یا تیمارهای خاص مقایسه گردد. این موضوع در فصل حاضر مورد بحث و بررسی واقع می‌شود.

تمرکز اصلی این فصل روی بسته‌ی DEXSeq از پروژه‌ی Bioconducror است (۱ و ۲). روش پیاده‌سازی شده در بسته‌ی DEXSeq مشابه روش‌های مورد استفاده در بسته‌ی DESeq است. این بدان معناست که بسته‌ی مزبور نیز می‌تواند ژن‌هایی که متفاوت بیان شده‌اند را شناسایی کند. علاوه بر این، ویژگی‌های اختصاصی بسته‌ی DEXSeq از بسته‌ی edgeR گرفته شده است. بنابراین در اینجا خصوصیات این روش مجدداً به طور کامل مورد بحث قرار نمی‌گیرد. در صورت لزوم، خوانندگان می‌توانند به فصل هشتم مراجعه نمایند.

چون هدف، مقایسه‌ی چند شرایط آزمایشی است، لذا اطمینان یافتن از این موضوع که به تعداد کافی تکرار برای هر کدام از شرایط مزبور ایجاد شده است، از اهمیت بالایی برخوردار است. روش‌هایی وجود دارند که امکان مقایسه‌ی دو شرایط مختلف با یک تکرار واحد در دو گروه را فراهم

می‌آورند. این کار شامل مقایسه‌ی ساده‌ی شمارش‌های خوانش‌ها در یک ژن یا رونوشت در یک زمان و با استفاده از آزمون دقیق فیشر[1] است. ولی این روش اجازه نمی‌دهد که حتی در صورت تکرار نمونه‌ها برای هر گروه، تنوع زیستی نیز در نظر گرفته شود. این روش در یک نرم‌افزار تحت عنوان MISO (۳) مورد استفاده قرار گرفته است. ولی در کتاب حاضر بیش از این مورد بحث و بررسی قرار نمی‌گیرد.

اگر نمونه‌های تکراری برای هر گروه در اختیار باشند، از نظر آماری احتمال یافتن اگزون‌های با بیان متفاوت بین شرایط آزمایشی بیشتر خواهد بود. توزیع شمارش خوانش‌ها را می‌توان از نوع توزیع پواسون در نظر گرفت. این توزیع آماری اغلب برای توصیف داده‌های شمارش به کار گرفته می‌شود. در عمل، اغلب شمارش‌های مختص ژن یا رونوشت بیش از حد پراکنده بوده و از توزیع دو جمله‌ای منفی پیروی می‌کنند. این توزیع به عنوان یک تعمیم از توزیع پواسون در نظر گرفته شده و انطباق بهتری با داده‌ها دارد. بنابراین داده‌ها می‌توانند توسط مدل (رگرسیون) خطی تعمیم یافته‌ی استاندارد آنالیز شوند. این مدل از مزیت استفاده از نمونه‌های تکراری در برآورد واریانس یا پراکندگی برای کلیه‌ی اگزون‌ها برخوردار است. ولی معمولاً تعداد کُل نمونه‌ها در یک آزمایش توالی‌یابی RNA کم است و برآورد مقادیر پراکندگی مختص اگزون دقیق نخواهد بود. بنابراین روش‌های پیاده شده در بسته‌ی DEXSeq امکان برآورد پراکندگی برای هر اگزون را با استفاده از روشی که اطلاعات را از سایر اگزون‌های با بیان مشابه دریافت نماید، فراهم می‌کند.

در DEXSeq یک مدل دو جمله‌ای منفی مجزا برای هر ژن برازش داده می‌شود. لگاریتم بیان هر ژن (یا لگاریتم شمارش خوانش‌های مکان‌یابی شده با آن) به صورت تابعی از: ۱) بیان پایه‌ی ژن، ۲) بخش مورد انتظار از خوانش‌های مکان‌یابی شده با ژن که با اگزون معینی مکان‌یابی گردیده‌اند، ۳) لگاریتم تغییر فولد در یک شرایط معین و ۴) اثر شرایط روی بخشی از شمارش‌ها که با آن اگزون معین مکان‌یابی می‌شوند، مدل‌بندی می‌گردد. این مدل امکان شناسایی ژن‌های با بیان متفاوت و اگزون‌های دارای بیان متفاوت را فراهم می‌آورد.

گردش کار یک آنالیز معمولی با DEXSeq شامل شمارش خوانش‌های هر اگزون و خواندن جدول شمارش در R ، نرمال‌سازی با کمک برآورد عوامل اندازه[2]، برآورد مقادیر پراکندگی مختص اگزون، آزمون افتراقی استفاده از اگزون و مصورسازی نتایج است. این مراحل به طور مفصل در بخش‌های زیر و با استفاده از مجموعه‌ی داده‌های ENCODE در مثال‌ها تشریح می‌گردند.

1- Fisher’s exact test
2- Size factor

۹–۲ آماده‌سازی فایل‌های ورودی برای DEXSeq

فایل‌های ورودی برای DEXSeq توسط دو اسکریپت پایتون که با بسته‌ی DEXSeq همراه هستند، ایجاد می‌شوند. علاوه بر این بسته‌ی پایتون HTSeq نیز مورد نیاز است. ابتدا توسط اسکریپت پایتون تحت عنوان dexseq_prepare_annotation.py یک فایل GTF مسطح ایجاد می‌شود. با استفاده از اسکریپت پایتون با عنوان dexseq_count.py ده خوانش به ازای هر اگزون (در واقع به ازای هر ناحیه‌ی اگزونی یا هر بِن اگزونی (ر.ک: Anders و همکاران (۲))) ایجاد می‌گردد. ایجاد جدول شمارش به تفصیل در فصل ششم تشریح شده است. همچنین می‌توانید از راهنمای DEXSeq که در آن مثال‌های تشریحی از نحوه‌ی کاربرد این نرم‌افزار ارائه شده و از سایت Bioconductor نیز قابل تهیه است، استفاده نمایید. زیرا ممکن است جزییات این فرآیندها دستخوش تغییر شوند. از طرف دیگر این فرآیندها می‌توانند به تنهایی و با استفاده از تابع `generateCountTable()` در Bioconductor انجام شوند (ر.ک: فصل هفتم). این کار نیز در راهنمای بسته‌ی paratyroidSE تشریح شده است.

اسکریپت پایتون dexseq_count.py نیازمند فایل GTF مسطح و همردیف‌هایی در فایل‌های با فرمت SAM یا BAM است. لازم است که فایل‌های حاوی داده‌های جفت انتهایی بر مبنای نام خوانش یا موقعیت ژنومی مرتب شوند. اگر فایل‌های شما در فرمت BAM است، می‌توانید آنها را با استفاده از SAMtools (http://samtools.sourceforge.net/) به فرمت SAM تبدیل کنید. یک فایل BAM از پروژه‌ی ENCODE را می‌توان با استفاده از دستور زیر تبدیل نمود:

```
samtools view -h Gm12892_1_chr18.bam -o Gm12892_1_chr18.sam
```

در اینجا، GM12892_1_chr18.bam ، نام فایل ورودی بوده و GM12892_1_chr18.sam ، نام فایل خروجی است.

هر کدام از فایل‌های SAM که در بالا ایجاد شده‌اند، به صورت زیر پردازش می‌گردند. توجه نمایید که لازم است که نشان دهید که آیا داده‌ها با دستورالعمل مختص زنجیره ایجاد شده‌اند یا خیر (`-s no`).

```
python dexseq_count.py -p yes -s no -r name
Homo_sapiens.GRCh37.70.chr18.chr.gtf Gm12892_1_chr18.sam
gm1.txt
```

این دستورات هفت فایل متنی با نام‌های gm1 ، gm2 ، gm3 ، h1 ، h2 ، h3 و h4 ایجاد می‌کنند. هر فایل متنی حاوی دو ستون بوده که نخستین ستون (id) شامل مشخصه‌ی ژن در

Ensembl است که توسط یک علامت دو نقطه‌ی بیانی (:) از نام اگزون جدا شده است و دومین ستون (count) نیز شامل شمارش خوانش‌های مکان‌یابی شده با آن اگزون است (در واقع این جدول دارای ردیف سرستون نبوده و در اینجا ردیف مزبور فقط برای تشریح کردن افزوده شده است):

Id	Count
ENSG00000235552:002	35,769
ENSG00000175886:001	15,732
ENSG00000235552:003	7515
ENSG00000235297:001	7275
ENSG00000215492:008	5882

توابع R در بسته‌ی DEXSeq با عملکرد تابعی بسته‌ی پایتون HTSeq تلفیق شده است. این کار سبب شده است که فایل‌های متنی بتوانند مستقیماً به R وارد شوند. این موضوع در بخش‌های بعدی توضیح داده می‌شود.

۹–۳ خواندن داده‌ها در R

خواندن کلیه‌ی جداول شمارش مختص نمونه‌ها توسط تابع `read.HTSeqCounts()` در بسته‌ی DEXSeq صورت می‌گیرد. این تابع فهرستی از نام فایل‌های وارد شده (یک نام از فایل GTF مسطح، مشابه نام استفاده شده در طی ایجاد فایل‌های شمارش) و یک چارچوب داده که تنظیمات و شرایط آزمایش را شرح داده و متغیرهایی که در طی آنالیز آماری افتراقی استفاده از اگزون مورد نیاز هستند، دریافت می‌دارد. اگر فایل GTF در طی ورود داده‌ها استفاده نشود، مصورسازی صورت نگرفته ولی هنوز آزمون آماری می‌تواند اجرا گردد.

نخست بایستی تنظیمات و شرایط آزمایش تشریح شوند. سه نمونه‌ی GM و چهار نمونه‌ی hESC در مجموعه‌ی داده‌های ENCODE وجود دارد. ابتدا نمونه‌های GM و سپس نمونه‌های hESC وارد می‌شوند. بردار[1] حاوی نام نمونه‌ی اصلی و شرایط بردار که منجر به گروه‌بندی نمونه‌ها به GM یا hESC می‌شوند، خواهد بود. نخست این بردارها ایجاد شده و سپس به یکدیگر متصل گردیده و یک چارچوب داده تحت عنوان فنودیتا[2] تشکیل می‌دهند:

1- Vector
2- Phenodata

```
samplename<-c("Gm12892_1", "Gm12892_2", "Gm12892_3",
                "hESC_1", "hESC_2", "hESC_3", "hESC_4")
condition<-c("gm", "gm", "gm", "esc", "esc", "esc", "esc")
phenodata<-data.frame(samplename, condition)
rownames(phenodata)<-c("gm1",  "gm2",  "gm3",  "h1",  "h2",
"h3", "h4")
```

پس از تشکیل فایل فنودیتا، جداول شمارش می‌توانند در R خوانده شوند. در اینجا داده‌ها در داخل یک شیء ExonCountSet تحت عنوان ec خوانده می‌شوند:

```
library(DEXSeq)
ec<read.HTSeqCounts(countfiles=c("gm1.txt","gm2.txt",
                        "gm3.txt","h1.txt","h2.txt",
                        "h3.txt","h4.txt"),
            design=phenodata)
```

گاهی اوقات، مجموعه‌ی داده‌های شمارش کُل اگزون در یک جدول داده می‌شود. در چنین وضعیتی این جدول می‌تواند: ۱) به فایل‌های جداگانه شکسته شده و طبق روش فوق‌الذکر در R خوانده شود، ۲) در R به صورت یک جدول خوانده شده و با دستور `newExonCountSet()` به یک ExonCountSet تبدیل شود. این روش نسبتاً ساده بوده و تنها مانع در اجرای آن ایجاد ID های ژن و ID های اگزون برای هر ردیف جدول است. ولی این ID ها را نیز می‌توان از نام ردیف‌های جدول و از طریق شکستن آنها در ناحیه‌ی علامت دو نقطه‌ی بیانی (:) ایجاد نمود. فرآیند این کار به طور کامل به شرح زیر است:

```
dat<-read.table("ENCODE_ngs-data-table_exons.tsv",
                    header=T,sep="\t")
nc<-nchar(rownames(dat))
geneids<-substr(rownames(dat), 1, nc-4)
exonids<-substr(rownames(dat),nc-2, nc)
ec2<-newExonCountSet(dat, phenodata, geneids, exonids)
```

۹-۴ دسترسی به شیء ExonCountSet

بعد از ایجاد شیء R حاوی داده‌ها، بهتر است که بررسی شود که آیا این شیء به درستی ایجاد شده و نیز حاوی اطلاعات صحیح در فرمتی صحیح است یا خیر. با کمک تابع `design()` می‌توان به فنودیتای حاوی حاشیه‌نگاری نمونه‌ها دسترسی داشت:

```
design(ec)
```

	samplename	condition
gm1.txt	Gm12892_1	gm
gm2.txt	Gm12892_2	gm
gm3.txt	Gm12892_3	gm
h1.txt	hESC_1	esc
h2.txt	hESC_2	esc
h3.txt	hESC_3	esc
h4.txt	hESC_4	esc

شمارش خوانش‌ها می‌تواند با تابع counts() در دسترس قرار گیرد. با تابع head() تعداد ردیف‌هایی که روی صفحه نمایش داده می‌شوند را می‌توان محدود نمود (مثلاً ۱۰ ردیف):

```
head(fData(ec), 10)
```

	gm1.txt	gm2.txt	gm3.txt	h1.txt	h2.txt	h3.txt	h4.txt
ENSG00000000003:E001	0	0	0	0	0	0	0
ENSG00000000003:E002	0	0	0	0	0	0	0
ENSG00000000003:E003	0	0	0	0	0	0	0
ENSG00000000003:E004	0	0	0	0	0	0	0
ENSG00000000003:E005	0	0	0	0	0	0	0
ENSG00000000003:E006	0	0	0	0	0	0	0
ENSG00000000003:E007	0	0	0	0	0	0	0
ENSG00000000003:E008	0	0	0	0	0	0	0
ENSG00000000003:E009	0	0	0	0	0	0	0
ENSG00000000003:E010	0	0	0	0	0	0	0

با کمک تابع fData() می‌توان به داده‌های ترکیب یا حاشیه‌نگاری‌ها برای ژن‌ها و اگزون‌ها دست یافت:

```
head(fData(ec), 10)
```

	geneID	exonID	testable	dispBefore Sharing	dispFitted
ENSG00000000003:E001	ENSG00000000003	E001	NA	NA	NA
ENSG00000000003:E002	ENSG00000000003	E002	NA	NA	NA
ENSG00000000003:E003	ENSG00000000003	E003	NA	NA	NA
ENSG00000000003:E004	ENSG00000000003	E004	NA	NA	NA
ENSG00000000003:E005	ENSG00000000003	E005	NA	NA	NA
ENSG00000000003:E006	ENSG00000000003	E006	NA	NA	NA
ENSG00000000003:E007	ENSG00000000003	E007	NA	NA	NA
ENSG00000000003:E008	ENSG00000000003	E008	NA	NA	NA
ENSG00000000003:E009	ENSG00000000003	E009	NA	NA	NA
ENSG00000000003:E010	ENSG00000000003	E010	NA	NA	NA

```
                    dispersion pvalue padjust chr start end strand transcripts
ENSG00000000003:E001         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E002         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E003         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E004         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E005         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E006         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E007         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E008         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E009         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
ENSG00000000003:E010         NA     NA      NA <NA>    NA  NA   <NA>        <NA>
```

با کمک تابع `geneIDs()` می‌توان به ID ژن‌ها دسترسی داشت. این موضوع از آن جهت اهمیت دارد که می‌توان فهمید که هر ژن چه تعداد اگزون دارد. اگر این شمارش برای چند ژن ابتدایی صورت گیرد، نتیجه به شکل زیر خواهد بود:

```
data.frame(head(table(geneIDs(ec))))
ENSG00000000003                15
ENSG00000000005                 9
ENSG00000000419                19
ENSG00000000457                21
ENSG00000000460                48
ENSG00000000938                29
```

در اینجا تابع `data.frame()` تنها برای دستیابی به نمایشی زیباتر از جدول مزبور به کار گرفته شده است. به عنوان مثال، ژن ENSG00000000003 دارای ۱۵ اگزون است.

به همین ترتیب می‌توان تعداد ژن‌های دارای تعداد مشخصی اگزون را شمارش نمود:

```
head(data.frame(table(table(geneIDs(ec)))))
  Var1  Freq
1    1 20436
2    2  7868
3    3  3394
4    4  2068
5    5  1771
6    6  1240
```

همان‌گونه که دیده می‌شود، به نظر می‌رسد که ۲۰۴۳۶ ژن با تنها یک اگزون در این مجموعه‌ی داده وجود دارد. حداکثر تعداد اگزون‌ها برای یک ژن در این مجموعه‌ی داده‌ها نیز برابر با ۳۹۴ اگزون است.

۹–۵ نرمال‌سازی و برآورد واریانس

معمولاً عمق توالی‌یابی در بین نمونه‌ها متفاوت بوده و این اُریبی در پوشش بایستی در طی آنالیز در نظر گرفته شود. برای این کار از نرمال‌سازی استفاده می‌شود. بسته‌ی DEXSeq از روش نرمال‌سازی مشابه بسته‌ی DESeq استفاده می‌کند. این بسته تلاش می‌کند که اندازه‌ی کتابخانه و ترکیب ترانسکریپتوم را بین نمونه‌ها نرمال‌سازی نماید. این نرمال‌سازیِ خاص، یک عامل اندازه برای هر نمونه در آزمایش برآورد می‌کند. عوامل اندازه عمق نسبی توالی‌یابی نمونه‌های مختلف را منعکس می‌کنند.

عوامل اندازه توسط تابع `estimateSizeFactors()` برآورد می‌گردند:

```
ec<-estimateSizeFactors(ec)
```

برای کنترل اینکه چه عواملی عوامل اندازه هستند، می‌توان از تابع دسترسی `sizeFactors()` بهره گرفت:

```
sizeFactors(ec)
gm1.txtgm2.txt gm3.txt h1.txt h2.txt h3.txt h4.txt
1.62556350.78236070.78643541.1872495 1.0721540 1.0778185 0.8696311
```

بعد از برآورد عوامل اندازه، لازم است که قبل از اجرای آزمون آماری افتراقی استفاده از اگزون، پراکنش (واریانس) مختص اگزون برآورد گردد. این پراکنش از تکرارهای زیستی در مجموعه‌ی داده‌ها برآورد می‌شود. ولی می‌تواند از تکرارهای فنی موجود در مجموعه‌ی داده‌ها نیز باشد. معمولاً پراکنش نمی‌تواند برای هر اگزون به صورت جداگانه برآورد شود. زیرا معمولاً تعداد تکرارهای زیستی در مجموعه‌ی داده‌ها نسبتاً کم است. بنابراین برآورد پراکنش با استفاده از اطلاعات حاصل از اگزون‌هایی که تقریباً در نرخ مشابهی بیان شده و پراکنش برای آنها محاسبه شده است، صورت می‌گیرد. این روش اغلب برآورد پراکنش با روش وابستگی شدید[1] نامیده می‌شود.

یک برآورد پراکنش برای هر ژن با استفاده از تابع `estimateDispersions()` محاسبه می‌گردد. محاسبه‌ی مقادیر پراکنش اندکی زمان می‌خواهد. ولی پیشرفت کار را می‌توان پایش نمود. زیرا یک نقطه به ازای هر ۱۰۰ ژن پردازش شده روی صفحه‌ی تصویر نمایش داده می‌شود.

```
ec<-estimateDispersions(ec)
Dispersion estimation. (Progress report: one dot per
100 genes)
..
```

1- Intensity-dependent manner

در این مورد خاص، در برآورد پراکنش دو پیام هشدار نیز دریافت می‌شود:

```
Warning messages:
1: In .local(object, ...) :
  Exons with less than 10 counts will not be tested.
For more details please see the manual page of
'estimateDispersions', parameter 'minCount'
2: In .local(object, ...) :
  Genes with more than 70 testable exons will be
omitted from the analysis. For more details please see
the manual page of 'estimateDispersions', parameter
'maxExon'.
```

هیچ‌کدام از این پیام‌های هشدار، حاوی نکته‌ی حیاتی و تعیین کننده نبوده و برآورد با موفقیت انجام شده است. این پیام‌ها یادآوری می‌کنند که دو پارامتر اضافی برای تابع estimateDispersions() وجود دارند که تعداد اگزون‌های پردازش شده را محدود می‌نمایند. پارامتر minCount که مقدار پیش‌فرض آن ۱۰ است، سبب می‌شود که در این مرحله تنها اگزون‌های با حداقل تعداد ۱۰ شمارش مورد پردازش قرار گیرند. همچنین پارامتر maxExons تعیین می‌کند که تعداد حداکثر اگزون‌ها در یک ژن به صورت پیش‌فرض، بیش از ۷۰ اگزون نباشد. اگر تعداد اگزون‌ها در یک ژن بیش از این مقدار باشد، هیچ پراکنشی برای آن برآورد نمی‌گردد.

نتایج در شیء ec ، شیار featureData و ستون DispBeforeSharing ذخیره می‌گردد. به عنوان مثال با دستور زیر می‌توان به مقادیر مزبور دسترسی داشت:

```
featureData(ec)@data$dispBeforeSharing
```

یا:

```
fData(ec)$dispBeforeSharing.
```

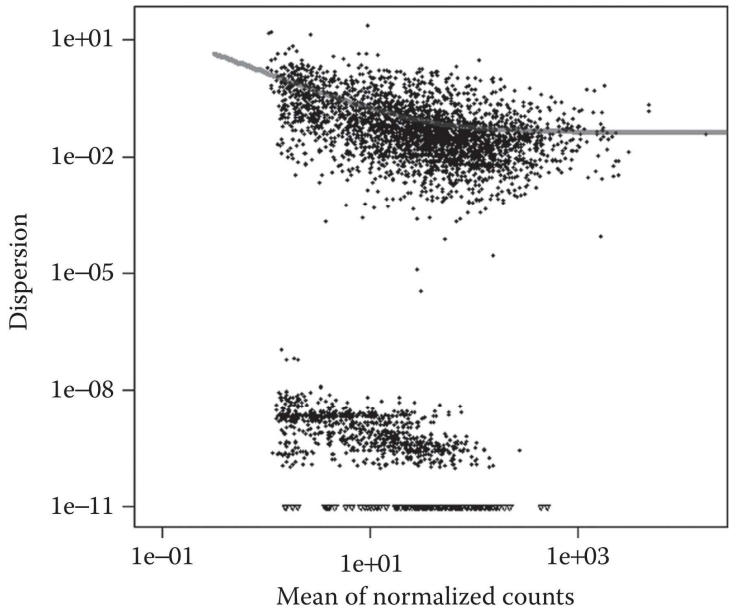
برآوردهای پراکنش به دست آمده لازم است که تصحیح شوند تا اطلاعات حاصل از شباهت رفتار اگزون‌ها نیز به آنها افزوده شود. این کار توسط دستور زیر که یک تابع ساده را با داده‌های پراکنش برازش می‌دهد، انجام می‌شود:

```
ec<-fitDispersionFunction(ec)
```

این دستور نیز برآوردهای پراکنش تصحیح شده را در ستون پراکنش شیار featureData ذخیره می‌نماید. بعد از تصحیح پراکنش برآورد شده، بهتر است که نتایج کنترل شوند. بهترین راه برای کنترل نتایج، استفاده از یک نمودار است که در آن میانگین خوانش‌های نرمال شده در محور افقی و پراکنش در محور عمودی واقع شده باشند. همچنین یک تابع میانگین پراکنش به صورت یک خط به نمودار افزوده می‌شود (نگاره‌ی ۹-۱). این نمودار می‌تواند با استفاده از تابع `plotDispEsts()` که در راهنمای بسته‌ی DEXSeq و نیز در کُدهای این کتاب موجود است، ایجاد شود:

```
plotDispEsts(ec)
```

نقاط نزدیک به کف نمودار، اگزون‌هایی هستند که دارای برآورد کوچکی از پراکنش می‌باشند. برخی از نقاط، خطی را در کف نمودار تشکیل می‌دهند. پراکنش این اگزون‌ها عملاً صفر یا دقیقاً صفر است.



نگاره‌ی ۹-۱: نمودار میانگین پراکنش. هر نقطه در این نمودار نشان دهنده‌ی یک اگزون است. خط ترسیم شده (میانگین تابع پراکنش) از شکل ابری داده‌ها تبعیت می‌کند.

۹–۶ آزمون افتراقی استفاده از اگزون

وقتی که عوامل اندازه (نرمال‌سازی) و مقادیر پراکنش برآورد شدند، شرایط لازم برای آزمون آماری افتراقی استفاده از اگزون مهیا خواهد بود. این آزمون بر اساس مقایسه‌ی دو مدل خطی تعمیم یافته با یک آزمون درستنمایی انجام می‌شود. در مورد مجموعه‌ی داده‌های ENCODE ، تنها دو گروه با هم مقایسه شده و در نتیجه این آزمون در عمل نسبتاً ساده‌تر خواهد بود.

برای هر اگزون، دو مدل متفاوت برازش می‌شود. مدل صفر[1] (نول)، بیان یک اگزون را از اثرات اصلی نمونه، اگزون و شرایط (یعنی گروهی که نمونه به آن تعلق دارد) مدل‌بندی می‌کند. مدل جایگزین[2] نیز اثر اصلی یک اگزون و اثر متقابل[3] آن با شرایط را در نظر می‌گیرد. به صورت خلاصه این مدل‌ها را می‌توان به صورت زیر نوشت:

مدل صفر: شمارش ~ نمونه + اگزون + شرایط

مدل جایگزین: شمارش ~ نمونه + اگزون + شرایط+ اگزون

اگزون ID = شرایط + اگزون × اگزون ID

اگزون‌های یک ژن می‌توانند با دستور `testGeneForDEU_BM()` آزمون شوند. این دستور دارای دو برهان است: شیء exonCountSet و یک نام ژن. به عنوان مثال:

```
testGeneForDEU_BM(ec, "ENSG00000017797")
```

	deviance	df	pvalue
E001	1.4925835	1	2.218161e-01
E002	0.0998600	1	7.519977e-01
E003	1.3569223	1	2.440716e-01
E004	2.6444047	1	1.039151e-01
E005	12.0979356	1	5.047768e-04
E008	26.6814441	1	2.399145e-07
E009	4.7567963	1	2.918282e-02
E010	8.6114130	1	3.340630e-03
E011	0.8170818	1	3.660348e-01
E012	0.4034728	1	5.253012e-01
E013	0.7998242	1	3.711459e-01
E014	3.0656931	1	7.996105e-02
E015	4.2166483	1	4.002916e-02
E016	4.9595151	1	2.594748e-02
E017	0.1039061	1	7.471916e-01
E018	2.2409165	1	1.344013e-01
E019	0.4172684	1	5.183032e-01

1- Null model
2- Alternative model
3- Interaction

خروجی این دستور یک جدول است که ردیف‌های آن شامل کلیه‌ی اگزون‌های ژن بوده و نتایج روی ستون‌ها فهرست شده‌اند. ستون انحراف (deviance) تفاوت نکویی برازش بین مدل‌های صفر و جایگزین را نشان می‌دهد. این مقادیر انحراف (deviance) دارای توزیع مربع کای (χ^2) بوده و هر کدام از مقادیر p (p-value) از طریق مقایسه‌ی انحراف (deviance) با توزیع مربع کای محاسبه شده و df نیز نشان دهنده‌ی درجه‌ی آزادی است. ستون آخر حاوی مقادیر p (p-value) است. اگر این مقادیر p با ۰/۰۵ مورد قضاوت واقع شوند (≤۰/۰۵) چهار اگزون (۵ ، ۸ ، ۹ و ۱۰) ژن ENSG00000017797 بین سلول‌های بنیادی جنینی انسان (ecs) و سلول‌های شاهد (gm) متفاوت بیان شده‌اند.

معمولاً بررسی یک ژن و اگزون‌هایش چندان جذابیت ندارد. ولی در اینجا کلیه‌ی ژن‌ها و اگزون‌های‌شان که در مجموعه‌ی داده‌ها در دسترس هستند، مورد آزمون واقع می‌شوند. تابع testForDEU() این کار را با فراخوانی تابع testGeneForDEU_BM() به صورت تکراری برای کلیه‌ی ژن‌ها در مجموعه‌ی داده‌ها انجام می‌دهد. ولی تنها ژن‌هایی که در طی برآورد پراکنش به عنوان قابل آزمون مشخص شده‌اند، آزمون می‌شوند. این ژن‌ها که به عنوان قابل آزمون مشخص گردیده‌اند، را می‌توان در ستون testable جدول داده‌های ترکیبات ملاحظه نمود:

```
head(fData(ec))
```

	geneID	exonID	testable	dispBeforeSharing	disp Fitted	dispersion
ENSG00000000003:E001	ENSG00000000003	E001	FALSE	NA	Inf	1e+08
ENSG00000000003:E002	ENSG00000000003	E002	FALSE	NA	Inf	1e+08
ENSG00000000003:E003	ENSG00000000003	E003	FALSE	NA	Inf	1e+08
ENSG00000000003:E004	ENSG00000000003	E004	FALSE	NA	Inf	1e+08
ENSG00000000003:E005	ENSG00000000003	E005	FALSE	NA	Inf	1e+08
ENSG00000000003:E006	ENSG00000000003	E006	FALSE	NA	Inf	1e+08

	pvalue	padjust	chr	start	end	strand	transcripts
ENSG00000000003:E001	NA	NA	<NA>	NA	NA	<NA>	<NA>
ENSG00000000003:E002	NA	NA	<NA>	NA	NA	<NA>	<NA>
ENSG00000000003:E003	NA	NA	<NA>	NA	NA	<NA>	<NA>
ENSG00000000003:E004	NA	NA	<NA>	NA	NA	<NA>	<NA>
ENSG00000000003:E005	NA	NA	<NA>	NA	NA	<NA>	<NA>
ENSG00000000003:E006	NA	NA	<NA>	NA	NA	<NA>	<NA>

آزمون حقیقی به آسانی و به روش زیر انجام می‌شود. ولی ممکن است که این روش اندکی وقت‌گیر باشد. اگر می‌خواهید تنها یک کروموزوم را آنالیز نمایید، به طور تقریبی زمان کافی برای درست کردن یک فنجان قهوه خواهید داشت! ولی اگر بخواهید یک ژنوم کامل را آنالیز کنید، به اندازه‌ی صرف کامل یک ناهار، دسر و قهوه طول خواهد کشید!

```
ec<-testForDEU(ec)
Testing for differential exon usage.(Progress report:
one dot per 100 genes)
..
```

علاوه بر نتیجه‌ی آزمون آماری، می‌توان تغییرات فولد را نیز برآورد نمود:

```
ec<-estimatelog2FoldChanges(ec)
```

پس از به دست آوردن نتایج آزمون آماری و در صورت تمایل محاسبه‌ی مقادیر تغییر فولد، یک جدول خلاصه از نتایج تلفیق شده تشکیل می‌گردد:

```
res<-DEUresultTable(ec)
```

تنها نتایج آماری معنی‌دار می‌توانند از نتایج غیرمعنی‌دار جدا شوند. تعداد کمی از نخستین نتایج آماری معنی‌دار را می‌توان در زیر ملاحظه نمود:

```
ind<-which(res$padjust <=0.05)
head(res[ind,])
```

	geneID	exonID	dispersionp	value
ENSG00000017797:E005	ENSG00000017797	E005	0.14779157	5.047768e-04
ENSG00000017797:E008	ENSG00000017797	E008	0.04967980	2.399145e-07
ENSG00000049759:E007	ENSG00000049759	E007	0.61660215	6.701749e-07
ENSG00000049759:E009	ENSG00000049759	E009	0.07053636	2.911027e-04
ENSG00000049759:E021	ENSG00000049759	E021	0.05289764	2.890047e-03
ENSG00000049759:E038	ENSG00000049759	E038	0.04466578	9.168722e-04

	padjust	meanBase	log2fold(esc/gm)
ENSG00000017797:E005	1.192748e-02	13.30575	-0.8620503
ENSG00000017797:E008	1.681800e-05	199.74625	0.4350362
ENSG00000049759:E007	4.239748e-05	20.98780	-2.3576794
ENSG00000049759:E009	8.002469e-03	50.20427	-0.4834733
ENSG00000049759:E021	4.639519e-02	136.85350	0.3015733
ENSG00000049759:E038	1.957545e-02	703.55049	0.2016261

نام برخی از ژن‌ها به صورت ترکیبی، نظیر ENSG00000119547+ ENSG00000266636 ، آمده است. این ژن‌ها در برخی از اگزون‌ها مشترک بوده و بنابراین نمی‌توان آن اگزون‌ها را تنها به یک ژن منتسب نمود. تفسیر نتایج چنین ترکیباتی مشکل است. زیرا این نتایج بیش از آنکه ناشی

از تفاوت در بیان اگزون‌ها باشند، از تفاوت در بیان ژن‌ها ناشی می‌شوند. به عنوان نمونه‌ای از این نوع نتایج، خروجی زیر را ملاحظه نمایید:

```
tail(head(res[ind,],69),3)
                                                                  geneID  exonID
ENSG00000119547+ENSG00000266636:E002 ENSG00000119547+ENSG00000266636    E002
ENSG00000119547+ENSG00000266636:E003 ENSG00000119547+ENSG00000266636    E003
ENSG00000119547+ENSG00000266636:E007 ENSG00000119547+ENSG00000266636    E007
                                      dispersion        pvalue       padjust
ENSG00000119547+ENSG00000266636:E002  0.9580874  6.039439e-06  3.175235e-04
ENSG00000119547+ENSG00000266636:E003  0.1404351  2.284703e-04  6.719902e-03
ENSG00000119547+ENSG00000266636:E007  0.0598415  2.437996e-08  2.050842e-06
                                          meanBase        log2fold(esc/gm)
ENSG00000119547+ENSG00000266636:E002      1.527859              -3.839978
ENSG00000119547+ENSG00000266636:E003     14.307039              -1.286007
ENSG00000119547+ENSG00000266636:E007     81.487062               1.266303
```

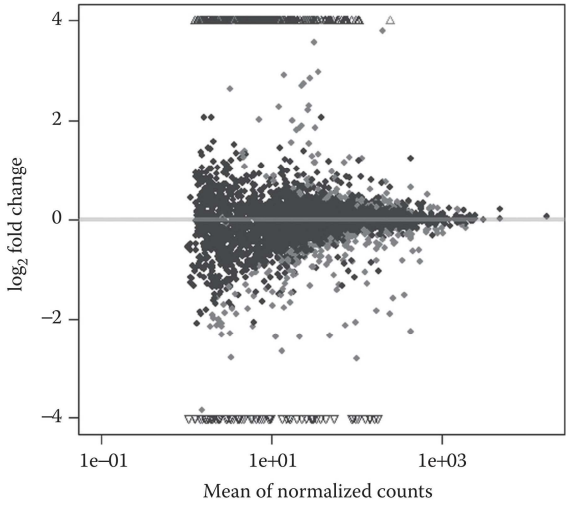
گاهی اوقات ترسیم نمودار پیرایش برای چنین ژن‌های خاصی و ملاحظه‌ی اینکه آیا این نتایج منعکس کننده‌ی بیان متفاوت ژن هستند یا اگزون، می‌تواند مفید واقع گردد.

۹-۷ مصورسازی

سه روش مصورسازی وجود دارد که به خوبی نتایج به دست آمده را تخلیص می‌نمایند. این روش‌ها عبارتند از: نمودار MA ، نمودار آتشفشانی و نقشه‌ی حرارتی. این نمودارها به طور مفصل در فصل یازدهم تشریح شده‌اند. ولی یک تابع نمودار MA اختصاصی برای نمایش نتایج حاصل از بسته‌ی DEXSeq وجود دارد. یک نمودار MA در واقع یک نمودار پراکنش نرمال است که در آن میانگین بیان یک اگزون روی محور x و تغییر فولد روی محور y در نظر گرفته می‌شود.

تابع نمودار MA برای یک شیء exonCountSeq در کُدهای این کتاب و در راهنمای بسته‌ی DEXSeq موجود است. این تابع اگزون‌های دارای مقادیر p تصحیح شده‌ی کمتر از ۰/۱ را با رنگ قرمز مشخص کرده و نقاطی که در خارج از محور y واقع می‌گردند را با مثلث‌های کوچکی محصور می‌کند. نمودار MA را می‌توان با استفاده از کُد زیر ایجاد نمود. نمودار حاصل در نگاره‌ی ۹-۲ نمایش داده شده است.

```
x<-data.frame(baseMean = res$meanBase,
              log2FoldChange = res$'log2fold(esc/gm)',
              padj = res$padjust)
plotMA(na.omit(x), ylim = c(-4,4), cex = 0.8)
```



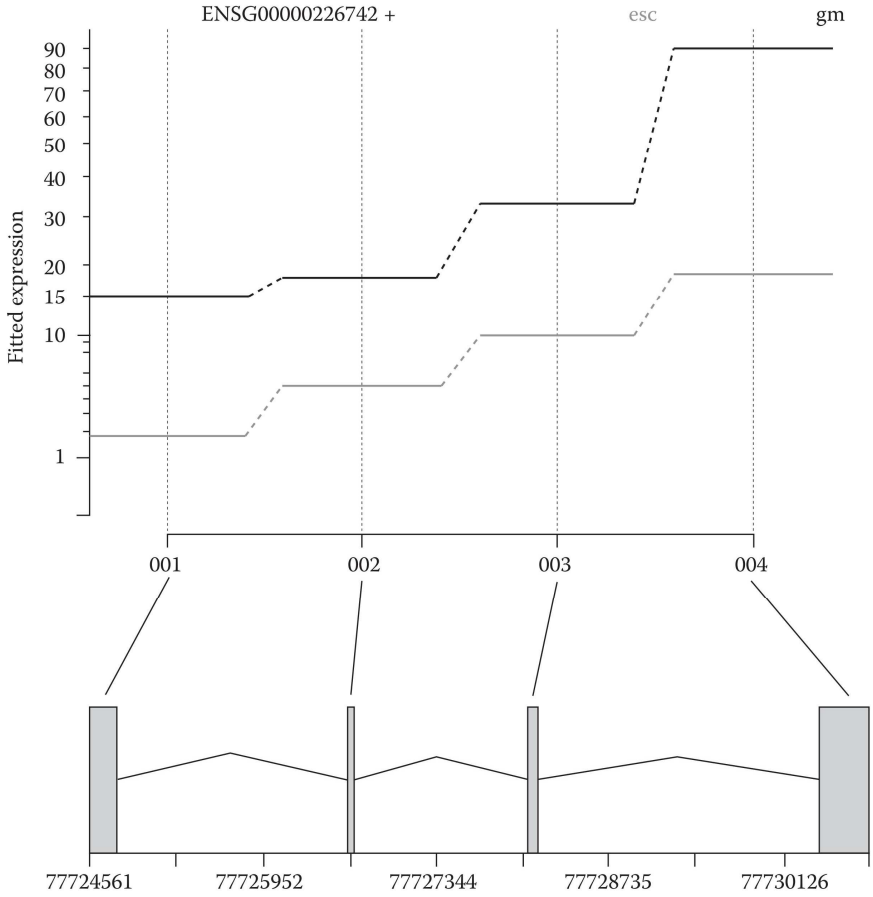
نگاره‌ی ۹-۲: نمودار MA برای نتایج حاصل از آنالیز افتراقی استفاده از اگزون

نموداری که می‌توان برای مصورسازی نتایج مختص اگزون برای یک ژن معین استفاده نمود، `plotDEXSeq()` است. ترسیم یک نمودار برای بیان برازش یافته، پیرایش برازش یافته یا شمارش‌های نرمال‌سازی شده امکان‌پذیر است. متاسفانه این تابع تنها زمانی به طور کامل عمل می‌کند که کلیه‌ی حاشیه‌نگاری‌های اگزونی (کروموزوم‌ها، موقعیت‌های ابتدایی و انتهایی، نام رونوشت‌ها و زنجیره‌ها) برای تمامی ژن‌های موجود در مجموعه‌ی داده‌ها در شیء exonCountSet حضور داشته باشند. چون همه‌ی حاشیه‌نگاری‌ها برای مجموعه‌ی داده‌های ENCODE یافت نگردید، لذا برای مواجهه با این مشکل، تابع مزبور بایستی تغییر داده می‌شد. تابع تغییر داده شده در فایل اضافی مربوط به این فصل در وبسایت این کتاب موجود است. تابع تغییر داده شده تحت عنوان `plotDEXSeqSimple()` نامیده شده و مشابه تابع `plotDEXSeq()` عمل می‌کند. ولی این تابع تغییر یافته نیز تنها می‌تواند نمودار مقادیر بیان برازش یافته را ترسیم کرده و اگر کلیه‌ی اگزون‌های ژنی که ترسیم می‌شود، به طور کامل حاشیه‌نگاری شده باشند، ساختارهای رونوشتی نیز قابل ترسیم خواهند بود.

جهت ایجاد نمودار یک ژن، تابع `plotDEXSeqSimple()` نیازمند نام exonCountSet ، نام ژنی که ترسیم می‌شود (به صورت یک بردار ویژگی) و یک شاخص منطقی برای درج یا عدم درج یک علامت[1] است. دستور زیر نموداری را که در نگاره‌ی ۹-۳ ارائه شده است، ایجاد می‌نماید:

```
plotDEXSeqSimple(ec, "ENSG00000226742", legend = TRUE)
```

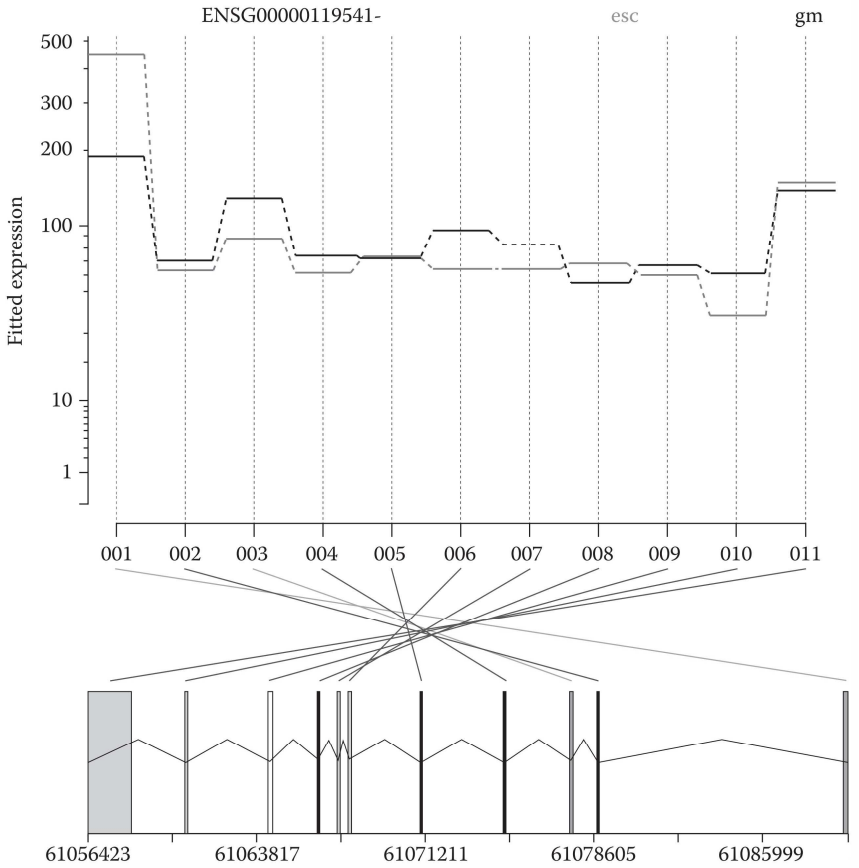
1- Legend



نگاره‌ی ۹-۳: نمودار نتایج حاصل از بسته‌ی DEXSeq. در بخش بالایی، بیان اگزون‌های ژن ENSG00000226742 در دو شرایط (esc و gm) با دو خط نشان داده شده است. هر اگزون به طور واضحی در سلول‌های gm در مقایسه با سلول‌های esc بیشتر بیان شده است. بنابراین این نتیجه احتمالاً ناشی از بیان متفاوت کُل ژن است. جدول نتایج نیز موید این موضوع است (`res[res$geneID == "ENSG00000226742",]`). زیرا هیچ‌یک از اگزون‌ها از نظر آماری بیان متفاوت معنی‌داری ندارند. اگزون‌ها به صورت میله‌های خاکستری و اینترون‌ها به صورت خط‌چین‌های سیاه در بین اگزون‌ها نمایش داده شده‌اند.

ژن ENSG00000119541 مثالی از یک ژن می‌باشد که حاوی یک اگزون با بیان متفاوت است. نتایج ترسیم شده برای این ژن با استفاده از کُد زیر، در نگاره‌ی ۹-۴ نمایش داده شده است.

```
plotDEXSeqSimple(ec, "ENSG00000119541", legend=TRUE)
```



نگاره‌ی ۹-۴: نتایج مربوط به ژن ENSG00000119541. در بین رده‌های سلولی esc و gm، بیان اگزون ۱ به طور واضحی متفاوت است. بیان اگزون دیگر نیز از نظر آماری به طور معنی‌داری متفاوت است. آیا می‌توانید حدس بزنید که این اگزون، کدام است؟ (پاسخ: اگزون شماره‌ی ۳).

علاوه بر تک نمودار، با کمک یک دستور ساده نیز می‌توان نمودارهایی برای کلیه‌ی نتایج در فرمت HTML ایجاد نمود:

```
DEXSeqHTML(ec, as.character(unique(res[ind,1])))
```

این دستور یک دایرکتوری (فولدر) در دایرکتوری فعال جاری ایجاد می‌کند. این فولدر حاوی یک صفحه‌ی وب خلاصه و یک فولدر فرعی حاوی نمودارهای جداگانه مختص ژن است. نخستین برهان تابع ، نام شیء exonCountSet بوده و دومین برهان نیز برداری از نام ژن‌هایی که بایستی ترسیم گردند، است. صفحه‌ی خلاصه در نگاره‌ی ۹-۵ نمایش داده شده است.

DEXSeq differential exon usage test

Experimental design

sample	samplename	condition
chip.sample001.tsv	Gm12892_1	gm
chip.sample002.tsv	Gm12892_2	gm
chip.sample003.tsv	Gm12892_3	gm
chip.sample004.tsv	hESC_1	esc
chip.sample005.tsv	hESC_2	esc
chip.sample006.tsv	hESC_3	esc
chip.sample007.tsv	hESC_4	esc

formulaDispersion = count ~ sample + condition * exon

formula0 = count ~ sample + exon + condition

formula1 = count ~ sample + exon + condition * I(exon == exonID)

testForDEU result table

geneID	chr	start	end	total_exons	exon_changes
ENSG00000017797	chr18			20	3
ENSG00000049759	chr18			38	6
ENSG00000060069	chr18			15	1
ENSG00000067900	chr18			43	5
ENSG00000074695	chr18	56995055	57027194	13	3

نگاره‌ی ۹-۵: صفحه‌ی خلاصه‌ی گزارش HTML ایجاد شده توسط بسته‌ی DEXSeq . اطلاعات نمونه‌ها و مدل‌های دقیقی که با این داده‌ها برازش یافته‌اند، نمایش داده شده است. جدول موجود در زیر صفحه حاوی اطلاعات هر ژن به طور جداگانه بوده و نخستین ستون هر ردیف از این جدول نیز حاوی یک لینک به صفحه‌ای است که جزییات بیشتری از آن ژن را در اختیار می‌گذارد.

آنالیز افتراقی بیان اگزون در Chipster

- خوانش‌ها را با استفاده از ابزار RNA-seq/Count aligned reads per exons for DEXSeq شمارش کرده و مطابق روش تشریح شده در فصل ششم، فایل‌های خوانش برای کلیه‌ی نمونه‌ها را با استفاده از ابزار Utilities/Defne NGS experiment در قالب یک جدول شمارش تلفیق نمایید. این ابزار علاوه بر یک جدول شمارش، یک فایل فنودیتا نیز ایجاد می‌کند که امکان توصیف شرایط و تنظیمات آزمایش را فراهم می‌آورد. با کمک ویرایشگر فنودیتا، کلیه‌ی نمونه‌های متعلق به یک گروه را با یک عدد یکسان در ستون گروه (group) مشخص نمایید.
- جدول شمارش و ابزار RNA-seq/Differential exon expression using DEXSeq را انتخاب کرده و در پارامترها، جاندار و آستانه‌ی مقدار p مورد نظر را تنظیم کرده و روی Run کلیک کنید.
- نتایج آزمون آماری در دو فایل گزارش می‌شود: یک فایل برای کلیه‌ی ژن‌ها و دیگری برای ژن‌هایی که اگزون‌های با بیان متفاوت دارند. فایل‌های نتایج نیز شامل یک نمودار MA ، یک نمودار پراکنش و مصورسازی ژن‌های دارای اگزون‌های با بیان متفاوت هستند.

۹–۸ خلاصه

یک ژن می‌تواند با استفاده از پروموتورهای دیگر و پیرایش‌های دیگر، چندین ایزوفرم رونوشت ایجاد نماید. توالی‌یابی RNA امکانات خوبی برای مطالعه‌ی بیان و تنظیم ایزوفرم‌ها در سطح کُل ژنوم فراهم می‌آورد. ولی اسمبل کردن و برآورد فراوانی ایزوفرم‌های رونوشت بر اساس اگزون‌های مشترک و پوشش رونوشت غیریکنواخت، کاری پیچیده است. یک روش برای اجتناب از عدم قطعیت ناشی از اسمبل کردن رونوشت در مطالعه‌ی تنظیم ایزوفرم‌های دیگر، بررسی تفاوت‌ها در استفاده از اگزون‌های جداگانه است.

بسته‌ی DEXSeq این امکان را برای کاربران محیا می‌کند که تفاوت در استفاده از اگزون را با یک مدل آماری برازش داده شده با داده‌ها و با کمک یک توزیع دو جمله‌ای منفی آزمون نمایند. برخی از ویژگی‌های DEXSeq مشابه بسته‌ی DESeq بوده و برخی از ویژگی‌های آن نیز با بسته‌ی edgeR مشابهت دارد. همچنین DEXSeq حاوی توابعی برای ترسیم نتایج آنالیز است. پس از انجام آنالیز توسط DEXSeq ، می‌توان نتایج را با استفاده از توابع ارائه شده توسط پروژه‌ی Bioconductor حاشیه‌نگاری و مصورسازی نمود.

منابع

1. Anders S. HTSeq documentation website, 2012. http://www-huber.embl.de/users/anders/HTSeq/doc/overview.html (Accessed 17 January 2014).
2. Anders S., Reyes A., and Huber W. Detecting differential usage of exons from RNA-seq data. Genome Res 22:2008, 2012. doi:10.1101/gr.133744.111
3. Katz Y., Wang E.T., Airoldi E.M., and Burge C.B. Analysis and design of RNA sequencing experiments for identifying isoform regulation. Nat Methods 7:1009–1015, 2010.

فصل دهم

حاشیه‌نگاری نتایج

۱۰–۱ مقدمه

عموماً حاشیه‌نگاری عبارت از هر نوع نظر، توضیح یا علامت‌گذاری است (مجموعاً تحت عنوان فراداده[1] شناخته می‌شوند) که به داده‌ها پیوست می‌گردد. فراداده‌ها اغلب به بخش‌های خاصی از داده‌ها پیوست می‌شوند. ولی می‌توانند نحوه، مکان یا زمان جمع‌آوری داده‌ها را نیز تشریح کنند. با این حال معنای حاشیه‌نگاری در بیوانفورماتیک، تخصصی‌تر است. حاشیه‌نگاری ژنوم به معنای فرآیند شناسایی و موقعیت‌یابی ژن‌ها و سایر عناصر عملکردی ژنوم جاندار بوده و در این فرآیند نکاتی از عملکردهای‌شان به آنها پیوست می‌گردد. معمولاً حاشیه‌نگاری‌هایی که مرتبط با یک ژن هستند، شامل موقعیت ژنومی (کروموزوم، نوار کروموزومی[2] و تعداد جفت باز)، ساختار اگزونی و اینترونی، رونوشت‌ها و برخی از نکات عملکردی ممکن در قالب واژگان کنترل شده نظیر هستی‌شناسی ژن[3] (GO) یا مسیرهای متابولیسمی که پروتئین‌های ترجمه شده در آن عمل می‌کنند (نظیر KEGG (دائره‌المعارف کیوتو در زمینه‌ی ژن‌ها و ژنوم‌ها[4]) و Reactome (پایگاه داده‌ی واکنش‌ها، مسیرها و فرآیندهای زیستی)) هستند.

معمولاً استفاده از حاشیه‌نگاری‌ها در یک آزمایش توالی‌یابی RNA دو برابر است. اگر مکان‌یابی خوانش‌ها با استفاده از یک ژنوم مرجع صورت گرفته باشد، حاشیه‌نگاری‌ها در این مرحله‌ی مکان‌یابی برای تخصیص خوانش‌ها به ژن‌ها یا رونوشت‌های صحیح به کار گرفته می‌شوند. معمولاً پس از آنالیز داده‌ها، حاشیه‌نگاری‌های مفصل‌تر برای رونوشت‌های مورد نظر یا رونوشت‌هایی که به طور متفاوت بیان شده‌اند، بازیابی[5] گردیده و بدین ترتیب می‌توان یک مفهوم زیستی را برای نتایج مشاهده شده تایید و تصدیق نمود. به عنوان مثال، استفاده از اطلاعات عملکردی ژن‌ها و رونوشت‌ها، نظیر دسته‌های هستی‌شناسی ژن، امکان آنالیز دقیق‌تر مسیرهای متابولیکی با افزایش بیان[6] یا کاهش بیان[7] را فراهم می‌آورند.

1- Metadata
2- Cytoband
3- Gene Ontology (GO)
4- Kyoto Encyclopedia of Genes and Genomes (KEGG)
5- Retrieve
6- Upregulated
7- Downregulated

در این فصل نحوه‌ی بازیابی حاشیه‌نگاری‌های اضافه (حاشیه‌نگاری مجدد) برای ژن‌های دارای بیان متفاوت و تعدادی از آنالیزهایی که به واسطه‌ی این حاشیه‌نگاری‌های جدید امکان‌پذیر می‌گردند، بحث و بررسی می‌شوند.

۱۰-۲ بازیابی حاشیه‌نگاری‌های اضافه

پایگاه‌های داده‌ی متعددی وجود دارند که حاوی حاشیه‌نگاری‌هایی برای ژن‌ها، رونوشت‌ها و محصولات‌شان هستند. در حال حاضر اغلب حاشیه‌نگاری‌ها از پایگاه‌های داده‌ی اولیه‌ی جداگانه نظیر GenBank (http://www.ncbi.nlm.nih.gov/genbank/)، در پایگاه‌های ثانویه و احتمالاً در قابل توجه‌ترین پایگاه‌های ژنومی نظیر UCSC Genome Browser (http://genome.ucsc.edu/) و Ensembl (http://www.ensembl.org/) گردآوری می‌شوند. Biomart (http://www.biomart.org/) یک رابط کاربری مناسب برای اصلاح اطلاعات حاصل از Ensembl و بیش از ۴۰ پایگاه داده‌ی دیگر دارد.

افزودن حاشیه‌نگاری‌های حاصل از این پایگاه‌ها و منابع متعدد دیگر به ژن‌های دارای بیان متفاوت هم از طریق جستجوی هر پایگاه داده به صورت جداگانه و سپس اسمبل کردن یک مجموعه‌ی داده‌ی جامع به صورت دستی و هم از طریق استفاده از بسته‌های آماده و در دسترس Bioconductor که تنها در پایگاه‌های داده‌ی خاصی جستجو را انجام می‌دهند (معمولاً پایگاه‌هایی که اطلاعات پایه از رونوشت‌ها و محصولات پروتئینی آنها دارند)، امکان‌پذیر است.

معمولاً هر ژن یا رونوشت دارای یک شماره‌ی دسترسی است. در برخی از پایگاه‌های داده این شماره‌ی دسترسی به یک نکته‌ی خاص اشاره می‌کند. به عنوان مثال، هر ژن دارای یک شماره‌ی دسترسی جداگانه و واضح در پایگاه داده‌ی Ensembl است. در پایگاه داده‌ی Ensembl شماره‌ی دسترسی مختص ژن با عبارت ESGN و شماره‌ی دسترسی مختص رونوشت با ENST شروع می‌شود. علاوه بر شماره‌ی دسترسی، جستجو در پایگاه‌های داده با استفاده از توالی ژن یا رونوشت (نظیر آنچه که در جستجوی BLAST انجام می‌شود)، نیز امکان‌پذیر است. یک سرویس سفارشی تحت عنوان Blast2GO (http://www.blast2go.com/b2ghome) وجود دارد که حاشیه‌نگاری عملکردی مناسب را برای توالی‌های شناخته شده یا توالی‌هایی که از قبل ناشناخته هستند، می‌یابد. گاهی اوقات بهترین کار استفاده از نام ژن‌ها برای یافتن حاشیه‌نگاری‌ها است. ولی متاسفانه این مسیر به ندرت خیلی ساده است. ژن‌ها می‌توانند چندین نام داشته باشند که مورد استفاده نیز واقع شوند. حتی در مورد ژن‌های انسانی که نام‌های‌شان استاندارد شده است نیز معمولاً هنوز چندین نام مترادف برای هر ژن به کار گرفته می‌شود. به دلیل ابهام در راهبردهای نام‌گذاری،

مقایسات بین گونه‌ای بر مبنای نام ژن‌ها معمولاً چندان مورد اعتماد نیستند. به طور خلاصه، در صورت موجود بودن شماره‌ی دسترسی، بهترین کار در هنگام جستجو برای اطلاعات اضافی، استفاده از شماره‌ی دسترسی است.

استفاده از شماره‌ی دسترسی برای حاشیه‌نگاری اطلاعات ژن و رونوشت، نسبتاً ساده است. گاهی اوقات لازم است که شماره‌ی دسترسی از مبنای قوانین یک پایگاه داده به پایگاه داده‌ی دیگر ترجمه شود که این امر سبب می‌شود که بخشی از داده‌ها از دست بروند. زیرا معمولاً همه‌ی شماره‌های دسترسی جفت‌شدگی مستقیمی در سایر پایگاه‌های داده ندارند. به عنوان مثال، هنوز تعداد برآورد شده‌ی ژن‌های انسان در مرورگرهای ژنومی Ensembl و UCSC متفاوت هستند. این موضوع ناشی از تفاوت اندک در فرآیندهای حاشیه‌نگاری ژنومی است.

در اینجا برخی اطلاعات اضافی برای ژن‌هایی که در فصل ۹ تشخیص داده شدند که بیان متفاوتی دارند، بازیابی می‌گردد. این اطلاعات جدید با کمک بسته‌های Bioconductor در دسترس قرار می‌گیرند.

۱۰-۲-۱ استفاده از یک بسته‌ی حاشیه‌نگاری مختص جاندار برای بازیابی حاشیه‌نگاری‌های ژن‌ها

یکی از ساده‌ترین روش‌ها برای افزودن حاشیه‌نگاری‌های جدید به ژن‌ها، گرفتن آنها از یک بسته‌ی حاشیه‌نگاری مختص جاندار در R است. این بسته برای انسان، org.Hs.eg.db نام داشته و بسته‌های مشابه دیگری نیز برای جانداران مدل دیگر وجود دارد که همگی آنها دارای پیشوند org می‌باشند. بسته‌های مختص جاندار امکان تبدیل شماره‌های دسترسی معین به شماره‌های دسترسی سایر پایگاه‌های داده را فراهم کرده و نیز حاوی اطلاعات پایه در مورد موقعیت ژن در ژنوم، حاشیه‌نگاری‌های عملکردی آن با استفاده از دسته‌های هستی‌شناسی (GO) ژن و غیره هستند. در اینجا دسته‌های GO برای ژن‌هایی که در فصل نهم به عنوان ژن‌های با بیان متفاوت تشخیص داده شده‌اند، یافت می‌گردند.

بسته‌ی مختص انسان بر مبنای شماره‌های دسترسی Entrez Gene بوده ولی حاوی ID های Ensembl نیز می‌باشد. ژن‌هایی که بیان متفاوت داشته و در فصل نهم شناسایی شده‌اند، دارای ID های Ensembl هستند. بنابراین قبل از پیوند دسته‌های هستی‌شناسی به این ژن‌ها، لازم است که ID های Ensembl به ID های Entrez Gene تبدیل شده و سپس با استفاده از این شماره‌های دسترسی جدید، بازیابی دسته‌های GO از بسته‌ی حاشیه‌نگاری صورت گیرد. در این فرآیند تا حدودی از دست رفتن اطلاعات نیز قابل انتظار است.

شیء res حاوی نتایج آنالیز افتراقی بیان که در فـصل نهم اجرا شده است، می‌باشد. این شیء دارای چندین ردیف به ازای هر ژن است. زیرا آنالیز آماری برای اگزون‌ها انجام شده است (نه برای ژن‌ها). ابتدا ژن‌هایی که از نظر آماری با تفاوت معنی‌داری بیان شده‌اند، در یک شیء با عنوان res2 استخراج گردیده و سپس کلیه‌ی مشخصه‌های انحصاری Ensembl ژن‌هایی که از آنها مجموعه‌ی نتایج به دست آمده است، در شیء deg استخراج می‌شوند:

```
res2<-res[!is.na(res$padjust) & res$padjust<=0.05,]
deg<-as.character(unique(res2$geneID))
```

مجموعاً ۸۷ ژن با بیان متفاوت در این مجموعه وجود دارند. تعداد واقعی ممکن است اندکی متفاوت باشد. ولی با همین ویرایش R و بسته‌های مربوط به آن روی سامانه‌ی ویندوز 7 بایستی همین تعداد ژن ایجاد شود. با استفاده از محیط org.Hs.egENSEMBL2EG از بسته‌ی مختص انسان org.Hs.eg.db می‌توان مشخصه‌های این ژن‌ها را به مشخصه‌های Entrez Gene تبدیل نمود. این بسته حاوی طرحی از تبدیل ID های Ensembl ژن‌ها به ID های Entrez Gene است.

کلیه‌ی زمینه‌های موجود در بسته‌ی حاشیه‌نگاری را می‌توان با استفاده از دستور ls() کنترل نمود:

```
library("org.Hs.eg.db")
ls("package:org.Hs.eg.db")
```

```
 [1] "org.Hs.eg"                "org.Hs.eg.db"
 [3] "org.Hs.eg_dbconn"         "org.Hs.eg_dbfile"
 [5] "org.Hs.eg_dbInfo"         "org.Hs.eg_dbschema"
 [7] "org.Hs.egACCNUM"          "org.Hs.egACCNUM2EG"
 [9] "org.Hs.egALIAS2EG"        "org.Hs.egCHR"
[11] "org.Hs.egCHRLENGTHS"      "org.Hs.egCHRLOC"
[13] "org.Hs.egCHRLOCEND"       "org.Hs.egENSEMBL"
[15] "org.Hs.egENSEMBL2EG"      "org.Hs.egENSEMBLPROT"
[17] "org.Hs.egENSEMBLPROT2EG" "org.Hs.egENSEMBLTRANS"
[19] "org.Hs.egENSEMBLTRANS2EG""org.Hs.egENZYME"
[21] "org.Hs.egENZYME2EG"       "org.Hs.egGENENAME"
[23] "org.Hs.egGO"              "org.Hs.egGO2ALLEGS"
[25] "org.Hs.egGO2EG"           "org.Hs.egMAP"
[27] "org.Hs.egMAP2EG"          "org.Hs.egMAPCOUNTS"
[29] "org.Hs.egOMIM"            "org.Hs.egOMIM2EG"
[31] "org.Hs.egORGANISM"        "org.Hs.egPATH"
[33] "org.Hs.egPATH2EG"         "org.Hs.egPFAM"
[35] "org.Hs.egPMID"            "org.Hs.egPMID2EG"
[37] "org.Hs.egPROSITE"         "org.Hs.egREFSEQ"
[39] "org.Hs.egREFSEQ2EG"       "org.Hs.egSYMBOL"
[41] "org.Hs.egSYMBOL2EG"       "org.Hs.egUCSCKG"
[43] "org.Hs.egUNIGENE"         "org.Hs.egUNIGENE2EG"
[45] "org.Hs.egUNIPROT"
```

۴۵ زمینه‌ی مختلف در این بسته موجود بوده و اینکه کدام‌یک از آنها مورد استفاده واقع شوند، طبیعتاً به اطلاعات مورد نظر بستگی دارد. هر کدام از این زمینه‌ها یک صفحه‌ی راهنما دارند. در اینجا زمینه‌ی org.Hs.egENSEMBL2EG انتخاب می‌شود. زیرا این زمینه حاوی بیشترین حجم اطلاعات حاشیه‌نگاری GO برای این ژن‌ها است.

در ادامه‌ی گردش کار، نخست اطلاعات مکان‌یابی از بسته در شیء xx استخراج شده و سپس تبدیل‌های بعدی در چارچوب داده‌ی xxd صورت می‌گیرد:

```
xx <- as.list(org.Hs.egENSEMBL2EG)
xxd <- as.data.frame(unlist(xx))
```

محتویات شیء xxd بایستی بدین صورت باشد:

```
head(xxd)
```

```
                    unlist(xx)
ENSG00000121410              1
ENSG00000175899              2
ENSG00000256069              3
ENSG00000171428              9
ENSG00000156006             10
ENSG00000196136             12
```

ردیف‌هایی که حاوی ID های Entrez Gene برای ژن‌های با بیان متفاوت هستند، می‌توانند پس از ایجاد یک شاخص برای بخش‌هایی که لازم است استخراج شوند و با استفاده از یک تنظیم فرعی ساده استخراج گردند:

```
ind<-match(deg, rownames(xxd))
degeg<-xxd[ind,]
```

اساساً ۸۷ ژن با بیان متفاوت وجود دارند ولی تنها برای ۷۴ ژن ID های Entrez Gene یافت شدند. در مرحله‌ی بعد می‌توان برای دسته‌های GO مربوط به این ۷۴ ژن جستجو نمود. اطلاعات دسته‌های GO در محیط org.Hs.eg.dbGO در بسته‌ی org.Hs.eg.db موجود هستند. مشابه کاری که برای مشخصه‌ها انجام شد، ابتدا دسته‌های GO در یک چارچوب داده استخراج شده و سپس با یک شاخص تنظیم گردیده و نتیجه‌ی نهایی در یک فهرست degeggo ذخیره می‌گردد:

```
xx <- as.list(org.Hs.egGO)
ind<-match(degeg, names(xx))
degeggo<-xx[ind]
```

مشکلی که در آنالیزهای پایین دستی با دسته‌های GO ایجاد می‌شود این است که یک ژن می‌تواند با چندین دسته‌ی GO مکان‌یابی گردد و تحکیم این نتایج در قالب یک جدول ساده یا یک چارچوب داده که دست‌ورزی آن آسان باشد، همواره ساده نخواهد بود.

با این حال یک راه ساده برای تنظیم نتایج در قالب یک جدول وجود دارد. بسته‌ی AnnotationDbi حاوی تابع `select()` بوده که می‌تواند برای انتخاب حاشیه‌نگاری‌های حاصل از بسته‌های حاشیه‌نگاری مختص جاندار به کار گرفته شود. این تابع نیازمند چهار برهان شامل: نام بسته‌ی حاشیه‌نگاری، شماره‌های دسترسی که بایستی مورد جستجو واقع گردند، زمینه‌هایی که باید استخراج شوند و نام بسته‌ی حاشیه‌نگاری است. زمینه‌های در دسترس بسته را می‌توان با دستور `keytypes()` یافت. به عنوان مثال:

```
library(AnnotationDbi)
keytypes(org.Hs.eg.db)
```

```
 [1] "ENTREZID"    "PFAM"        "IPI"         "PROSITE"     "ACCNUM"
 [6] "ALIAS"       "CHR"         "CHRLOC"      "CHRLOCEND"   "ENZYME"
[11] "MAP"         "PATH"        "PMID"        "REFSEQ"      "SYMBOL"
[16] "UNIGENE"     "ENSEMBL"     "ENSEMBLPROT" "ENSEMBLTRANS" "GENENAME"
[21] "UNIPROT"     "GO"          "EVIDENCE"    "ONTOLOGY"    "GOALL"
[26] "EVIDENCEALL" "ONTOLOGYALL" "OMIM"        "UCSCKG"
```

هر زمینه‌ای را می‌توان با هر کدام از زمینه‌های دیگر مکان‌یابی نمود. بنابراین می‌توان ژن‌های با بیان متفاوت را هم توسط ID های Entrez Gene و هم توسط ID های Ensembl Gene با دسته‌های GO مکان‌یابی کرد. برای این کار می‌توان از هر کدام از دستورات زیر استفاده نمود:

```
degeggo<-select(org.Hs.eg.db, as.character(degeg),
                    "GO", keytype = "ENTREZID")
degeggo<-select(org.Hs.eg.db, as.character(deg),
                    "GO", keytype = "ENSEMBL")
```

خروجی حاصل از این دستور احتمالاً چند ردیف را برای هر ژن بازیابی می‌کند (یک دسته‌ی GO به ازای هر خط):

```
head(degeggo)
           ENSEMBL         GO    EVIDENCE     ONTOLOGY
1   ENSG00000017797 GO:0005096        IDA           MF
2   ENSG00000017797 GO:0005515        IPI           MF
3   ENSG00000017797 GO:0005829        TAS           CC
4   ENSG00000017797 GO:0006200        IDA           BP
5   ENSG00000017797 GO:0006810        IDA           BP
6   ENSG00000017797 GO:0006935        TAS           BP
```

هر دسته‌ی GO نیز با یک کُد مستند[1] که در http://www.geneontology.org/GO.evidence .shtml قابل تایید است، مرتبط می‌باشد. از موارد نشان داده شده در فوق، هم IDA و هم IPI به صورت آزمایشی قابل تایید و اعتبارسنجی بوده و TAS بر مبنای توضیح نویسنده قابل ردیابی است. هر دسته نیز متعلق به یکی از سه دسته‌ی هستی‌شناسی، شامل فرآیند زیستی (BP) عملکرد مولکولی (MF) و مولفه‌ی سلولی (CC) که در ستون ONTOLOGY فهرست شده‌اند، می‌باشد. در حال حاضر، این داده‌ها در قالب مورد نیاز برای آنالیزهای بیشتر هستند.

اگر بخواهید بعداً برخی از آنالیزهای مقایسه‌ای را برای پی بردن به اینکه آیا برخی از دسته‌های GO در ژن‌های با بیان متفاوت غنی می‌شوند یا خیر، انجام دهید، لازم است که یک فهرست ژنی کُلی که حاوی کلیه‌ی ژن‌های حاصل از کروموزوم ۱۸ (آزمایش ENCODE تنها شامل داده‌های کروموزوم ۱۸ بوده است) و حاشیه‌نگاری‌های GO آنها ایجاد کنید. شیء `res` (که در فصل نهم ایجاد گردید) حاوی ژن‌های با بیان متفاوت بوده و بنابراین شما می‌خواهید حاشیه‌نگاری‌های GO را با اطلاعات کروموزومی برای این ژن‌ها بازیابی کنید. بدین ترتیب ژن‌هایی که روی کروموزوم ۱۸ واقع نشده یا تخصیص کروموزومی آنها مفقود شده است پاکسازی گردیده و ستون اطلاعات کروموزومی از جدول حذف می‌شود:

```
univ<-as.character(unique(res$geneID))
univgo<-select(org.Hs.eg.db, univ, c("CHR", "GO"),
                  keytype="ENSEMBL")
univgo<-univgo[univgo$CHR=="18" & !is.na(univgo$CHR),
                  c(1, 3:5)]
```

۱۰-۲-۲ استفاده از BioMart برای بازیابی حاشیه‌نگاری‌های ژن‌ها

بسته‌ی biomaRt از پروژه‌ی Bioconductor یک رابط کاربری برای کار با پایگاه داده‌ی BioMart ارائه می‌کند. قبل از هر جستجویی، لازم است که یک پایگاه داده انتخاب شود. فهرستی از کلیه‌ی مارت‌ها[2] (پایگاه‌های داده) را می‌توان با دستور زیر ایجاد نمود:

```
library(biomaRt)
listMarts()
```

در بین مارت‌ها بایستی به Ensembl نگاه کنید. زیرا می‌خواهید حاشیه‌نگاری‌هایی برای ژن‌های انسان را بازیابی کنید. سپس باید با کمک تابع `useMart()` به پایگاه داده مرتبط گردید. این تابع نیازمند دریافت نام مارت در قالب یک برهان است:

```
ensembl=useMart("ensembl")
```

1- Evidence code
2- Mart

پس از انتخاب پایگاه داده، لازم است که یک مجموعه‌ی داده نیز انتخاب گردد. برای Ensembl این مجموعه‌های داده بر مبنای جانداران تفکیک شده‌اند. مجموعه‌ی داده‌های در دسترس را می‌توان با تابع زیر فهرست نمود:

```
listDatasets():
listDatasets(ensembl)
```

برای انسان، مجموعه‌ی داده‌ها با عنوان hsapiens_gene_ensembl نامیده شده و در هنگام چاپ این کتاب، ویرایش GRCh37.p12 آن در دسترس بوده است. پس از انتخاب مجموعه‌ی داده‌ها، می‌توان شیء ensembl را به‌روزرسانی نمود تا این داده‌ها را نیز در بر بگیرد:

```
ensembl = useDataset("hsapiens_gene_ensembl",
                     mart= ensembl)
```

وقتی که ارتباط با پایگاه داده به درستی برقرار شد، لازم است که اطلاعات حاشیه‌نگاری انتخاب شده و به طور مشابه در صورت لزوم چند فیلتر بایستی به کار گرفته شود. به عنوان مثال، شماره‌ی دسترسیِ Ensembl ، کروموزوم و دسته‌های GO برای هر ژنِ دارای بیان متفاوت دانلود خواهند شد. مجموعه‌ی این موارد اصطلاحاً ویژگی‌ها[1] نامیده می‌شوند. علاوه بر این می‌توان جستجو را تنها به کروموزوم ۱۸ محدود نمود. ویژگی‌ها و فیلترهای در دسترس را می‌توان با توابع listAttributes() و listFilters() فهرست نمود. چون صدها ویژگی و فیلتر وجود دارد، لذا برای آسانی کار در دو شیء جداگانه ذخیره می‌شوند:

```
filters = listFilters(ensembl)
attributes = listAttributes(ensembl)
```

ویژگی‌هایی که لازم است مورد استفاده قرار گیرند عبارتند از: ensembl_gene_id ، chromosome_name و go_id . همچنین از chromosome_name به عنوان یک فیلتر نیز استفاده می‌شود. زیرا هدف این است که تنها حاشیه‌نگاری‌های مربوط به ژن‌های روی کروموزوم ۱۸ دریافت گردند. جستجو در پایگاه داده با تابع getBM() انجام می‌شود. این تابع چهار برهان دارد: ویژگی‌ها (attributes)، فیلترها (filters)، مقادیر (values) و مارت (mart). ویژگی‌ها و فیلترها قبلاً تشریح شده‌اند. برهانِ مقادیر فهرستی از شماره‌های دسترسی که در حال حاضر در شیء deg حضور دارند، و همچنین نام کروموزومی را که به عنوان یک فیلتر استفاده شده و در شیء chrom ذخیره شده است، را دریافت می‌دارد. مارت نیز همان شیء ensembl که در بالا ساخته شده است، می‌باشد. بدین ترتیب جستجو در شکل کُلی‌اش عبارت است از:

1- Attribute

```
chrom<-c(18)
query<-getBM(attributes=c("ensembl_gene_id",
                          "chromosome_name",
                          "go_id"),
             filters="chromosome_ name",
             values=list(deg, chrom),
             mart=ensembl)
```

وقتی که جستجو انجام شد، R به پرومپت باز می‌گردد و می‌توان ردیف‌ها را از ابتدای شیء query حاصل آمده بررسی نمود:

```
head(query)
 ensembl_gene_id  chromosome_name       go_id
1ENSG00000101574               18  GO:0006139
2ENSG00000101574               18  GO:0003676
3ENSG00000101574               18  GO:0008168
4ENSG00000101574               18
5ENSG00000154065               18  GO:0005515
6ENSG00000080986               18  GO:0008608
```

برای هر ژن چند خط می‌تواند وجود داشته باشد. زیرا همان‌گونه که در بالا در استفاده از بسته‌های مختص جاندار نیز اشاره شد، ممکن است که چند دسته‌ی GO به یک ژن تخصیص داده شوند و هر کدام از آنها نیز خط اختصاصی خود را اشغال خواهند نمود. ۸۷ ژن در فهرست اصلی ژن‌ها وجود داشته که بر همان مبنای شیء deg مرتب شده‌اند. اگر ژن‌های منحصر به فردی را که جستجو برگردانده است، کنترل کنید، ملاحظه خواهید کرد که بیشتر ژن‌های برگردانده شده در فهرست ژن‌های اصلی حضور دارند:

```
length(unique(query$ensembl_gene_id))
[1]289
```

این موضوع نشان می‌دهد که اگر بخواهید تنها نتایج مربوط به ژن‌های با بیان متفاوت را دریافت کنید، لازم است که آنها را به صورت یک فیلتر در جستجو اعمال نمایید. به عنوان مثال:

```
query<-getBM(attributes=c("ensembl_gene_id",
                          "chromosome_name",
                          "go_id"),
             filters=c("ensembl_gene_ id",
                       "chromosome_name"),
             values=list(deg,chrom),
             mart=ensembl)
length(unique(query$ensembl_gene_id))
[1]72
```

می‌توان همین دستور را برای دریافت حاشیه‌نگاری‌های کلیه‌ی ژن‌های تخصیص یافته به کروموزم ۱۸ اجرا نمود:

```
query2<-getBM(attributes=c("ensembl_gene_id",
                           "chromosome_name",
                           "go_id"),
               filters=c("chromosome_ name"),
               values=list(chrom),
               mart=ensembl)
```

باید توجه شود که لازم است که فیلترها با همان ترتیب مقادیر مرتب شوند. در اینجا، ابتدا ID های ژنی Ensembl قرار داشته و سپس در هر دو برهان، نام کروموزوم قرار گرفته است.

۱۰–۳ استفاده از حاشیه‌نگاری‌ها برای آنالیز هستی‌شناسی مجموعه‌های ژنی

معمولاً چند اصطلاح برای روش‌هایی که غنی‌شدگی یا فزون‌نمایی ژن‌ها در یک آزمایش پروفایل‌بندی بیان در برخی دسته‌های معنی‌دار زیستی (نظیر دسته‌های هستی‌شناسی GO ، مسیرهای (متابولیکی) زیستی، یا سایر گروه‌های عملکردی) را بررسی می‌کنند، وجود دارد. اصطلاحاتی که به طور معمول استفاده می‌شوند عبارتند از: آنالیز مجموعه‌ی ژن[1] (GSA)، آنالیزغنی‌سازی ژن[2]، یا آنالیز هستی‌شناختی[3]. این روش‌ها در چندین مقاله بررسی و مرور شده‌اند (۱ ، ۲ و ۳).

در اغلب موارد یک جدول ۲ × ۲ برای آزمون فزون‌نمایی به کار گرفته می‌شود. برای کُل آزمایش، نخست ژن‌ها هم به گروه‌های فزون‌بیان[4] (فرابیان) و هم به گروه‌های غیرفزون‌بیان تخصیص داده می‌شوند. سپس هر دو گروه به بخش‌هایی که در یک گروه عملکردی معین قرار دارند یا قرار ندارند، تقسیم می‌گردند. بدین ترتیب می‌توان یک جدول ساده تشکیل داد که تعداد ژن‌ها در هر خانه‌ی این جدول با حروف a تا d مشخص می‌شوند:

	فزون‌بیان شده	غیرفزون‌بیان شده
در گروه عملکردی	a	b
در غیر گروه عملکردی	c	d

1- Gene Set Analysis (GSA)
2- Gene enrichment analysis
3- Ontological analysis
4- Overexpressed

در اینجا یک مثال واقعی با استفاده از حاشیه‌نگاری‌های به دست آمده در اشیاء query و query2 ارائه می‌شود. به طور کُلی ۷۲ ژن با بیان متفاوت و مجموعاً ۷۲۸۹ ژن در کروموزوم ۱۸ وجود دارد. این عدد تعداد واقعی ژن‌های موجود روی کروموزوم ۱۸ نبوده ولی تعداد بازگشت‌های[1] biomaRt را نشان می‌دهد. دسته‌ی 0005515 در GO به معنای یک پروتئین مرتبط با چاپرون بوده و کُد زیر تعداد ژن‌های مرتبط با این دسته را در هر دو فهرست ژنی (query و query2) شمارش می‌کند. نخست ردیف‌های جدول حاشیه‌نگاری که ژن‌ها را با حاشیه‌نگاری GO در دسته‌ی 0005515 نگه داشته‌اند، یافت شده و سپس آنها را در بردارهای شاخص جداگانه ذخیره می‌کند:

```
indq<-which(query$go_id=="GO:0005515")
indq2<-which(query2$go_id=="GO:0005515")
```

با کمک این شاخص‌ها می‌توان تعداد ژن‌های متفاوت در هر دو فهرست ژنی که برای آن دسته حاشیه‌نگاری شده‌اند، را شمارش نمود. ستون ensembl_gene_id holds مشخصه‌های ژنی را که واضح باشند، نگه می‌دارد. سپس تنها انتخاب مشخصه‌های ژن‌های منحصر به فرد، این تعداد را نشان خواهد داد:

```
length(unique(query$ensembl_gene_id[indq])) # 44
length(unique(query2$ensembl_gene_id[indq2)) # 132
```

۱۳۲ ژن در کروموزوم ۱۸ با این دسته‌ی GO مرتبط بوده و ۴۴ ژن نیز بیان متفاوتی داشته‌اند. بدین ترتیب جدول ۲ × ۲ برای تعیین فزون‌نمایی عبارت است از:

	فزون‌بیان شده	**غیرفزون‌بیان شده**
در گروه عملکردی	۴۴	۱۳۲
در غیر گروه عملکردی	۷۲ - ۴۴ = ۲۸	۲۸۹ - ۱۳۲ = ۱۵۷

با کمک تابع matrix() می‌توان جدول فوق را به صورت یک ماتریس در R نمایش داد:

```
mat<-matrix(ncol=2, data=c(44,28,132,157))
mat
```

```
      [,1]   [,2]
[1,]    44    132
[2,]    28    157
```

1- Return

فزون‌نمایی در ساده‌ترین شکلش می‌تواند توسط آزمون دقیق فیشر مورد بررسی واقع شود. این آزمون در R با کمک تابع `fisher.test()` به سادگی برای یک ماتریس اجرا می‌گردد:

```
fisher.test(mat)
      Fisher's Exact Test for Count Data
data: mat
p-value = 0.02471
alternative hypothesis: true odds ratio is not equalto 1
95 percent confidence interval:
  1.069865 3.296576
sample estimates:
odds ratio
  1.86583
```

مقدار p برای این آزمون کمتر از مقداری است که به طور معمول برای معنی‌داری آماری (۰/۰۵) به کار گرفته می‌شود. بنابراین دسته‌ی GO پروتئین مرتبط با چاپرون در بین ژن‌های با بیان متفاوت، به صورت فزون‌نمایی شده ظاهر می‌گردد.

مثال بعدی یک دیدگاه نامناسب از نحوه‌ی انجام این آنالیز برای یک دسته‌ی عملکردی منفرد را ارائه می‌دهد. در عمل، این آنالیز بایستی برای هر دسته به صورت جداگانه انجام گیرد. آزمون فیشر که در بالا مورد بحث و بررسی قرار گرفت، تنها یکی از امکانات متعدد برای آنالیز مجموعه‌ی ژنی بوده و سایر روش‌ها با جزییات بیشتر در فصل بعد تشریح می‌گردند.

۱۰–۴ جزییات بیشتر از آنالیز مجموعه‌های ژنی

Goeman و Bühlmann (۲۰۰۵) روش‌های مختلف آنالیز را به دو دسته‌ی آزمون‌های رقابتی[1] و جامع[2] تقسیم کردند. در روش‌های جامع، نخست کلیه‌ی ژن‌های موجود در آزمایش به دو دسته‌تقسیم می‌شوند: ژن‌هایی که در یک گروه معین (نظیر یک دسته‌ی GO) قرار داشته و کلیه ژن‌های دیگر که در این گروه واقع نشده‌اند. سپس تعداد ژن‌های دارای بیان متفاوت در هر دو گروه با هم مقایسه شده و اگر نتیجه معنی‌دار باشد نشان دهنده‌ی آن است که این دسته‌ی GO فعال شده است. آزمون‌های جامع تنها از اطلاعات ژن‌ها در یک گروه معین (نظیر یک دسته‌ی GO) استفاده کرده و این فرض که هیچ‌یک از ژن‌های موجود در گروه مزبور به صورت متفاوت بیان نمی‌شوند را مورد آزمون قرار می‌دهد.

1- Competitive test
2- Self-contained test

علاوه بر این، Goeman و Bühlmann (۲۰۰۵) این آزمون‌ها را از دیدگاه واحد نمونه‌گیری نیز تقسیم می‌کنند. در آزمون‌های معمول که از جدول ۲ × ۲ استفاده می‌کنند، ژن (یعنی ژن‌هایی که در خانه‌های جدول تقسیم می‌گردند) به عنوان واحد نمونه‌گیری استفاده می‌شود. این وضعیت با روش‌های آماری متداول که در آنها از نمونه‌ها یا افراد به عنوان واحدهای نمونه‌گیری استفاده می‌شود، در تقابل و تضاد است. بر همین اساس Goeman و Bühlmann (۲۰۰۵) مدل‌های نمونه‌گیری ژنی[1] و نمونه‌گیری موضوعی[2] را از هم تفکیک می‌کنند. معمولاً روش‌های جامع مبتنی بر نمونه‌گیری موضوعی بوده و روش‌های رقابتی بر مبنای نمونه‌گیری ژنی پایه‌گذاری شده‌اند.

هم Goeman و Bühlmann (۲۰۰۵) و هم Maciejewski (۲۰۱۳) استدلال کرده‌اند که روش‌های جامع و مبتنی بر نمونه‌گیری موضوعی معتبرتر از بقیه‌ی روش‌ها بوده و تفسیرپذیری آسان‌تری نیز دارند. مثال‌هایی از الگوریتم‌هایی که از این نوع روش‌ها استفاده می‌کنند، عبارتند از: SAFE (۴) و Globaltest (۵). هر دو الگوریتم نیز توسط بسته‌های Bioconductor مورد استفاده قرار می‌گیرند.

روش‌های رقابتی در بسته‌های Bioconductor نظیر GOstats ، topGO و چند بسته‌ی دیگر در دسترس هستند که در اینجا تنها به برخی از آنها اشاره می‌شود. goseq که از بسته‌های Bioconductor است، از روشی که مبتنی بر تصحیحات برای طبیعت اُریبی احتمالی طول در داده‌های توالی‌یابی RNA است، استفاده می‌کند. بسته‌ی GSVA روشی را به کار می‌گیرد که غنی‌سازی نسبی مسیرها در کُل آزمایش را مورد بررسی قرار می‌دهد. بسته‌ی limma مشتمل بر روشی است که ابتدا داده‌های توالی‌یابی RNA را به یک مقیاس مشابه داده‌های ریزآرایه‌ی DNA (voom) تبدیل کرده و سپس از دو روش غنی‌سازی مجموعه‌ی ژنی تحت عنوان roast و camera استفاده می‌کند.

روش‌های آنالیز مختص توالی‌یابی RNA به طور وسیعی در دسترس نبوده و در حال حاضر شاید بهترین کار این باشد که داده‌های توالی‌یابی RNA مشابه داده‌های ریزآرایه‌ی DNA تیمارشده و طبیعتاً پس از آن لازم است که مراحل نرمال‌سازی (نظیر limma/voom یا edgeR) و تبدیل (تثبیت واریانس یا لگاریتم) را نیز طی کنند. به عنوان مثال، یک روش رقابتی، یک روش جامع و یک روش تصحیح اُریبی طول برای داده‌های توالی‌یابی RNA مورد بررسی قرار می‌گیرد. در مثال‌های زیر از مجموعه‌ی داده‌های parathyroid استفاده شده و نتایج مبتنی بر ژن‌ها (نه اگزون‌ها) است.

1- Gene-sampling model
2- Subject-sampling model

قبل از اینکه مثال‌هایی از روش‌های مختلف ارائه گردد، فهرستی از ژن‌های با بیان متفاوت برای مجموعه داده‌های parathyroid ایجاد می‌گردد. کُد زیر آنالیزی است که توسط بسته‌ی edgeR اجرا شده و به دنبال آن شامل مراحلی است که با جزییات بیشتر در فصل هشتم توضیح داده شده‌اند. نتایج شامل جدولی حاوی تغییرات فولد و مقادیر p (sig.edger) و برداری از ID های Ensembl برای ژن‌های با بیان متفاوت (sig.edger.names) است:

```
# Differential analysis using edgeR
library(edgeR)
library(parathyroid)
data(parathyroidGenes)
d<-DGEList(counts=counts(parathyroidGenes))
d<-d[rowSums(d$counts)>=
      ncol(counts(parathyroidGenes)),]
d<-calcNormFactors(d,method="TMM")
meta <- pData(parathyroidGenes)
design <- model.matrix(~treatment+time, data=meta)
d <- estimateGLMCommonDisp(d,design)
d <- estimateGLMTrendedDisp(d,design)
d <- estimateGLMTagwiseDisp(d,design)
fit <- glmFit(d,design)
# Differential expression through time
lrt <- glmLRT(fit,coef=4)
temp <- topTags(lrt,n=100000)$table
sig.edger <- temp[temp$FDR < 0.01,]
sig.edger.names <- rownames(sig.edger)
```

۱۰-۴-۱ روش رقابتی با استفاده از بسته‌ی GOstates

بسته‌ی Bioconductor تحت عنوان GOstats از مدت‌ها پیش در دسترس بوده و معمولاً وقتی که فهرستی از ژن‌های با بیان متفاوت موجود باشد، به نحو قابل اعتمادی عمل می‌کند. علاوه بر این، یک فهرست جداگانه و کُلی از ژن‌ها نظیر ژنوم یک جاندار، نیز لازم است. پیش‌پردازش داده‌ها در دو مرحله صورت می‌گیرد: نخست کلیه‌ی ژن‌ها در بسته‌های حاشیه‌نگاری مختص جاندار (در اینجا: org.Hs.eg.db) در شیء reference.genes به صورت ID های Entrez Gene ذخیره می‌گردد. سپس ID های Ensembl برای ژن‌های با بیان متفاوت (در شیء sig.edger.names) با استفاده از بسته‌ی حاشیه‌نگاری org.Hs.eg.db به ID های Entrez Gene تبدیل می‌شود. کُدی که این دستورزی را اجرا می‌کند، در زیر نمایش داده شده است:

```
library(org.Hs.eg.db)
library(GOstats)
ensembl.to.entrez <- as.list(org.Hs.egENSEMBL2EG)
reference.genes <- unique(unlist(ensembl.to.entrez))
selected.genes<-na.omit(
                      unique(
                        select(
                            org.Hs.eg.db,
                            sig.edger.names,
                            c("ENTREZID"),
                            keytype="ENSEMBL")$ENTREZID
                        )
                      )
```

بسته‌ی GOstats امکان آزمون فزون‌نمایی یا فرونمایی[1] عبارات GO در بین ژن‌های با بیان متفاوت در مقایسه با کُل ژن‌ها را فراهم می‌آورد. معمولاً یک آزمون برای فزون‌نمایی اجرا می‌شود. فزون‌نمایی یا فرونمایی را می‌توان با در نظر گرفتن طبیعت آشیانه‌ای هستی‌شناسی آزمون نمود. زیرا اغلب دیده می‌شود که اگر یک سطح بالاتر از نظر آماری به عنوان فزون‌نمایی معنی‌دار تشخیص داده شود، در نُتاجش نیز فزون‌نمایی دیده می‌شود. این وضعیت در GOstats اصطلاحاً آزمون شرطی[2] نامیده می‌شود. علاوه بر این، لازم است که یک هستی‌شناسی از فرآیند زیستی (BP)، عملکرد مولکولی (MF) و مولفه‌ی سلولی (CC) انتخاب شده و یک مقدار p مناسب نیز برای حد معنی‌داری آماری تنظیم گردد. این آنالیز با تنظیم پارامترها شروع شده و لازم است که فهرستی از ژن‌های با بیان متفاوت و کُل ژن‌ها همراه با بسته‌ی حاشیه‌نگاری که این موارد از آن در دسترس قرار گیرند، تنظیم گردد. به عنوان مثال، کُد زیر آنالیز مورد نظر را اجرا می‌کند:

```
params <- new('GOHyperGParams', geneIds=selected.genes,
universeGeneIds=reference.genes,
annotation='org. Hs.eg.db', ontology='BP',
pvalueCutoff=0.01, conditional=TRUE,
testDirection="over")
```

این آنالیز با دستور `hyperGTest()` اجرا شده و نتایج حاصل از آنرا می‌توان با کمک دستور `summary()` در یک جدول مناسب نمایش داد:

```
go <- hyperGTest(params)
go.table <- summary(go, pvalue=2)
```

1- Underpresentation
2- Conditional testing

شیء go.table حاوی نتایج کلیه‌ی دسته‌های ممکن GO است:

```
head(go.table)
```

```
      GOBPID        Pvalue    OddsRatio     ExpCount  Count   Size
1 GO:0048285  3.061584e-10     6.398889    3.8920412     21    373
2 GO:0051783  1.386607e-05     8.197938    1.0956148      8    105
3 GO:0006950  1.654383e-05     2.113116   32.3467230     55   3100
4 GO:0000070  1.792116e-05    12.560139    0.5530246      6     53
5 GO:0007067  2.069431e-05     4.829758    2.7525349     12    279
6 GO:0010564  2.560388e-05     6.435160    1.5518320      9    155

                                  Term
1                    organelle fission
2         regulation of nuclear division
3                   response to stress
4  mitotic sister chromatid segregation
5                               mitosis
6       regulation of cell cycle process
```

این جدول را می‌توان به گونه‌ای فیلتر نمود که تنها نتایجی که از نظر آماری معنی‌دار هستند، باقی بمانند:

```
go.table.sig<-go.table[go.table$Pvalue<=0.01,]
```

مجموعاً ۸۷ دسته‌ی GO که از نظر آماری معنی‌دار هستند، وجود دارند. یکی از این دسته‌ها تقسیم اندامک است که در جدول فوق نیز دیده می‌شود.

۱۰-۴-۲ روش جامع با استفاده از بسته‌ی Globaltest

استفاده از بسته‌ی Globaltest نسبتاً ساده است. ولی این بسته نیازمند داده‌های واقعی بیان به عنوان ورودی بوده و لازم است که این داده‌ها به یک مقیاس مناسب برای آنالیز تبدیل گردند. بسته‌ی limma با استفاده از تابع voom() این کار را انجام می‌دهد. ولی این تابع نیز نیازمند عوامل نرمال‌سازی و یک ماتریس طرح به عنوان ورودی است. عوامل نرمال‌سازی را می‌توان با استفاده از تابع calcNormFactors() در بسته‌ی edgeR محاسبه نمود. ماتریس طرح قبلاً در همین فصل و زمانی که edgeR برای تشخیص معنی‌داری ژن‌های با بیان متفاوت مورد استفاده قرار گرفت، ایجاد شده است. مراحل پیش‌پردازش را می‌توان با استفاده از کُد زیر اجرا نمود:

```
library(edgeR)
library(limma)
nf <- calcNormFactors(counts(parathyroidGenes))
y <- voom(counts(parathyroidGenes), design, plot=TRUE,
            lib.size=colSums(counts(parathyroidGenes))*nf)
```

سپس می‌توان آنالیز جامع مجموعه‌ی ژنی را با استفاده از تابع gtGO() انجام داد. انتظار می‌رود که این توابع ماتریسی را دریافت کنند که در آن نمونه‌ها در ردیف‌ها و متیغرها در ستون‌ها قرار گرفته‌اند. لذا می‌توان با تنظیم یک گزینه، تابع را به نحوی نوشت که این کار را به صورت خودکار انجام دهد:

```
library(globaltest)
gt.options(transpose=TRUE)
```

اجرای آنالیز نیازمند دریافت بردار پاسخ (در اینجا: زمان)، یک ماتریس مقادیر بیان که توسط voom() تولید شده است، هستی‌شناسی آزمون شده (در اینجا: BP) و نام بسته‌ی حاشیه‌نگاری که امکان تبدیل probe names به ID های Entrez Gene را فراهم نماید، است. چون از یک بسته‌ی حاشیه‌نگاری مختص جاندار برای اجرای آنالیز بهره گرفته می‌شود، یک پارامتر اضافی تحت عنوان probe2entrez نیز لازم است. در این پارامتر فهرستی از ID های ژنی Ensembl با ID های متناظرشان در Entrez Gene وجود دارد. شیء ensemble.to.entrez در طی آنالیز رقابتی ایجاد شده است. این آنالیز با کمک دستور زیر اجرا می‌گردد:

```
go2<-gtGO(meta$time, y$E, ontology="BP",
      annotation="org.Hs.eg.db",
      probe2entrez=as.list(ensembl.to.entrez))
```

نتیجه‌ی آنالیز، فهرستی با مقادیر p برای هر دسته‌ی GO است:

```
head(go2)
```

	holm	alias	p-value	Statistic
GO:0071168	1.94e-05	protein localization to chromatin	1.58e-09	55.9
GO:0051303	2.10e-05	establishment of chromosome localization	1.71e-09	40.7
GO:0050000	2.62e-05	chromosome localization	2.14e-09	37.1
GO:0046104	2.04e-04	thymidine metabolic process	1.67e-08	50.3
GO:0046125	2.04e-04	pyrimidined eoxyribo nucleoside metabolic process	1.67e-08	50.3
GO:0060138	2.07e-04	fetal process involved in parturition	1.69e-08	72.6

```
             Expected      Std.dev      #Cov
GO:0071168       3.85         3.92         7
GO:0051303       3.85         2.92        22
GO:0050000       3.85         2.70        23
GO:0046104       3.85         3.82         4
GO:0046125       3.85         3.82         4
GO:0060138       3.85         5.33         1
```

این جدول می‌تواند فیلتر گردد تا تنها دسته‌هایی از GO که از نظر آماری معنی‌دار هستند، باقی بمانند:

```
go2.table.sig<-go2@result[go2@result[,1]<=0.01,]
```

جدول فیلتر شده حاوی ۹۹۷ دسته‌ی GO بوده که ۱۰ برابر بیشتر از تعداد دسته‌های حاصل از روش رقابتی است.

۱۰-۴-۳ روش تصحیح اُریبی طول

بسته‌ی goseq روشی را پیاده‌سازی می‌کند که زمانی که آنالیز یک مجموعه‌ی ژنی در آزمایش توالی‌یابی RNA انجام می‌شود، برای اُریبی طول تصحیح انجام می‌دهد. ورودی آنالیز goseq یک بردار نامگذاری شده متشکل از صفر و یک بوده که یک‌ها نشان‌دهنده‌ی ژن‌های دارای بیان متفاوت هستند. چنین برداری می‌تواند به آسانی از نتایج قبلیِ به دست آمده در این فصل ایجاد شود. ولی باید توجه نمود که تکرار نام ژن در این بردار مجاز نیست:

```
gene.vector<-as.numeric(reference.genes %in%
                        selected.genes)
names(gene.vector)<-reference.genes
```

شیء gene.vector حاوی اطلاعات لازم است. این آنالیز در سه مرحله انجام می‌شود. در مرحله‌ی نخست، با کمک تابع ()nullp مقدار وزن‌دهی به هر ژن محاسبه می‌گردد. علاوه بر بردار ژنی نام‌گذاری شده، دو پارامتر دیگر شامل: نسخه‌ی ویرایش ژنوم و نوع مشخصه‌ی ژن نیز مورد نیاز است. موارد موجود را می‌توان با کمک توابع ()supportedGenomes و ()supportedGeneIDs بررسی نمود. پس از برآورد شدن، می‌توان آزمون را با استفاده از تابع ()goseq اجرا کرد. این آزمون تنها با استفاده از برهان test.cats به دسته‌ی GO فرآیندهای بیولوژیکی (BP) محدود شده است. goseq به طور خودکار یک تصحیح آزمون چندگانه برای نتایج انجام نمی‌دهد. بنابراین لازم است که یک مرحله‌ی

جداگانه به آن افزوده گردد. تابع p.adjust() با کمک نرخ غلطیابی بنجامینی و هوکبرگ[1] (BH) این تصحیح را انجام می‌دهد. آنالیز بایستی به صورت زیر ادامه یابد:

```
library(goseq)
pwf<-nullp(gene.vector,"hg19","knownGene")
GO.wall<-goseq(pwf,"hg19","knownGene",
                  test.cats=c("GO:BP"))
GO.wall$padj<-p.adjust(
                  GO.wall$over_represented_ pvalue,
                  method="BH")
```

شیء GO.wall یک چارچوب داده‌ی ساده است که حاوی نتایجی برای کلیه‌ی دسته‌های GO می‌باشد. ۷۸ دسته‌ی فزون‌نمایی که از نظر آماری معنی‌دار هستند، وجود دارند که می‌توان آنها را با کمک دستور زیر در یک جدول جداگانه استخراج نمود:

```
go3.table.sig<-GO.wall[GO.wall$padj<=0.01,]
```

۱۰–۵ خلاصه

برای به‌روزرسانی حاشیه‌نگاری‌های ژن‌ها، Bioconductor علاه بر بسته‌های حاشیه‌نگاری مختص جاندار، ارتباط مستقیمی نیز با مجموعه‌ی پایگاه‌های داده‌ی بین‌المللی BioMart دارد. حاشیه‌نگاری‌ها را می‌توان با آنالیز تابعی نتایج به دست آورد. آنالیز مجموعه‌ی ژنی تکنیکی است که اغلب برای کمک به تفسیر زیستی نتایج به کار گرفته می‌شود. حداقل سه روش اصلی برای آنالیز مجموعه‌ی ژنی وجود دارد: روش‌های رقابتی (نظیر GOstats)، روش‌های جامع (نظیر globaltest) و روش‌های تصحیح اُریبی طول (نظیر goseq). همچنین این روش‌ها می‌توانند بر مبنای واحد نمونه‌گیری (ژن یا نمونه) نیز تقسیم‌بندی شوند.

منابع

1. Goeman J. and Bühlmann P. Analyzing gene expression data in terms of gene sets: Methodological issues. Bioinformatics 23:980–987, 2005.
2. Khatri P. and Draghici S. Ontological analysis of gene expression data: Current tools, limitations, and open problems. Bioinformatics 21:3587–3595, 2005.
3. Maciejewski H. Gene set analysis methods: Statistical models and methodological differences. Briefngs Bioinform 1–15, 2013. http://m.bib.oxfordjournals.org/content/early/2013/02/09/bib.bbt002.abstract.

1- Benjamini and Hochberg's false discovery rate (BH)

4. Barry W.T., Nobel A.B., and Wright F.A. Signifcance analysis of functional categories in gene expression studies: A structured permutation approach. Bioinformatics 21(9):1943–1949, 2005.
5. Goeman J.J., van de Geer S.A., de Kort F., and van Houwelingen J.C. A global test for groups of genes: Testing association with a clinical outcome. Bioinformatics 20(1):93–99, 2004.

فصل یازدهم

مصورسازی

۱۱–۱ مقدمه

مصورسازی یک اصطلاح کُلی است که هر چیزی را از نمودارهای شناسایی ساده تا نمودارهای با کیفیت انتشار بهبود یافته در بر می‌گیرد. نمودارهای شناسایی اغلب در طی آنالیز و با هدف شناخت داده‌ها یا کنترل خروجی مراحل مختلف آنالیز ایجاد می‌شوند. مثال‌هایی از چنین نمودارهایی عبارتند از: هیستوگرام[1] (بافت نگار)، نمودار پراکنش[2] و نمودار ستونی. این نمودارها یک دیدگاه کُلی ساده و سریع از مجموعه‌ی داده‌ها ارائه می‌دهند. نمودارهای با کیفیت انتشار بسیار دقیق ترسیم می‌شوند. این نمودارها اغلب برای برجسته نمودن نتایج یا تکمیل نمودن نتیجه‌گیری‌های مطالعه ترسیم شده و از چنان کیفیتی برخوردار هستند که می‌توان آنها را در یک کتاب یا در یک مقاله در مجلات علمی چاپ نمود.

اکثر مجلات علمی قواعد خاصی برای فایل‌های تصویری که برای چاپ پذیرش می‌کنند، دارند. این قواعد در راهنمای نویسندگان ذکر می‌گردد. معمولاً مجلات علمی تصاویر را حداقل در فرمت TIFF یا PDF می‌پذیرند. ولی این موضوع در مورد همه‌ی مجلات علمی صدق نمی‌کند. علاوه بر این، ممکن است که ملاحظاتی در مورد مدل رنگِ به کار گرفته شده در تصاویر داشته باشند و محدودیت‌هایی نیز روی وضوح تصویر اعمال نمایند.

در این فصل تعدادی از انواع نمودارها که می‌توانند برای انتقال پیام‌های یک مطالعه در یک ارائه یا انتشار به کار گرفته شوند، مورد بحث و بررسی قرار می‌گیرند. نمودارهای شناسایی[3] در فصل‌هایی که انواع آنالیزها مورد بحث قرار می‌گیرند، پوشش داده می‌شوند. چون ملزومات پایه‌ی یک نمودار دارای کیفیت انتشار شامل وضوح مناسب و نوع فایل تصویر است، لذا راه حل‌های مختلف در R برای این ملزومات نیز مورد بحث و بررسی قرار می‌گیرند.

ابتدا مروری بر مفاهیم پایه‌ی گرافیک رایانه‌ای، انواع فایل، وضوح و مدل‌های رنگ صورت می‌گیرد. پس از درک مبانی پایه‌ی این مفاهیم، اصول ایجاد نمودارهای معین با استفاده از R ارائه می‌گردد.

1- Histogram
2- Scatter plot
3- Exploratory plot

۱۱-۱-۱ انواع فایل‌های تصویر

به طور کلی فرمت فایل‌های تصویری به دو دسته‌ی بزرگ تقسیم می‌شوند: بیتمپ‌ها[1] و وکتور گرافیک‌ها[2]. تصاویر بیتمپ که گاهی مواقع پیکسمپ یا تصاویر شطرنجی نیز خوانده می‌شوند، از نقاط منفردی تشکیل می‌شوند که پیکسل[3] نام دارند. چون تعداد پیکسل‌ها در هر تصویر ثابت است، تصاویر بیتمپ نمی‌توانند بدون از دست دادن جزییات‌شان بزرگ شوند. ولی در مقابل، تصاویر وکتور از شکل‌های هندسی برای نشان دادن تصاویر استفاده می‌کنند. چون تصاویر وکتور از اجزا و عبارات ریاضی تشکیل می‌شوند، لذا اندازه‌ی آنها را می‌توان بدون کاهش کیفیت تغییر داد. ولی یک نکته‌ی احتیاطی کوچک را باید در نظر گرفت. اگر تعداد عناصر (مثلاً تعداد نقاط در یک نمودار پراکنش) زیاد باشد، تصاویر وکتور گرافیک بسیار حجیم شده و فایل‌ها به آهستگی باز می‌شوند. بنابراین اگر تعداد عناصر زیاد باشد، بهتر است که از فرمت‌های تصویر شطرنجی برای ذخیره‌ی نمودارها استفاده شود.

مثال‌هایی از فرمت‌های بیتمپ عبارتند از: بیتمپ ویندوز[4] (BMP)، فرمت فایل تصویر نشانمند[5] (TIFF)، گروه مشترک خبرگان عکاسی[6] (JPEG) و گرافیک‌های شبکه‌ای ترابرپذیر[7] (PNG). بیشترین استفاده از فرمت‌های وکتور گرافیک نیز در پستاسکریپت[8] و فرمت اسناد ترابرپذیر[9] (PDF) است. به عبارت دقیق‌تر، پستاسکریپت و PDF فرمت‌های فرافایل هستند که می‌توانند هم تصاویر شطرنجی و هم تصاویر وکتور را ترکیب نمایند. ولی اگر یک فایل پستاسکریپت یا PDF را بر مبنای راهنمایی‌های موجود در این فصل از R بسازید، فایل‌های مزبور تنها شامل تصاویر وکتور خواهند بود.

۱۱-۱-۲ وضوح تصویر

وضوح تصویر جزییات آن تصویر را اندازه گرفته و برای تصاویر بیتمپ بر مبنای پیکسل سنجیده می‌شود. وضوح در وکتور گرافیک، تفسیر مستقیمی ندارد. به عنوان مثال، اگر یک تصویر TIFF حاوی ۸۰۰ ستون و ۶۰۰ ردیف پیکسلی باشد، گفته می‌شود که وضوحش ۶۰۰ × ۸۰۰ بوده

1- Bitmap
2- Vector graphics
3- Pixel
4- Windows bitmap (BMP)
5- Tagged Image File Format (TIFF)
6- Joint Photographic Experts Group (JPEG)
7- Portable Network Graphics (PNG)
8- PostScript
9- Portable Document Format (PDF)

و دارای ۴۸۰۰۰۰ پیکسل است. در دوربین‌های دیجیتال، چنین تصویری تقریباً ۰/۵ مگاپیکسل خواهد بود.

در چاپ دیجیتال، اغلب وضوح بر مبنای تعداد نقاط در هر اینچ[1] (PPI) یا نقطه‌های موجود در هر اینچ[2] (DPI) سنجیده می‌شود. این واحدها را می‌توان به صورت تعداد پیکسل در هر اینچ (۲۵/۴ میلی‌متر) تفسیر نمود. از این اطلاعات می‌توانید برای محاسبه‌ی اندازه‌ی تصویری که می‌خواهید تولید نمایید، استفاده کنید. به عنوان مثال، اگر بخواهید تصویری با طول ۴ اینچ (حدوداً ۱۰ سانتی‌متر) چاپ کنید، و مجله نیز تاکید دارد که تصاویر با وضوح ۶۰۰ DPI باشند، لازم است که تصویر مزبور حداقل ۲۴۰۰ پیکسل طول داشته باشد تا بتواند به استاندارد مورد نظر مجله برسد.

۱۱-۱-۳ مدل‌های رنگ

دو مدل رنگی که اغلب مورد استفاده قرار می‌گیرند، عبارتند از: RGB و CMYK. مدل رنگ RGB روی تصاویر رسانه‌ای مبتنی بر انتقال نور نظیر تلویزیون یا نمایشگر رایانه مورد استفاده قرار می‌گیرد. هر پیکسل یک تصویر شامل سه رنگ مختلف قرمز (R)، سبز (G) و آبی (B) بوده و وقتی که این سه رنگ در مقادیر معین با هم مخلوط می‌شوند، رنگ‌های مرئی (فرمت رنگ افزوده[3]) را تشکیل می‌دهند. چون نمایشگرهای رایانه در وضعیت RGB کار می‌کنند، تصاویر ایجاد شده توسط رایانه‌ها نیز اغلب در فایل‌های با فرمت RGB ذخیره می‌شوند.

CMYK در صنایع چاپ مورد استفاده واقع می‌شود. از طریق ترکیب مرکب‌های فیروزه‌ای (C)، ارغوانی (M) و زرد (Y)، طیف وسیعی از رنگ‌های مرئی برای انسان ایجاد می‌شود. علاوه بر این، مرکب سیاه (K) نیز اغلب به دلایل اقتصادی و فنی مورد استفاده قرار می‌گیرد. CMYK یک فرمت رنگ کاهشی[4] است. زیرا مرکب‌ها طول‌های مختلف رنگ را جذب کرده و رنگ مرئی از طول موج‌هایی که جذب نشده‌اند، تشکیل می‌شود.

برخی از مجلات تصاویر را تنها در فرمت CMYK می‌پذیرند. ولی سایر مجلات تصاویر را در فرمت RGB نیز پذیرش می‌نمایند.

۱۱-۲ گرافیک در R

R می‌تواند تصاویر را در چندین فرمت ذخیره نماید. در این بین، BMP ، JPEG ، TIFF ، PNG ، PDF و پست‌اسکریپت به راحتی در دسترس هستند. هم فایل‌های پست‌اسکریپت و هم

1- Points Per Inch (PPI)
2- Dots Per Inch (DPI)
3- Additive color format
4- Subtractive color format

فایل‌های PDF در فرمت‌های رنگ RGB و CMYK ایجاد می‌شوند. ولی کلیه‌ی فرمت‌های دیگر تنها از مدل رنگ RGB پشتیبانی می‌کنند. وضوح این تصاویر را می‌توان به راحتی تغییر داده و اکثر ملزومات وضوح نیز به آسانی قابل اجرا هستند.

دو سامانه‌ی گرافیکی مختلف در R وجود دارد: گرافیک پایه[1] و گرافیک شبکه‌ای[2] (۱). گرافیک پایه شامل هر دو تابع سطح بالا و سطح پایین است. توابع سطح بالا یک نمودار کامل ایجاد کرده و توابع سطح پایین این امکان را به کاربر می‌دهند که نمودارهایی را از ابتدا ایجاد نموده یا مواردی را به یک نمودار موجود بیفزاید. گرافیک شبکه‌ای شامل هیچ‌گونه تابع سطح بالایی نبوده ولی برخی از بسته‌ها (نظیر lattice و ggplot2) از گرافیک شبکه‌ای برای پیاده‌سازی انواع تابع سطح بالای ترسیم نمودار استفاده می‌کنند. هم گرافیک پایه و هم گرافیک شبکه‌ای ایستا بوده و نمودار ایجاد شده توسط آنها را نمی‌توان به وسیله‌ی ابزارهای دیگر تغییر داد.

علی‌رغم اینکه انواع زیادی از گرافیک ایستا[3] در دسترس است، ولی چنین وضعیتی در مورد گرافیک تعاملی[4] صادق نیست. تعداد بسیار اندکی از بسته‌ها نظیر iplots ، playwith و rgl چنین توابعی را به R می‌افزایند. ولی انواع نمودارهای موجود برای گرافیک تعاملی چندان زیاد نیستند. این محدودیت با تعامل بین R و برنامه‌ی ggobi یا با ایجاد نمودارها توسط برخی از کتابخانه‌های جاوا اسکریپت نظیر rCharts برطرف می‌شود. کلیه‌ی گرافیک‌های ایستا را می‌توان در فرمت‌های بیتمپ یا وکتور گرافیک که در بالا اشاره گردیدند، ذخیره نمود. ولی چنین کاری برای گرافیک تعاملی امکان‌پذیر نمی‌باشد.

بخش‌های زیر معمول‌ترین مصورسازی‌های مشاهده شده در مقالات مرتبط با بیان ژن و نحوه‌ی ایجاد آنها در R را مورد بحث و بررسی قرار می‌دهند.

۱۱-۲-۱ نقشه‌ی حرارتی

نقشه‌ی حرارتی نموداری است که مقادیر بیان ترکیبات (ژن‌ها، اگزون‌ها و غیره) را با استفاده از یک مقیاس رنگی نمایش می‌دهد. معمولاً ترکیبات در ستون‌ها (نمونه‌ها) و ردیف‌ها (ترکیبات) و همانند ماتریس داده‌های اصلی تنظیم می‌شوند. هر جفت ترکیب – نمونه با یک مستطیل کوچک که بر مبنای بیانش رنگ شده است، نشان داده می‌شود. اغلب پیش از ایجاد نمودار حرارتی، هم

1- Base graphics
2- Grid graphics
3- Static graphics
4- Interactive graphics

نمونه‌ها و هم ترکیبات به صورت آشیانه‌ای خوشه‌بندی شده و این خوشه‌بندی با یک نمایش درختی در سمت چپ و بالای ماتریس داده‌های رنگی نمایش داده می‌شود.

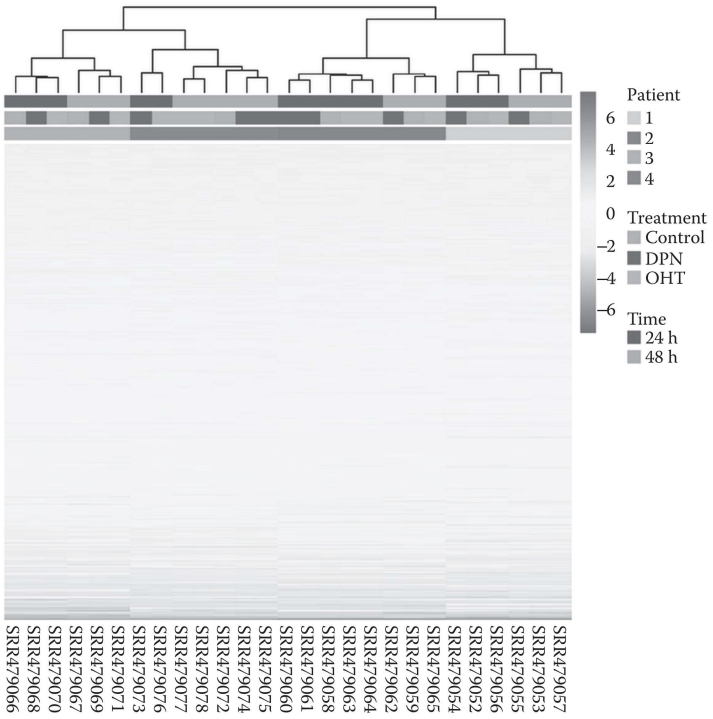
نمودار حرارتی اغلب برای نشان دادن دو یا چند گروهی که در بیان ژن‌های معینی تفاوت دارند، استفاده می‌شود. اگر ترسیم نمودار پیش از اجرای آزمون آماری انجام شود، مناسب خواهد بود. ولی اگر این نمودار تنها برای ترکیبات (ژن‌هایی) ترسیم شود که تفاوت بیان‌شان از نظر آماری معنی‌دار تشخیص داده شده است، نمودار حرارتی نمی‌تواند برای نمایش دو گروهی که واقعاً متفاوت هستند، به کار گرفته شود. در چنین مواردی این نمودار می‌تواند تنها به عنوان روشی برای مصورسازی نتایج به کار گرفته شود. هر دو کاربرد فوق با مثال‌هایی از مجموعه داده‌ی parathyroid در R تشریح می‌شوند.

برای ایجاد نمودارهای حرارتی در R به صورت پیش‌فرض از تابع `heatmap()` استفاده می‌شود. این تابع اجازه‌ی اعمال تغییرات چندانی در نمودار نمی‌دهد. لذا به جای آن از تابع `pheatmap()` از بسته‌ی pheatmap (ترکیبی از دو عبارت pretty heatmap به معنای نمودار حرارتی زیبا) استفاده می‌گردد. یک نمودار حرارتی زیبا برای مجموعه داده‌ی parathyroidGenes را می‌توان به صورت زیر ایجاد نمود:

```
# Loads the data
library(parathyroid)
data(parathyroidGenes)
# Filtering
keep <-rowSums(counts(parathyroidGenes) >100)
                  >=ncol(counts(parathyroidGenes))
dooku <-counts(parathyroidGenes)[keep,]
rsd <-rowSums(dooku)
dooku <-dooku[order(rsd),]
# Plotting
library(pheatmap)
pheatmap(log2(dooku),cluster_rows = FALSE,
            show_ rownames = FALSE,
            annotation = data.frame(
                  (pData(parathyroidGenes)[,3:5])),
            border_color = "grey95",
            scale = "column")
```

ابتدا داده‌ها از بسته‌ی parathyroid بارگذاری می‌شوند. سپس برای حذف ژن‌هایی که شمارش پایینی در نیمی از نمونه‌ها دارند، مجموعه‌ی داده‌ها پاکسازی می‌گردد. جدول شمارش بر مبنای بیان کُل ژن‌ها در بین نمونه‌ها مرتب می‌شود تا از این طریق خواندن نمودار حرارتی آسان‌تر گردد.

شمارش‌ها با لگاریتم‌گیری در مبنای ۲ تبدیل می‌شوند. سرانجام نموداری ترسیم می‌شود که بایستی شبیه نگاره‌ی ۱۱-۱ باشد. توجه شود که ستون‌ها مقادیر تبدیل شده‌ی لگاریتمی بر مبنای ۲ بوده و در طی ترسیم نمودار با میانگین یکسانی مقیاس‌بندی شده‌اند. اگر راه حل بهتری برای نرمال‌سازی وجود نداشته باشد، این کار می‌تواند به عنوان یک نرمال‌سازی تقریبی نیز در نظر گرفته شود.



نگاره‌ی ۱۱-۱: یک نمودار حرارتی ایجاد شده از مجموعه داده‌ی parathyroidGenes پاکسازی شده. به صورت پیش‌فرض از یک طرح رنگی شامل دامنه‌ی از قرمز تا آبی استفاده شده است.

از طرف دیگر می‌توان نتایج حاصل از آنالیز DESeq که در فصل هشتم انجام شده است را مورد استفاده قرار داد. کُدهای زیر با استفاده از داده‌های شمارش با واریانس تثبیت شده (نرمال‌سازی شده) یک نمودار حرارتی ترسیم می‌نمایند (نمودار مزبور در اینجا نمایش داده نشده است):

```
library(parathyroid)
library(DESeq2)
data(parathyroidGenes)
```

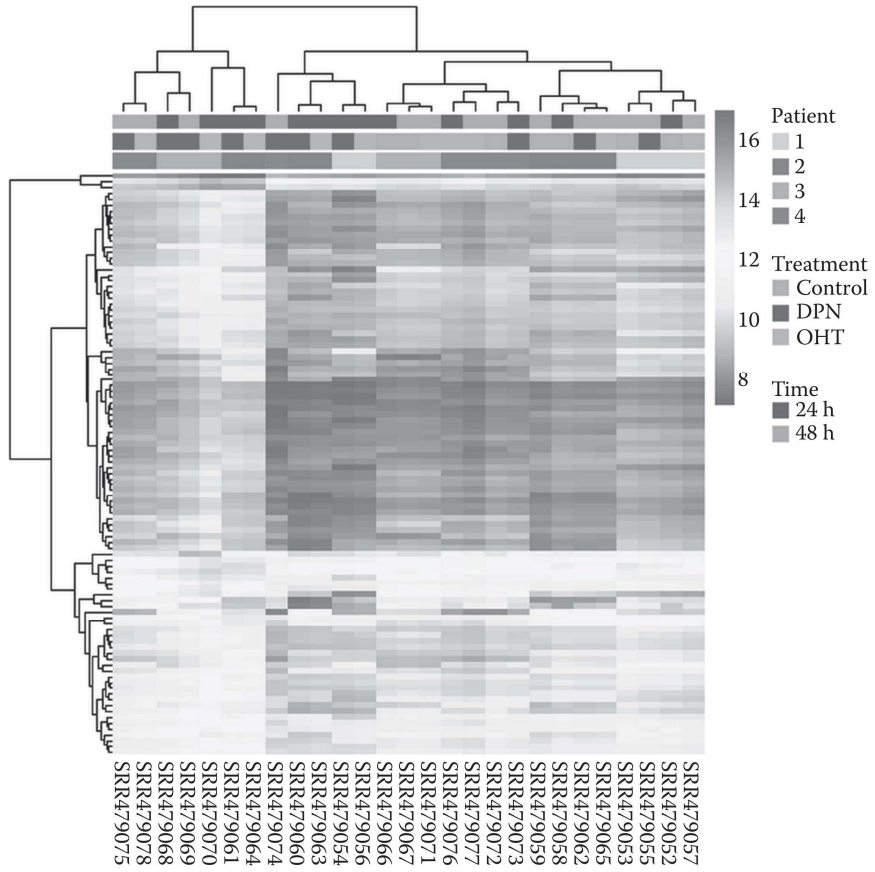
```
d.deseq <- DESeqDataSetFromMatrix(
                countData = counts (parathyroidGenes),
                colData=pData(parathyroidGenes),
                design=~treatment)
d.deseq <-estimateSizeFactors(d.deseq)
d.deseq <-DESeq(d.deseq)
resultsNames(d.deseq)
res <- results(d.deseq,"treatment_OHT_vs_Control")
sig <- res[which(res$pvalue < 0.01),]
vsd <- getVarianceStabilizedData(d.deseq)
vsdp <- vsd[rownames(vsd) %in% rownames(sig),]
library(pheatmap)
pheatmap(vsdp, cluster_rows = TRUE,
          show_ rownames = TRUE,
          annotation = data.frame(
               (pData(parathyroidGenes)
               [,4,drop = FALSE])
          ),
          border_color = "grey95", scale = "none")
```

تابع pheatmap() یک ویژگی جالب و در بسیاری از مواقع مفید دارد. این تابع ترسیمِ نمودار می‌تواند تعداد معینی از شبه‌ژن‌ها[1] را ایجاد کرده و سپس از آنها به عنوان مجموعه داده برای ترسیم نمودار استفاده کند. شبه‌ژن‌ها با کمک الگوریتم خوشه‌بندی K-means که کلیه‌ی ژن‌های با رفتار مشابه را با یکدیگر خوشه‌بندی می‌کند، ایجاد می‌شوند. همچنین استفاده از شبه‌ژن‌ها در این نمودار، خوشه‌بندی ردیف‌ها را نیز امکان‌پذیر می‌سازد. این کار اغلب برای کُل مجموعه داده‌های متشکل از ده‌ها و صدها هزار ترکیب، غیرممکن است. این نمودار را می‌توان با تنظیم برهان kmeans_k روی یک عدد (در اینجا: ۱۰۰) و فعال نمودن خوشه‌بندی ردیف‌ها ایجاد نمود:

```
pheatmap(log2(dooku), cluster_rows = TRUE,
            show_ rownames = FALSE,
            annotation = data.frame(
                (pData(parathyroidGenes)
                [,3:5])
            ),
            border_color = "grey95",
            scale = "none", kmeans_k = 100)
```

نمودار حاصل در نگاره‌ی ۱۱-۲ نشان داده شده است.

1- Pseudogene

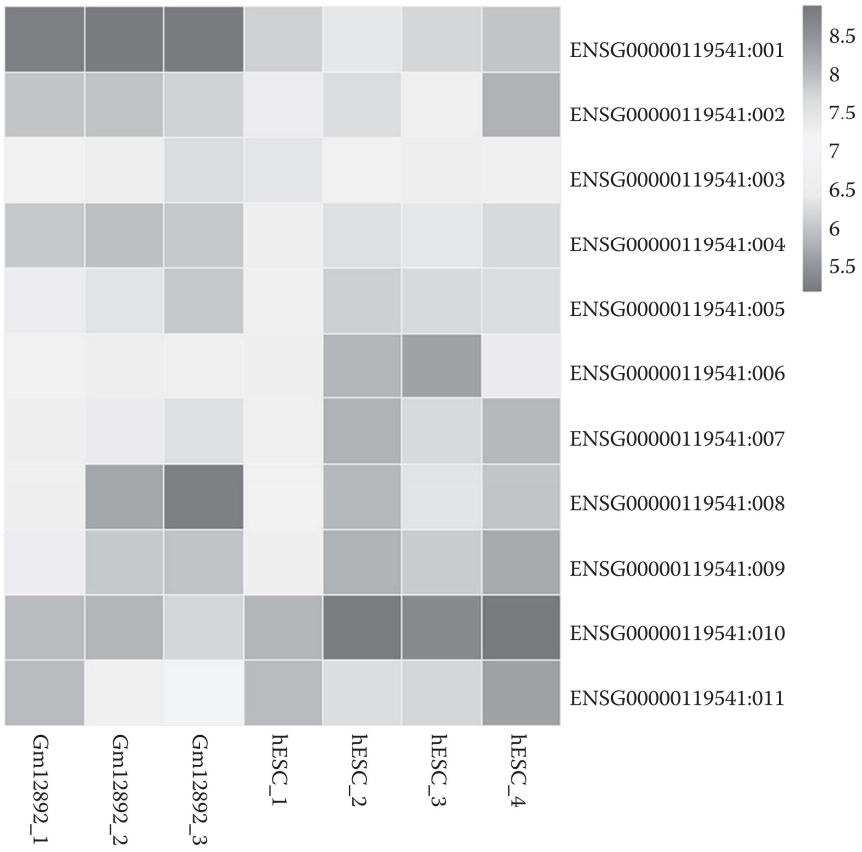


نگاره‌ی ۱۱-۲: نمودار حرارتی ایجاد شده از مجموعه داده‌ی parathyroidGenes با استفاده از شبه‌ژن‌ها برای ترسیم نمودار

همچنین یک نمودار حرارتی می‌تواند برای مصورسازی نتایج آنالیز بیان مختص اگزون نیز به کار گرفته شود. هر اگزون یک ردیف جداگانه در نمودار تشکیل داده و بر مبنای جدول شمارش رنگ‌آمیزی می‌شود. برای ژن ENSG00000119541 که در فصل نهم شناسایی شده است، می‌توان نمودار حرارتی را از شیء ecs (ایجاد شده در فصل نهم) و با کمک کُد زیر ترسیم نمود:

```
vismat <-counts(ecs)[fData(ecs)$geneID==
                    "ENSG00000119541",]
colnames(vismat)<-pData(ecs)$samplename
pheatmap(log2(vismat), cluster_rows = FALSE,
           cluster_cols = FALSE, border_col = "grey95")
```

نخست کلیه‌ی شمارش‌های مختص اگزون از شیء exonCountSet (ecs) استخراج شده و نمودار بدون خوشه‌بندی نمونه‌ها یا ردیف‌ها ترسیم می‌گردد تا ترتیب مکانی اگزون‌ها در طول ژن مشخص گردد. نمودار حاصل در نگاره‌ی ۱۱-۳ نمایش داده شده است.



نگاره‌ی ۱۱-۳: نمودار حرارتی ایجاد شده از یک جدول شمارش برای اگزون‌های یک ژن. این نگاره را با نگاره‌ی ۹-۴ مقایسه نمایید.

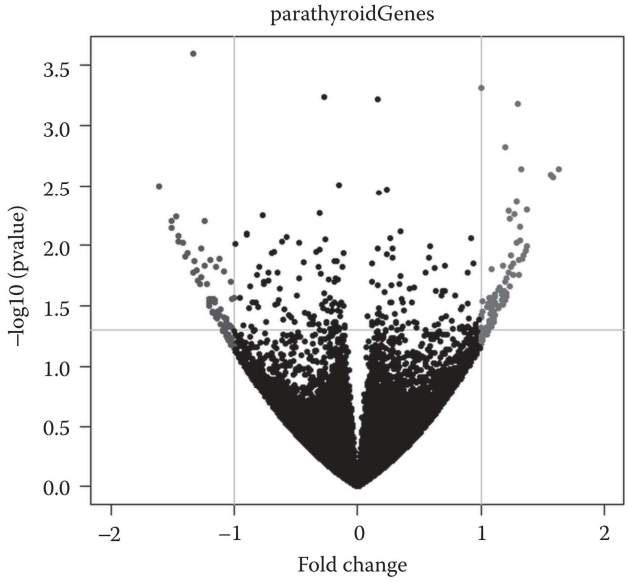
۱۱-۲-۲ نمودار آتشفشانی

به طور ساده یک نمودار آتشفشانی (۲) در واقع یک نمودار پراکنش است که در آن مقادیر تغییرات فولد برای کلیه‌ی ترکیبات روی محور افقی (x) و مقادیر p تبدیل شده با کمک منفی لگاریتم در مبنای ۱۰ روی محور عمودی (y) ترسیم می‌گردد. در برخی از موارد چنین نموداری را

نمودار آتشفشانی نیز می‌نامند. زیرا شبیه یک آتشفشان است که فوران کرده و گدازه‌ها را به همه جا می‌پاشد. اگر ترکیبات دارای افزایش و کاهش بیان به رنگ‌های مختلف رنگ‌آمیزی شده و خطوط راهنما به زمینه‌ی نمودار افزوده شوند، نمودار آتشفشانی جذاب‌تر می‌شود. نمودار آتشفشانی به آسانی یک دیدگاه کُلی قابل رمزگشایی از تعداد تقریبی ترکیبات دارای افزایش و کاهش بیان و معنی‌داری آماری آنها ارائه می‌دهد.

در اینجا از شیء res که برای مجموعه داده‌ی parathyroidGenes ایجاد شده بود، در ترسیم نمودار آتشفشانی استفاده می‌شود. آنالیز مربوط به این داده‌ها در فصل هشتم به صورت مفصل تشریح شده و در مباحث بخش قبل در مورد ترسیم نمودار حرارتی نیز مورد بحث قرار گرفت. شیء res حاوی ستون‌های log2FoldChange و pvalue است که برای ترسیم یک نمودار به کار گرفته می‌شوند. کُد زیر، نمودار نشان داده شده در نگاره ۱۱-۴ را ترسیم می‌کند:

```
# Analysis library(parathyroid)
library(DESeq2)
data(parathyroidGenes)
d.deseq <- DESeqDataSetFromMatrix(countData =
                counts (parathyroidGenes),
                colData = pData(parathyroidGenes),
                design = ~treatment)
d.deseq <- estimateSizeFactors(d.deseq)
d.deseq <- DESeq(d.deseq)
resultsNames(d.deseq)
res <- results(d.deseq, "treatment_OHT_vs_Control")
sig <- res[which(res$pvalue < 0.01),]
# Generate the colors
cols <-rep("#000000", nrow(res))
cols[res$log2FoldChange >=1]<-"#CC0000"
cols[res$log2FoldChange <=-1]<-"#0000CC"
# Produce the plot
plot(res$log2FoldChange,-log10(res$pvalue),
    pch = 16, cex = 0.75, col = cols, las = 1,
    xlab = "FoldChange", ylab = "-log10(pvalue)",
    xlim = c(-2,2))
# Add the vertical and horizontal
lines abline(h = -log10(c(0.05)),col = "grey75")
abline(v = c(-1,1),col = "grey75")
# Plot the points again to overlay the lines
points(res$log2FoldChange, -log10(res$pvalue),
  pch = 16,cex = 0.75, col = cols)
# Add a title
title(main = "parathyroidGenes")
```



نگاره‌ی ۱۱-۴: یک نمودار آتشفشانی که از مجموعه داده‌ی parathyroidGenes ایجاد شده است. تعداد نسبتاً کمی از ژن‌ها از نظر آماری بیان متفاوت معنی‌داری داشته و این موضوع به وضوح قابل مشاهده است.

۱۱-۲-۳ نمودار MA

نمودار MA (۳) نیز یک نمودار پراکنش است که در آن میانگین بیان ژن روی محور x و تغییر فولد یا لگاریتم نسبت روی محور y ترسیم می‌گردد. نمودار MA به صورت وسیعی در زمینه‌ی بیوانفورماتیک ریزآرایه‌های DNA به کار گرفته شده است. همچنین این نمودار در آزمایشات توالی‌یابی RNA نیز قابل استفاده است. در اینجا مثالی از مجموعه داده‌ی parathyroidGenes ارائه می‌شود. نخست باید ژن‌هایی که بیان متفاوتی دارند، یافت شوند:

```
# Analysis library(parathyroid)
library(DESeq2)
data(parathyroidGenes)
d.deseq <- DESeqDataSetFromMatrix(countData =
    counts (parathyroidGenes),
    colData = pData(parathyroidGenes),
    design = ~treatment)
d.deseq <- estimateSizeFactors(d.deseq)
d.deseq <- DESeq(d.deseq)
resultsNames(d.deseq)
res <-results(d.deseq, "treatment_OHT_vs_Control")
sig <- res[which(res$pvalue < 0.01),]
```

شیء res که حاصل شده است، حاوی ستون‌های baseMean (میانگین بیان) و log2FoldChange بوده که در بردارنده‌ی مقادیر تغییر فولد است. همچنین ستون pvalue حاوی مقادیر خام p بوده که برای رنگ‌آمیزی علایم موجود در نمودار به کار گرفته می‌شوند. نخست نمودار پراکنشی که شبکه‌ی خاکستری روی آن قرار دارد، ترسیم می‌گردد. سپس برای پاک کردن خطوط خاکستری شبکه که نقاطی را می‌پوشانند، نقاط مزبور مجدداً ولی به صورت جداگانه برای ژن‌هایی که به صورت متفاوت بیان شده یا بیان نشده‌اند، ترسیم می‌گردند. ژن‌های با بیان متفاوت در انتها و با مربع‌های قرمز ترسیم می‌شوند تا نسبت به ژن‌های بیان نشده متمایز گردند. سرانجام برای تکمیل ظاهر نمودار، ناحیه‌ی مزبور توسط کادری احاطه می‌شود. کُد R برای ترسیم چنین نموداری در زیر آورده شده و نمودار مزبور نیز در نگاره‌ی ۱۱-۵ نمایش داده شده است.

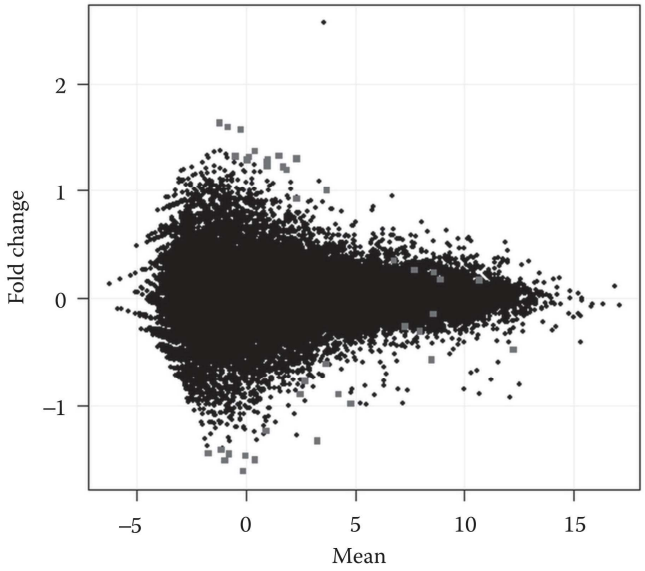
```
# Plot
plot(x = log2(res$baseMean), res$log2FoldChange, pch = 16,
   cex = 0.5, las = 1, xlab = "Mean", ylab = "Fold Change")
grid(lty = 1, col = "grey95")
cols <-ifelse(res$pvalue>0.01|is.na(res$pvalue),
   "#000000", "#CC0000")
points(x = log2(res$baseMean)[cols = ="#000000"],
   res$log2FoldChange[cols = ="#000000"],pch = 16,
   cex = 0.5)
points(x = log2(res$baseMean)[cols = ="#CC0000"],
   res$log2FoldChange[cols = ="#CC0000"],pch = 15,
   cex = 0.75, col = "#CC0000")
box()
```

۱۱-۲-۴ ایدئوگرام

ایدئوگرام یک ترسیم یا یک عکس از کروموزوم‌های یک سلول (کاریوتایپ) است. این نمودار اکثراً برای تجسم یک نمودار ایده‌آل از کروموزوم‌های جاندار به کار گرفته شده و گاهی اوقات با اغماض و البته اندکی گمراه کننده، ایدئوگرام نامیده می‌شود. در مطالعات بیان ژن این نمودار در برخی از مواقع برای نمایش موقعیت ژن‌های بیان شده در ژنوم جاندار به کار گرفته می‌شود. علاوه بر این، در اغلب موارد به طور مشابهی موقعیت SNP ها و واریانت‌های ساختاری نیز توسط یک ایدئوگرام مصورسازی می‌گردد.

کروموزوم‌های انسان پس از رنگ‌آمیزی گیمسا در زیر میکروسکوپ به صورت راه‌راه یا نواربندی شده دیده می‌شوند. این نوارها اصطلاحاً ایزوکور[1] نامیده شده و شامل نواحی بزرگی از ژنوم هستند

1- Isochore



نگاره‌ی ۱۱-۵: مثالی از یک نمودار MA برای مجموعه داده‌های parathyroidGenes. ژن‌های بیان شده با مربع مشخص شده و متمایز گردیده‌اند.

که محتوای GC نسبتاً متفاوتی دارند. ایزوکورها اغلب برای مشخص کردن موقعیت ژن‌ها در کروموزوم‌ها استفاده می‌شوند. قبل از توالی‌یابی ژنوم انسان، ایزوکورها سامانه‌ی موقعیت‌یابی غالب بودند. امروزه به جای آن اغلب از موقعیت‌های دقیق نوکلئوتیدها استفاده می‌شود. نقطه‌ی ضعف کوچک سامانه‌ی موقعیت‌یابی مبتنی بر نوکلئوتید این است که این موقعیت‌ها می‌توانند بین نسخه‌های ویرایش مختلف اسمبل یک ژنوم، دچار تغییر شوند. ایزوکورها نسبتاً پایدارتر بوده ولی حجم نوارها تقریباً به مرحله‌ی چرخه‌ی سلولی کروموزوم‌های رنگ‌آمیزی شده بستگی دارد. با استفاده از ایزوکورها (نوارها) موقعیت‌های ژنی با کمک یک سامانه‌ی ساده از شماره‌ی کروموزوم (۱ تا ۲۲ ، X و Y)، بازوی کروموزوم (q برای بازوی بلند و p برای بازوی کوتاه) و نوار ایزوکور (شماره‌بندی از سانترومر به سمت انتهای کروموزوم) مشخص می‌گردد. به عنوان مثال، 7q31 نشان دهنده‌ی ژنی است که در بازوی بلند کروموزوم شماره‌ی ۷ و در نوار ۳۱ واقع شده است. گاهی اوقات نوارهای فرعی نیز مورد استفاده واقع شده و با استفاده از مقادیر اعشاری نشان داده می‌شوند (به عنوان مثال: 7q31.2).

در اینجا ایدئوگرامی برای نشان دادن موقعیت ژن‌های با بیان متفاوت که در مجموعه داده‌های parathyroidGenes یافت شده‌اند، ترسیم می‌گردد. نخست لازم است که ژن‌های با بیان

متفاوت یافت گردند. این کار طبق روش توضیح داده شده برای نمودار حرارتی صورت گرفته ولی در اینجا نیز تکرار می‌گردد:

```
library(parathyroid)
library(parathyroid)
library(DESeq2)
data(parathyroidGenes)
d.deseq <- DESeqDataSetFromMatrix(
    countData = counts(parathyroidGenes),
    colData = pData(parathyroidGenes),
    design = ~treatment)
d.deseq <- estimateSizeFactors(d.deseq)
d.deseq <- DESeq(d.deseq)
resultsNames(d.deseq)
res <- results(d.deseq, "treatment_OHT_vs_Control")
sig <- res[which(res$pvalue < 0.01),]
```

پس از ذخیره‌ی ژن‌های دارای تفاوتِ بیانِ معنی‌دار در شیء sig ، لازم است که موقعیت‌های کروموزومی این ژن‌ها نیز یافت شوند. امکانات متعددی برای انجام این کار وجود دارد. ولی در اینجا از یک بسته‌ی حاشیه‌نگاری مختص جاندار استفاده می‌شود.

```
library(org.Hs.eg.db)
keys <- keys(org.Hs.eg.db, keytype = "ENSEMBL")
columns <-c ("CHR","CHRLOC", "CHRLOCEND")
sel <-select(org.Hs.eg.db, keys, columns,
    keytype = "ENSEMBL")
sel2 <-sel[sel$ENSEMBL %in% rownames(sig),]
sel3 <-na.omit(sel2[!duplicated(sel2$ENSEMBL),])
sel3$strand <-ifelse(sel3$CHRLOC < 0, "-", " + ")
sel3$start <-abs(sel3$CHRLOC)
sel3$end <-abs(sel3$CHRLOCEND)
```

پس از مشخص شدن موقعیت ژن‌های دارای بیان متفاوت، می‌توان از این داده‌ها برای ترسیم یک ایدئوگرام استفاده کرد. ابزارها و امکانات معدودی برای ترسیم یک ایدئوگرام وجود دارد. پروژه‌ی Bioconductor حداقل دو بسته برای این کار پیشنهاد می‌کند که توابعی برای ترسیم ایدئوگرام دارند. بسته‌ی قدیمی‌تر idiogram نام داشته که به خوبی با بسته‌ی geneplotter کار می‌کند. در اینجا از یک بسته‌ی جدیدتر تحت عنوان ggbio استفاده می‌شود. قبل از اقدام به ترسیم، لازم است که شیء جی‌رنجز که حاوی موقعیت ژن‌های با بیان متفاوت است، ایجاد شود. زیرا ggbio عمدتاً با اشیاء جی‌رنجز کار می‌کند:

```
library(GenomicRanges)
tt <-org.Hs.egCHRLENGTHS
read <-GRanges(
      seqnames = Rle(paste("chr", sel3$CHR, sep = "")),
      ranges = IRanges(start = sel3$start, end = sel3$end),
      strand = Rle(sel3$strand)
)
```

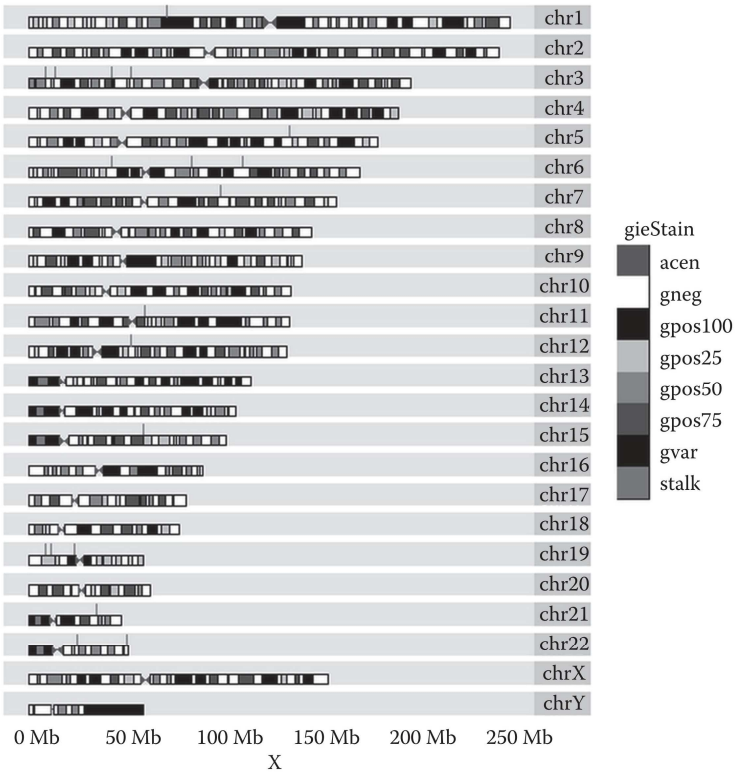
پس از ساخته شدن شیء جی‌رنجز، یک ایدئوگرام به روش زیر ترسیم می‌گردد. نخست لازم است که داده‌ها روی نواربندی کروموزوم‌های انسانی پیاده شوند. این کار در بسته‌ی biovizBase در شیء hg19IdeogramCyto تعبیه شده است. لازم به ذکر است که این داده‌ها همراه با اسمبل ژنوم نسخه‌ی ویرایش hg19 هستند. چون در اینجا هدف تنها تکمیل کروموزوم‌ها است (سایر قطعات نیز در داده‌ها حضور دارند)، لذا تنها کروموزوم‌ها با استفاده از تابع keepSeqlevels() انتخاب می‌شوند. سپس ایدئوگرام با استفاده از تابع ggplot() و با کمک طرح‌بندی مختص کاریوگرام ترسیم می‌گردد. پس از آن برای تشکیل نمودار نهایی، این موقعیت‌های ژنی به صورت میله‌های قرمز به نمودار افزوده می‌شوند. برای نمایش نمودار حاصل، پس از ذخیره کردن شیء، نام آن تایپ می‌شود (p):

```
library(biovizBase)
data(hg19IdeogramCyto, package = "biovizBase")
data(hg19Ideogram, package = "biovizBase")
hg19 <- keepSeqlevels(hg19IdeogramCyto,
    paste0 ("chr", c(1:22, "X", "Y"))) library(ggbio)
p <- ggplot(hg19) + layout_karyogram(cytoband =
RUE)
seqlengths(read) <-
    seqlengths(hg19Ideogram)[names(seqlengths(read))]
p <- p + layout_karyogram(read, geom = "rect",
    ylim = c(11,21), color = "red")
p
```

نمودار حاصل در نگاره‌ی ۱۱-۶ نمایش داده شده است.

۱۱-۲-۵ مصورسازی ساختارهای ژن و رونوشت

مصورسازی رونوشت‌های معلوم توسط بسته‌ی ggbio نسبتاً ساده است. رونوشت‌های انسانی معلوم برای اسمبل hg19 ژنوم انسان به صورت یک بسته در Bioconductor در دسترس است. یک نمودار شامل ساختار ژنی (اگزون‌ها و اینترون‌ها) اطلاعات زنجیره (مستقیم و معکوس) و موقعیت کروموزومی در چندین مرحله ایجاد می‌شود.



نگاره‌ی ۱۱-۶: یک ایدئوگرام از کروموزوم‌های انسانی با موقعیت‌های مربوط به ژن‌های دارای بیان متفاوت که از مجموعه داده‌های parathyroidGenes استنتاج گردیده‌اند.

نخست کلیه‌ی رونوشت‌های معلوم از بسته‌ی Bioconductor تحت عنوان `TxDb.Hsapiens.UCSC.hg19.knownGene` استخراج می‌شوند. سپس موقعیت‌های شروع و پایان (بر حسب جفت باز) از بسته‌ی حاشیه‌نگاری مختص جاندار `org.Hs.eg.db` گرفته می‌شوند. پس از آن همان‌گونه که در بخش قبل نیز برای ایدئوگرام توضیح داده شد، این موقعیت‌ها به یک شیء جی‌رنجز تبدیل شده و با استفاده از تابع `autoplot()` ترسیم می‌شوند. هر رونوشت روی یک خط جداگانه ترسیم می‌شود. ولی این ساختار ژنی می‌تواند کاهش نیز داده شود. یک نسخه‌ی کاهش یافته کلیه‌ی رونوشت‌ها را در بالای یکدیگر ترسیم کرده و کلیه‌ی اگزون‌ها و اینترون‌های آن ژن را روی یک خط واحد دربرمی‌گیرد. موقعیت کروموزومی می‌تواند با کمک تابع `plotIdeogram()` ترسیم شود. سرانجام می‌توان یک نمودار کامل حاوی کلیه‌ی

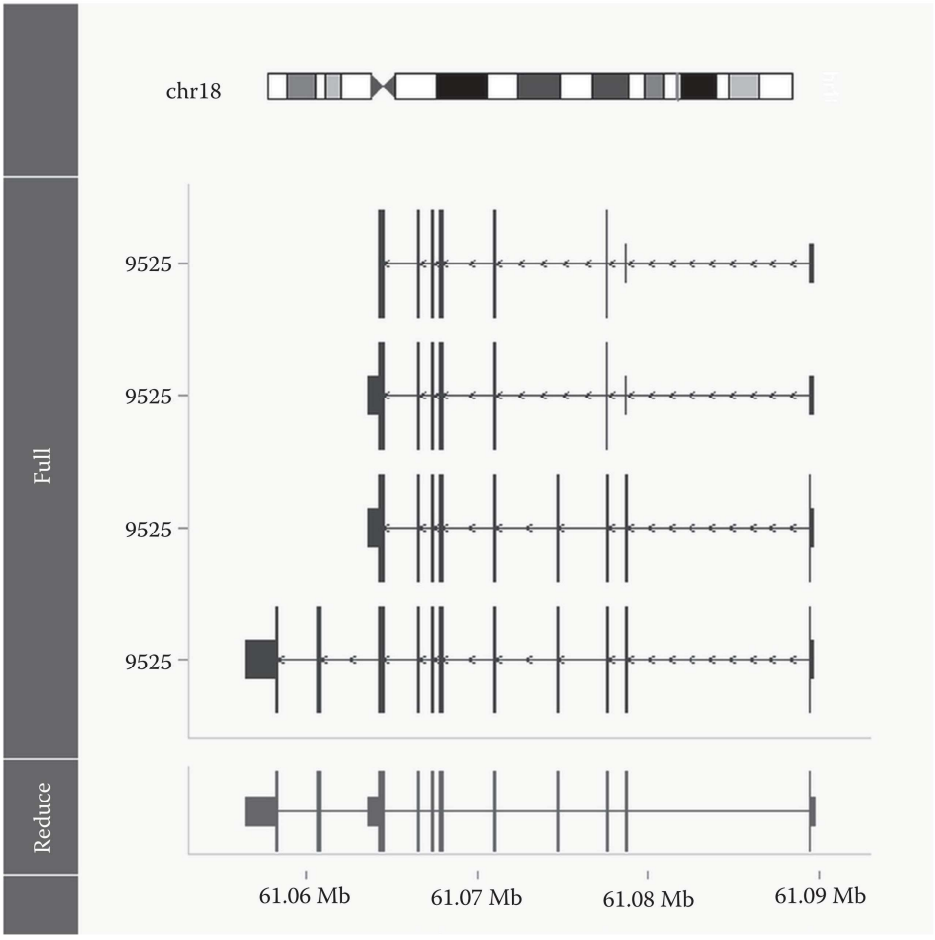
عناصر را با استفاده از تابع tracks() ترسیم نمود. نمودار حاصل در نگاره‌ی ۱۱-۷ نمایش داده شده و کُدنویسی کامل R برای آن نیز در زیر ارائه شده است:

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db) library(ggbio)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
columns <-c ("CHR", "CHRLOC", "CHRLOCEND")
sel <-select(org.Hs.eg.db, "ENSG00000119541",
    columns,keytype = "ENSEMBL")
wh <- GRanges("chr18", IRanges(61056425, 61089752),
    strand = Rle("-"))
p1 <- autoplot(txdb, which = wh, names.expr = "gene_id")
p2 <- autoplot(txdb, which = wh, stat = "reduce",
    color = "brown",fill = "brown")
p.ideo <- plotIdeogram(genome = "hg19", subchr =
"chr18")
tracks(p.ideo, full = p1, reduce = p2,
    heights = c(1.5, 5, 1)) + ylab("") +
    theme_tracks_sunset()
```

۱۱-۳ نهایی کردن نمودارها

کلیه‌ی کُدهای فوق نمودار را روی صفحه‌ی نمایش ایجاد می‌کنند. ذخیره‌ی نمودار به صورت یک فایل تصویری روی سیستم عامل ویندوز و مَک از طریق انتخاب منوهای مناسب امکان‌پذیر است. ولی توصیه می‌شود که کنترل بهتری روی جزییات نمودار ایجاد شده اعمال گردد. همان‌گونه که در مقدمه‌ی این فصل نیز اشاره شد، تغییر وضوح تصاویر شطرنجی یا بیتمپ امکان‌پذیر بوده و علاوه بر آن برای تصاویر وکتور گرافیک نیز می‌توان مدل رنگ تخصیص داد.

در کلیه‌ی موارد، ابتدا باید ابزاری که برای ترسیم نمودار در نظر گرفته شده است، گشوده شود. چندین ابزار گرافیکی برای این منظور می‌تواند مورد استفاده واقع شود. اگر هدف ایجاد یک نمودار با کیفیت انتشار است، احتمالاً فرمت‌های TIFF یا PDF معمول‌ترین فرمت‌های قابل پذیرش هستند. ابزارهای گرافیکی برای این نوع از فایل‌ها با استفاده از دستور tiff() یا pdf() گشوده می‌شوند. پس از گشودن ابزار گرافیکی، دستورات ترسیم نمودار مطابق مثال‌های بالا صادر می‌شوند. پس از صدور دستورات مزبور، ابزار گرافیکی باید بسته شود. این کار توسط دستور dev.off() انجام می‌شود. لازم به یادآوری است که بسته شدن ابزار گرافیکی در این مرحله، حیاتی است. در غیراینصورت کلیه‌ی خروجی‌های بعدی نیز به ابزار گرافیکی و ترسیم نمودار رهنمون می‌شوند.



نگاره‌ی ۱۱-۷: نموداری از ساختارهای ژنی (کاهش یافته) و رونوشت (کامل) برای ژن ENSG00000119541 که دارای دو اگزون با بیان متفاوت در مجموعه داده‌های ENCODE است. این ژن در بازوی بلند کروموزوم ۱۸ واقع شده است.

به عنوان مثال اگر بخواهید برای نتایج حاصل از یک ژن که از یک آنالیز اگزون به دست آمده است، یک نقشه‌ی حرارتی ترسیم کنید، می‌توانید از کُد زیر که یک تصویر tiff ایجاد می‌کند، استفاده نمایید:

```
tiff(filename = "heatmap1.tif", width = 1000, height =
1000)
    vismat <-counts(ecs)
```

```
        [fData(ecs)$geneID = ="ENSG00000119541",]
    colnames(vismat) <-pData(ecs)$samplename
    pheatmap(log2(vismat), cluster_rows = FALSE,
        cluster_cols = FALSE, border_col = "grey95")
dev.off()
```

این کُد یک تصویر tiff ایجاد می‌کند که در عرض و ارتفاع شامل ۱۰۰۰ پیکسل است. این تصویر روی کاغذ احتمالاً اندکی بیشتر از سه اینچ (۸ سانتی‌متر) خواهد بود. اگر کُد بالا را مورد آزمون قرار دهید، احتمالاً خواهید دید که این متن در قالب نمودار بسیار کوچک خواهد شد. بنابراین لازم است که حاشیه‌های نمودار و اندازه‌ی متن را تصحیح کنید. این کارها را می‌توان با تنظیم پارامترهای معمول گرافیکی در R و از طریق تابع `par()` و یا در دستور ترسیم نمودار همانند مثال زیر انجام داد:

```
tiff(filename = "heatmap1.tif", width = 1000, height =
1000)
    vismat <-counts(ecs)
        [fData(ecs)$geneID = ="ENSG00000119541",]
    colnames(vismat) <-pData(ecs)$samplename
    par(mar = c(4,1,1,4))
    pheatmap(log2(vismat), cluster_rows = FALSE,
    cluster_cols = FALSE, border_col = "grey95",
    cex = 1.5)
dev.off()
```

برای یک ابزار گرافیکی PDF ، عرض و ارتفاع بر مبنای واحد پیکسل تخصیص داده نشده و به جای آن از واحد اینچ استفاده می‌شود. در اصل هنوز هم می‌توانید از پیکسل استفاده نمایید. ولی باید آنرا بر وضوح تقسیم نموده و یا اینکه در انتها با یک فایل بسیار بزرگ مواجه خواهید شد. به عنوان مثال، برای داشتن فایلی که با وضوح DPI ۳۰۰ چاپ شود، می‌توانید حجم پیکسل‌ها را بر ۳۰۰ تقسیم کنید:

```
pdf(file = "heatmap1.pdf", width = 1000/300,
    height = 1000/300)
        vismat <-counts(ecs)
            [fData(ecs)$geneID = ="ENSG00000119541",]
        colnames(vismat) <-pData(ecs)$samplename
        pheatmap(log2(vismat), cluster_rows = FALSE,
          cluster_cols = FALSE,
          border_col = "grey95",
          cex = 1)
dev.off()
```

علاوه بر این در صورت لزوم، ایجاد یک فایل PDF در مدل رنگ CMYK امکان‌پذیر است. این کار نیازمند یک برهان اضافی بوده که مدل رنگ را به صورت cmyk دریافت می‌دارد:

```
pdf(file = "heatmap1.pdf", width = 1000/300,
    height = 1000/300, colormodel="cmyk")
        vismat <-counts(ecs)
            [fData(ecs)$geneID = ="ENSG00000119541",]
        colnames(vismat) <-pData(ecs)$samplename
        pheatmap(log2(vismat), cluster_rows = FALSE,
           cluster_cols = FALSE,
           border_col = "grey95",
           cex = 1)
dev.off()
```

۱۱-۴ خلاصه

R می‌تواند انواع متعددی از نمودارها، نظیر نقشه‌ی حرارتی، نمودار آتشفشانی، نمودار MA و ایدئوگرام را ایجاد نماید. همچنین نمودارهای تخصصی‌تر که مصورسازی ساختارهای ژن و رونوشت را امکان‌پذیر می‌سازند، می‌توانند توسط R ایجاد شوند. علاوه براین، R توابعی برای ذخیره‌سازی نمودارها در فرمت‌های مختلف، نظیر پستاسکریپت، PDF و TIFF داشته ولی مدل‌های رنگ مختلف تنها توسط فرمت‌های وکتور گرافیک پشتیبانی می‌گردند.

منابع

1. Murrell P. R Graphics, Boca Raton: CRC Press, 2011, ISBN9781439831762.
2. Cui X. and Churchill G.A. Statistical tests for differential expression in cDNA microarray experiments. Genome Biol 4(4):210, 2003. Epub Mar 17, 2003 [Review]. PubMed PMID: 12702200; PubMed Central PMCID: PMC154570.
3. Yang Y.H., Dudoit S., Luu P., Lin D.M., Peng V., Ngai J., and Speed T.P. Normalization for cDNA microarray data: A robust composite method addressing single and multiple slide systematic variation. Nucleic Acids Res 30(4):e15, 2002. PubMed PMID: 11842121; PubMed Central PMCID: PMC100354.

فصل دوازدهم

RNA های نارمزگر کوچک

۱۲–۱ مقدمه

RNA های نارمزگر کوچک شامل چند دسته از مولکول‌های فعال زیستی بوده که طیف وسیعی از فعالیت‌های زیستی، از زمان‌بندی تکامل اولیه تا مرگ برنامه‌ریزی شده‌ی سلول، را کنترل کرده و تغییر می‌دهند. آنها از پیش از نخستین تقسیم سلولی تا مراحل نهایی پیری بیان می‌شوند. این RNA ها در اووسیت‌ها، سلول‌های بنیادی، سلول‌های گلیال، سلول‌های سوماتیک و نیز در سلول‌های سرطانی یافت می‌شوند. علی‌رغم اینکه مطالعات اولیه روی شناسایی و توضیح عملکرد RNA های کوچک در ترجمه (tRNA ها و snoRNA ها) متمرکز بوده است، ولی مطالعات اخیر بر RNA های نارمزگری که رونویسی را کنترل کرده و تغییر می‌دهند (miRNA ها، piRNA ها و endo-siRNA ها) تمرکز یافته‌اند. همچنین مطالعات اولیه عمدتاً روی روش‌های متداول زیست‌شناسی مولکولی نظیر همسانه‌سازی و توالی‌یابی دی‌دئوکسی برای شناسایی اعضای هر دسته تاکید کرده‌اند. ولی در حال حاضر روش‌های توالی‌یابی RNA گسترش و عمق بی‌سابقه‌ای در توالی‌یابی اعضای دسته‌ی RNA های کوچک ایجاد کرده‌اند. با پیشرفت‌های عظیمی که در کارآمدی و دقت روش‌های توالی‌یابی نسل جدید ایجاد شده است، چالش فعلی عبارت از شناسایی، حاشیه‌نگاری و تهیه‌ی پروفایل RNA های کوچک شناخته شده و کشف RNA های جدید در داخل یک مجموعه داده‌ی توالی‌یابی است. روش‌های فعلی توالی‌یابی RNA های کوچک به آسانی نمی‌توانند بین دسته‌های مختلف RNA های کوچک تمایز قائل شده و بنابراین آگاهی و شناخت دسته‌های مختلف، ویژگی‌های‌شان و رابطه‌ی آنها با یکدیگر برای طرح‌ریزی و اجرای آزمایشات با هدف شناسایی و کمی‌سازی اعضای یک دسته‌ی خاص از RNA های نارمزگر کوچک، مفید خواهند بود. در بخش‌های بعد دسته‌های اصلی RNA های نارمزگر کوچک توضیح داده می‌شوند. در این توضیحات روی سامانه‌های حیوانی تاکید شده است. خلاصه‌ای از دسته‌های مختلف در جدول ۱۲-۱ ارائه شده است. همزمان با افزایش روزافزون داده‌های توالی‌یابیRNA های کوچک این موضوع بیشتر نمایان می‌شود که دسته‌های متعدد مختلفی از RNA های نارمزگر موجود بوده و دسته‌های جدیدی نیز در حال شناسایی هستند. در بخش‌های بعدی تلاش می‌شود که حداقل دسته‌هایی که تاکنون به خوبی شناسایی شده‌اند، بهتر نیز معرفی شوند. فهرست زیر فهرستی جامع

از کلیه‌ی دسته‌های RNA های نارمزگر کوچک نیست. ولی نقطه‌ی شروع قابل قبولی برای فهم دسته‌های اصلی که در حال حاضر شناخته شده‌اند، محسوب می‌گردد.

جدول ۱۲-۱: دسته‌های اصلی RNA های نارمزگر کوچک

دسته	اندازه	زیست‌زایی	تعداد اعضا (در انسان)	عملکرد
ریز RNA (miRNA)	۲۱ تا ۲۳ نوکلئوتید	پردازش شده توسط دایسر از پیش‌سازهای ۶۵ تا ۷۰ نوکلئوتیدی	>۲۵۰۰	تنظیم بیان ژن
RNA پیوی (piRNA)	۲۵ تا ۳۳ نوکلئوتید	پیش‌سازهای هسته‌ای تکثیر شده توسط سازوکار پینگ‌پنگ در سیتوپلاسم	>۲۰۰۰۰	تنظیم رتروترانسپوزان‌ها
RNA سرکوبگر درون‌زاد (endo-siRNA)	۲۱ تا ۲۶ نوکلئوتید	پردازش شده از رونوشت‌های RNA حامل	نامعلوم	تنظیم بیان ژن
RNA هستکی کوچک (snoRNA)	۶۰ تا ۳۰۰ نوکلئوتید	پردازش شده از اینترون‌های RNA حامل	>۲۶۰	مشارکت در تغییر شیمیایی سایر RNA ها
RNA هسته‌ای کوچک (snRNA)	۱۵۰ نوکلئوتید	RNA پُلی‌مراز II و III	۹ خانواده	مشارکت در سرکوب سایر RNA ها
RNA ناقل (tRNA)	۷۳ تا ۹۳ نوکلئوتید	RNA پُلی‌مراز III	>۵۰۰	ترجمه‌ی mRNA به پروتئین
RNA تعدیل کننده‌ی ریز RNA (moRNA)	۱۹ تا ۲۳ نوکلئوتید	پردازش شده از پیش‌ساز miRNA	نامعلوم	نامعلوم
RNA تقویت کننده (eRNA)	۵۰ تا ۲۰۰۰ نوکلئوتید	رونویسی نوآیند RNA	>۲۰۰۰	تنظیم بیان ژن مبدا

۱۲-۲ ریز RNA ها (miRNA ها)

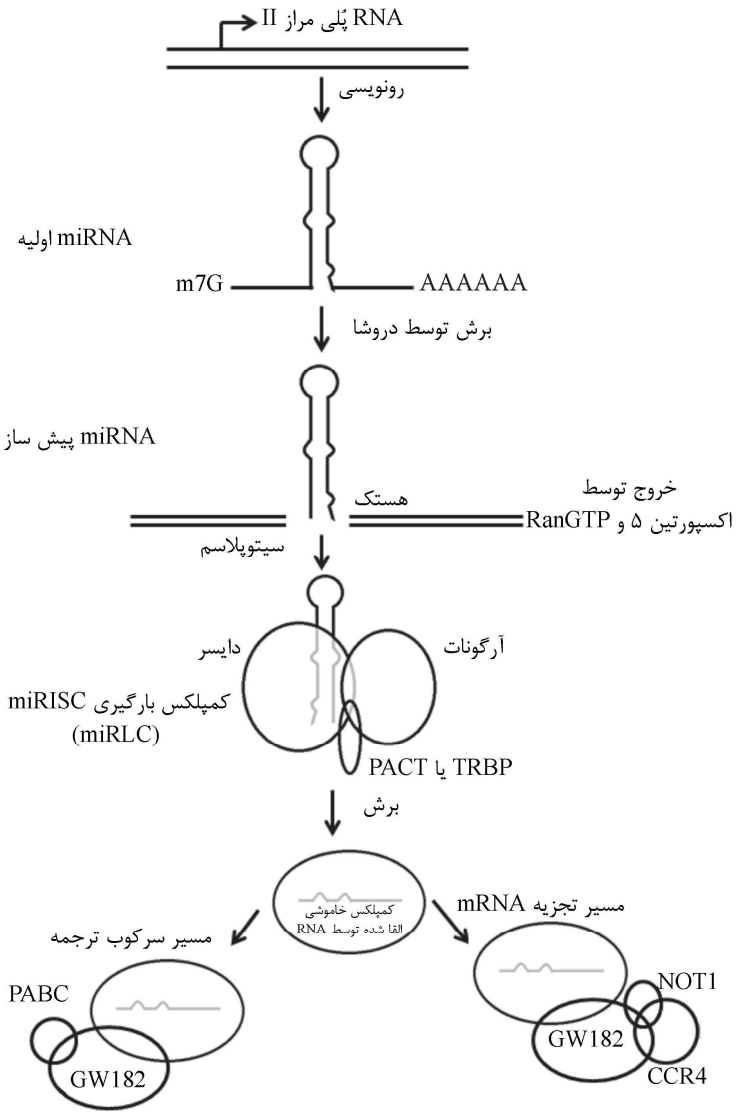
ریز RNA ها (miRNA ها) مولکول‌های کوچک ۲۱ تا ۲۳ نوکلئوتیدی هستند که از پردازشmiRNA های اولیه و سپس مولکول‌های پیش‌ساز miRNA با طول حدود ۷۰ نوکلئوتید ایجاد می‌شوند. این RNA ها متنوع، فراوان و از نظر تکاملی حفاظت شده هستند. نخستین عضو این خانواده که *lin-4* نام داشت، به طور تصادفی در غربال‌های ژنتیکی که برای شناسایی ژن‌های جهش یافته‌ی عامل ایجاد عیوب دودمانی در تکامل *Caenorhabditis elegans* صورت می‌گرفت، یافت شد (۱). حیوانات جهش یافته‌ی *lin-4* متعلق به دسته‌ای از جهش یافته‌های هتروکرونیک هستند که به دلیل نقایص زمان‌بندی فرآیندهای تکاملی، در تکامل ساختارهای بالغ نظیر واژن دچار اختلال و مشکل هستند. مشخص شده است که نوع وحشی ژن *lin-4* که حیوانات جهش یافته را از این نقایص رهایی می‌بخشد، پروتئینی را رمز نکرده ولی یک RNA کوچک ۲۲ نوکلئوتیدی را که از یک پیش‌ساز سنجاق‌سری کوچک پردازش می‌شود، رمز می‌نماید. همچنین مشخص شده است که توالی ژن RNA کوچک *lin-4* با چندین موقعیت در ناحیه‌ی ناترجمان '3 ژن *lin-14* مکملی بالایی دارد. چون پروفایل بیان *lin-4* و *lin-14* همبستگی معکوس دارند و *lin-4* سرکوبگر *lin-14* است، این فرضیه مطرح شده است که *lin-4* از طریق شناسایی و سرکوب فرآیند ترجمه، *lin-14* را هدف قرار داده و نهایتاً منجر به کاهش بیان پروتئین *lin-14* می‌گردد. اندازه‌ی برآورد شده‌ی رونوشت *lin-4* تقریباً ۲۲ نوکلئوتید بوده و پیش‌ساز احتمالی آن نیز حدود ۶۱ نوکلئوتید طول دارد. پس از نخستین مقالات منتشر شده، گزارشات بسیار اندکی در این زمینه ارائه گردید تا اینکه ژن دیگری تحت عنوان *let-7* که سبب نقص دودمان سلولی در *C. elagans* می‌گردید، همسانه‌سازی شده و مشخص گردید که یک RNA نارمزگر ۲۲ نوکلئوتیدی است. پس از آن اورتولوگ‌های *let-7* در انسان، مگس سرکه و سایر جانداران عالی‌تر شناسایی شده و این امر منجر به جذب شدید محققین و تلاش مضاعف آنها در جهت همسانه‌سازی و شناسایی همزمان صدها miRNA جدید گردید (۲، ۳، ۴ و ۵). در حال حاضر مشخص شده است که miRNA ها توالی‌های حفاظت شده‌ای بوده که در گونه‌های متعدد دیگر شامل انسان، گیاهان و سایر نماتدها یافت می‌شوند. تاکنون بیش از ۲۴۰۰۰ مدخل در پایگاه miRBase ثبت شده است.

بخش زیادی از تلاش‌ها بر شناسایی سازوکار زیست‌زایی[1] miRNA ها معطوف شده است. ابتدا ژن‌های miRNA توسط RNA پُلی‌مراز II (همان RNA پُلی‌مرازی که mRNA ها را تولید می‌کند) رونویسی می‌شوند. ممکن است که برخی از miRNA ها توسط RNA پُلی‌مراز III رونویسی شوند. ژن miRNA که در ابتدا رونویسی می‌شود (miRNA اولیه یا pri-miRNA)، ممکن

1- Biogenesis

است که به صورت بینابینی در حد فاصل بین ژن‌های رمزگر پروتئین‌ها، در داخل اینترون، در ناحیه‌ی رمزگر یا در ناحیه‌ی ناترجمان یک mRNA واقع شده باشد. miRNA های متعددی نیز از پروموتورهای اختصاصی خودشان رونویسی می‌شوند. اندازه‌ی miRNA اولیه شامل دامنه‌ای وسیع، از چند صد نوکلئوتید تا چندین کیلو جفت باز است. مرحله‌ی بعدی شامل ایجاد یک ساختار حد واسط سنجاق‌سری[1] بالغ به طول ۶۰ تا ۷۰ نوکلئوتید است که اصطلاحاً miRNA پیش‌ساز یا pre-miRNA نامیده می‌شود. مرحله‌ی بلوغ توسط اندونوکلئاز DroshaRNase III در یک کمپلکس بزرگ تقریباً ۶۵۰ کیلو دالتونی با DGCR8/Pasha (ژن ۸ ناحیه‌ی حیاتی سندروم دی‌جورج) که حاوی دو RNA دو رشته‌ای (dsRNA) متصل به دُمین‌ها است، صورت می‌گیرد. این کمپلکس اغلب کمپلکس پردازش pri-miRNA یا کمپلکس ریزپردازنده[2] نامیده می‌شود. نتیجه‌ی پردازش pri-miRNA ایجاد یک ساختار ساقه و حلقه با یک برآمدگی دو نوکلئوتیدی در انتهای '3 است که اصطلاحاً pre-miRNA نامیده می‌شود. سپس این pre-miRNA با کمک Ran-GTP و اِکسپورتین ۵ از هستک به سیتوپلاسم منتقل می‌شود. در سیتوپلاسم، اندونوکلئاز RNAase دیگری تحت عنوان دایسر[3] pre-miRNA دو رشته‌ای را شناسایی کرده و هر دو رشته‌ی آنرا برش می‌زند. دایسر هر دو انتهای برآمدگی‌های '5 و '3 را بریده و حلقه را برش داده و در نتیجه این کار یک RNA دو رشته‌ای ۲۲ نوکلئوتیدی با یک برآمدگی ۲ نوکلئوتیدی در '3 در هر دو رشته ایجاد می‌شود. سپس از این دو رشته‌ای، مولکول راهنمای[4] فعال در داخل یک پروتئین آرگونات[5] موجود در کمپلکس خاموشی RNAi که تحت عنوان کمپلکس RISC شناخته می‌شود، بارگیری[6] شده و رشته‌ی دیگر که تحت عنوان زنجیره‌ی مسافر[7] نامیده می‌شود نیز تخریب می‌گردد. پس از آن کمپلکس RISC که حاوی چندین پروتئین دیگر از جمله GW182 (که در *C. elegans* تحت عنوان AIN-1 شناخته می‌شود) است، mRNA را مورد هدف قرار می‌دهد. مولکول راهنما، کمپلکس RISC را به نزدیکی mRNA هدف هدایت کرده و سپس آنزیم‌ها و کوفاکتورهایی را که در خاموشی RNA از طریق سرکوب ترجمه، از بین بردن پایداری mRNA ، تخریب mRNA و یا ترکیبی از این سه روش موثر هستند، به کار می‌گیرد. کلیات مراحل مختلف پردازش miRNA در نگاره‌ی ۱۲-۱ ارائه شده است.

1- Hairpin
2- Microprocessor complex
3- Dicer
4- Guide molecule
5- Argonaute
6- Load
7- Passenger strand



نگاره‌ی ۱۲-۱: مسیر زیست‌زایی و پردازش miRNA

در نگاره‌ی ۱۲-۱، ژن‌های miRNA برای تولید pri-miRNA ها، با کمک RNA پُلی‌مراز II در هستک رونویسی می‌شوند. سپس این مولکول‌ها توسط آنزیم دروشا[1] (یک اندونوکلئاز RNAase III

1- Drosha

در کمپلکس پردازش pri-miRNA برای تولید ساختار سنجاق‌سری pre-miRNA) که بخشی از کمپلکس DGCR8 است، برش داده می‌شود. این ساختار سنجاق‌سری توسط اِکسپورتین ۵ و RanGTP به سیتوپلاسم ارسال می‌گردد. در سیتوپلاسم، ساختار سنجاق‌سری توسط دایسر (اندونوکلئاز RNAase III دیگر در کمپلکس بارگیری miRNA (miRLC)) تحت پردازش بیشتر قرار گرفته و رشته‌ی راهنما و رشته‌ی مسافر miRNA بالغ تولید می‌گردد. سپس رشته‌ی راهنما در داخل پروتئین آرگونات که با GW182 مرتبط است، بارگیری می‌شود. سایر پروتئین‌های مرتبط به مسیر کمپلکس خاموشی القا شده توسط RNA (مسیر RISC) وابسته هستند. در مسیر سرکوب ترجمه، پروتئین متصل شونده به پُلیA ، PABP ، GW182 و پروتئین‌های ماشین ترجمه برای توقف ترجمه برهمکنش دارند. در مسیر تجزیه‌ی mRNA ، کمپلکس CCR4:NOT که حاوی حداقل پنج پروتئین CCR یا NOT است، برای دِآدنیلاسیون mRNA های پُلی‌آدنیله نقش ایفا می‌کنند. البته بعداً آنزیم کلاهک‌بردار mRNA 1/2 (DCP1/2)[1] برای حذف کلاهک m7G در mRNA ها عمل می‌نماید. در یک مدل نیز پیشنهاد شده است که تجزیه‌ی mRNA پس از سرکوب ترجمه و به صورت متوالی و پشت سر هم صورت می‌گیرد (۶).

با توجه به شناسایی تعداد زیادی miRNA ، همپوشانی توالی‌های‌شان و نیز حفاظت فیلوژنتیکی آنها، یک فهرست اصطلاحات برای این مولکول‌ها تهیه شده و در جدول ۱۲-۲ ارائه شده است.

ایزومیرها[2]، ایزوفرم‌های miRNA های بالغ بوده و احتمالاً به دلیل اشتباه در پردازش در طی زیست‌زایی miRNA ها، در تعداد بسیار اندکی از نوکلئوتیدها در جهت '5 یا '3 با آنها تفاوت دارند (۷). همچنین ایزومیرها می‌توانند از طریق ویرایش کردن که ممکن است یک یا چند یوریدین به انتهای '3 یک miRNA بالغ بیافزاید یا از طریق ویرایش آدنوزین به اینوزین در انتهای مولکول‌های RNA دو رشته‌ای توسط آنزیم آدنوزین دِآمیناز، نیز ایجاد شوند. ایزومیرهای اختصاصی، فراوان، تنظیم شده، حفاظت شده و عملکردی بوده و بنابراین از نظر زیستی معنی‌دار و مهم تلقی می‌شوند.

علاوه بر این miRNA ها می‌توانند از توالی‌های اینترونی و از طریق پیرایش و نیز پس از تاب خوردن و تشکیل ساختارهای سنجاق‌سری pre-miRNA در مسیر میرترون[3] و از طریق شاخه‌زُدایی توسط آنزیم شاخه‌زُدای لریت[4]، پردازش گردند. هنگامی‌که ساختارهای سنجاق‌سری کوتاه از

1- mRNA-decapping enzyme 1/2 (DCP1/2)
2- IsomiR
3- Mirtron
4- Lariat debranching enzyme

جدول ۱۲-۲: فهرست اصطلاحات miRNA

اصطلاحات	تعریف	مثال
مشخصه‌ی گونه	مخففی از ترکیب سه یا چهار حرف پیشوند است.	mus-miR-101، (*Homo sapiens*) has-miR-101 (*Mus musculus*)، dme-miR-101 (*Drosophila melanogaster*)، cbr-miR-101 (*Caenorhabditis briggsae*)
مشخصه‌ی بالغ	توالی‌های بالغ پردازش شده توسط دایسر که با حرف بزرگ R مشخص می‌شوند.	hsa-miR-7
مشخصه‌ی پیش‌ساز	پیش‌سازهای سنجاق سری که با حرف کوچک r مشخص می‌شوند.	hsa-mir-7
مشخصه‌ی عددی	اعداد متوالی که بر مبنای سابقه‌ی تاریخی تخصیص داده می‌شوند (miRNA هایی که زودتر کشف شده‌اند، با اعداد کوچک‌تر و آنهایی که اخیراً شناسایی شده‌اند، با اعداد بزرگ‌تر مشخص می‌گردند). اورتولوگ‌ها در گونه‌های مختلف شماره‌های یکسانی دارند.	mir-1 ، mir-2 ، mir-3
مشخصه‌ی پارالوگ	پارالوگ‌هایی که در توالی بالغ در ۱ تا ۲ نوکلئوتید تفاوت دارند، یک حرف به صورت پسوند بعد از مشخصه‌ی عددی دریافت می‌دارند.	mmu-miR-10a و mmu-miR-10a پارالوگ‌های mir-10 در موش هستند.
پیش‌سازهای متمایز سنجاق‌سری با توالی miRNA بالغ یکسان	پسوند عددی دارند.	dme-mir-281-1 و dme-mir-281-2
شکل اصلی و شکل جزیی	شکل اصلی که بیشتر بیان می‌شود، بر مبنای اصولی که در فوق تشریح گردیدند، نام‌گذاری شده و شکل فرعی علاوه بر آن یک ستاره (*) نیز دریافت می‌کند.	mmu-miR-124 شکل اصلی بوده و mmu-miR-124* شکل فرعی را نشان می‌دهد.
بازوی ’5 و بازوی ’3	به جای شکل ستاره‌دار (*) از یک سامانه با ابهام کمتر استفاده می‌شود. در این سامانه miRNA بالغ در بازوی ’5 پیش‌ساز با 5p مشخص شده و miRNA بالغ حاصل از بازوی ’3 ، پسوند 3p دریافت می‌کند.	miR-124-5p و miR-124-3p
خوشه‌ی miRNA	چند miRNA که از روی یک RNA رونویسی می‌شوند، یا ژن چند miRNA که روی یک کروموزم و در نزدیکی هم واقع می‌گردند.	miR-124-5p یک رونوشت است که از آن شش ژن miRNA و دوازده miRNA بالغ پردازش می‌شود: miR-17 ، miR-18a ، miR-19a ، miR-20a ، miR-19b-1 و miR-92a-1
خانواده‌های miRNA	مطابق با توالی‌های سید تعریف می‌شوند.	let-7 مهره‌داران که در خانواده‌ی *let-7* قرار دارد، شامل mir-98 و mir-202 می‌شود.

پیرایش اینترون‌ها حاصل می‌آیند، کمپلکس ریزپردازنده را کنار گذاشته و miRNA های بالغ در سیتوپلاسم توسط دایسر برش داده شده و پس از آن می‌توانند در مسیر پردازش miRNA همراه با سایر miRNA ها مورد پردازش واقع شوند (۸). در ابتدا میرترون‌ها از *D. melanogaster* و *C. elegans* شناسایی و کلون شدند. زیرا این جانداران دارای ژنوم‌های متراکم و در نتیجه اینترون‌های سنجاق‌سری کوتاه متعدد هستند. پس از آن میرترون‌ها در انواع مختلفی از واریته‌ها شامل انسان، نخستی‌های دیگر و گیاهان یافت شدند.

۱۲-۳ RNA های تعدیل کننده‌ی ریز RNA ها (moRNA ها)

RNA های تعدیل کننده‌ی ریز RNA ها (moRNA ها) مولکول‌های کوچک RNA هستند که از ساختار سنجاق‌سری pre-miRNA همانند miRNA ها مشتق می‌شوند. moRNA ها در مجاورت miRNA بالغ در هر دو بازوی '5 و یا '3 قرار گرفته و اندازه‌ای برابر با اندازه‌ی miRNA های بالغ دارند. این RNA ها ابتدا در طنابدار ساده‌ی *Ciona intestinalis* شناسایی شده و سپس در کتابخانه‌های حاصل از مغز انسان و از طریق آنالیزهای توالی‌یابی RNA شناسایی گردیدند (۹ ، ۱۰ و ۱۱). این RNA ها در مقایسه با miRNA ها از فراوانی کمتری برخوردار بوده و در حال حاضر عملکرد و جزییات پردازش آنها از طریق ساختار سنجاق‌سری نامعلوم است.

۱۲-۴ RNA های مرتبط با پیوی (piRNA ها)

RNA های مرتبط با پیوی[1] (piRNA ها) نیز RNA های نارمزگر کوچک هستند. ولی این RNA ها الگوهای بیان، زیست‌زایی و عملکرد منحصر به فردی دارند. piRNA ها تقریباً ۲۵ تا ۳۳ نوکلئوتید طول داشته و در سلول‌های رده‌ی زاینده (به ویژه در بیضه‌ها) بیان می‌شوند. ولی می‌توانند در سلول‌های ماده نیز یافت شوند. این RNA ها بر مبنای ارتباطشان با پروتئین‌های PIWI که دسته‌ای از آرگونات‌ها هستند، تعریف می‌شوند. همچنین این توالی‌ها در نخستین نوکلئوتیدشان دارای اُریبی به سمت U هستند. این RNA ها توالی‌های متحرک DNA ژنومی رده‌های زاینده را که معمولاً عناصر قابل انتقال[2] هستند، خاموش می‌کنند. عناصر قابل انتقال، توالی‌های متحرک DNA در یک ژنوم هستند. این عناصر به رتروترانسپوزان‌ها (رونویسی این عناصر نیازمند RNA ، رونویسی معکوس به DNA و سپس الحاق مجدد به ژنوم است) وترانسپوزان‌های DNA (نیازمند یک ژن فعال آنزیم ترانسپوزاز است که محصول این ژن، جدا شدن و پیوستن یک

1- Piwi-associated RNAs (piRNAs)
2- Transposable element

عنصر قابل انتقال DNA را کاتالیز می‌کند) دسته‌بندی می‌شوند. معمول‌ترین عنصر قابل انتقال در ژنوم انسان توالی Alu است که طول آن تقریباً ۳۰۰ جفت باز بوده و چند صد هزار کُپی از آن وجود دارد. piRNA ها در ابتدا به صورت piRNA های اولیه از نواحی خاصی از کروموزوم که می‌توانند از چند کیلو جفت باز تا بیش از ۲۰۰ کیلو جفت باز متغیر باشند، رونویسی می‌شوند. این نواحی اصطلاحاً خوشه‌های piRNA نامیده شده و هر خوشه می‌تواند شامل ده‌ها یا هزاران توالی piRNA باشد. توالی‌های piRNA از هر خوشه می‌توانند همپوشانی داشته و همچنین از هر دو زنجیره در داخل آن خوشه رونویسی شوند. سپس piRNA های اولیه در یکی از سه آرگونات MILI ، MIWII یا MIWIII در موش و PIWI ، AUB یا AGO3 در *D. melanogaster* بارگیری می‌شوند. لازم به ذکر است که هر آرگونات از نظر اندازه اختصاصی عمل می‌کند. به عنوان مثال در *D. melanogaster* ، اندازه‌ی piRNA ها متناظر با آرگونات‌های PIWI ، AUB و AGO3 به ترتیب ۲۵ ، ۲۴ و ۲۳ نوکلئوتید است. پس از آن piRNA های اولیه یک مرحله زیست‌زایی ثانویه‌ی منحصر به فردی را می‌گذرانند که در طی آن تکثیر می‌گردند. piRNA های بامعنا[1] که در AUB بارگیری شده‌اند، با زنجیره‌های پادمعنا[2]، مکمل گردیده و ۱۰ نوکلئوتید از انتهای '5 آنها برش داده می‌شود. سپس piRNA ثانویه در AGO3 بارگیری شده و با زنجیره‌ی مستقیم یک رونوشت ترانسپوزان فعال پیوند حاصل کرده و ۱۰ نوکلئوتید از انتهای '5 آن برش خورده و سپس یک piRNA از توالی ترانسپوزان بازتولید می‌شود. این فعالیت برشی اصطلاحاً برش دهنده[3] (اسلایسر) نامیده شده و در داخل AUB و AGO3 رُخ می‌دهد. این حلقه‌ی تکثیر نیز اصطلاحاً چرخه‌ی تکثیر پینگ‌پنگ[4] نامیده می‌شود.

۱۲–۵ RNA های سرکوبگر درون‌زاد (endo-siRNA ها)

RNA های سرکوبگر درون‌زاد (endo-siRNA ها) با منشاء درونی به RNA های کوچک ۲۱ نوکلئوتیدی رونویسی می‌شوند. این RNA ها از محصولات RNA دو رشته‌ای که می‌توانند از رونویسی جفت‌های کوتاه بامعنا – پادمعنا، تکرارهای معکوس شده‌ی تک زنجیره‌ی RNA و یا هیبرید شدن شبه‌ژن‌های پادمعنا با ژن‌های رمزگر بامعنا ایجاد شوند. جفت‌های بامعنا – پادمعنا می‌توانند از رونویسی یک ژنگاه در هر دو جهت ایجاد شده (که اصطلاحاً *cis*-dsRNA نامیده می‌شوند) یا از ژنگاه‌های متمایز با توالی‌های مکمل ایجاد گردند (که اصطلاحاً *trans*-dsRNA

1- Sense
2- Antisense
3- Slicer
4- Ping-pong amplifcation cycle

نامیده می‌شوند). پیش‌سازهای بامعنا – پادمعنا نیز می‌توانند از ساختارهای سنجاق‌سری ایجاد شوند. ولی به دلیل طول بیشتر در بخش ساقه و نیز پردازش توسط DICER2 به جای DICER1 ، با ساختارهای سنجاق‌سری حاصل از miRNA ها تفاوت دارند. ممکن است که مکمل بودن *trans*-dsRNA و زنجیره‌ی بلند ساختار سنجاق‌سری دقیق نبوده و منجر به برآمدگی‌های متعدد در dsRNA شده و همچنین ممکن است ویرایش گردد. سپس در *Drosophila* ، مولکول endo-siRNA اولیه در AGO2 بارگیری می‌شود. علاوه بر *Drosophila* ، مولکول‌های endo-siRNA در اووسیت‌های موش‌ها، سلول‌های بنیادی جنینی، گیاهان و *C. elegans* نیز یافت می‌شوند (۱۲). در *C. elegans*، این endo-siRNA ها می‌توانند از طریق RNA پُلی‌مرازهای وابسته به RNA و برای ایجاد endo-siRNA های ثانویه تکثیر شوند. علی‌رغم اینکه ماشین‌های هدف‌گیری کمتر شناخته شده هستند، بر خلاف جفت شدن سیدها در miRNA ها، توالی‌های endo-siRNA به دقت با توالی‌های مکمل هدف‌شان انطباق می‌یابند.

۱۲–۶ RNA های سرکوبگر برون‌زاد (exo-siRNA ها)

RNA های سرکوبگر برون‌زاد[1] (exo-siRNA ها) با منشاء بیرونی منبع RNA های کوچکی که تقریباً ۲۱ نوکلئوتید طول دارند، می‌شوند. این RNA ها از طریق آلودگی ویروسی در یک وضعیت طبیعی و از طریق ترازهش[2] و تراآلایی[3] توالی‌های DNA به سلول وارد شده و سپس به صورت درون‌زاد به توالی‌های dsRNA رونویسی می‌گردند. ویروس هپاتیت C که یک ویروس با RNA دو رشته‌ای است، مثالی از یک منبع برون‌زاد می‌باشد. توالی‌های dsRNA توسط سلول شناسایی شده و به وسیله‌ی DICER2 هضم می‌شوند. در *C. elagans* ، مولکول‌های exo-siRNA اولیه‌ی ایجاد شده توسط DICER می‌توانند به وسیله‌ی RNA پُلی‌مرازهای وابسته به RNA تکثیر شوند. این موضوع نشان دهنده‌ی آن است که مسیرهای exo-siRNA و endo-siRNA در *C. elegans* ممکن است که در برخی از مولفه‌های محدود کننده مشترک باشند (۱۳).

۱۲–۷ RNA های ناقل (tRNA ها)

RNA های ناقل (tRNA ها) در کلیه‌ی جانداران حضور داشته و طول‌شان بین ۷۳ تا ۹۵ نوکلئوتید متغیر است. این RNA ها توسط RNA پُلی‌مراز III رونویسی شده و عملکردشان بر

1- Exogenous silencing RNA (exo-siRNA)
2- Transduction
3- Transfection

مبنای ساختارهای دوم و سوم استوار است. tRNA ها با آوردن اسیدهای آمینه به ریبوزوم برای ساخت پروتئین نقش حیاتی در ترجمه ایفا می‌کنند. تعداد ژن‌های tRNA یک جاندار از ۱۷۰ تا ۵۷۰ ژن در معمول‌ترین گونه‌های مطالعه شده متغیر بوده و در انسان نیز تعداد ژن‌های مزبور ۴۹۷ ژن است (۱۴).

۱۲–۸ RNA های هستکی کوچک (snoRNA ها)

RNA های هستکی کوچک[1] (snoRNA ها) RNA های هسته‌ای بلند ۶۰ تا ۱۵۰ نوکلئوتیدی بوده که نقش‌شان هدایت پردازش مولکول‌های RNA ریبوزومی است. این کار از طریق برقراری ارتباط با پروتئین‌های موجود در کمپلکس ریبوزومی کوچک هستکی صورت می‌گیرد. تاکنون تقریباً ۴۰۰ مولکول snoRNA در انسان شناسایی شده است (۱۵). یک snoRNA حاوی ۱۰ تا ۲۰ نوکلئوتید پادمعنایِ هدفِ RNA ریبوزومی بوده و از این نوکلئوتیدها برای هدایت تغییرات rRNA ها (شامل متیلاسیون‌ها[2] و سودویوریدیلاسیون‌ها[3]) استفاده می‌کند. معمولاً snoRNA هایی که در اینترون‌های پروتئین‌ها واقع شده‌اند، در ساختن ریبوزوم نقش داشته و بنابراین توسط RNA پُلی‌مراز II رونویسی می‌شوند. ولی می‌توانند توسط پروموتور احتصاصی خودشان نیز رونویسی گردند. snoRNA ها ساختارهای حفاظت شده‌ای شامل یک جعبه‌ی C/D با دو توالی موتیف حفاظت شده و یک جعبه‌یH/ACA دارند. جعبه‌یH/ACA در snoRNA ها دو ساختار سنجاق‌سری تشکیل می‌دهد.

۱۲–۹ RNA های هسته‌ای کوچک (snRNA ها)

RNA های هسته‌ای کوچک (snRNA ها) RNA های هسته‌ای بلند با طول تقریبی ۱۵۰ نوکلئوتید بوده که به دلیل برخورداری از محتوای بالای یوریدین، تحت عنوان U-RNA ها نیز شناخته می‌شوند. این RNA ها عمدتاً توسط RNA پُلی‌مراز II یا III (U6) رونویسی شده و وظیفه‌ی پردازش RNA هسته‌ای ناهمگون[4] در کمپلکس‌های ریبوپروتئین هسته‌ای کوچک را بر عهده دارد. بنابراین snRNA هایی نظیر U1 ، U2 ، U4 ، U5 و U6 بخشی از اسپلیسوزوم[5] که اینترون‌ها را در طی پردازش mRNA برش می‌دهد، هستند. جانداران عالی‌تر دارای ۵ تا ۳۰ مولکول snRNA هستند.

1- Small nucleolar RNAs (snoRNA)
2- Methylation
3- Pseudouridylation
4- Heteronuclear RNA
5- Spliceosome

۱۲–۱۰ RNA های تقویت کننده (eRNA ها)

RNA های تقویت کننده (eRNA ها) RNA های کوتاهی هستند که از توالی‌یابی محصولات در طی رونویسی نوآیند[1] شناسایی گردیده و با مناطق تقویت کننده مکان‌یابی می‌شوند (۱۶). محصولات رونویسی نوآیند را می‌توان با استفاده از رسوب ایمنی[2] (ایمیونوپرسیپیتاسیون) و نیز توالی‌یابی مداوم کُلی[3] (توالی‌یابی GRO) شناسایی نمود. تقویت‌کننده‌ها در مبداء[4] ژن‌های رمزگر قرار داشته و eRNA ها می‌توانند در هر دو جهت بامعنا و پادمعنا با توجه به ژن مبداء رونویسی شوند. مطالعات اخیر نشان داده است که چند هزار eRNA در بافت‌های سلولی پس از القای هورمونی یا الکتروفیزیولوژی تولید می‌شوند (۱۷). مطالعات کشت سلولی نیز نشان می‌دهند که eRNA ها می‌توانند از طریق به کارگیری یا برهمکنش با عوامل کنترل کننده‌ی ساختار حلقه‌های کروماتین که بین تقویت‌کننده و ژن مبداءشان تشکیل می‌شوند، هم رونویسی ژن‌های مبداء را تقویت کرده و هم آنرا سرکوب نمایند.

۱۲–۱۱ سایر RNA های نارمزگر کوچک

سایر RNA های نارمزگر کوچک شامل tRNA های مشتق از قطعات RNA[5] (tRF)، tRNA های مشتق از RNA های کوچک[6] (tsRNA)، snoRNA های مشتق از RNA[7] (sdRNA)، RNA های کوچک مرتبط با پروموتور[8] و RNA های مرتبط با محل آغاز رونویسی[9] (TSSa-RNA) هستند. این قطعات ممکن است بین ۱۷ تا ۲۶ نوکلئوتید طول داشته باشند. در ابتدا تصور می‌شد که این قطعات از تجزیه‌ی tRNA ها، snoRNA ها یا رونوشت‌ها ایجاد می‌شوند. ولی فراوانی بالا، الگوهای بیان متمایز و اثرات فنوتیپی آنها پس از مختل شدن توسط siRNA ها نشان داد که این RNA ها ممکن است که عملکردهایی در سلول داشته باشند (۱۸).

در حال حاضر به سرعت مشخص می‌شود که دسته‌های مختلفی از RNA های کوچک وجود دارند. این RNA ها با اندازه، زیست‌زایی، پردازش، الگوهای بیان، موقعیت درون سلولی و عملکرد

1- Nascent
2- Immunoprecipitation
3- Global run-on sequencing (GRO-seq)
4- Proximal
5- tRNA-derived RNA fragment (tRF)
6- tRNA-derived small RNA (tsRNA)
7- snoRNA-derived RNA (sdRNA)
8- Promoter-associated small RNA
9- Transcription start site-associated RNA (TSSa-RNA)

مولکولی‌شان تشخیص داده می‌شوند. بنابراین در زمان طراحی آزمایشات توالی‌یابی RNA، ضرورت دارد که محقق موقعیت و اندازه‌ی دسته‌های RNA های کوچکی را که انتظار دارد مشاهده نماید، پیش‌بینی کند. برآورد تقریبی از فراوانی نیز می‌تواند در تعیین امکان و گستره‌ی عملیات آزمایشگاهی و مواد مورد نیاز، به کار گرفته شود. همچنین آگاهی از این موضوع می‌تواند در تصمیم‌گیری برای عمق پوشش مورد نیاز، مفید واقع شود. علاوه بر این آگاهی از توالی‌های مورد انتظار می‌تواند به حاشیه‌نگاری خوانش‌های توالی حاصل از یک آزمایش توالی‌یابی RNA کمک نماید. بررسی دقیق و جزیی هر کدام از دسته‌های RNA های نارمزگر کوچک فراتر از اهداف این کتاب است. با این حال مقالات مروری خوبی در دسترس هستند که می‌توانند برای خوانندگان کنجکاو مفید واقع شوند. علاوه بر این، جدول ۱۲-۳ که اورتولوگ‌های اجزای مسیر miRNA را به صورت مقایسه‌ای نشان می‌دهد، نیز می‌تواند به خوانندگان برای جستجوی واژگان مختلف مورد استفاده‌ی محققین در پژوهش‌های علمی مربوط به انسان، *Drosophila* و *C. elegans* کمک نماید. این موارد مبنایی همراه با جزییات مراحل زیست‌زایی، بلوغ، بیان و اسمبل شدن RISC برای miRNA ها در مقاله‌ی مروری Bartel ارائه شده است (۱۹). یک مقاله‌ی مروری نیز توسط Siomi و همکاران در مورد piRNA ها ارائه شده است (۲۰). خلاصه‌ای از زیست‌زاییendo-siRNA ها نیز در مقاله‌ی مروری Kim و همکاران ارائه گردیده است (۲۱). یک مقاله‌ی مروری نیز در زمینه‌ی مسیرهای miRNA ، siRNA و آرگونات‌ها توسط Meister منتشر شده است (۲۲).

۱۲-۱۲ روش‌های توالی‌یابی برای یافتن RNA های نارمزگر کوچک

اصول کُلی روش‌های توالی‌یابی در توالی‌یابی RNA های نارمزگر کوچک شباهت زیادی به روش‌های توالی‌یابی تشریح شده در فصل اول دارد. تفاوت اصلی در انتخاب و مجموعه‌ی RNA برای در بر گرفتن RNA های کوچک است. مشابه توالی‌یابی RNA ، در اینجا نیز RNA از یک منبع زیستی تخلیص گردیده، آداپتورها به انتهاهای آنها افزوده شده، یک زنجیره‌ی cDNA تولید شده و cDNA با کمک آغازگرهای PCR که حاوی نمایه‌ها و آغازگرهای توالی‌یابی برای ایجاد کتابخانه هستند، تکثیر می‌گردد. انتخاب اندازه هم در زمانی که RNA ورودی خالص‌سازی می‌شود و هم در زمانی که برای RNA های کوچک غنی‌سازی گردیده و نیز بعد از ساخت زنجیره‌ی cDNA، انجام می‌شود. در عمل، انتخاب اندازه می‌تواند پس از تکثیر کتابخانه نیز صورت گیرد. ابزارهای آزمایشگاهی جدید نظیر سامانه‌ی Pippen-prep که آزمایشگاه را روی فناوری‌های الکتروفورز تراشه‌ای به کار می‌گیرند، می‌توانند برای خالص‌سازی و انتخاب اندازه در هر مرحله از تشکیل کتابخانه مورد استفاده قرار گیرند. در زیر گردش کار روش‌های اصلی توالی‌یابی RNA های

جدول ۱۲-۳: ژن‌های اجزای مسیر miRNA و ارتولوگ‌های‌شان در *Drosophila* و *C. elegans*

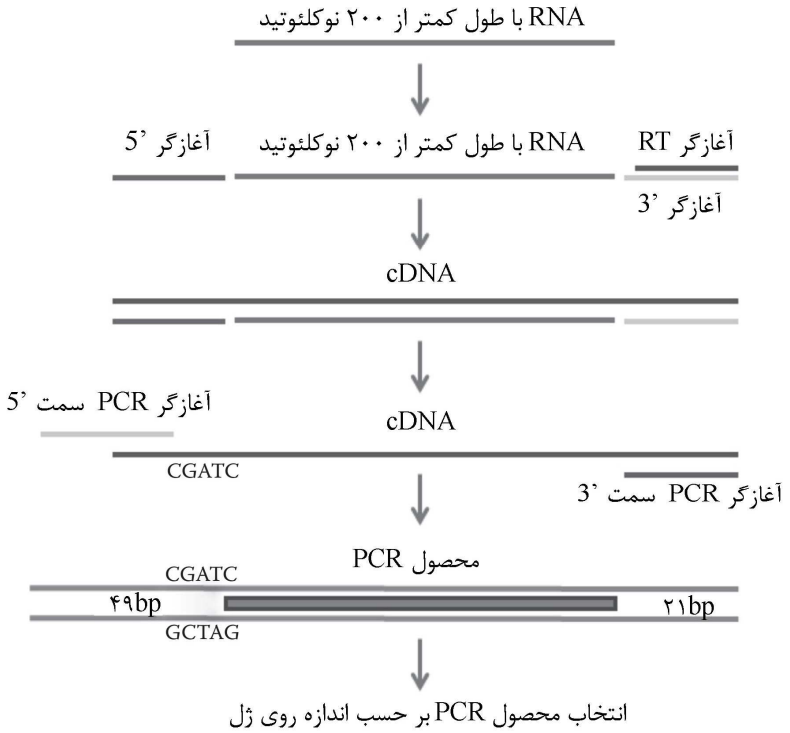
جزء / عملکرد	نام ژن در انسان	ارتولوگ در *Drosophila*	ارتولوگ در *C. elegans*
دروشا: شکافتن رونوشت‌های miRNA اولیه در کمپلکس با DCGR8	DROSHA	drosha	*drsh-1*
ناحیه‌ی حیاتی سندروم دی‌جورج: شناسایی سوبسترای RNA و کوفاکتور برای DROSHA	DGCR8	pasha	*pash-1*
اکسپورتین ۵: انتقال پیش‌ساز miRNA ها از هسته به سیتوپلاسم	XPO5	Ranbp21	نامعلوم
دایسر: پروتئینی از خانواده‌ی RNAase III با عملکرد شکافتن dsRNA	DICER1	Dcr-1	dcr-1
آرگونات: اتصال به dsRNA و پردازش RNA های پیش‌ساز به رشته‌های راهنما و مسافر، هدایت رشته‌ی راهنما به سمت mRNA هدف	AGO1-4	AGO1-2	*alg-1* و *alg-2*
پروتئین‌های پیوی: یک پروتئین آرگونات تخصصی با عملکرد اتصال به piRNA ها و کمک به تکثیر و تولید piRNA های ثانویه	PIWI1-4	AUB ، AGO3 و PIWI	نامعلوم
TRBP : پروتیئن متصل شونده به RNA پاسخ فعال‌سازی ترانس (TAR)، تشخیص dsRNA برای شکافتن آن همراه با دایسر، تغییر نرخ شکافتن و مشارکت در ویرایش	TRBP	Loqs	*rde-4*
PACT : فعال‌کننده‌ی پروتئین PKR ، یک پروتئین متصل شونده به dsRNA ، با عملکرد مشارکت احتمالی در انتخاب زنجیره و در کمپلکس با دایسر، فاقد عملکرد شکافتن	PACT	نامعلوم	نامعلوم
GW182 : کمک به هدف قرار دادن mRNA و خاموش کردن mRNA ها در کمپلکس با فاکتورهای آغاز ترجمه، آدنوزین دآمیناز و آنزیم‌های کلاهک‌بردار	GW182	Gawky	*ain-1* و *ain-2*

کوچک به طور مفصل توضیح داده شده است. به طور کلی این روش‌ها در مرحله‌ی ابتدایی که تخلیص RNA های کوچک بوده و در طی آن یک جمعیت هدف خاص از RNA های نارمزگر کوچک (نظیر RNA های کوچک متصل شونده به آرگونات) پیش از تشکیل کتابخانه خالص‌سازی می‌گردند، با هم تفاوت دارند. سپس تشکیل کتابخانه با استفاده از انواع مختلف کیت‌های تجاری انجام می‌شود. این کیت‌ها برای تولید کتابخانه‌های RNA های کوچک جهت توالی‌یابی با یک پلتفرم مشخص بهینه‌سازی شده‌اند. در زیر جزییات روش‌های اصلی توالی‌یابی RNA های کوچک ارائه گردیده است.

۱-۱۲-۱۲ توالی‌یابی ریز RNA ها

توالی‌یابی miRNA مشابه توالی‌یابی RNA است. تفاوت کار در این است که ماده‌ی ورودی پُلی‌آدنیله نبوده و اندازه‌یRNA که توالی‌یابی می‌گردد، کوچک است. بر مبنای جدولی که در قبل ارائه گردید، RNA های کوچک بسته به دسته‌ای که پژوهشگر می‌خواهد در کتابخانه توالی‌یابی کند، می‌توانند از ۲۱ نوکلئوتید تا چند صد نوکلئوتید متغیر باشند. بنابراین تهیه‌ی کتابخانه‌ای که در آن RNA ورودی خالص‌سازی یا انتخاب شود، امری ضروری است. برای یک آزمایش معمولی توالی‌یابی miRNA که در آن miRNA ها اهدافی برای توالی‌یابی هستند، miRNA غنی‌سازی شده یا از نظر اندازه در بین یک مجموعه از کُل RNA ها انتخاب می‌شود. این کار به آسانی و با کمک کیت‌های تجاری موجود و یا روش‌های خالص‌سازی از روی ژل امکان‌پذیر است. در حال حاضر یک سامانه‌ی تراشه‌ی میکروفلوییدی تجاری نیز برای این منظور در دسترس است. تعیین کیفیت و کمیت RNA های کوچک یکی از مراحل ابتدایی و حیاتی است. RNA های تجزیه شده نه تنها تعداد ورساخته‌ها در کتابخانه را افزایش می‌دهند، بلکه در تفسیر داده‌های توالی‌یابی نیز مشکل ایجاد می‌کنند. در عمل، وقتی که تعداد نمونه‌ها برای آماده‌سازی زیاد باشد، از یک پایدار کننده‌ی RNA تحت عنوان RNAlater (امبیون، اوستین، تگزاس/لایف تکنولوژیز، کارلزبد، کالیفرنیا) استفاده می‌شود. با این کار می‌توان نمونه‌ها را در یک روز جمع‌آوری کرده و در روز دیگر پردازش نمود. این ماده تاثیری بر کیفیت نمونه‌های RNA نداشته و انتقال و ذخیره نمونه‌ها را تسهیل می‌نماید. برای جداسازی RNA های کوچک، از یک روش مبتنی بر ستون تخلیص که RNA های کوچک‌تر از ۲۰۰ نوکلئوتید را غنی می‌کند، استفاده می‌شود. کیت‌های تجاری برای این کار مناسب هستند. به عنوان نمونه گزارش‌های خوبی از کیت mirVana (امبیون/لایف تکنولوژیز) ارائه شده است. پس از ساخت کتابخانه‌ی RNA های کوچک بر مبنای مراحل دستورالعمل زیر و پیش از ارسال برای توالی‌یابی، کتابخانه‌ی حاصل باید در دمای ۸۰- درجه‌ی سلسیوس یا در یخ خشک نگهداری شود.

دستورالعمل تهیه‌ی کتابخانه که در زیر ارائه شده است، برای توالی‌یابی بر مبنای دستورالعمل اِلومِنا مناسب است. توجه شود که دستگاه‌های GAIIx و Hi-Seq 2000 با انواع آداپتورهای کتابخانه سازگار بوده و بنابراین می‌توان اطمینان داشت که توالی آداپتورها در دستورالعمل تهیه‌ی کتابخانه با دستگاه مورد نظر می‌تواند به کار گرفته شود. معمولاً دستورالعمل تهیه‌ی کتابخانه سه روز زمان نیاز دارد. گردش کار معمول برای تهیه‌ی کتابخانه‌ی cDNA در نگاره‌ی ۱۲-۲ نشان داده شده است. یک مثال دقیق بدون استفاده از کیت‌ها و بر مبنای این گردش کار و توالی‌های آداپتور را می‌توان در مقاله‌ی Juhila و همکاران ملاحظه نمود (۲۳).



نگاره‌ی ۱۲-۲: طرح کُلی تهیه‌ی کتابخانه‌ی RNA های کوچک. نخست RNA های کوچک با استفاده از روش‌های بیوشیمیایی غنی می‌شوند. یک آداپتور '3 و یک آداپتور '5 به مولکول‌های RNA چسبانده می‌شود. رونویسی معکوس با استفاده از یک آغازگر مکملِ آداپتور '3 صورت گرفته و یک cDNA از RNA ساخته می‌شود. سپس روی cDNA و با کمک آغازگرهای '5 و '3 بر مبنای توالی‌های آداپتور چسبانده شده، PCR صورت می‌گیرد. آغازگر '5 در جایی که نمایه‌ها برای فرآیند چند عضوی می‌توانند استفاده شوند، یک برآمدگی نسبت به توالی cDNA دارد.

گردش کار تهیه‌ی کتابخانه‌ی توالی‌یابی miRNA به شرح زیر است:

۱- (اختیاری) نمونه را همراه با ۱۰ حجم RNAlater در تیوب میکروفیوژ قرار دهید. به مدت یک شب در دمای ۴ درجه‌ی سلسیوس نگهداری کنید. پس از آن نمونه‌ها را می‌توان در فریزر با دمای ۲۰- درجه‌ی سلسیوس به مدت چند ماه یا در فریزر با دمای ۸۰- درجه‌ی سلسیوس به صورت نامحدود نگهداری کرد.

۲- (اختیاری) نمونه را سانتریفیوژ و پایپت کرده و هر اندازه که برایتان مقدور است، RNAlater را از نمونه خارج کرده و دور بریزید.

۳- RNA های کوچک غنی شده با اندازه‌ی کمتر از ۲۰۰ نوکلئوتید را با کمک کیت mirVana و بر مبنای راهنمای سازنده‌اش جدا نمایید. در مرحله‌ی رقیق‌سازی نهایی، به جای ۱۰۰ میکرولیتری که در دستور کیت نوشته شده است، از ۶۰ میکرولیتر به ازای هر ستون استفاده کنید. استفاده از کیت miRNeasy (شرکت کایاژن[1]) نیز برای این مرحله مناسب است. هر دو کیت مزبور کاربری آسانی داشته، سریع بوده و مقدار کافی از RNA کوچک برای تهیه‌ی کتابخانه فراهم می‌آورند.

۴- مقدار ۲ میکرولیتر از نمونه‌ی حاصل را برای سنجش غلظت RNA توسط دستگاه نانودراپ[2] (ترمو ساینتیفیک[3]) یا دستگاه کیبِت فلورومتر[4] (لایف تکنولوژیز) استفاده کنید. استفاده از نانودراپ راحت‌تر بوده ولی کیبِت به ویژه با مقادیر اندک، از صحت بالاتری برخوردار است. معمولاً بایستی ۲ تا ۴ میکروگرم RNA کوچک غنی شده در حجم ۵۰ میکرولیتر از یک پلت بافتی ۰/۵ میلی‌لیتری به دست آمده باشد.

۵- پلت RNA کوچک تا حجم تقریبی ۵ میکرولیتر در یک خشک کن خلاء بدون استفاده از گرما خشک می‌شود.

۶- کتابخانه‌ی RNA کوچک نمونه را توسط NEBNext Small RNA Sample Prep Set 1 Kit (نیوانگلند بایولبز[5]) یا توسط کیت TruSeq (اِلومِنا) و بر مبنای راهنمای ارائه شده توسط کیت آماده کنید. در مرحله‌ی تکثیر PCR کتابخانه، از ۱۲ چرخه استفاده نمایید. کیتی که برای تهیه‌ی کتابخانه مورد استفاده قرار می‌دهید باید با پلتفرمی که برای توالی‌یابی استفاده خواهید کرد، سازگاری داشته باشد. این موضوع در راهنمای کیت ذکر شده است. به عنوان مثال، شرکت نیوانگلند بایولبز کیت‌های مختلفی برای تهیه‌ی

1- Qiagen
2- Nanodrop
3- Thermo Scientific
4- Quibit fluorometer
5- New England Biolabs

نمونه‌ی RNA کوچک داشته که با پلتفرم‌های اِلومِنا، سولاید، آیون تورنت و ۴۵۴ سازگار هستند. هر پلتفرمی کیت‌های مختص خود را داشته و شما می‌توانید مطمئن باشید که کیت‌های مزبور با آن پلتفرم خاص سازگاری دارند.

۷- در انتخاب اندازه و تخلیص کتابخانه‌ی cDNA تکثیر شده از روی ژل، نوارهای ۸۰ تا ۱۱۰ نوکلئوتیدی را برش بزنید. از برش زدن نوار ۷۰ تا ۷۵ نوکلئوتیدی پرهیز کنید. زیرا این نوار حاوی دایمر آداپتور است. miRNA های بالغ به دلیل اتصال آداپتورها، آغازگرهای توالی و نمایه‌ها، ۹۱ تا ۹۳ نوکلئوتید طول دارند. اگر می‌خواهید pre-miRNA ها را نیز جدا نمایید، نوارهای با اندازه‌ی بزرگ‌تر را برش دهید.

۸- کتابخانه‌ی cDNA تکثیر شده از ژل پُلی‌اکریل‌آمید را بر مبنای راهنمای کیت تخلیص نموده و پلت را در ۱۰ میکرولیتر بافر TE حل کنید.

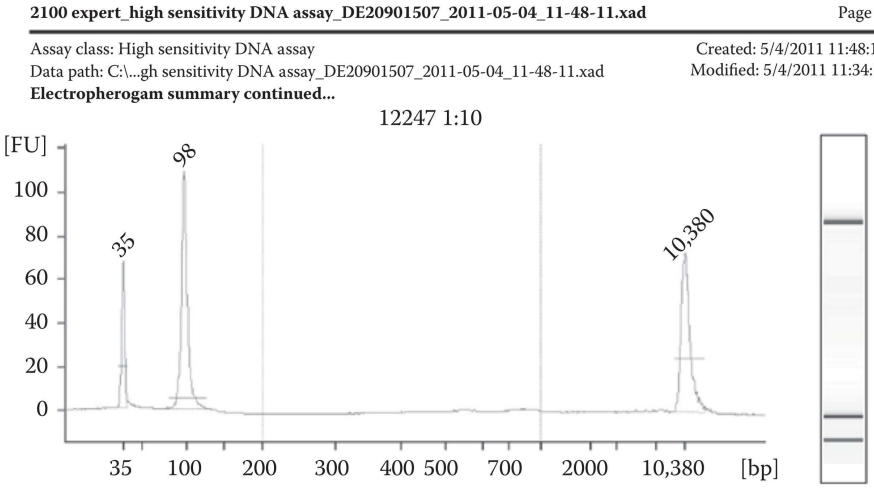
۹- مقدار ۱ میکرولیتر از کتابخانه‌ی cDNA را برای مشاهده‌ی اندازه روی یک دستگاه بایوآنالایزر (اجیلنت تکنولوژیز) به کار گیرید. نگاره‌ی ۱۲-۳ مثالی از یک کتابخانه‌ی miRNA آنالیز شده برای اندازه است.

۱۰- اگر اندازه‌ی کتابخانه با اندازه‌ی مورد انتظارتان مطابقت دارد، کار را با ایجاد خوشه‌ها توسط کیتِ خوشه‌بندی مختص پلتفرم (TruSeq SR Cluster Kit ، اِلومِنا) و سپس توالی‌یابی توسط کیتِ ساخت (TruSeq SBS Kit ، اِلومِنا) ادامه دهید. این مرحله که شامل ایجاد خوشه و توالی‌یابی است، معمولاً توسط شرکت‌هایی که توالی‌یابی را انجام می‌دهند، اجرا می‌گردد.

۱۲-۱۲-۲ توالی‌یابی CLIP

توالی‌یابی رسوب ایمنی همبر[1] (توالی‌یابی CLIP) نوع خاصی از توالی‌یابی RNA است که در آن از منبع متفاوتی از RNA برای تهیه کتابخانه استفاده می‌شود. در مطالعات اولیه برای شناسایی مولکول‌های miRNA:mRNA از رسوب ایمنی به عنوان یک ابزار برای غنی‌سازی مولکول‌های RNA کوچک در کمپلکس با پروتئین‌ها استفاده می‌شد. در این مطالعات از آنتی‌بادی‌ها علیه پروتئین‌های آرگونات نشانمند شده با myc استفاده به عمل می‌آمد تا از این طریق مولکول‌های متصل به RNA رسوب داده شده و سپس نسبت به شناسایی مولکول‌های هدف mRNA (۲۴) یا آنالیز آنها با استفاده از ریزآرایه (۲۵) اقدام می‌گردید. هدف از این کار جداسازی برای توالی‌یابی miRNA های بالغ و اهدافشان بوده است. پیشرفت بعدی در این روش نخست با همبرسازی

1- Cross-linked immunoprecipitation sequencing (CLIP-seq)



نگاره‌ی ۱۲-۳: الکتروفوروگرام یک کتابخانه‌ی miRNA ران شده در دستگاه اجیلنت بایوآنالایزر. نشانگرهای اندازه در ۳۵ و ۱۰۳۸۰ جفت باز بوده ولی کتابخانه یک اوج در ۹۸ جفت باز نشان می‌دهد. در اوج ۹۸ جفت بازی، طول RNA های کوچک بین ۲۱ تا ۲۳ نوکلئوتید است. سایر dsDNA ها آداپتورهای کتابخانه هستند.

مولکول‌های RNA با استفاده از پروتئین‌ها در نزدیکی فیزیکی فوری با کمک تابش UV و سپس با توالی‌یابی RNA به جای کلون کردن یا آنالیز ریزآرایه برای شناسایی مولکول‌های RNA به دست آمده است. این روش در گذشته توالی‌یابی پُربروندا‌د (کارآمد) RNA جداسازی شده توسط رسوب ایمنی همبر[1] (HITS-CLIP) نامیده می‌شد که در حال حاضر به صورت ساده و کوتاه CLIP-seq (توالی‌یابی CLIP) نامیده می‌شود (۲۶). علی‌رغم اینکه مطالعه‌ی اولیه روی رسوب ایمنی عامل

1- High-throughput sequencing of RNA isolated by crosslinking immunoprecipitation (HITS-CLIP)

پیرایش مختص نرون تحت عنوان نووا[1] متمرکز بود، ولی مطالعه‌ی بعدی به همبر شدن و رسوب پروتئین آرگونات (Ago) از مغز موش در یک کمپلکس با miRNA و mRNA پرداخت (۲۷). نتیجه‌ی این مطالعه، خواندن توالی mRNA از ۸۲۹ رونوشت بود. مطالعه‌ی مشابهی در *C. elegans* با استفاده از تیپ وحشی و جهش‌یافته‌های آرگونات (*alg-1*) به عنوان شاهد صورت گرفته و با کمک روش فوق ۳۹۹۳ ژن مختص تیپ وحشی شناسایی شدند (۲۸). اصطلاح توالی‌یابی رسوب ایمنی پروتئین متصل شونده به RNA[2] (توالی‌یابی RIP) نیز در مواردی که اِپی‌توپ یک پروتئین متصل شونده به RNA است، برای توصیف توالی‌یابی CLIP به کار گرفته می‌شود (۲۹). همچنین از توالی‌یابی CLIP برای شناسایی RNA ها در کمپلکس پردازشگر pri-miRNA از طریق رسوب ایمنی DGCR8 استفاده می‌شود (۳۰). Macias و همکاران با این روش pri-miRNA ها، mRNA ها و snoRNA هایی را یافتند که عملکردهای بیشتری برای DGCR8 پیشنهاد می‌نمودند. چون بازیابی RNA های کوچک ازنمونه‌های رسوب ایمنی بسیار اندک است، لذا استفاده از شاهدهای منفی اهمیت زیادی دارد. به همین دلیل، در دستورالعمل‌های توالی‌یابی CLIP چندین تکرار همراه با شاهدهای زمینه به طور وسیعی مورد استفاده قرار می‌گیرند. یک دستورالعمل گام به گام مناسب برای توالی‌یابی CLIP توسط Murigneux و همکاران ارائه شده است (۳۱).

توالی‌یابی CLIP ریبونکلئوزید قابل فعال‌سازی نوری تقویت شده[3] (توالی‌یابی PAR-CLIP) نوعی از توالی‌یابی CLIP است که هدفش مکان‌یابی دقیق نواحی برهمکنش بین پروتئین و RNA در سطح نوکلئوتیدی است. PAR-CLIP از ریبونوکلئوتیدهای تغییر یافته نظیر ۴-تیویوریدین در طی بیوسنتز مولکول RNA استفاده می‌کند. سپس نوکلئوزیدهای فعال نوری تغییر یافته از طریق نور UV و در یک واکنش با کارآیی بیشتر از نوکلئوتیدهای تغییر نیافته، به پروتئین متصل شونده به RNA همبر می‌شوند (۳۲). مزیت دیگر این روش این است که محل دقیق برهمکنش RNA:protein را می‌توان مکان‌یابی نمود. پس از همبرسازی توسط UV ، رسوب ایمنی و سپس توالی‌یابی نسل جدید با همان روشی که در توالی‌یابی CLIP مورد استفاده قرار می‌گیرد، اجرا می‌شود. یکی از نقاط ضعف این روش این است که نیازمند انکوباسیون اولیه‌ی سلول‌ها/نمونه‌ها با آنالوگ‌های نوکلئوزید بوده و بدین ترتیب مناسب بودن نمونه‌ها برای این روش محدود می‌گردد. علاوه بر این ممکن است که برخی از آنالوگ‌های ریبونوکلئوزیدها سمی بوده و در نتیجه در هنگام مقایسه نمودن نمونه‌ها با یکدیگر، نتایج اُریب به دست می‌آید.

1- Nova
2- RNA-binding protein immunoprecipitation sequencing (RIP-seq)
3- Photoactivatable ribonucleoside-enhanced CLIPSeq (PAR-CLIPSeq)

وضوح تک نوکلئوتیدی در همبرسازی UV و رسوب ایمنی[1] (توالی‌یابی iCLIP) یکی دیگر از انواع توالی‌یابی CLIP است. در مطالعات اولیه‌ی مبتنی بر توالی‌یابی CLIP ملاحظه شد که برخی از RNA های رسوب داده شده، زمانی که با ژنوم متناظرشان مکان‌یابی می‌شوند، حاوی توالی‌های گمشده[2] هستند. این موضوع ناشی از آن بود که رونویسی معکوس از توالی‌های RNA که اتصال پروتئین همبر به آن هنوز باقی مانده بود، صورت نمی‌گرفت. برای بهبود وضوح، روش توالی‌یابی iCLIP ابداع گردید. در این روش برهمکنش RNA:protein در سطح تک نوکلئوتید مکان‌یابی می‌گردد (۳۳). این روش مبتنی بر شواهدی است که در طی تهیه‌ی کتابخانه به دست آمده است. این شواهد نشان می‌دهند که در مرحله‌ای که RNA رسوب داده شده و جدا شده به cDNA رونویسی معکوس می‌گردد، نواحی که RNA با پروتئین در چند مولکول همبر شده و تیمار با آنزیم پروتئیناز K نتوانسته ارتباط بین پروتئین و RNA را به طور کامل از هم بگسلد، آنزیم رونوشت‌بردار معکوس[3] یا دقیقاً آن نقطه از RNA را نادیده گرفته و یا متوقف می‌شود. در این موارد، cDNA قطع شده‌ی حاصل، حلقوی شده، خطی شده، تکثیر شده و سپس توالی‌یابی می‌شود (۳۴). توالی حاصل نه تنها RNA های رسوب داده شده را شناسایی می‌کند، بلکه نواحی برهمکنش پروتئین در سطح تک نوکلئوتید را نیز مشخص می‌نماید.

۱۲-۱۲-۳ توالی‌یابی تخریبی

آگاهی از محل و نحوه‌ی برش خوردن مولکول‌های mRNA توسط اندوریبونوکلئازها منجر به ایجاد مجموعه‌ای از روش‌ها گردید که امروزه تحت عنوان توالی‌یابی تخریبی[4] شناخته می‌شوند (۳۵ و ۳۶). بیشترین استفاده‌ی عملی از این روش در مکان‌یابی، انطباق و شناسایی miRNA ها و محصولات حاصل از برش خوردن mRNA های هدف آنها صورت می‌گیرد. این روش بر مبنای این اصل تفاوت بیوشیمیایی استوار است که mRNA های یوکاریوتی بالغ دارای ۷-متیل‌گوانازین در انتهای '5 کلاهک‌گذاری شده ولی mRNA های دارای یک '5-منوفسفات توسط اندوریبونکلئاز برش زده می‌شوند. در این روش کتابخانه‌های RNA کوچک از RNA های سلولی دارای یک '5-منوفسفات که RNA های برش نخورده را مستثنی می‌کند، ایجاد می‌شوند. زیرا در طی تشکیل کتابخانه نوکلئوتیدهای ۷-متیل‌گوانازین نمی‌توانند به آداپتورها متصل گردند. سپس کتابخانه‌ها

1- Individual-Nucleotide Resolution UV Crosslinking and Immunoprecipitation (iCLIPSeq)
2- Missing sequence
3- Reverse transcriptase
4- Degradome-seq

تحت فرآیند توالی‌یابی قرار می‌گیرند. پس از آن خوانش‌ها با ژنوم مکان‌یابی شده و جایی که انتهای '5 مکان‌یابی می‌گردد، یک محل بالقوه برای برش خوردن با واسطه‌ی miRNA خواهد بود. اگر این برش توسط دایسر (یک اندوریبونوکلئاز دسته‌ی RNase III) صورت گیرد، ابتدای miRNA بایستی دقیقاً ۱۰ نوکلئوتید از انتهای '5 آن miRNA باشد. بنابراین توالی miRNA می‌تواند از روی محل برش استنباط گردد. کنترل متقابل توالی‌های احاطه کننده‌ی ناحیه‌ی برش با پایگاه‌های داده‌ی مختلف miRNA می‌تواند توالی miRNA را تایید و شناسایی کند. سایر روش‌های بیوانفورماتیکی نظیر محاسبه‌ی ساختار ثانویه‌ی RNA و حداقل انرژی آزاد ساختارهای سنجاق‌سری در مجاورت ناحیه‌ی برش نیز می‌تواند برای شناسایی miRNA های جدید به کار گرفته شود. این ترکیب از مجموعه‌ی روش‌های تشکیل کتابخانه، توالی‌یابی و آنالیز بیوانفورماتیکی اصطلاحاً PARE (مخفف آنالیز موازی انتهاهای RNA[1]) نامیده می‌شود. این روش نخستین بار در گیاهان استفاده شد. ولی در حال حاضر در اکثر گونه‌های پُرسلولی مورد استفاده واقع می‌گردد. یک نکته مهم این است که در این روش باید جانب احتیاط را رعایت نمود. زیرا '5-منوفسفات‌ها می‌توانند توسط سایر مولکول‌های دسته‌ی RNase III سلولی غیر از دایسر نیز ایجاد شوند. بنابراین فعالیت تخریبی درون‌زاد RNA بیشتر نمایان می‌شود. به همین علت به نظر می‌رسد که انتخاب توالی‌یابی تخریبی به عنوان نام این روش مناسب بوده است. مکان‌یابی و شناسایی miRNA ها یک چالش قابل توجه‌ی محاسباتی و تکنیکی برای این روش محسوب می‌گردد. مسیر بیوانفورماتیکی تحت عنوان CleaveLand ایجاد شده است که می‌تواند هدف‌های miRNA برش خورده‌ی حاصل از پروژه‌های توالی‌یابی تخریبی را تشخیص دهد (۳۵). CleaveLand را می‌توان از http://axtell-lab- psu. weebly.com/cleaveland.html دانلود نمود.

۱۲-۱۲-۴ توالی‌یابی مداوم کُلی (توالی‌یابی GRO)

توالی‌یابی مداوم کُلی (توالی‌یابی GRO) روشی برای شناسایی رونوشت‌های نوآیند (تازه ساخته شده) در حین ساختن و طویل شدن است. این روش رونوشت‌هایی را که در لحظه‌ی آزمایش در حال ساخته شدن هستند، شناسایی می‌کند (در مقایسه با سطوح حالت ثابت رونوشت‌ها در یک نمونه‌ی زیستی). در این روش از توقف رونویسی استفاده می‌شود. در این فرآیند RNA پُلی‌مراز شروع به رونویسی کرده یا وارد فعالیت رونویسی می‌شود. ولی ممکن است کمپلکس رونویسی در انتظار عوامل اضافی بوده و در نتیجه متوقف گردد. توقف یا تزاحم رونویسی[2] می‌تواند توسط

1- Parallel analysis of RNA ends (PARE)
2- Transcriptional interference

سمومی نظیر اکتینومایسین D نیز ایجاد شود. در این روش برای به دست آوردن قطعات جهت توالی‌یابی، تدام هسته‌ای[1] اجرا شده و ۵-بروموریدین '5 فسفات به کشت افزوده گردیده و در داخل RNA نوآیند تولید شده در سلول جای می‌گیرد (۳۷). این روش معمولاً برای شناسایی ژن‌هایی که به صورت فعال رونویسی می‌شوند، استفاده می‌گردد. ولی در سال‌های اخیر برای شناسایی eRNA ها نیز مورد استفاده قرار گرفته است.

۱۲-۱۳ خلاصه

شیوه‌ی مرسوم در روش‌های توالی‌یابی RNA های کوچک شامل جداسازی RNA های کوچک و سپس توالی‌یابی آنها با استفاده از پلتفرم‌های مختلف است. همگام با تلاش مداوم دانشمندان برای یافتن دسته‌های جدید RNA های کوچک که حتی در سال‌های اخیر نیز منجر به کشف eRNA ها گردیده و به نظر می‌رسد که هر جانداری miRNA اُم منحصر به فرد خود را دارد، روش‌های مزبور نیز دائماً بسط و گسترش یافته‌اند. در حال حاضر روش‌های متعددی شامل مراحل قطعه‌بندی می‌گردند که برای غنی‌سازی RNA ها در تشکیل کتابخانه‌های RNA های کوچک و mRNA ها به کار گرفته می‌شوند. با توجه به فناوری‌های ارزشمندی که هم اکنون در دسترس هستند، به نظر می‌رسد که اکتشافات جدید و هیجان‌انگیز متعددی در راه باشد.

منابع

1. Lee R.C., Feinbaum R.L., and Ambros V. Te C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. Cell 75(5):843–854, 1993.
2. Pasquinelli A.E., Reinhart B.J., Slack F. et al. Conservation of the sequence and temporal expression of let-7 heterochronic regulatory RNA. Nature 408(6808):86–89, 2000.
3. Lagos-Quintana M., Rauhut R., Lendeckel W. et al. Identification of novel genes coding for small expressed RNAs. Science 294(5543):853–858, 2001.
4. Lau N.C., Lim L.P., Weinstein E.G. et al. An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans. Science 294(5543):858–862, 2001.
5. Lee R.C. and Ambros V. An extensive class of small RNAs in Caenorhabditis elegans. Science 294(5543):862–864, 2001.
6. Djuranovic S., Nahvi A., and Green R. A parsimonious model for gene regulation by miRNAs. Science 331(6017):550–553, 2011.

1- Nuclear run-on

7. Neilsen C.T., Goodall G.J., and Bracken C.P. IsomiRs—the overlooked repertoire in the dynamic microRNAome. Trends Genet 28(11):544–549, 2012.
8. Westholm J.O. and Lai E.C. Mirtrons: MicroRNA biogenesis via splicing. Biochimie 93(11):1897–1904, 2011.
9. Shi W., Hendrix D., Levine M. et al. A distinct class of small RNAs arises from pre-miRNA-proximal regions in a simple chordate. Nat Struct Mol Biol 16(2):183–189, 2009.
10. Langenberger D., Bermudez-Santana C., Hertel J. et al. Evidence for human microRNA-offset RNAs in small RNA sequencing data. Bioinformatics 25(18):2298–2301, 2009.
11. Bortoluzzi S., Biasiolo M., and Bisognin A. MicroRNA-offset RNAs (moRNAs): By-product spectators or functional players? Trends Mol Med 17(9):473–474, 2011.
12. Asikainen S., Heikkinen L., Wong G. et al. Functional characterization of endogenous siRNA target genes in Caenorhabditis elegans. BMC Genomics 9:270, 2008.
13. Duchaine T.F., Wohlschlegel J.A., and Kennedy S. Functional proteomics reveals the biochemical niche of C. elegans DCR-1 in multiple small-RNAmediated pathways. Cell 124(2):343–354, 2006.
14. Goodenbour J.M. and Pan T. Diversity of tRNA genes in eukaryotes. Nucleic Acids Res 34(21):6137–6146, 2006.
15. Lestrade L. and Weber M.J. snoRNA-LBME-db, a comprehensive database of human H/ACA and C/D box snoRNAs. Nucleic Acids Res 34(Database issue):D158–D162, 2006.
16. Redmond A.M. and Carroll J.S. Enhancer-derived RNAs: “spicing up” transcription programs. EMBO J 32(15):2096–2098, 2013.
17. Kim T.K., Hemberg M., Gray J.M. et al. Widespread transcription at neuronal activity-regulated enhancers. Nature 465(7295):182–187, 2010.
18. Aalto A.P. and Pasquinelli A.E. Small non-coding RNAs mount a silent revolution in gene expression. Current Opinion Cell Biol 24(2):333–340, 2012.
19. Bartel D. MicroRNAs: Genomics, biogenesis, mechanism, and function. Cell 116(2):281–297, 2004.
20. Siomi M.C., Sato K., Pezic D. et al. PIWI-interacting small RNAs: Te vanguard of genome defence. Nat Rev Mol Cell Biol 12(4):246–258, 2011.
21. Kim V.N., Han J., and Siomi M.C. Biogenesis of small RNAs in animals. Nat Rev Mol Cell Biol 10(2):126–139, 2009.
22. Meister G. Argonaute proteins: Functional insights and emerging roles. Nat Rev Genet 14(7):447–459, 2013.
23. Juhila J., Sipilä T., Icay K. et al. MicroRNA expression profling reveals miRNA families regulating specifc biological pathways in mouse frontal cortex and hippocampus. PLoS ONE 6(6):e21495, 2011.
24. Karginov F.V., Conaco C., Xuan Z. et al. A biochemical approach to identifying microRNA targets. Proc Natl Acad Sci USA 104(49):19291–19296, 2007.
25. Easow G., Teleman A.A., and Cohen S.M. Isolation of microRNA targets by miRNP immunopurifcation. RNA 13(8):1198–1204, 2007.

26. Licatalosi D.D., Mele A., Fak J.J. et al. HITS-CLIP yields genome-wide insights into brain alternative RNA processing. Nature 456(7221):464–469, 2008.
27. Chi S.W., Zang J.B., Mele A. et al. Argonaute HITS-CLIP decodes microRNA–mRNA interaction maps. Nature 460(7254):479–486, 2009.
28. Zisoulis D.G., Lovci M.T., Wilbert M.L. et al. Comprehensive discovery of endogenous Argonaute binding sites in Caenorhabditis elegans. Nat Struct Mol Biol 17(2):173–179, 2010.
29. Zhao J., Ohsumi T.K., Kung J.T. et al. Genome-wide identifcation of polycomb-associated RNAs by RIP-seq. Mol Cell 40(6):939–953, 2010.
30. Macias S., Plass M., Stajuda A. et al. DGCR8 HITS-CLIP reveals novel functions for the microprocessor. Nat Struct Mol Biol 19(8):760–766, 2012.
31. Murigneux V., Saulière J., Roest Crollius H. et al. Transcriptome-wide identifcation of RNA binding sites by CLIP-seq. Methods 63(1):32–40, 2013.
32. Hafner M., Lianoglou S., Tuschl T. et al. Genome-wide identifcation of miRNA targets by PAR-CLIP. Methods 58(2):94–105, 2012.
33. König J., Zarnack K., Rot G. et al. iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. Nat Struct Mol Biol 17(7):909–915, 2010.
34. Sugimoto Y., König J., Hussain S. et al. Analysis of CLIP and iCLIP methods for nucleotide-resolution studies of protein–RNA interactions. Genome Biol 13(8):R67, 2012.
35. Addo-Quaye C., Eshoo T.W., Bartel D.P. et al. Endogenous siRNA and miRNA targets identifed by sequencing of the Arabidopsis degradome. Current Biol 18(10):758–762, 2008.
36. German M.A., Pillay M., Jeong D.H. et al. Global identifcation of microRNAtarget RNA pairs by parallel analysis of RNA ends. Nat Biotechnol 26(8):941–946, 2008.
37. Core L.J., Waterfall J.J., and Lis J.T. Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. Science 322(5909):1845–1848, 2008.

فصل سیزدهم

آنالیز محاسباتی داده‌های توالی‌یابی RNA های نارمزگر کوچک

۱–۱۳ مقدمه

پس از جدا نمودن RNA های کوچک از نمونه‌ها، کتابخانه‌ها تشکیل شده و داده‌های توالی‌یابی به دست آمده و معمولاً یک یا چند فایل بزرگ حاوی توالی‌ها در فرمت FASTA دریافت می‌گردد. بخش هیجان‌انگیز کار از اینجا شروع می‌شود. ابزارهای متعددی برای آنالیز داده‌های مرتبط با اهداف چنین آزمایشاتی در دسترس هستند. این ابزارها از جنبه‌های مختلف شامل سهولت کاربری، الگوریتم‌های مورد استفاده برای شناسایی انواع RNA های کوچک، وابستگی به سایر فایل‌های داده‌ها و حاشیه‌نگاری‌ها، روش‌های آماری، افزونه‌ها و فرمت‌های ورودی و خروجی با هم فرق دارند. در این فصل روش عملی گام به گام با استفاده از miRDeep2 تشریح می‌شود. این نرم‌افزار به خوبی به کار گرفته شده و توسط سایر محققین در این زمینه مورد استناد قرار گرفته و حاوی ابزارهای لازم برای مکان‌یابی خوانش‌ها، فولدبندی و مصورسازی miRNA هایی که به تازگی یافت شده‌اند، بوده و خواندن خروجی‌های آن آسان است. ولی کار کردن با این نرم‌افزار مستلزم آشنایی با خط فرمان می‌باشد. ابزار دیگر که مورد بحث قرار می‌گیرد، miRanalyzer است که یک ابزار تحت وب بوده و می‌تواند روی هر مرورگر مبتنی بر وب اجرا شده، کاربری آسانی داشته و ابزارهایی را برای آنالیز افتراقی فراهم می‌آورد که با کمک آنها می‌توان مطالعات مختلفی را ترتیب داد. این ابزارها، تنها ابزارهای موجود برای آنالیز داده‌های توالی‌یابی RNA های کوچک نیستند. دو مثال ارائه شده در فوق راه حل‌های عملی مطلوبی ارائه می‌دهند که پیاده‌سازی آنها نسبتاً یا بسیار ساده است. خوانندگانی که به سایر ابزارهای موجود علاقه دارند، می‌توانند به مقالات مروری که اخیراً منتشر شده‌اند، مراجعه نمایند. در این مقالات، تفاوت‌ها و نقاط قوت و ضعف ابزارهای مزبور و ابزارهایی که در این کتاب نیز مورد استفاده قرار گرفته‌اند، بحث و بررسی می‌گردند (۱ و ۲). در بخش دوم این فصل، روش‌های عملی برای انجام آنالیزهای پایین‌دستی پس از شناسایی و کمی‌سازی miRNA های حاصل از نمونه‌ها ارائه شده است. این آنالیزها شامل تعیین اهداف miRNA ها هستند. در انتها نیز برخی از منابع اطلاعات عمومی راجع به RNA های کوچک شامل miRBase و Rfam مورد بحث و بررسی قرار می‌گیرند.

۱۳-۲ شناسایی RNA های کوچک: miRDeep2

miRDeep2 یک ابزار جامع است که این امکان را فراهم می‌آورد که داده‌های توالی‌یابی RNA را دریافت کرده و در خروجی، شمارش miRNA های شناخته شده، شناسایی miRNA های جدید و تفاوت‌های موجود در بین مجموعه‌ی داده‌ها از نظر miRNA ها را ارائه دهد (۳). این نرم‌افزار در لینوکس اجرا می‌شود. ماژول‌های این نرم‌افزار در پوشش پِرل شامل یک ابزار مکان‌یابی خوانش‌ها (Bowtie) و ابزارهایی برای فولدبندی و مصورسازی پیش‌سازهای miRNA (Randfold و Viennarna) هستند. رابط کاربری خط فرمان و خروجی گرافیکی این نرم‌افزار را به ابزاری بسیار مفید تبدیل کرده است.

روش پایه برای اجرای miRDeep2 به شرح زیر است:

۱- لینوکس را روی رایانه‌ی شخصی نصب کنید. برای آگاهی از نحوه‌ی انجام این کار، به فصل دوم، بخش ۲-۶ مراجعه نمایید.

۲- پس از نصب لینوکس، mirDeep2 را از نشانی https://www.mdc-berlin.de/8551903/en/research/research_teams/systems_biology_of_gene_regulatory_elements/projects/miRDeep دانلود کنید.

۳- فایل حاشیه‌نگاری GFF را برای گونه‌ای که می‌خواهید آنالیز کنید، دانلود نمایید.

۴- فایل FASTA مربوط به miRNA های شناخته شده برای گونه‌ای که می‌خواهید آنالیز نمایید، را دانلود کنید.

۵- محیط اجرا را تنظیم کنید. به عنوان مثال، کلیه‌ی فایل‌های مورد نیاز را در miRDeep2 working directory قرار داده و اسکریپت‌ها را بر طبق نیازتان پیکره‌بندی کنید.

۶- سپس miRDeep2 را اجرا نمایید.

۱۳-۲-۱ فایل‌های GFF

فایل‌های GFF (مخفف شده‌ی Generic Feature Foemat (فرمت عمومی ترکیب)) حاوی ترکیبات ژنومی یک توالی هستند. ترکیبات در قالب یک فایل متنی ساده با نُه ستون ذخیره می‌گردند که هر ستون نشان دهنده‌ی یک ترکیب متمایز از یک توالی است. ستون‌های مزبور با تَب از یکدیگر جدا می‌شوند. نخستین خط این فایل شامل یک متن سرآغاز به صورت ##gff-version 3 است. خطوط بعدی نیز می‌توانند شامل متن‌های توضیحی نظیر شرحی از فایل، منشاء شماره‌ی نسخه‌ی ویرایش، نکات، مراجع و توضیحات دیگر باشند. این توضیحات تا جایی که خطوط با علامت # مشخص می‌شوند، ادامه خواهند داشت. بعد از خطوط توضیحی، ترکیبات ژنومی را

می‌توان در هر کدام از نُه ستون ملاحظه نمود. در مثالی که در جدول ۱۳-۱ ارائه شده است، هشت خط انتهایی حاوی اطلاعات ترکیبات ژنومی است.

ستون ۱: شامل Landmark ID است: در این مثال، همگی I خورده و نشان دهنده‌ی کروموزوم شماره ۱ است.

ستون ۲: منبعی که این ترکیب از آن ایجاد شده است: در مواردی نظیر این مثال که هیچ منبعی برای این مورد ذکر نشده باشد، ستون مزبور خالی گذاشته شده و تنها یک نقطه در آن درج می‌شود.

ستون ۳: نوع توالی: در این مثال یا رونوشت اولیه‌ی miRNA (miRNA_primary_transcript) یا miRNA بوده است.

ستون ۴: نقطه‌ی شروع توالی بر مبنای سامانه‌ی مختصات: در این مثال، 1738637 نشان دهنده‌ی آن است که توالی مزبور در کروموزوم شماره‌ی ۱ و نوکلئوتید ۱۷۳۸۶۳۷ شروع می‌شود.

ستون ۵: نقطه‌ی پایان توالی بر مبنای سامانه‌ی مختصات: در این مثال، 1738735 نشان دهنده‌ی آن است که توالی مزبور در کروموزوم شماره‌ی ۱ و نوکلئوتید ۱۷۳۸۷۳۵ پایان می‌پذیرد.

ستون ۶: امتیاز: در این مثال، این ستون با درج یک نقطه (.)، خالی گذاشته شده است.

ستون ۷: زنجیره (رشته): در این مثال، علامت «+» نشان دهنده‌ی زنجیره‌ی مثبت و علامت «-» نشان دهنده‌ی زنجیره‌ی منفی است.

ستون ۸: فاز: در این مثال، این ستون با درج یک نقطه (.)، خالی گذاشته شده است.

ستون ۹: ویژگی‌های توالی: سامانه‌ی استفاده شده در اینجا شامل نشان = مقدار است. مقادیر مختلف با علامت «;» از هم جدا می‌شوند. هر نشان شامل یک ویژگی بوده و می‌تواند یک مقدار به خود تخصیص دهد. در اینجا می‌توان از چندین نشان استفاده نمود. در این مثال، نخستین نشان ID بوده که مقدار آن MI0000021-1 است. دومین نشان نیز Name بوده که مقدار آن cel-mir-50 می‌باشد.

فایل‌های GFF نسخه‌های ویرایش مختلف دارند که در این مثال از نسخه‌ی ویرایش 3 استفاده شده است. باید توجه نمود که حتی اگر نسخه‌ها مشابه باشند، همواره با هم سازگار نخواهند بود. فایل‌های GFF به کار برده شده در این مثال از miRBase دانلود شده‌اند. ولی می‌توان آنها را از منابع مختلف دیگر نظیر انواع پایگاه‌های داده‌ی ژنومی نیز دانلود کرد. خصوصیات دقیق این فایل‌ها را می‌توان در www.sequenceontology.org/gff3.shtml مطالعه نمود.

جدول ۱۳-۱: یک فایل GFF از miRNA های *C. elegans*

```
##gff-version 3
##date
 2012-7-23
#
# Chromosomal coordinates of C. elegans microRNAs
# microRNAs                                    miRBase v19
#                    genome-build-id WBcel215
#
# Hairpin precursor sequences have type "miRNA_primary_transcript"
# Note, these sequences do not represent the full primary transcript
# rather a predicted stem-loop portion that includes the precursor
# miRNA. Mature sequences have type "miRNA."
#
I    .    miRNA_primary_transcript    17,38,637    17,38,735    .    +    .    ID = MI0000021_1;Name = cel-mir-50
I    .    miRNA                       17,38,652    17,38,675    .    +    .    ID = MIMAT0000021_1;Name = cel-miR-50-5p
I    .    miRNA                       17,38,694    17,38,715    .    +    .    ID = MIMAT0020310_1;Name = cel-miR-50-3p
I    .    miRNA_primary_transcript    28,88,450    28,88,559    .    -    .    ID = MI0019067_1;Name = cel-mir-5546
I    .    miRNA                       28,88,514    28,88,536    .    -    .    ID = MIMAT0022183_1;Name = cel-miR-5546-5p
I    .    miRNA                       28,88,472    28,88,493    .    -    .    ID = MIMAT0022184_1;Name = cel-miR-5546-3p
I    .    miRNA_primary_transcript    29,21,188    29,21,292    .    +    .    ID = MI0017717_1;Name = cel-mir-4931
I    .    miRNA                       29,21,258    29,21,277    .    +    .    ID = MIMAT0020137_1;Name = cel-miR-4931
```

علامت هَش (#) نشان دهنده خطوط توضیح است. پس از خطوط توضیحی، نُه ستون داده ارائه شده است. در این جدول تنها ۸ مدخل (خط) از داده‌ها ذکر گردیده است.

۱۳-۲-۲ فایل‌های FASTA مربوط به miRNA های شناخته شده

فرمت فایل FASTA یک فرمت برای توالی‌های DNA ، RNA و پروتئین است. نخستین خط توالی با فرمت FASTA شامل یک علامت بزرگ‌تر (>) بوده و به دنبال آن متنی می‌آید که توالی را توضیح می‌دهد. سپس توالی در سایر خطوط آورده می‌شود. توالی بعدی زمانی شروع می‌شود که علامت > بعدی درج گردد. فایل‌های FASTA برای miRNA های شناخته شده‌ی هر گونه را می‌توان مستقیماً از miRBase دانلود نمود. به عنوان مثال، فایل FASTA برای miRNA های *Caenorhabditis elegans* در جدول ۱۳-۲ نشان داده شده است.

۱۳-۲-۳ تنظیم محیط اجرا

۱- اطمینان حاصل کنید که ابزار GCC نصب شده است. در اوبونتو بسته‌ی build-essential را نصب کنید:

```
$ sudo apt-get install build-essential
```

۲- نرم‌افزار Bowtie را دریافت نمایید (bowtie-bio.sourceforge.net). بسته را در مسیر /usr/local/share از حالت فشرده خارج کرده و لینک‌های نمادین را در /usr/local/bin ایجاد کنید تا بر فایل‌های اجرایی غیرفشرده متمرکز گردد.

```
$ sudo unzip name_of_the_bowtie_package.zip -d/usr/local/share
```

```
$ ln -s/usr/local/share/name_of_the_bowtie_directory/bowtie*/usr/local/bin
```

۳- بسته‌ی Vienna RNA را دریافت کنید (http://www.tbi.univie.ac.at/~ivo/RNA/). برای کامپایل و نصب این بسته، به صورت زیر تایپ کنید:

```
$./configure
```

```
$ make
```

```
$ sudo make install
```

۴- SQUID را نصب کنید. برای این کار در اوبونتو به صورت زیر تایپ کنید:

```
$ sudo apt-get install biosquid
```

جدول ۱۳-۲: یک فایل FASTA از miRNA های *C. elegans*

>cel-let-7 MI0000001 *C. elegans* let-7 stem-loop
UACACUGUGGAUCCGGUGAGGUAGUAGGUUGUAUAGUUUGGAAUAUUACCACCGGUGAAC
UAUGCAAUUUUCUACCUUACCGGAGACAGAACUCUUCGA
>cel-lin-4 MI0000002 *C. elegans* lin-4 stem-loop
AUGCUUCCGGCCUGUUCCCUGAGACCUCAAGUGUGAGUGUACUAUUGAUGCUUCACACCU
GGGCUCUCCGGGUACCAGGACGGUUUGAGCAGAU
>cel-mir-1 MI0000003 *C. elegans* miR-1 stem-loop
AAAGUGACCGUACCGAGCUGCAUACUUCCUUACAUGCCCAUACUAUAUCAUAAAUGGAUA
UGGAAUGUAAAGAAGUAUGUAGAACGGGGUGGUAGU
>cel-mir-2 MI0000004 *C. elegans* miR-2 stem-loop
UAAACAGUAUACAGAAAGCCAUCAAAGCGGUGGUUGAUGUGUUGCAAAUUAUGACUUUCA
UAUCACAGCCAGCUUUGAUGUGCUGCCUGUUGCACUGU
>cel-mir-34 MI0000005 *C. elegans* miR-34 stem-loop
CGGACAAUGCUCGAGAGGCAGUGUGGUUAGCUGGUUGCAUAUUUCCUUGACAACGGCUAC
CUUCACUGCCACCCCGAACAUGUCGUCCAUCUUUGAA
>cel-mir-35 MI0000006 *C. elegans* miR-35 stem-loop
UCUCGGAUCAGAUCGAGCCAUUGCUGGUUUCUUCCACAGUGGUACUUUCCAUUAGAACUA
UCACCGGGUGGAAACUAGCAGUGGCUCGAUCUUUUCC
>cel-mir-36 MI0000007 *C. elegans* miR-36 stem-loop
CACCGCUGUCGGGGAACCGCGCCAAUUUUCGCUUCAGUGCUAGACCAUCCAAAGUGUCUA
UCACCGGGUGAAAAUUCGCAUGGGUCCCCGACGCGGA
>cel-mir-37 MI0000008 *C. elegans* miR-37 stem-loop
UUCUAGAAACCCUUGGACCAGUGUGGGUGUCCGUUGCGGUGCUACAUUCUCUAAUCUGUA
UCACCGGGUGAACACUUGCAGUGGUCCUCGUGGUUUCU
>cel-mir-38 MI0000009 *C. elegans* miR-38 stem-loop
GUGAGCCAGGUCCUGUUCCGGUUUUUUCCGUGGUGAUAACGCAUCCAAAAGUCUCUAUCA
CCGGGAGAAAAACUGGAGUAGGACCUGUGACUCAU
>cel-mir-39 MI0000010 *C. elegans* miR-39 stem-loop
UAUACCGAGAGCCCAGCUGAUUUCGUCUUGGUAAUAAGCUCGUCAUUGAGAUUAUCACCG
GGUGUAAAUCAGCUUGGCUCUGGUGUC

۵- نسخه‌ی 2 (نسخه‌ی ویرایش C) Randfold را دریافت کنید (http://bioinformatics.psb.ugent.be/software/details/Randfold). نخست Makefile را تغییر داده و -I/usr/include/biosquid را به خط INCLUDE اضافه کنید (حال این خط بایستی INCLUDE = -I.-I/usr/include/biosquid را بخواند). برای کامپایل و نصب، به صورت زیر تایپ کنید:

```
$ make

$ sudocprandfold/usr/local/bin
```

۶- بسته‌ی PDF::API2 را نصب کنید. بدین منظور در اوبونتو به صورت زیر تایپ کنید:

```
$ sudo apt-get install libpdf-api2-perl
```

۷- miRDeep2 را دریافت کنید (www.mdc-berlin.de/8551903/en/research/research_teams/systems_biology_of_gene_regulatory_elements/projects/miRDeep). سپس آنرا از حالت فشرده خارج کرده و اجرا نمایید.

۱۳-۲-۴ اجرای miRDeep2

```
mapper.pl ctrl_trimmed.fasta -c -p/home/wong/rna_seq/genomes/ws220/genome -t
ctrl_trimmed_mapped.arf -o 4 -n -s ctrl_trimmed_processed.fa -v –m

miRDeep2.pl ctrl_trimmed_processed.fa/home/wong/rna_seq/genomes/ws220/
genome.fa ctrl_trimmed_mapped.arfmature.fa none hairpin.fa
```

نکات مرتبط با miRDeep2 عبارتند از:

۱- برنامه‌ها و فایل‌های مورد نیاز را به دایرکتوری public/common منتقل کنید تا همواره در مسیر باشند.

۲- یک دایرکتوری فرعی جدید برای اجرای miRDeep2 و تخلیه‌ی فایل‌های خروجی در آن بسازید.

۳- فایل‌های ورودی مورد نیاز را به دایرکتوری فرعی جدید miRDeep2 انتقال دهید.

۱۳-۲-۴-۱ خروجی *miRDeep2*

یکی از بهترین ویژگی‌های miRDeep2 این است که فایل‌های خروجی آن شامل یک صفحه‌ی html است که کلیه‌ی خروجی‌ها را جمع‌آوری کرده و به شما این امکان را می‌دهد که آنرا در قالب یک صفحه‌ی وب مطالعه نمایید. در صفحه‌ی وب مزبور لینک‌هایی به کلیه‌ی خروجی‌ها از جمله گرافیک‌هایی که در فایل‌های PDF جداگانه ذخیره شده‌اند، وجود دارد. البته می‌توانید نتایج مزبور را به صورت جداگانه نیز جستجو نمایید. فایل‌های خروجی در دایرکتوری کاری که برنامه در آن اجرا می‌شود، قرار می‌گیرند (مگر آنکه دایرکتوری دیگری به این کار تخصیص داده شود). در دایرکتوری کاری روی آیکون html با نشان تاریخ و زمان، کلیک کنید. در این مثال، این فایل عبارت از /mirdeepworking/expression_02_04_2013_t_15_15_09.html است (نگاره‌ی ۱۳-۱). نخستین مجموعه‌ی خروجی‌ها جزییات پارامترهای مورد استفاده شامل نسخه‌ی ویرایش miRDeep2 ، خوانش برنامه، نام فایلی که خوانش‌ها از آن گرفته شده‌اند، موقعیت و نام فایل ژنوم، نام فایل مکان‌یابی، نام فایل مرجع miRNA بالغ و سایر miRNA های بالغ را ارائه می‌کند. بخش بعدی خروجی‌ها عملکرد miRDeep2 را با توجه به تعداد miRNA های پیش‌بینی شده‌ی جدید که شناسایی شده و miRNA های شناخته شده بررسی می‌کند. بخش سوم شامل فهرستی از miRNA های جدید پیش‌بینی شده توسط miRDeep است که شامل ID موقت، امتیاز miRDeep2 ، شمارش خوانش‌ها در نواحی بالغ، نواحی حلقه‌ای و ستاره‌ای، معنی‌داری randfold ، لینک‌هایی به پایگاه‌های داده‌ی خارجی، لینک به بخش blastn سایت NCBI برای نتایج، توالی‌های بالغ اجماعی، توالی‌های ستاره‌ای اجماعی، توالی‌های پیش‌ساز اجماعی و مختصات پیش‌سازها بر اساس کروموزوم و موقعیت می‌باشد (نگاره‌ی ۱۳-۲). بخش چهارم فهرستی از miRBasemiRNA های بالغ با یک ID نشان، امتیاز miRDeep ، برآورد احتمال اینکه miRNA یک مثبت واقعی است، اعلانی در مورد موافقت با توالی بالغ miRBase ، شمارش کُل خوانش‌ها در قالب‌های بالغ، حلقه‌ای و ستاره‌ای، معنی‌داری randfold ، نام miRBasemiRNA بالغ، لینک به بخش جستجوی blastn در سایت NCBI و توالی‌های بالغ، ستاره‌ای و سنجاق‌سری اجماعی می‌باشد.

شما می‌توانید با کلیک روی ID موقت یک miRNA جدید، به یک گرافیک ایجاد شده توسط RNA فولد دسترسی داشته باشید که شامل ساختار سنجاق‌سری واقعی فولد شده توسط شبیه‌سازی رایانه‌ای، با تعداد خوانش‌ها برای هر بخش از ساختار سنجاق‌سری، امتیاز حداقل انرژی آزاد، امتیاز randfold و امتیاز توالی سید حفاظت شده است. علاوه بر این، یک همردیفی برای خوانش‌های مکان‌یابی شده با ساختار سنجاق‌سری شامل خوانش‌هایی که خیلی نزدیک مکان‌یابی

Parameters used

miRDeep2 version	2.0.0.5
Program call	/home/wong/mirdeep2/miRDeep2.pl ctrl_processed.fa /home/wong/rna_seq/genomes/ws220/genome.fa ctrl_mapped.arf mature.fa none hairpin.fa
Reads	ctrl_processed.fa
Genome	/home/wong/rna_seq/genomes/ws220/genome.fa
Mappings	ctrl_mapped.arf
Reference mature miRNAs	mature.fa
Other mature miRNAs	none

Survey of miRDeep2 performance for score cut-offs -10 to 10

	novel miRNAs			known miRBase miRNAs				
miRDeep2 score	predicted by miRDeep2	estimated false positives	estimated true positives	in species	in data	detected by miRDeep2	estimated signal-to-noise	excision gearing
10	3	2 ± 1	1 ± 1 (46 ± 37%)	112	112	2 (2%)	2.3	2
9	3	2 ± 1	1 ± 1 (45 ± 37%)	112	112	2 (2%)	2.2	2
8	3	2 ± 1	1 ± 1 (42 ± 37%)	112	112	2 (2%)	2.1	2
7	3	2 ± 2	1 ± 1 (41 ± 37%)	112	112	2 (2%)	2	2
6	3	2 ± 2	1 ± 1 (39 ± 36%)	112	112	2 (2%)	1.9	2
5	3	2 ± 2	1 ± 1 (38 ± 35%)	112	112	2 (2%)	1.8	2
4	3	2 ± 2	1 ± 1 (36 ± 35%)	112	112	2 (2%)	1.7	2
3	5	2 ± 2	3 ± 2 (53 ± 30%)	112	112	2 (2%)	2.4	2
2	22	4 ± 2	18 ± 2 (83 ± 9%)	112	112	62 (55%)	14.6	2
1	54	8 ± 3	46 ± 3 (85 ± 5%)	112	112	90 (80%)	12.3	2
0	69	40 ± 6	29 ± 6 (42 ± 9%)	112	112	93 (83%)	2.9	2
-1	79	64 ± 8	15 ± 7 (19 ± 9%)	112	112	93 (83%)	2	2
-2	99	91 ± 8	9 ± 7 (9 ± 7%)	112	112	97 (87%)	1.6	2
-3	124	123 ± 9	4 ± 6 (3 ± 5%)	112	112	97 (87%)	1.4	2
-4	137	166 ± 10	0 ± 0 (0 ± 0%)	112	112	98 (88%)	1.1	2
-5	150	223 ± 11	0 ± 0 (0 ± 0%)	112	112	100 (89%)	0.9	2
-6	166	271 ± 12	0 ± 0 (0 ± 0%)	112	112	100 (89%)	0.8	2
-7	188	319 ± 12	0 ± 0 (0 ± 0%)	112	112	100 (89%)	0.8	2
-8	220	369 ± 13	0 ± 0 (0 ± 0%)	112	112	100 (89%)	0.7	2
-9	257	419 ± 15	0 ± 0 (0 ± 0%)	112	112	100 (89%)	0.7	2
-10	281	474 ± 16	0 ± 0 (0 ± 0%)	112	112	100 (89%)	0.7	2

نگاره‌ی ۱۳-۱: فایل خروجی حاصل از miRDeep2 که امتیازات اجرا را نشان می‌دهد.

شده و این امکان را فراهم می‌آورند که بتوان هر نوع ایزومیری را مشاهده نمود، ایجاد می‌شود. مثالی از این مورد در نگاره‌ی ۱۳-۳ ارائه شده است.

۳-۱۳ miRanalyzer

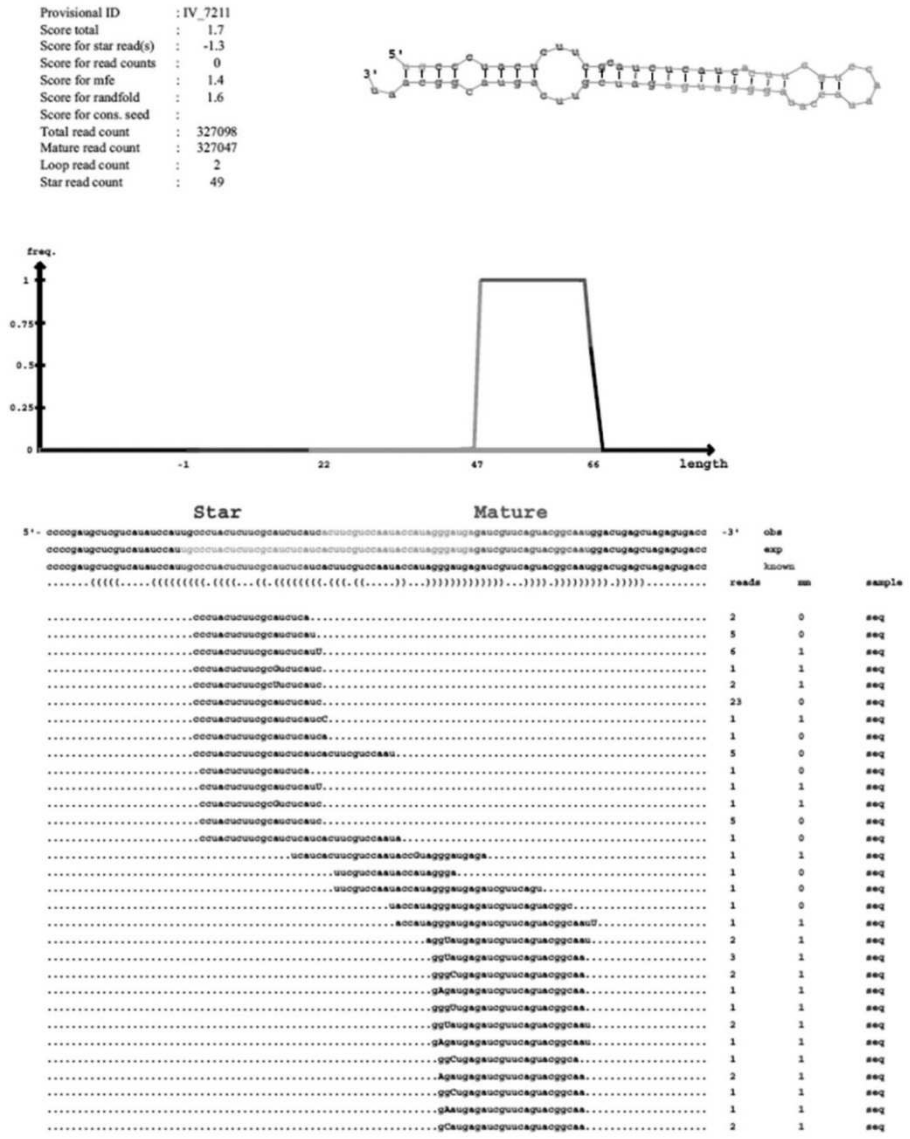
ابزار دیگری که miRNA های شناخته شده را شناسایی کرده و ساختارهای سنجاق‌سری miRNA های جدید که از داده‌های توالی‌یابی RNA حاصل آمده‌اند را پیشنهاد می‌دهد، miRanalyzer است (۴). همچنین این ابزار امکان مقایسات افتراقی بیان miRNA ها را بین دو مجموعه داده فراهم می‌آورد. در شرایط عملی، یک تفاوت بزرگ miRanalyzer با miRDeep2 در این است که miRanalyzer مبتنی بر وب است. همچنین این ابزار می‌تواند دانلود شده و به صورت موضعی اجرا گردد. در هر دو حالت، کاربر باید داده‌های توالی‌یابی RNA را خوشه‌بندی کرده و فرمت آنرا به Read-Count (شمارش خوانش) یا Multi-FASTA (چند FASTA) تغییر دهد. جدول ۱۳-۳ این فرمت‌ها را نمایش می‌دهد.

novel miRNAs predicted by miRDeep2

provisional id	miRDeep2 score	estimated probability that the miRNA candidate is a true positive	rfam alert	total read count	mature read count	loop read count	star read count	significant randfold p-value
X_16668	4.2e+2	0.46 ± 0.37		826	753	0	73	yes
IV_7271	2.1e+1	0.46 ± 0.37		37	35	1	1	yes
IV_12490	2.1e+1	0.46 ± 0.37		36	35	0	1	yes
X_17044	3.2	0.53 ± 0.30		818	748	11	59	yes
X_16689	3.2	0.53 ± 0.30		804	748	0	56	yes
IV_14843	2.9	0.83 ± 0.09		121	121	0	0	yes
IV_5013	2.8	0.83 ± 0.09		594	573	0	21	yes
II_2882	2.7	0.83 ± 0.09		42	41	0	1	yes
X_16990	2.4	0.83 ± 0.09		70	70	0	0	yes
V_15838	2.4	0.83 ± 0.09		345	328	0	17	yes
X_16661	2.3	0.83 ± 0.09		155	155	0	0	yes
IV_5316	2.3	0.83 ± 0.09		46064	46028	0	36	yes
IV_6089	2.2	0.83 ± 0.09		94	93	0	1	yes
X_16561	2.1	0.83 ± 0.09		696	696	0	0	yes
X_16952	2.0	0.83 ± 0.09		386	319	0	67	yes
III_4385	2.0	0.83 ± 0.09		12	8	0	4	yes
III_4580	2.0	0.83 ± 0.09		6828	5813	0	1015	yes
X_16998	2.0	0.83 ± 0.09		111	111	0	0	yes
X_16598	2.0	0.83 ± 0.09		26	26	0	0	yes
X_16902	2.0	0.83 ± 0.09		76	76	0	0	yes
I_1297	2.0	0.83 ± 0.09		204	202	0	2	yes
V_16054	2.0	0.83 ± 0.09		9890	9872	0	18	yes
I_44	1.9	0.85 ± 0.05		10	10	0	0	yes
V_15924	1.9	0.85 ± 0.05		62	62	0	0	yes
II_3022	1.9	0.85 ± 0.05		5197	5193	0	4	yes
IV_5318	1.9	0.85 ± 0.05		119	118	0	1	yes
X_16559	1.9	0.85 ± 0.05		496	496	0	0	yes

نگاره‌ی ۱۳-۲: فایل خروجی حاصل از miRDeep2 که miRNA های جدید پیش‌بینی شده را نشان می‌دهد. این جدول برش داده شده و توالی‌های miRNA بالغ و سنجاق‌سری در ستون‌های سمت راست حذف شده‌اند.

این فرمت‌ها توالی و تعداد خوانش‌ها در مجموعه‌ی داده را فهرست کرده و بدین ترتیب فشرده‌سازی مجموعه داده‌ها که برای یک برنامه‌ی مبتنی بر وب از اهمیت بالایی برخوردار است، انجام می‌شود. برای خوشه‌بندی و تبدیل فرمت، یک برنامه‌ی پِرل ارائه شده است. پس از آن کاربر آمادگی اجرای برنامه‌ی miRanalyzer را خواهد داشت. کاربر باید از صفحه‌ی وب، جاندار / ژنوم مورد نظر را از بین ۴۰ موردی که تاکنون تحت پشتیبانی قرار گرفته است (نظیر انسان، موش،



نگاره‌ی ۱۳-۳: خروجی گرافیکی ساختار سنجاق‌سری و خوانش‌های مکان‌یابی شده از miRDeep2

مگس سرکه و *C. elegans*) انتخاب نموده، پارامترهایی نظیر تعداد عدم تطابق‌های مجاز و تصمیم‌گیری در مورد پیش‌بینی miRNA های جدید یا شناسایی miRNA های شناخته شده را از یک منوی پایین‌رو انتخاب کرده و روی دکمه‌ی launch کلیک کـند. سِروِر miRanalyzer داده‌ها را

جدول ۱۳-۳: فرمت FASTA ، فرمت Read-Count و فرمت Muiti-FASTA

فرمت FASTA:

```
>gene1
ACTCTCGATCTATTT
>gene2
TCTCACGTGCGGTAAGC
>gene3
GTGATTGCATATCAT
…
```

فرمت Read-Count:

ACTCTCGATCTATTT	57882	
TCTCACGTGCGGTAAGC	23815	
GTGATTGCATATCAT	432	

فرمت Multi-FASTA:

```
  >gene1 57882
ACTCTCGATCTATTT
  >gene2 23815
TCTCACGTGCGGTAAGC
  >gene3 432
GTGATTGCATATCAT
```

این جدول تفاوت بین فرمت‌ها را نشان می‌دهد. miRanalyzer تنها فایل‌های داده با فرمت Read-Count و Multi-FASTA را به عنوان ورودی می‌پذیرد.

پردازش و آنالیز کرده و در پایان کار یک لینک miRanalyzerjobID ارائه می‌کند که از طریق آن نتایج در دسترس قرار می‌گیرند. این آنالیز بسته به اندازه‌ی مجموعه داده‌ها بین نصف روز تا یک هفته زمان می‌برد. خروجی که در صفحه‌ی وب ارائه می‌شود، پارامترهای استفاده شده و خلاصه‌ای از نتایج را نمایش می‌دهد. سایر بخش‌ها تعداد خوانش‌های مکان‌یابی شده با miRNA های شناخته شده، سایر دسته‌های RNA و RNA های پیش‌بینی شده‌ی جدید را نشان می‌دهند. با کلیک روی جعبه‌ی جزییات در هر کدام از این بخش‌ها، miRNA ها یا RNA های دقیق و یا خوانش‌های مربوط به هر کدام از آنها ارائه می‌گردد. مقایسه‌ی بین دو مجموعه‌ی داده می‌تواند به طور مستقیم و با استفاده از ابزارهای آنالیز افتراقی بیان صورت گیرد. در اینجا به طور ساده miRanalyzerjobID مربوط به دو مجموعه داده وارد شده است. مجموعه داده‌های ساده همراه با راهنمای عیب‌یابی در سایت miRanalyzer ارائه گردیده است. ابزار miRanalyzer را می‌توان از http://bioinfo5.ugr.es/miRanalyzer/miRanalyzer.php دریافت کرد. اسکریپت پِرل جهت خوشه‌بندی و تبدیل فرمت داده‌های ورودی miRanalyzer از http://web.bioinformatics.cicbiogune.es/microRNA/miRanalyser.php قابل دریافت است (نگاره‌ی ۱۳-۴).

Queing and Execution

Analysis completed
You can bookmark this page
Download all results in plain text here

Parameters

Species:	Cel	Assembly:	Ce6
Input:	mehg_ctrl.txt..	Mismatches (known):	1
Mismatches (library):	1	Mismatches (genome):	1
Score threshold:	0.9	Min. positives:	3
Type:	Full analysis	Solid:	no

Brief summary

unique reads:	443720	read count	18450502
filtered unique reads:	20056	filtered read count	101762
No known microRNA	274	No. known microRNAs	---
No microRNA (not miRBase)	---	No. new microRNAs	115
unique reads (after known)	395939	read count (after known)	1
unique reads (after lb)	393256	read count (after lb)	1
unique reads matched	277749	read count matched	1
unique reads not-matched	115507	read count not-matched	1477974

Mapping to known microRNA (miRBase 19)

Library/ Parameters	Mature	ambiguous mature	Mature-star	ambiguous mature-star	unobs. mature-star	ambiguous unobs. mature-star	hairpin	ambiguous hairpin
No.microRNA	274	7	0	0	0	0	121	3
fraction (number) of known microRNAs	74.7% (367)	---	0.0% (57)	---	---	---	54.3% (223)	---
unique reads	26882	70	0	0	0	0	768	7
fraction of unique reads	6.3%	0.017%	0.000%	0.000%	0.000%	0.000%	0.181%	0.002%
read count	2371520	488	0	0	0	0	2639	9
fraction of read count	12.9%	0.003%	0.000%	0.000%	0.000%	0.000%	0.014%	0.000%
links to detail pages	details	details	no results	no results	no results	no results	details	details

Alignment to other transcribed entities

Library/ Parameters	RefSeq_genes	Rfam
number of unique reads	58333	1196
fraction of unique reads	13.77%	0.28%
number of reads	1	3711
fraction of reads	67.55%	0.02%
Links	details	details

Predicted candidate microRNAs

No. of read clusters:	173783			
No. of checked candidates	46844			
No. new microRNAs:	115	Unique reads (read count):	629 (44440)	details
No. new microRNAs (trans filtered):	109	Unique reads (read count):	623 (4434)	details

نگاره‌ی ۱۳-۴: خروجی miRanalyzer

۱۳-۳-۱ اجرای miRanalyzer

۱- اسکریپت پِرل را برای خوشه‌بندی و تبدیل فرمت داده‌ها دانلود کنید

۲- اسکریپت پِرل را برای خوشه‌بندی و تبدیل فرمت داده‌های توالی‌یابی RNA اجرا کنید.

۳- از داده‌های با فرمت جدید به عنوان ورودی در miRanalyzer استفاده نمایید.

۱۳-۴ آنالیز هدف miRNA

miRNA ها mRNA های هم‌ریشه‌ی خود را با سازوکاری که هنوز به طور دقیق و کامل شناسایی نشده است، مورد هدف قرار می‌دهند. با این حال برخی از جنبه‌های کلیدی این سازوکار شناسایی هدف، شناخته شده است. مکمل بودن ۷ تا ۸ نوکلئوتید ابتدایی miRNA با mRNA هدف، پایداری ترمودینامیکی کمپلکس miRNA-mRNA و موقعیت انطباق‌های GC و AU مهم است. نوکلئوتیدهای ۲ تا ۹ واقع در انتهای '5 مولکول miRNA بالغ اصطلاحاً توالی سید نامیده می‌شود. انطباق‌های مکمل توالی‌های سید در این موقعیت‌ها با mRNA های هدف به 7-mer-A1 (۷ نوکلئوتید منطبق با یک آدنوزین در موقعیت A1)، 7-mer-m8 (۷ انطباق با یک عدم انطباق در موقعیت ۸) یا 8-mer (۸ نوکلئوتید منطبق در سید) دسته‌بندی می‌شوند. موقعیت هدف روی mRNA در ناحیه‌ی ناترجمان (UTR) انتهای '3 مولکول mRNA واقع شده است. البته شواهدی نیز وجود دارد که حاکی از آن است که miRNA ها می‌توانند اگزون‌ها و UTR های انتهای '5 را نیز هدف قرار دهند. حداقل سه روش مختلف برای آنالیز هدف miRNA به کار گرفته شده است. نخستین روش‌ها، پیش‌بینی‌های محاسباتی بر مبنای امتیازدهی و آنالیز آماری ترکیبات توالی مناسب بوده و از نظر تاریخی، قدیمی‌ترین روش‌ها هستند. این روش‌های پیش‌بینی به عنوان روش‌های مبنایی در دستورالعمل‌های پایه‌ی حاصل از جفت‌شدگی lin-4 miRNA:lin-14 mRNA باقی مانده و بعداً با شناسایی miRNA های بیشتر و اهداف‌شان بازنگری شدند. دوم، روش‌های هوش مصنوعی که بر مبنای مجموعه‌های آموزش مثبت و منفی برای دسته‌بندی mRNA ها به عنوان هدف یا غیرهدف بنا نهاده شده و جایگزین دقیق‌تری برای روش‌های ساده‌تر امتیازدهی انطباق توالی محسوب می‌گردند. سوم، روش‌های آزمایشی که صحیح‌تر و معتبرتر از روش‌های محاسباتی بوده ولی بُرونداد ضعیفی داشته و در نتیجه نمونه‌های اندکی از آنها می‌توان ذکر نمود. مطالعات اولیه مبتنی بر ساختارهای سنجاق‌سری miRNA های تراآلایی شده در داخل سلول‌ها و سپس آنالیز کاهش بیان ژن‌ها بوده است. در حال حاضر روش‌های توالی‌یابی RNA از جمله رسوب ایمنی همبر (CLIP) آرگونات و سایر پروتئین‌های متصل شونده به RNA که برای شناسایی کمپلکس miRNA:mRNA در محل[1] به کار گرفته می‌شوند، روش‌های پیشگام در اعتبارسنجی هدف هستند.

۱۳-۴-۱ روش‌های پیش‌بینی محاسباتی

اصیل‌ترین برنامه‌های پیش‌بینی محاسباتی مبتنی بر دستورالعمل‌های زیر است. این

1- *in situ*

دستورالعمل‌ها به تنهایی برای آنالیز هدف ضروری و کافی نبوده و اُریبی در امتیازدهی به اهداف بالقوه ایجاد می‌کنند. در عمل، ابزارهای محاسباتی متعددی مورد استفاده قرار گرفته و پیش‌بینی‌های مورد توافق در قالب بهترین پیش‌بینی ارائه می‌گردند.

این دستورالعمل‌ها عبارتند از:

۱- تطابق‌های مکمل در ناحیه‌ی سید که به صورت موقعیت‌های ۲ تا ۷ در miRNA تعریف می‌شوند.

۲- تطابق‌های مکمل جبرانی فراتر از ناحیه‌ی سید در حضور تطابق‌های ضعیف ناحیه‌ی سید

۳- حضور آدنوزین‌ها در مجاورت توالی جفت شده با سید در miRNA

۴- انطباق‌های چندگانه‌ی ناحیه‌ی سید یک miRNA در 3'UTR

۵- انطباق‌های چندگانه‌ی ناحیه‌ی سید miRNA های متفاوت در 3'UTR

۶- انطباق‌های توالی مکمل کُلی miRNA به 3'UTR

۷- حفاظت‌شدگی توالی‌های سید در بین گونه‌ها

چون ناحیه‌ی سید تنها تا ۸ نوکلئوتید طول داشته و این دستورالعمل‌ها نیز قدری مبهم هستند، لذا ابزارهای محاسباتی متعددی وجود دارند که از اختصاصی نبودن رنج برده و پیش‌بینی صدها هدف برای هر miRNA توسط آنها امری متداول و مرسوم است. با این حال این ابزارها هنوز به عنوان یک نقطه‌ی شروع برای ایجاد فرضیه‌های جدید و آنالیز آزمایشات پایین دستی مفید هستند. نمونه‌هایی از ابزارهای پیش‌بینی هدف miRNA در زیر ارائه شده است.

TargetScan (www.targetscan.org) یک ابزار مبتنی بر وب برای یافتن اهداف پیش‌بینی شده‌ی miRNA در حیوانات است. این ابزار از گونه‌های زیادی از جمله انسان، موش‌ها، مگس‌ها و نماتدها پشتیبانی می‌کند. این ابزار یک پایگاه داده‌ی دستیار[1] است. بنابراین پیش‌بینی‌ها با استفاده از اصول کلی یافتن اهداف miRNA که در فوق تشریح گردیدند، یافت می‌شوند (۵). miRNA ها در دسته‌هایی گروه‌بندی شده و سپس اهداف مربوط به کلیه‌ی خانواده‌ها فهرست می‌شوند. خروجی شامل جدولی است که می‌توان به آسانی آنرا به یک صفحه‌ی گسترده منتقل نمود. یک مزیت این ابزار این است که علاوه بر فهرست نمودن mRNA های پیش‌بینی شده، یک امتیاز بر مبنای حفاظت‌شدگی (Pct تجمعی[2]) و یک امتیاز جداگانه بر مبنای زمینه‌ی توالی (امتیاز زمینه کُل[3]) برای ارزیابی میزان اطمینان به پیش‌بینی‌ها تخصیص می‌دهد. به عنوان مثال، miRNA های با حفاظت‌شدگی فیلوژنتیکی بیشتر به طرف امتیاز Pct تجمعی سوق می‌یابند. یکی از عیوب این

1- Curated database
2- Aggregate Pct
3- Total Context Score

ابزار تعداد محدود گونه‌های پشتیبانی شده و عمق کم پشتیبانی برای توالی‌های miRNA ستاره‌ای است. با این حال اگر بخواهید با استفاده از پایگاه داده‌ی TargetScan پیش‌بینی‌های دلخواهی انجام دهید، کُل این پایگاه داده از سایت مزبور قابل دانلود خواهد بود. همچنین TargetScan در موارد دیگر، به عنوان مثال زمانی که یک mRNA را داشته باشید و بخواهید محل‌های هدف miRNA پیش‌بینی شده را بدانید، نیز قابل استفاده است. بخش FAQ (سوالات متداول) در این پایگاه به خوبی طراحی شده و اگر کاربر بخواهد از الگوریتم‌های دقیقِ موردِ استفاده آگاه شود، می‌تواند از ارجاعات موجود در این بخش، استفاده کند.

دستورالعمل TargetScan به شرح زیر است:

۱- مرورگر را روی TargetScan (www.targetscan.org) قرار دهید.

۲- گونه‌ی مورد نظر را با حرکت در منو و کلیک روی آن انتخاب نمایید.

۳- miRNA مورد نظر را با حرکت در منو و کلیک روی خانواده‌ی آن انتخاب کنید.

۴- جدول خروجی در صفحه‌ی وب را کپی کرده و در یک صفحه‌ی گسترده برگردان نمایید.

ویژگی منحصر به فرد سِروِر DIANA-microT (www.microrna.gr/microT) این است که کاربر می‌تواند توالی miRNA خود را وارد کرده و سپس سِروِر مزبور اهداف بالقوه را محاسبه نماید. این ابزار می‌تواند اهداف miRNA های شناخته شده و بالعکس را نیز جستجو نماید. الگوریتم‌های امتیازدهی این سِروِر بسته به انطباق ۷ ، ۸ یا ۹ نوکلئوتید سید در miRNA ، جفت وابل G:U در انتهای '3 مولکول miRNA یا انطباق ۶ نوکلئوتید در ۹ نوکلئوتید ابتدایی انتهای '5 مولکول miRNA عمل کرده و امتیاز می‌دهند. حفاظت‌شدگی موقعیت هدف در بین ۲۷ گونه برای این محاسبه منظور می‌گردد. با شبیه‌سازی miRNA ها جهت محاسبه‌ی نسبت سیگنال به نویز، موارد مثبت غلط نیز کنترل می‌شود (۶). جستجوی miRNA در miRBase به سادگی انجام شده و در خروجی اهداف پیش‌بینی شده همراه با Pct تجمعی و امتیاز زمینه‌ی کُلی ارائه می‌گردد.

miRBase شامل پیش‌بینی‌هایی برای هر miRNA حاصل از TargetScan است. همچنین خروجی گرافیکی جزییات انطباق مکمل و عدم انطباق در توالی‌های miRNA در طول 3'UTR را نشان می‌دهد. چون در حال حاضر بهترین الگوریتم‌های محاسباتی با ابزارهای بیوانفورماتیکی که می‌توانند برای پیش‌بینی به کار گرفته شوند، سازگار نیستند، لذا آزمایشگاه‌های مختلف از چندین ابزار برای دستیابی به پیش‌بینی‌های اجماعی استفاده می‌کنند. از طریق www.mirbase.org نیز می‌توان به miRBase دسترسی یافت. توضیحات مفصل‌تری راجع به miRBase در ادامه‌ی این فصل ارائه می‌گردد.

۱۳-۴-۲ روش‌های مبتنی بر هوش مصنوعی

ماشین‌های بردار پشتیبان[1] (SVM ها) ابزارهای پایه‌ی یادگیری ماشین هستند. ویژگی‌های داده‌ها در فضای بردار با ابعاد بزرگ مکان‌یابی می‌شوند. تعداد ویژگی‌ها نامحدود بوده ولی می‌تواند شامل مواردی نظیر انطباق‌ها در سید، عدم انطباق‌ها در سید، انرژی آزاد سید یا بخش '3 و توالی‌های مبتنی بر موقعیت باشد. وقتی که همه‌ی ویژگی‌ها بردار‌بندی شدند، نمونه‌های مختلف می‌توانند از مجموعه‌ی آموزش[2] مجزا شده و یک مرتب کننده[3] که بهترین بردارها را در نمونه جدا می‌کند، ایجاد می‌شود. این کار می‌تواند مستقیماً به عنوان یک فراطرح اختیاری که بردارها را در آموزش جدا می‌نماید، تلقی گردد. سپس داده‌های واقعی و مرتب کننده که بر مبنای مجموعه‌ی آموزش ایجاد شده‌اند، برای تمایز یا مجزا نمودن داده‌ها به کار گرفته می‌شوند. در عمل مجموعه‌ی آموزش می‌تواند شامل مثال‌های مثبت از miRNA هایی که بیان mRNA را کاهش داده و نیز مثال‌های منفی از ژن‌هایی که تغییری در بیان‌شان ایجاد نشده و هر دو از یک آزمایش ریزآرایه اقتباس شده‌اند، باشد. ویژگی‌ها از miRNA و توالی 3'UTR مولکول mRNA که بیانش کاهش یافته است، استخراج می‌شوند. توالی‌های تصادفی نیز گاهی اوقات به عنوان یک مثال منفی در آموزش استفاده می‌شوند. SVM ها با کمک نُه مجموعه‌ی آموزش با ۰ تا ۵۱ مثال مثبت و ۰ تا ۱۱۴ مثال منفی با در نظر گرفتن پنج ویژگی نخست به عنوان ویژگی‌های مهم (شامل: انطباق نوکلئوتید پنجم، انرژی آزاد '5 ، انطباق نوکلئوتید ششم، انطباق نوکلئوتید چهارم و انطباق‌های AU در بخش '5) پیاده‌سازی شد (۷). یک مطالعه‌ی SVM که اخیراً با استفاده از یک مجموعه‌ی آموزش صورت گرفته است نشان داد که حفاظت‌شدگی انطباق سید، انطباق باز انتهایی و ناحیه‌ی انطباق 7a سید ویژگی‌های مهمی هستند (۸). نتایج این مطالعه در ابزار مرورگر www.mirdb.org قرار داده شده و می‌توان در آن اهداف miRNA های مورد نظر یا miRNA هایی که ژن‌های مورد نظر را هدف قرار می‌دهند، برای انسان، موش، موش صحرایی، سگ یا مرغ جستجو نمود (۸ و ۹).

نقشه‌های خودسازمان‌ده[4] (SOM ها) روش دیگری از هوش مصنوعی است که بر مبنای یک الگوریتم یادگیری نظارت‌ناپذیر بنا نهاده شده و برای یافتن اهداف miRNA ها به کار گرفته می‌شوند. فرآیند اولیه‌ی یادگیری شامل خوشه‌بندی داده‌ها (به عنوان مثال: توالی‌ها) در فضای چند بُعدی است. فرآیند مکان‌یابی بعدی شامل قرار دادن داده‌های ورودی جدید در نقشه است. در mirSOM توالی‌های فرعی 3'UTR دریافت شده و خوشه‌بندی گردیدند (۱۰). نتایج شامل یک

1- Support Vector Machine (SVM)
2- Training set
3- Classifier
4- Self-Organizing Map (SOM)

نقشه‌ی خودسازمان‌ده با ۳۲ × ۳۲ = ۹۶۴ نرون بوده که هر نرون دارای صفر، یک یا بیشتر از ۱/۸ میلیون زنجیره‌ی فرعی ۲۲ نوکلئوتیدی از 3'UTR ژن‌های شناخته شده‌ی *C. elegans* بوده است. سپس miRNA ها در SOM مکان‌یابی شده‌ی ژن‌های متناظر با توالی‌های 3'UTR در نرون به عنوان اهداف کاندیدا در نظر گرفته شدند. mirSOM در مقایسه با اکثر ابزارهای دیگر از حساسیت بالایی برخوردار بوده و اختصاصی بودن آن نیز به میزان زیادی بهبود یافته است. متاسفانه در حال حاضر این ابزار تنها از داده‌های *C. elegans* پشتیبانی می‌کند. رابط کاربری mirSOM این امکان را برای کاربر فراهم می‌کند که یک miRNA را وارد کرده و mRNA های پیش‌بینی شده را به عنوان خروجی دریافت کند. mirSOM را می‌توان از www.oppi.uef.f/bioinformatics/mirsom/ دریافت نمود.

۱۳-۴-۳ روش‌های مبتنی بر پشتیبانی آزمایشی

اگر پیش‌بینی‌های محاسباتی با شواهد آزمایشگاهی پشتیبانی شوند، بیشتر مورد استقبال زیست‌شناسان قرار می‌گیرند. خوشبختانه در طی سال‌های اخیر، تلاش‌های متعددی در این جهت صورت گرفته است. اکثر روش‌های آزمایشگاهی محدودیت در بُروندادد دارند. ولی این روش‌ها به دلیل یافتن برهمکنش‌های miRNA:mRNA ارزشمند هستند. در حال حاضر برخی بُروندادهای حاصل از روش‌های آزمایشگاهی در حدی هستند که یک پایگاه داده‌ی تخصصی از نتایج آنها ایجاد شده است. درسال‌های اخیر یک پایگاه داده‌ی دستیار، miRNA های دارای پشتیبانی آزمایشگاهی از مقالات را پوشش داده است. در زیر چند پایگاه داده که حاوی برهمکنش‌های miRNA:mRNA بوده و بر مبنای داده‌های آزمایشگاهی پیش‌بینی و تایید شده‌اند، معرفی می‌گردند.

mirWIP به عنوان یک جایگزین برای پیش‌بینی‌های محاسباتی خالص محسوب می‌گردد. با این ابزار می‌توان فهرستی از جفت‌های miRNA-mRNA که از آزمایشات رسوب ایمنی حاصل از کمپلکس RISC در *C. elegans* به دست آمده‌اند، فراهم آورد (۱۱ و ۱۲). این ابزار نخستین بار برای فراهم آوردن فهرستی از کمپلکس‌های miRNA:mRNA که با شواهد آزمایشگاهی تایید شده بودند، و سپس برای بسط و گسترش معیارهای امتیازدهی بهبود یافته برای پیش‌بینی هدف miRNA به کار گرفته شد. پیش‌بینی هدف اولیه در mirWIP متناظر بر حداقل انرژی آزاد، حفاظت‌شدگی فیلوژنتیکی و جفت‌شدگی سید بوده است. سپس mirWIP از داده‌های آزمایشگاهی رسوب ایمنی استفاده کرده تا الگوریتم امتیازدهی خود راکه در حال حاضر شامل ویژگی‌های انطباق سید در سمت '5 ، دسترسی ساختاری و انرژی نقاط اتصال است، بهبود بخشد. mirWIP از www.mirwip.org در دسترس است.

TarBase یک پایگاه دستیار برای اهداف miRNA دارای شواهد آزمایشگاهی است. TarBase از یک روش نیمه خودکار مبتنی بر متن‌کاوی در مقالات بهره می‌برد (۱۳). این پایگاه داده حاوی ۶۵۸۱۴ برهمکنش ژن-miRNA دارای تایید آزمایشگاهی است. داده‌های آزمایشگاهی از روش‌های مختص ژن miRNA نظیر ژن‌های گزارشگر، qRT-PCR و وسترن بلات تا روش‌های پُربرونداد‌تر نظیر ریزآرایه‌ها و توالی‌یابی RNA (نظیر HIST-CLIP ، PAR-CLIP و توالی‌یابی تخریبی) به دست می‌آیند. شواهد محاسباتی پشتیبانی کننده‌ی نتایج آزمایشگاهی نیز از طریق امتیازات هدف DIANA microTmiRNA در دسترس هستند. دسترسی به داده‌ها و یک رابط کاربری آسان و جذاب از TarBase از طریق وبسایت www.microrna.gr/tarbase امکان‌پذیر است.

miRTarBase یک پایگاه داده‌ی دستیار دیگر برای هدف-miRNA است که از مقالات بهره می‌گیرد. با استفاده از حدود ۲۰۰۰ مقاله، ۳۵۷۶ برهمکنش هدف-miRNA در ۱۷ گونه از جمله انسان تایید شده است (۱۴). در مقایسه با TarBase ، این پایگاه حاوی برهمکنش‌های هدف-miRNA حاصل از مطالعات توالی‌یابی RNA نبوده و بنابراین ممکن است که مورد توجه کاربرانی که تمایل دارند از داده‌های تک ژن حاصل از مطالعات اعتبارسنجی استفاده نمایند، قرار گیرد. miRTarBase از طریق وبسایت www.miRTarBase.mbc.nctu.edu.tw در دسترس است.

۱۳–۵ تلفیق داده‌های توالی‌یابی miRNA و توالی‌یابی mRNA

هدف محققین در اکثر مطالعات توالی‌یابی miRNA این است که علاوه بر شناسایی miRNA های بیان شده در یک نمونه‌ی بافتی خاص، mRNA های تنظیم شده توسط miRNA های مزبور را نیز شناسایی کنند. این مساله سال‌های متمادی و حتی پیش از ابداع NGS نیز مطرح بوده است. همان‌گونه که در بالا نیز اشاره شد، برای miRNA های منفرد، به سادگی می‌توان آن miRNA را از فهرست miRNAهایی که بیان‌شان تغییر یافته است انتخاب کرده و به شیوه‌ی محاسباتی mRNA هدف را پیش‌بینی نمود. بسیاری از آزمایشگاه‌ها یک گام جلوتر رفته و پس از انتخاب یک miRNA خاص، سلول‌هایی را با ساختار سنجاق‌سری آن miRNA تراآلایی یا تراریخته نموده و سپس با کمک روش‌هایی نظیر qRT-PCR ، ریزآرایه یا توالی‌یابی RNA ، mRNA های تنظیم شده توسط miRNA مزبور را شناسایی نموده و نقش عملکردی آنرا تایید می‌نمایند. همچنین این کار می‌تواند با روش‌های دیگری نظیر ناک‌اوت یا ناک‌داون کردن miRNA و بررسی بیان mRNA های هدف کاندیدا صورت گیرد. علی‌رغم اینکه این روش‌های آزمایشگاهی زمان‌بر و هزینه‌بردار می‌باشند، ولی برای تایید آزمایشگاهی یک miRNA و هدفش قابل استفاده هستند.

در یک مطالعه‌ی معمولی توالی‌یابی miRNA ممکن است که ده‌ها miRNA با بیان تغییر یافته حاصل آید. به همین ترتیب در مطالعات توالی‌یابی RNA روی همان بافت نیز ممکن است صدها mRNA با بیان تغییر یافته مشاهده گردد. تلفیق این نتایج به آسانی صورت نمی‌گیرد. یک روش ساده، همبسته نمودن سطوح بیان miRNA ها در یک بافت یا یک نوع سلول با سطوح بیان mRNA ها در همان بافت است. یک روش پیچیده‌تر شامل افزودن پیش‌بینی هدف عامل رونویسی و ساختن شبکه‌های ژنی بر مبنای این داده‌ها است (۱۵). نمونه‌ای از این روش شامل ایجاد حلقه‌های بازخورد[1] miRNA-فاکتور رونویسی است (۱۶). در این روش، شبکه‌های mRNA-عامل رونویسی-miRNA از طریق پایگاه‌های داده‌ی عامل رونویسی و ابزارهای پیش‌بینی هدف-miRNA ایجاد می‌شوند. سپس این شبکه‌ها با داده‌های بیان miRNA و mRNA حاصل از پایگاه‌های داده‌ی بیان ژن نظیر Gene Expression Omnibus (GEO) و ArrayExpress تلفیق شده و داده‌های غنی‌سازی هدف به روش آماری امتیازدهی می‌شوند. بالاترین امتیاز به شبکه‌های mRNA-عامل رونویسی-miRNA که با شواهد حاصل از پایگاه‌های بیان ژن پشتیبانی می‌شوند، تعلق می‌گیرد.

تلیفق داده‌های miRNA-mRNA با ایجاد چالش ادامه می‌یابد. یک عامل زمینه‌ای این است که برهمکنش‌های miRNA و mRNA در بُعد چهارم رُخ می‌دهند. این بدان معناست که این برهمکنش‌ها علاوه بر ملاحظات مکانی، به زمان در طی تکامل و پیری نیز بستگی دارند. افزایش دو برابری miRNA در یک آزمایش توالی‌یابی miRNA الزاماً به معنای دو برابر شدن بیان آن miRNA یا به معنای دو برابر شدن تعداد سلول‌هایی که آن miRNA را بیان می‌کنند، نیست. اگر miRNA بیان شود، آیا عوامل دیگر اجازه‌ی هدف‌گیری و تنظیم mRNA را به آن خواهند داد؟ در حال حاضر کارهای زیادی در این زمینه باید صورت گیرد و به نظر نمی‌رسد که پاسخ‌دهی به این پرسش، آسان باشد.

۱۳-۶ پایگاه‌های داده و منابع RNA های کوچک

۱۳-۶-۱ خوانش‌های توالی‌یابی RNA مربوط به miRNA در miRBase

miRBase یک پایگاه داده‌ی دستیار miRNA برای اهداف تحقیقاتی است. این پایگاه حاوی توالی‌ها، ساختارها، اهداف احتمالی و مراجعی برای گونه‌های متعدد است (۱۷). miRBase امکان جستجوی توالی‌ها و حاشیه‌نگاری‌های miRNA بر مبنای نام یا کلمات کلیدی آن را فراهم می‌آورد. در حال حاضر این پایگاه حاوی بیش از ۲۰۰۰۰ مدخل miRNA است (نسخه‌ی 19، آگوست ۲۰۱۲). کاربر می‌تواند miRN ها را بر مبنای گونه جستجو نماید. به عنوان مثال نسخه‌ی

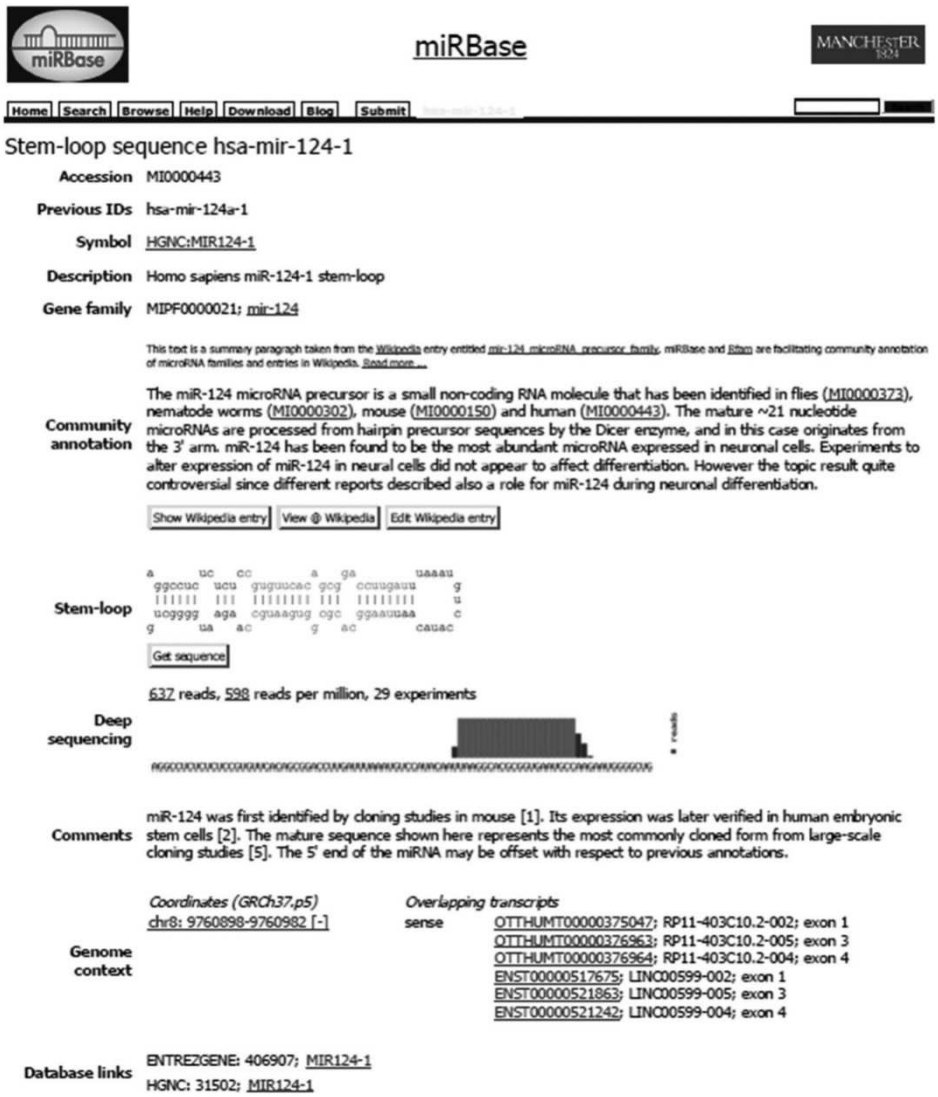
1- Feed-forward loop

فعلی miRBase (نسخه‌ی 20) دارای ۱۸۴۲ پیش‌ساز و ۲۵۷۸ توالی بالغ miRNA انسانی است. بخش دانلود به کاربر اجازه می‌دهد که کلیه‌ی توالی‌های miRNA در پایگاه داده را دانلود کرده یا تنها توالی‌های پیش‌ساز یا بالغ را دانلود نماید. همچنین این پایگاه فایل‌های قابل دانلود برای خانواده‌های miRNA دارد. یک ویژگی جدید که اخیراً به این پایگاه افزوده شده است، داده‌های دستیار روی خوانش‌هایی است که از موجودیت هر miRNA پشتیبانی می‌کنند (۱۸). در ادامه، حاشیه‌نگاری و حفظ تعداد خوانش‌های تایید شده به روش آزمایشگاهی برای miRNA ها چالش‌برانگیز خواهد بود. ولی miRBase در مواجهه با حجم بسیار بالای داده‌های جدید کاری استثنایی انجام می‌دهد (۱۹). به عنوان یک مثال Cel-mir-124 به عنوان یکی از مدخل‌های miRBase در نگاره‌ی ۱۳-۵ نمایش داده شده است. این مدخل شامل شماره‌ی دسترسی، توضیح، خانواده‌ی ژنی، حاشیه‌نگاری عمومی، نواحی ساقه-حلقه همراه با انطباق‌یافتگی‌ها، حلقه و برآمدگی، تعدادخوانش‌های توالی‌یابی عمیق که از حاشیه‌نگاری پشتیبانی می‌کنند، موقعیت خوانش‌های توالی‌یابی عمیق و زمینه‌ی ژنومی که ضرورتاً مختصات دقیق و موقعیت آنرا نسبت به رونوشت‌ها نشان می‌دهد، است.

نتایج دیگر شامل شواهد آزمایشگاهی برای miRNA ، اهداف تایید شده و اهداف پیش‌بینی شده هستند. شواهد آزمایشگاهی می‌توانند از کلون کردن، نُورزِن بلات، توالی‌یابی یا توالی‌یابی CLIP به دست آیند. اهداف تایید شده از TARBASE و اهداف پیش‌بینی شده از MICRORNA ، ORG ، RNA22-CEL یا TARGETSCAN-WORM دریافت می‌گردند. در پایان نیز مهم‌ترین منابع و مراجع مرتبط با آن miRNA خاص ذکر می‌گردد.

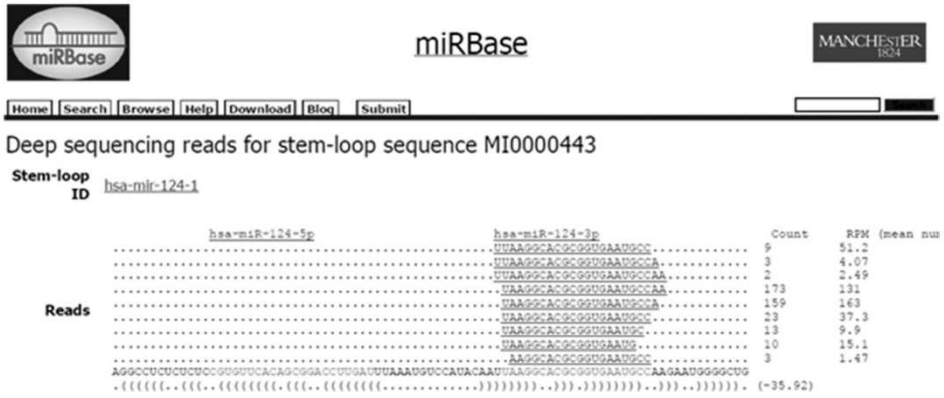
از دیدگاه ساقه - حلقه، به دست آوردن تعداد دقیق خوانش‌ها برای miRNA ها در ساقه - حلقه امکان‌پذیر است. با کلیک روی عدد موجود در بالای خوانش‌ها، نمای دیگری از خوانش‌های توالی‌یابی عمیق برای توالی ساقه - حلقه باز می‌شود (نگاره‌ی ۱۳-۶). در اینجا می‌توان توالی خوانش‌ها و تعداد دقیق خوانش / شمارش برای یک توالی خاص را هم در مقادیر مطلق و هم بر مبنای تعداد شمارش‌ها در واحد RPM (میانگین تعداد خوانش‌ها در هر میلیون) نرمال‌سازی شده را مشاهده نمود. مزیت دیگر این نوع نمایش آن است که کاربر می‌تواند ایزومیرهای توالی‌یابی شده و سهم آنها در کل شمارش‌ها را ملاحظه نماید. در نمای دیگر می‌توان به آزمایش و شمارش خوانش حاصل از داده‌ها دسترسی داشت.

مرورگر ژنومی UCSC یک مرورگر همه‌کاره برای یافتن ویژگی‌های یک توالی در سطح ژنومی و با استفاده از مسیرهای حاشیه‌نگاری است. هر مسیری یک حاشیه‌نگاری خاص را بر مبنای آزمایشات زیستی فهرست نموده و دسته‌بندی می‌نماید. صدها مسیر حاشیه‌نگاری ممکن برای هر



نگاره‌ی ۱۳-۵: نمایی از miRBase مربوط به یک miRNA

توالی وجود داشته و کاربر می‌تواند با کلیک کردن روی مسیرهایِ مختلفِ حاشیه‌نگاری، به آنها دست یابد. برای داده‌های توالی‌یابی RNA ، کاربر می‌تواند خوانش‌های حاصل از آزمایشات تـوالی‌یابی در سرتاسر یک ناحیه‌ی خاص از ژنوم را ملاحظه نماید. داده‌ها از آزمایشات مختلـف که



نگاره‌ی ۱۳-۶: مدخل miRBase برای خوانش‌های توالی‌یابی RNA مربوط به has-mir124-1 تایید شده از آزمایش‌ها

در پروژه‌ی Encode وجود دارند، اخذ گردیده‌اند. مرورگر ژنومی UCSC از طریق http://genome.ucsc.edu/ قابل دسترسی است.

۱۳-۶-۲ اطلس‌های بیان miRNA ها

microRNA در وبسایتی به همین نام (microRNA.org) شامل دو ابزار است (۲۰). این ابزار نخست نتایج پیش‌بینی‌های محاسباتی را برای محل‌های هدف miRNA:mRNA بر مبنای یک الگوریتم رگرسیون بردار پشتیبان هوشمند که از آزمایشات ترانسکریپتومی آموزش دیده است، فهرست می‌نماید. کاربر می‌تواند بر مبنای miRNA ، اهداف را جستجو کرده و یا بر مبنای اهداف، miRNA ها را جستجو نماید. یک ویژگی منحصر به فرد و قدرتمند در این ابزار وجود داشته که امکان دستیابی به پروفایل بیانی miRNA هایی که از تشکیل کتابخانه‌های RNA کوچک و توالی‌یابی بیش از ۳۰۰۰۰۰ همسانه‌ی حاصل از ۲۵۶ کتابخانه‌ی RNA کوچک مربوط به ۲۶ سامانه‌ی عضو یا رده‌های سلولی در انسان، موش صحرایی و موش حاصل آمده‌اند، را فراهم می‌آورد. الگوی بیان یک miRNA می‌تواند در قالب نمودار حرارتی، نمودار میله‌ای یا انواع نمودارهای میله‌ای سه بُعدی رویت گردد.

piRNABank شامل توالی‌های پیوی قابل جستجو، موقعیت‌شان، نقشه‌ها و ابزارهایی برای آنالیز است (۲۱). این پایگاه از piRNA های انسان، موش، موش صحرایی، مگس سرکه، ماهی زِبرا و پلاتی‌پوس پشتیبانی می‌کند. در حال حاضر بیش از ۲۰۰۰۰ توالی piRNA انسانی غیرهمپوشان در این پایگاه داده موجود است. یکی از ویژگی‌های مفید این پایگاه، قابلیت مشاهده و دانلود

توالی‌های piRNA در یک خوشه است. بنابراین اگر نیازمند کلیه‌ی piRNA های بیان شده در یک ناحیه‌ی ژنومی خاص هستید، می‌توانید کلیه‌ی صدها یا هزاران توالی piRNA دانلود شده در فرمت FASTA را دریافت نمایید. برای دسترسی به piwiDB می‌توان از وبسایت pirnabank.ibab.ac.in/index.shtml استفاده نمود.

Rfam یکی از قدیمی‌ترین پایگاه‌های داده‌ی RNA موجود از ۱۰ سال قبل تاکنون بوده است (۲۲). این پایگاه داده، RNA ها را بر مبنای خانواده‌های‌شان دسته‌بندی کرده و آنها را از طریق همردیفی‌ها، ساختارهای ثانویه و مدل‌های کوواریانس ارائه می‌کند. این پایگاه علاوه بر ژن‌های RNA های نارمزگر، RNA های کاتالیزوری را نیز شامل می‌شود. این پایگاه داده به کاربر اجازه می‌دهد که علاوه بر بازیابی داده‌های RNA ، انطباق‌ها با توالی‌های ورودی RNA را یافته و حاشیه‌نگاری و همردیفی‌ها را رویت کند. آخرین ویرایش Rfam (نسخه‌ی 11.0) حاوی ۲۲۰۸ خانواده‌ی RNA بوده است. از طریق نشانی http://rfam.sanger.ac.uk/ می‌توان به Rfam دسترسی داشت.

miRGator را می‌توان در نشانی mirgator.kobic.re.kr یافت (۲۳). این پایگاه یک پورتال تخصصی برای RNA بوده که شامل داده‌های NGS ، بیان و mRNA هدف است.

۱۳-۶-۳ پایگاه‌های داده برای داده‌های توالی‌یابی CLIP و توالی‌یابی تخریبی

starBase یک پایگاه داده است که داده‌های ۲۱ توالی‌یابی آرگونات یا TNRC6 CLIP و ۱۰ توالی‌یابی تخریبی از شش جاندار شامل انسان، موش، *C. elegans* ، *Arabidopsis thaliana* ، *Oryza sativa* (برنج) و *Vitis vinifera* (انگور) را جمع‌آوری و آنالیز کرده است (۲۴). این پایگاه داده از نوع پایگاه‌های تعاملی بوده و دورنمای ژنومی محل‌های اتصال آرگونات و محل‌های برش miRNA را نمایش می‌دهد. ترکیب مجموعه‌ی داده‌های حاصل از توالی‌یابی CLIP و توالی‌یابی تخریبی با پیش‌بینی‌های محاسباتی به طور تقریبی ۶۶۰۰۰ ارتباط تنظیمی هدف-miRNA ایجاد کرده و دانش موجود در این زمینه را به نحو چشمگیری افزایش داده است. starBase قابل جستجو بوده و بنابراین می‌توان یک توالی سید را وارد کرده و موقعیت هدف حاصل از توالی‌یابی CLIP را از طریق سِرور CLIPSearch یافته و یا یک توالی کوتاه RNA را وارد کرده و یک توالی حاصل از توالی‌یابی تخریبی را از طریق سِرور Degradome Search موقعیت‌یابی نمود. این پایگاه اخیراً نیز به‌روزرسانی شده است (۲۵). starBase از طریق http://starbase.sysu.edu.cn/ قابل دسترس است.

۱۳-۶-۴ پایگاه‌های داده برای miRNA ها و بیماری‌ها

miRò یک رابط کاربری تحت وب بوده که به کاربران برای یافتن ارتباط بین miRNA ها و بیماری‌ها کمک می‌کند (۲۶). این پایگاه، داده‌های حاصل از miRNA ها را با پیش‌بینی‌های mRNA هدف (TargetScan، PicTar ، miRanda) و اعتبارسنجی‌های آزمایشی (miRecords) تلفیق می‌کند. سپس ژن‌هایی که با miRNA ها مرتبط هستند، به Gene Ontology و Genetic Association Database پیوند زده می‌شوند. پس از آن این پایگاه miRNA مورد نظر را با یک بیماری انسانی مرتبط می‌نماید. چهار نوع جستجو می‌تواند انجام شود: بازیابی اطلاعات درباره‌ی یک miRNA ، ژن، اصطلاح حاشیه‌نگاری بیماری یا بافت، یافتن یک ارتباط بین یک miRNA و یک بیماری یا راهی دیگر در اطراف آن، یافتن کلیه‌ی miRNA های مرتبط با یک بیماری، آزمون جفت‌های جدید هدف-miRNA برای ارتباطات، و اجرای یک جستجوی پیشرفته با انتخاب یک موضوع و تخصیص قیود و محدودیت‌هایی به آن برای محدود ساختن خروجی‌ها. به عنوان مثال، جستجو با Parkinson's disease فهرستی از ۷۶۵۸ مدخل مربوط به miRNA ، ژنی که توسط آن هدف‌گیری می‌شود و وضعیت هدف مزبور (اعتبارسنجی شده‌ی آزمایشی یا پیش‌بینی‌شده) و برنامه‌ای که آنرا پیش‌بینی کرده است، را ارائه می‌دهد. برای دسترسی به miRò می‌توان از نشانی http://ferrolab.dmi.unict.it/miro/index.php استفاده نمود.

miRdSNP یک پایگاه داده برای کاربرانی است که مایل هستند ارتباط بین miRNA ها و موقعیت‌های هدف در 3'UTR که چندشکلی‌های تک نوکلئوتیدی (SNP ها) مرتبط با بیماری‌ها در نزدیکی آن واقع شده است، را بیابند (۲۷). در حال حاضر این پایگاه داده دارای ۷۸۶ مورد SNP مرتبط با بیماری‌ها و ۲۰۴ مورد بیماری است. این ابزار به ویژه برای کاربرانی که می‌خواهند یک SNP بالقوه‌ی مرتبط با یک miRNA بیابند، مفید خواهد بود. miRdSNP از طریق نشانی mirdsnp.ccr.buffalo.edu قابل دسترسی است.

۱۳-۶-۵ پایگاه‌های عمومی برای جوامع کاربری و منابع پژوهشی

RNAcentral تلاشی در جهت ایجاد یک پایگاه داده‌ی فدراتیو از پایگاه‌های داده‌ی مرتبط با توالی‌های RNA است. همگام با افزایش تعداد خانواده‌های RNA های نارمزگر کوچک و افزایش پژوهش‌ها در زمینه‌ی عملکرد آنها، پایگاه‌های داده برای تطبیق توالی‌ها و حاشیه‌نگاری‌های RNA مربوط به عملکرد زیستی ایجاد شدند. این امر منجر به ایجاد پایگاه‌های داده‌ی بسیار مجرب و تخصصی برای اکثر خانواده‌های RNA گردید (نظیر: miRBase ، Rfam ، starBase و غیره). یکی از مزایای این روند، ایجاد دانش تخصصی در هر زمینه و توانایی انطباق سریع با اطلاعات جدید

موثر در آن است. ولی عیب این کار هم قطعه قطعه شدن اطلاعات متناظر با توالی‌های RNA و رابطهای کاربری و مدل‌های پایگاه‌های داده‌ی مختلف است که باید یک پژوهشگر زیست‌شناسی RNA با آنها آشنا باشد. بنابراین اکثر مدیران پایگاه‌های داده‌یRNA به این موضوع توجه کرده و در تلاش هستند که یک پایگاه داده‌ی فدراتیو از توالی‌های RNA تشکیل دهند (۲۸). این روش فدراتیو به هر پایگاه داده اجازه می‌دهد که شناسه، مدیریت، سیاست‌ها و تعامل خود با کاربران را حفظ کرده ولی از مزیت یک پایگاه داده و پورتال متمرکز که می‌تواند منبعی برای جامعه‌ی کاربری جهت دسترسی به توالی‌های RNA ، ذخیره و ارائه‌ی آنها باشد، نیز برخوردار شوند. چنین روشی در زمینه پروتئین‌ها موفقیت‌آمیز بوده است (نظیر پایگاه داده‌ی InterPro). این پروژه کمک‌های مالی خوبی دریافت کرده و در حال حاضر در انستیتو بیوانفورماتیک اروپایی EMBL راه‌اندازی شده است.

miRNAblog ۶-۶-۱۳

miRNAblog یک منبع متمرکز برای دستیابی به مطالعات به‌روز در زمینه‌ی miRNA ، یافتن کنفرانس‌های مرتبط با این زمینه، فهرست مشاغل موجود، یافتن شغل و حل مشکلات مشترک در زمینه‌ی miRNA است (mirnablog.com). این پایگاه توسط تبلیغات و جوامع علمی محققین miRNA پشتیبانی می‌گردد. گزارشات پژوهشی دانشگاهی همراه با اخبار صنعت در این بلاگ یافت شده و مورد بحث و بررسی قرار می‌گیرد.

بدون شک اکثر پورتال‌های miRNA در تلاش برای وارد کردن ابزارهای بازیابی، نمایش و آنالیز داده‌های توالی‌یابی RNA در سامانه‌ی خود هستند. علی‌رغم اینکه ابعاد این تلاش‌ها بلندپروازنه بوده و نیازمند مهندسی سطح بالا همراه با پشتیبانی‌های زیستی و محاسباتی است، ولی پیش‌بینی می‌شود که ثمره‌ی حاصل از این سرمایه‌گذاری بسیار عظیم بوده و به سرعت و شدت مورد تقاضای کاربران پایین‌دستی در آینده‌ی نزدیک خواهد بود. خلاصه‌ای از کلیه‌ی سایت‌ها و منابعی که می‌توان در این زمینه یافت، در جدول ۱۳-۴ ارائه شده است.

۱۳-۷ خلاصه

در حال حاضر مجموعه‌ی بزرگی از ابزارها برای آنالیز داده‌های miRNA به راحتی در دسترس قرار گرفته و به آسانی نیز قابل استفاده هستند. این ابزارها شامل ابزارهای مدیریت خوانش‌های خام NGS و آنالیزهای پایین دستی می‌باشند. علاوه بر این، چندین پایگاه داده که مشهورترین‌شان miRBase اسـت، منبعی غنی از داده‌ها را در اختیار جوامع علمی و پژوهشی قرار می‌دهند. برخـی

جدول ۱۳-۴: منابع آنالیز داده‌های توالی‌یابی miRNA

منبع	نوع	توضیح	نشانی
miRBase	پایگاه داده	پایگاه miRNA ها، خوانش‌های نشان داده شده برای توالی‌های miRNA	mirbase.org
Rfam	پایگاه داده	پایگاه خانواده‌های RNA	rfam.sanger.ac.uk
UCSC	پایگاه داده	پایگاه داده و مرورگر ژنومی	genome.ucsc.edu/
piRNABank	پایگاه داده	پایگاه توالی‌های piwiRNA	pirnabank.ibab.ac.in/index.shtml
starBase	پایگاه داده	پایگاه توالی‌یابی CLIP و تخریبی	starbase.sysu.edu.cn
microRNA.org	پایگاه داده	پایگاه کتابخانه‌های miRNA و پروفایل‌های بیان	microRNA.org
miRò	پایگاه داده	پایگاه miRNA های مرتبط با اهدافشان و بیماری‌ها	ferrolab.dmi.unict.it/miro/index.php
miRdSNP	پایگاه داده	پایگاه miRNA ها و اهدافشان مرتبط با SNP ها و بیماری‌ها	mirdsnp.ccr.buffalo.edu
miRDeep2	آنالیز	آنالیز داده‌های توالی‌یابی RNA های کوچک: شناسایی، حاشیه‌نگاری و نمایش	mdc-berlin.de/8551903/en/research/research_teams/systems_biology_of_gene_regulatory_elements/projects/miRDeep
miRanalyzer	آنالیز	آنالیز داده‌های توالی‌یابی RNA های کوچک: شناسایی، حاشیه‌نگاری و نمایش	http://bioinfo5.ugr.es/miRanalyzer/miRanalyzer.php
TargetScan	آنالیز	ابزار یافتن هدف miRNA	targetscan.org
mir-WIP	آنالیز	ابزار یافتن هدف miRNA بر مبنای پیش‌بینی‌های بهبود یافته‌ی حاصل از داده‌های رسوب ایمنی	146.189.76.171/query.php
mirSom	آنالیز	ابزار یافتن هدف miRNA بر مبنای نقشه‌های خودسازمان‌ده	www.oppi.uef.f/bioinformatics/mirsom/
mirTarBase	آنالیز	ابزار یافتن هدف miRNA بر مبنای مقالات	miRTarBase.mbc.nctu.edu.tw
TarBase	آنالیز	ابزار یافتن هدف miRNA بر مبنای شواهد آزمایشی	microrna.gr/tarbase
DIANA-microT	آنالیز	ابزار یافتن هدف miRNA با امکان وارد نمودن miRNA و دریافت اهداف بالقوه‌ی آن	ferrolab.dmi.unict.it/miro/index.php
microRNAblog	عمومی	اخبار، کنفرانس‌ها، اعلانات، فرصت‌های شغلی	mirnablog.com

از ابزارها نیازمند آشنایی با محیط خط فرمان بوده ولی برخی از ابزارها که تعدادشان هم رو به افزایش است، مبتنی بر وب هستند. به طور کلی این ابزارها مجموعه‌ای قدرتمند از پیاده‌سازی‌ها را فراهم می‌آورند که به کاربر امکان شناسایی، حاشیه‌نگاری، مصورسازی و یافتن تفاوت‌ها در مجموعه‌ی داده‌های miRNA را می‌دهند. چون ابزارهای مورد استفاده جهت آنالیز دسته‌های غیر از miRNA ها دچار عقب‌ماندگی و تاخیر هستند، لذا در آینده‌ی نزدیک بیشتر تلاش‌ها معطوف به این بخش خواهد شد. در حال حاضر آنالیز توالی miRNA ها و RNA های نارمزگر کوچک از نظر عملی به خوبی امکان‌پذیر است.

منابع

1. Li Y., Zhang Z., Liu F. et al. Performance comparison and evaluation of sofware tools for microRNA deep-sequencing data analysis. Nucleic Acids Research 40(10):4298–4305, 2012.
2. Williamson V., Kim A., Xie B. et al. Detecting miRNAs in deep-sequencing data: A sofware performance comparison and evaluation. Briefngs Bioinformatics 14(1):36–45, 2013.
3. Friedländer M.R., Mackowiak S.D., Li N., Chen W. et al. miRDeep2 accurately identifes known and hundreds of novel microRNA genes in seven animal clades. Nucleic Acids Research 40(1):37–52, 2012.
4. Hackenberg M., Sturm M., Langenberger D. et al. miRanalyzer: A microRNA detection and analysis tool for next-generation sequencing experiments. Nucleic Acids Research 37(Web Server issue):W68–W76, 2009.
5. Lewis B.P., Burge C.B., and Bartel D.P. Conserved seed pairing, ofen flanked by adenosines, indicates that thousands of human genes are microRNA targets. Cell 120(1):15–20, 2005.
6. Maragkakis M., Reczko M., Simossis V.A. et al. DIANA-microT web server: Elucidating microRNA functions through target prediction. Nucleic Acids Research 37(Web Server issue):W273–W276, 2009.
7. Kim S.K., Nam J.W., Rhee J.K. et al. miTarget: microRNA target gene prediction using a support vector machine. BMC Bioinformatics 7:411, 2006.
8. Wang X. and El Naqa I.M. Prediction of both conserved and nonconserved microRNA targets in animals. Bioinformatics 24(3):325–332, 2008.
9. Wang X. miRDB: A microRNA target prediction and functional annotation database with a wiki interface. RNA 14(6):1012–1017, 2008.
10. Heikkinen L., Kolehmainen M., and Wong G. Prediction of microRNA targets in Caenorhabditis elegans using a self-organizing map. Bioinformatics 27(9):1247–1254, 2011.
11. Zhang L., Ding L., Cheung T.H. et al. Systematic identifcation of C. elegans miRISC proteins, miRNAs, and mRNA targets by their interactions with GW182 proteins AIN-1 and AIN-2. Molecular Cell 28(4):598–613, 2007.

12. Hammell M., Long D., Zhang L. et al. mirWIP: MicroRNA target prediction based on microRNA-containing ribonucleoprotein-enriched transcripts. Nature Methods 5(9):813–819, 2008.
13. Vergoulis T., Vlachos I.S., Alexiou P. et al. TarBase 6.0: Capturing the exponential growth of miRNA targets with experimental support. Nucleic Acids Research 40(Database issue):D222–D229, 2012.
14. Hsu S.D., Lin F.M., Wu W.Y. et al. miRTarBase: A database curates experimentally validated microRNA-target interactions. Nucleic Acids Research 39(Database issue):D163–D169, 2011.
15. Mestdagh P., Lefever S., Pattyn F. et al. Te microRNA body map: Dissecting microRNA function through integrative genomics. Nucleic Acids Research 39(20):e136, 2011.
16. Yan Z., Shah P.K., Amin S.B. et al. Integrative analysis of gene and miRNA expression profles with transcription factor-miRNA feed-forward loops identifes regulators in human cancers. Nucleic Acids Research 40(17):e135, 2012.
17. Grifths-Jones S., Saini H.K., van Dongen S. et al. miRBase: Tools for microRNA genomics. Nucleic Acids Research 36(Database issue):D154–D158, 2008.
18. Kozomara A. and Grifths-Jones S. miRBase: Integrating microRNA annotation and deep-sequencing data. Nucleic Acids Research 39(Database Issue):D152–D157, 2011.
19. Kozomara A. and Grifths-Jones S. miRBase: Annotating high confdence microRNAs using deep sequencing data. Nucleic Acids Research 42(1):D68–D73, 2014.
20. Betel D., Wilson M., Gabow A. et al. Te microRNA.org resource: Targets and expression. Nucleic Acids Research 36(Database issue):D149–D153, 2008.
21. Lakshmi S. and Agrawal S. piRNABank: A web resource on classifed and clustered Piwi-interacting RNAs. Nucleic Acids Research 36(Database issue):D173–D177, 2008.
22. Burge S.W., Daub J., Eberhardt R. et al. Rfam 11.0: 10 years of RNA families. Nucleic Acids Research 41(Database issue):D226–D232, 2013.
23. Cho S., Jang I., Jun Y., Yoon et al. MiRGator v3.0: A microRNA portal for deep sequencing, expression profling and mRNA targeting. Nucleic Acids Research 41(Database issue):D252–D257, 2013.
24. Yang J.H., Li J.H., Shao P. et al. starBase: A database for exploring microRNA–mRNA interaction maps from Argonaute CLIP-seq and Degradome-seq data. Nucleic Acids Research 39(Database issue):D202–D209, 2011.
25. Li J.H., Liu S., Zhou H. et al. starBase v2.0: Decoding miRNA–ceRNA, miRNA–ncRNA and protein–RNA interaction networks from large-scale CLIP-seq data. Nucleic Acids Research 42(1):D92–D97, 2014.
26. Laganà A., Forte S., Giudice A. et al. miRò: A miRNA knowledge base. Database (Oxford):bap008 2009.

27. Bruno A.E., Li L., Kalabus J.L. et al. miRdSNP: A database of disease-associated SNPs and microRNA target sites on 3′UTRs of human genes. BMC Genomics 13:44, 2012.
28. Bateman A., Agrawal S., Birney E. et al. RNAcentral: A vision for an international database of RNA sequences. RNA 17(11):1941–1946, 2011.

واژه‌نامه (انگلیسی به فارسی)

A

ab initio	از آغاز
Accession number	شماره‌ی دسترسی
Accessor function	تابع دسترسی
Adaptor	آداپتور (سازگارساز)
Additive color format	فرمت رنگ افزایشی
Adenylate	آدنیله
Aggregate Pct	Pct تجمعی
Alignment	همردیفی
Alternative model	مدل جایگزین
Amplicon	آمپلیکون
Annotation	حاشیه‌نگاری
Antisense	پادمعنا
Arc	قوس
Argonaute	آرگونات
Argument	برهان
Artifact	ورساخته
Assembly	اسمبل
Attribute	ویژگی

B

Bar plot	نمودار میله‌ای
Barcoding	بارکُد زدن
Base graphics	گرافیک پایه
Batch	دسته‌ای
Batch effect	اثر دسته
Bead	مُهره

Benjamini and Hochberg's false discovery rate (BH)	نرخ غلط‌یابی بنجامینی و هوکبرگ (BH)
Bin	بِن، قطعه
Biogenesis	زیست‌زایی
Biostring	بایواسترینگ
Biotype	بیوتیپ
Biplot	نمودار دوتایی
Bitmap	بیتمپ
Bonus	پاداش
Box plot	نمودار ستونی
Bubble	حباب
Bundle	دسته
Burrows-Wheeler transform	تبدیل باروز - ویلر

C

Capillary electrophoresis	مویین الکتروفورز
Capture-seq	توالی‌یابی تسخیری
Case-control study	مطالعه‌ی مورد - شاهد
Cell line	رده‌ی سلولی
Chamber	اتاقک
Chargecoupled camera	دوربین مجهز به شارژ
cis-eQTL	eQTL سیس
Classifier	مرتب کننده
Clip	گیرایی
Cluster generation	تشکیل خوشه
Coding	رمزگر
Command	دستور
Command line	خط فرمان
Competitive Endogenous RNA (ceRNA)	RNA درون‌زاد رقابتی (ceRNA)
Competitive test	آزمون رقابتی

Compile	همگردانی
Comprehensive R Archive Network (CRAN)	شبکه آرشیو جامع R (CRAN)
Concordant	منطبق
Conditional testing	آزمون شرطی
Contig	کانتیگ
Contrast matrix	ماتریس مقایسه
Converge	همگرایی
Copy number	تعداد کُپی
Count	شمارش
Coverage	پوشش
Cross-linked immunoprecipitation sequencing (CLIP-seq)	توالی‌یابی رسوب ایمنی هَمبَر (توالی‌یابی CLIP)
Curated database	پایگاه داده‌ی دستیار
Cytoband	نوار کروموزومی

D

Data frame	چارچوب داده‌ها
de Bruijn graph	نمودار دوبران
de novo	از نو
de novo assembly	اسمبل کردن از نو
Deconvolution	کاهش همتابی
Degradome-seq	توالی‌یابی تخریبی
Delimiter	جدا کننده
Design matrix	ماتریس طرح
Dicer	دایسر
Differential expression	بیان متفاوت
Differential Expression Analysis (DE)	آنالیز افتراقی بیان (DE)
Dots Per Inch (DPI)	نقطه‌های موجود در هر اینچ (DPI)
Downregulated	کاهش بیان
Downstream analysis	آنالیز پایین دستی

Drosha	دروشا
Duplicate	مضاعف‌شدگی
Dysregulate	اختلال در تنظیم

E

Edge	مرز، لبه
Effective length	طول موثر
Encoding	رمزگذاری
Endogenous silencing RNA (endo-siRNA)	RNA سرکوبگر درون‌زاد (endo-siRNA)
End-repair	ترمیم انتها
Enhancer RNA (eRNA)	RNA تقویت کننده (eRNA)
Entropy	انتروپی
Entry	مدخل
Evidence code	کُد مستند
Exogenous silencing RNA (endo-siRNA)	RNA سرکوبگر برون‌زاد (exo-siRNA)
Exome sequencing	توالی‌یابی اگزوم
Expectation Maximum (EM)	حداکثرسازی امید ریاضی
Exploratory plot	نمودار شناسایی
Expressed Sequence Tag (EST)	نشانه‌ی توالی بیان شده (EST)

F

Feature	ترکیب، ویژگی
Feed-forward loop	حلقه بازخورد
Filter	پاکسازی
Fisher’s exact test	آزمون دقیق فیشر
Flag	نشانه، نما
Flanking	مجاور
Flatten	مسطح شدن
Flow cell	سلول جریان
Flow chip	تراشه‌ی جریان

Fold	فولد
Fuorescent absorbance	جذب فلورسنت
Fusion break point	نقطه‌ی انفصال تلفیقی
Fusion genes	ژن‌های تلفیقی

G

Gamma-Poisson distribution	توزیع گاما - پواسون
Gap	شکاف
Gene enrichment analysis	آنالیز غنی‌سازی ژن
Gene Ontology (GO)	هستی‌شناسی ژن (GO)
Gene Set Analysis (GSA)	آنالیز مجموعه‌ی ژن (GSA)
Gene set enrichment analysis	آنالیز غنی‌سازی مجموعه‌ی ژنی
Generalized Linear Model (GLM)	مدل خطی تعمیم یافته (GLM)
Gene-sampling model	مدل نمونه‌گیری ژنی
Genome viewer	بیننده‌ی ژنومی
Genome-wide	گستره‌ی ژنومی
Genome-Wide Association Study (GWAS)	مطالعه پیوستگی گستره‌ی ژنوم (GWAS)
Global run-on sequencing (GRO-seq)	توالی‌یابی بی‌وقفه‌ی کُلی (توالی‌یابی GRO)
Granges	جی‌رنجز
Grid graphics	گرافیک شبکه‌ای
Guide molecule	مولکول راهنما

H

Hairpin	ساختار سنجاق‌سری
Hard-coded	اختصاصی
Hash table	جدول هَش
Header	سرتیتر
Heatmap	نقشه‌ی حرارتی
Heteronuclear RNA	RNA هسته‌ای ناهمگون
Hidden Markov Model (HMM)	مدل مارکوف مخفی (HMM)

Hierarchical	آشیانه‌ای
High-throughput	پُربروندا‌د، کارآمد
High-throughput sequencing of RNA isolated by crosslinking immunoprecipitation (HITS-CLIP)	توالی‌یابی پُربروندا‌د (کارآمد) RNA جداسازی شده توسط رسوب ایمنی هَمبَر (HITS-CLIP)
Histogram	هیستوگرام (بافت‌نگار)
Hit	توفیق
Homogeneous	همگن
Hybrid	هیبرید، آمیخته

I

Identifier	شناساگر
Immunoprecipitation	رسوب ایمنی (ایمیونوپرسیپیتاسیون)
Individual-Nucleotide Resolution UV Crosslinking and Immunoprecipitation (iCLIPSeq)	وضوح تک نوکلئوتیدی در همبرسازی UV و رسوب ایمنی (توالی‌یابی iCLIP)
in situ	در محل
Indel	ایندل
Index	نمایه
Indicator variable	متغیر شاخص
Insert	الحاق
Intensity-dependent manner	روش وابستگی شدید
Interaction	اثر متقابل، برهمکنش
Interface	رابط کاربری
Intergenic region	مناطق بین ژنی
Interleave	میان جادهی
Isochore	ایزوکور
Isolation	جداسازی (استخراج)
IsomiR	ایزومیر

J

Java	جاوا

Joint Photographic Experts Group (JPEG)	گروه مشترک خبرگان عکاسی (JPEG)
Junction	اتصال

K

Kyoto Encyclopedia of Genes and Genomes (KEGG)
دائره‌المعارف کیوتو در زمینه‌ی ژن‌ها و ژنوم‌ها (KEGG)

L

Label	برچسب
Lariat debranching enzyme	آنزیم شاخه‌زدای لریِت
Legend	علامت
Ligate	متصل کردن
Ligation	اتصال
Linux	لینوکس
Load	بارگیری
Log	لاگ
Long non-coding RNA	RNA نارمزگر بلند
Low-compexity sequence	توالی با پیچیدگی پایین
Low confidence	با اطمینان پایین

M

Mappable length	طول قابل مکان‌یابی
Mapping-based assembly	اسمبل کردن مبتنی بر مکان‌یابی
Maximum mappable length	حداکثر طول قابل مکان‌یابی
Measurement	سنجه
Metadata	فراداده
Method	روش
Methylation	متیلاسیون
Microarray	ریزآرایه
Microbiome	میکروبیوم
Microfluidics	میکروفلوییدی

Microprocessor complex	کمپلکس ریزپردازنده
MicroRNA off-set RNA (moRNA)	RNA تعدیل کننده‌ی RNA کوچک (moRNA)
Mirtron	میرترون
Mismatch	عدم تطابق
Missing sequence	توالی گمشده
Mode	وضعیت
Moor's law	قانون مور
mRNA-decapping enzyme 1/2 (DCP1/2)	آنزیم کلاهک‌بردار mRNA 1/2 (DCP1/2)
Multiplex	چند عضوی
Multithreated	چند مسیری

N

Nagative binomial distribution	توزیع دوجمله‌ای منفی
Nanodrop	نانودراپ
Nanopore technologies	فناوری‌های نانوپور
Nascent	نوآیند
Next Generation Sequencing (NGS)	توالی‌یابی نسل جدید (NGS)
Nimblegen	نیمبلِگِن
Node	گره
Noise	اختلال
Nonnegative Matrix Factorization (NMF)	تجزیه‌ی ماتریس نامنفی (NMF)
Normalize	نرمال کردن
Nova	نووا
Nuclear run-on	تداوم هسته‌ای
Null model	مدل صفر

O

Object-Oriented Programming (OOP)	برنامه‌نویسی شیءگرا (OOP)
Ontological analysis	آنالیز هستی‌شناختی
Open source	متن باز

Operator	عملگر
Overamplification	فزون‌تکثیری
Overrepresentation	فزون‌نمایی
Overcorrection	فوق تصحیح
Overdispersed	فوق پراکنده
Over-expressed	فرابیان، فزون‌بیان
Overhang	پیش‌آمدگی
Overlap-Layout-Consensus (OLC)	همپوشانی - طرح‌بندی - اجماع (OLC)

P

Paired-end read mode	وضعیت خوانش جفت انتهایی (هر دو انتها)
Pairwise overlap	همپوشان جفتی
Palindrome approach	روش پالیندروم
Parallel analysis of RNA ends (PARE)	آنالیز موازی انتهاهای RNA (PARE)
Passenger strand	زنجیره‌ی مسافر
Pathway analysis	آنالیز مسیر
PCR-specific gene	ژن اختصاصی PCR
Peak	اوج
Penalty	تاوان
Perl	پرل
Phenodata	فنودیتا
Photoactivatable ribonucleoside-enhanced CLIP (PAR-CLIPSeq)	توالی‌یابی CLIP ریبونکلئوزید قابل فعال‌سازی نوری تقویت شده (توالی‌یابی PAR-CLIP)
Picotiter plate	پلیت پیکوتیتر
Ping-pong amplifcation cycle	چرخه‌ی تکثیر پینگ‌پنگ
Pipeline	مسیر
Pipetting	پایپت کردن
Pipettor	پایپتور
Piwi-associated RNA (piRNA)	RNA مرتبط با پیوی (piRNA)
Pixel	پیکسل

Platform	پلتفرم
Points Per Inch (PPI)	تعداد نقاط در هر اینچ (PPI)
Poison-Tweedie	پواسون - توییدی
Poisson distribution	توزیع پواسون
Polyacrylamide gel electrophoresis	الکتروفورز روی ژل پُلی‌اکریلامید
Polymerase Chain Reaction (PCR)	واکنش زنجیره‌ای پلیمراز (PCR)
Pool	تلفیق کردن
Portable Document Format (PDF)	فرمت اسناد ترابرپذیر (PDF)
Portable Network Graphics (PNG)	گرافیک‌های شبکه‌ای ترابرپذیر (PNG)
PostScript	پُست اسکریپت
pre-miRNA	miRNA پیش‌ساز
Preprocessing	پیش‌پردازش
Primer	آغازگر
pri-miRNA	miRNA اولیه
Principal Component Analysis (PCA)	آنالیز مولفه‌های اصلی (PCA)
Probe	کاوشگر
Promoter-associated small RNA	RNA کوچک مرتبط با پروموتور
Prompt	پرومپت
Protocol	دستورالعمل
Proximal	مبداء
Pseudogene	شبه ژن
Pseudouridylation	سودویوریدیلاسیون
p-value	مقدار p
Pyrosequencing	پیروسکوئنسینگ
Python	پایتون

Q

Quality control	کنترل کیفیت
Quantifying	کمی‌سازی
Quibitfluorometer	کیبِت فلورومتر

R

Read-through	خوانش سراسری
Real-time	به‌هنگام
Reconstruction	بازسازی
Reference genome	ژنوم مرجع
Reset	بازنشانی
Retrieve	بازیابی
Return	بازگشت
Reverse transcriptase	رونوشت‌بردار معکوس
Reverse transcription	رونویسی معکوس
RNA-binding protein immunoprecipitation sequencing (RIP-seq)	توالی‌یابی رسوب ایمنی پروتئین متصل شونده به RNA (توالی‌یابی RIP)
RNA-Seq	توالی‌یابی RNA
Run	اجرا
Running sum method	روش مجموع اجرا

S

Sanger dideoxy sequencing	توالی‌یابی دی‌دئوکسی سَنگِر
Scatter plot	نمودار پراکنش
Self-contained test	آزمون جامع
Self-Organizing Map (SOM)	نقشه‌ی خود سازمان‌ده (SOM)
Sense	بامعنا
Sequence range	دامنه‌ی توالی
Sequencing by synthesis	توالی‌یابی مبتنی بر سنتز
Sharp	تیز
Short seed sequence	توالی کوتاه سید
Shot noise	تداخل تلاش
Single Nucleotide Polymorphism (SNP)	چندشکلی تک نوکلئوتیدی (SNP)
Single read mode	وضعیت خوانش تکی
Size-selected/fractionated small RNA	RNA کوچک بخش‌بندی شده/انتخاب شده بر مبنای اندازه

Sizing	اندازه‌بندی
Slicer	برش دهنده (اسلایسر)
Sliding window approach	روش پنجره‌ی لغزان
Small noncoding RNA	RNA نارمزگر کوچک
Small nucleolar RNAs (snoRNA)	RNA هستکی کوچک (snoRNA)
Small nuclear RNA (snRNA)	RNA هسته‌ای کوچک (snRNA)
Small RNA	RNA کوچک
Smear	اسمیر
snoRNA-derived RNA (sdRNA)	snoRNA مشتق از RNA (sdRNA)
Solid-Phase Reversible Immobilization (SPRI) bead	مُهره‌ی تثبیت کننده برگشت‌پذیر فاز جامد (SPRI)
Space separated	جدا کننده‌ی فاصله
Splice junction	اتصال پیرایشی
Spliced aligner	همردیف‌ساز پیرایش شده
Spliceosome	اسپلیسوزوم
Splicing	پیرایش
Splite-read	خوانش خُرد شده
Stack size	اندازه‌ی برآمدگی
Statistical power	توان آماری
Stream	جریان
Strictness	سخت‌گیری
Subject-sampling model	مدل نمونه‌گیری موضوعی
Subtractive color format	فرمت رنگ کاهشی
Support Vector Machine (SVM)	ماشین بردار پشتیبان (SVM)

T

Tab separated	جدا کننده‌ی تَب
Tag	نشان
Tagged Image File Format (TIFF)	فرمت فایل تصویر نشانمند (TIFF)
Target read length	طول خوانش هدف

Thread	جزء
Tip	نوک
Total Context Score	امتیاز زمینه‌ی کُلی
Trade off	تبادل
Training set	مجموعه‌ی آموزش
Transcript	رونوشت
Transcription Start Site (TSS)	محل آغاز رونویسی (TSS)
Transcription start site-associated RNA (TSSa-RNA)	RNA مرتبط با محل آغاز رونویسی (TSSa-RNA)
Transcriptional interference	تزاحم رونویسی
Transcriptomics	ترانسکریپتومیک
Transduction	ترازهش
trans-eQTL	eQTL ترانس
Transfection	تراآلایی
Translocation	جابجایی
Transposable element	عنصر قابل انتقال
Trimme	پیرایش
tRNA-derived RNA fragment (tRF)	tRNA های مشتق از قطعات RNA (tRF)
tRNA-derived small RNA (tsRNA)	tRNA های مشتق از RNA های کوچک (tsRNA)

U

Underestimate	زیربرآورد
Underpresentation	فرونمایی
Uniqueness	یگانگی
Univariate	تک متغیری
Universal	عمومی
Unix	یونیکس
Untranslated Region (UTR)	منطقه‌ی ناترجمان (UTR)
Upregulated	افزایش بیان

V

Validate	اعتبارسنجی
Vector	بردار
Vector graphics	وکتور گرافیک
Visualization	مصورسازی
Visualizing	آشکارسازی
Viterbi algorithm	الگوریتم ویتربی
Volcano plot	نمودار آتشفشانی

W

Windows bitmap (BMP)	بیتمپ ویندوز (BMP)
Workflow	گردش کار

Z

Zero inflation	تورم صفر
Zeromode waveguide (ZMW)	موجبَر حالت صفر (ZMW)

واژه‌نامه (فارسی به انگلیسی)

الف

اتاقک	Chamber
اتصال	Junction
اتصال	Ligation
اتصال پیرایشی	Splice junction
اثر دسته	Batch effect
اثر متقابل، برهمکنش	Interaction
اجرا	Run
اختصاصی	Hard-coded
اختلال	Noise
اختلال در تنظیم	Dysregulate
آداپتور (سازگارساز)	Adaptor
آدنیله	Adenylate
آرگونات	Argonaute
از آغاز	*ab initio*
آزمون جامع	Self-contained test
آزمون دقیق فیشر	Fisher’s exact test
آزمون رقابتی	Competitive test
آزمون شرطی	Conditional testing
از نو	*de novo*
اسپلیسوزوم	Spliceosome
اسمبل	Assembly
اسمبل کردن از نو	*de novo* assembly
اسمبل کردن مبتنی بر مکان‌یابی	Mapping-based assembly
اسمیر	Smear
آشکارسازی	Visualizing
آشیانه‌ای	Hierarchical

اعتبارسنجی	Validate
آغازگر	Primer
افزایش بیان	Upregulated
eQTL سیس	cis-eQTL
eQTL ترانس	trans-eQTL
الحاق	Insert
الکتروفورز روی ژل پُلی‌اکریلامید	Polyacrylamide gel electrophoresis
الگوریتم ویتربی	Viterbi algorithm
آمپلیکون	Amplicon
امتیاز زمینه‌ی کُلی	Total Context Score
آنالیز افتراقی بیان (DE)	Differential Expression Analysis (DE)
آنالیز پایین دستی	Downstream analysis
آنالیز غنی‌سازی ژن	Gene enrichment analysis
آنالیز غنی‌سازی مجموعه‌ی ژنی	Gene set enrichment analysis
آنالیز مجموعه‌ی ژن (GSA)	Gene Set Analysis (GSA)
آنالیز مسیر	Pathway analysis
آنالیز موازی انتهاهای RNA (PARE)	Parallel analysis of RNA ends (PARE)
آنالیز مولفه‌های اصلی (PCA)	Principal Component Analysis (PCA)
آنالیز هستی‌شناختی	Ontological analysis
انتروپی	Entropy
اندازه‌ی برآمدگی	Stack size
اندازه‌بندی	Sizing
آنزیم شاخه‌زدای لریِت	Lariat debranching enzyme
آنزیم کلاهک‌بردار mRNA 1/2 (DCP1/2)	mRNA-decapping enzyme 1/2 (DCP1/2)
اوج	Peak
ایزوکور	Isochore
ایزومیر	IsomiR
ایندل	Indel

ب

با اطمینان پایین	Low confidence
بارکُد زدن	Barcoding
بارگیری	Load
بازسازی	Reconstruction
بازگشت	Return
بازنشانی	Reset
بازیابی	Retrieve
بامعنا	Sense
بایواسترینگ	Biostring
برچسب	Label
بردار	Vector
برش دهنده (اسلایسر)	Slicer
برنامه‌نویسی شیءگرا (OOP)	Object-Oriented Programming (OOP)
برهان	Argument
بِن، قطعه	Bin
به‌هنگام	Real-time
بیان متفاوت	Differential expression
بیتمپ	Bitmap
بیتمپ ویندوز (BMP)	Windows bitmap (BMP)
بیننده‌ی ژنومی	Genome viewer
بیوتیپ	Biotype

پ

پاداش	Bonus
پادمعنا	Antisense
پاکسازی	Filter
پایپت کردن	Pipetting
پایپتور	Pipettor

پایتون	Python
پایگاه داده‌ی دستیار	Curated database
پُربروندداد، کارآمد	High-throughput
پرل	Perl
پرومپت	Prompt
پُست اسکریپت	PostScript
Pct تجمعی	Aggregate Pct
پلتفرم	Platform
پلیت پیکوتیتر	Picotiter plate
پواسون - توییدی	Poison-Tweedie
پوشش	Coverage
پیرایش	Splicing
پیرایش	Trimme
پیروسکوئنسینگ	Pyrosequencing
پیش‌آمدگی	Overhang
پیش‌پردازش	Preprocessing
پیکسل	Pixel

ت

تابع دسترسی	Accessor function
تاوان	Penalty
تبادل	Trade off
تبدیل باروز - ویلر	Burrows-Wheeler transform
تجزیه‌ی ماتریس نامنفی (NMF)	Nonnegative Matrix Factorization (NMF)
تداخل تلاش	Shot noise
تداوم هسته‌ای	Nuclear run-on
تراآلایی	Transfection
ترازهش	Transduction
تراشه‌ی جریان	Flow chip

ترانسکریپتومیک	Transcriptomics
ترکیب، ویژگی	Feature
ترمیم انتها	End-repair
tRNA های مشتق از RNA های کوچک (tsRNA)	tRNA-derived small RNA (tsRNA)
tRNA های مشتق از قطعات RNA (tRF)	tRNA-derived RNA fragment (tRF)
تزاحم رونویسی	Transcriptional interference
تشکیل خوشه	Cluster generation
تعداد کُپی	Copy number
تعداد نقاط در هر اینچ (PPI)	Points Per Inch (PPI)
تک متغیری	Univariate
تلفیق کردن	Pool
توالی با پیچیدگی پایین	Low-compexity sequence
توالی گمشده	Missing sequence
توالی‌یابی اگزوم	Exome sequencing
توالی‌یابی بی‌وقفه‌ی کُلی (توالی‌یابی GRO)	Global run-on sequencing (GRO-seq)
توالی‌یابی تخریبی	Degradome-seq
توالی‌یابی تسخیری	Capture-seq
توالی‌یابی دی‌دئوکسی سَنگِر	Sanger dideoxy sequencing
توالی‌یابی RNA	RNA-Seq
توالی‌یابی رسوب ایمنی پروتئین متصل شونده به RNA (توالی‌یابی RIP)	RNA-binding protein immunoprecipitation sequencing (RIP-seq)
توالی‌یابی رسوب ایمنی هَمبَر (توالی‌یابی CLIP)	Cross-linked immunoprecipitation sequencing (CLIP-seq)
توالی‌یابی پُربروندا (کارآمد) RNA جداسازی شده توسط رسوب ایمنی هَمبَر (HITS-CLIP)	High-throughput sequencing of RNA isolated by crosslinking immunoprecipitation (HITS-CLIP)
توالی‌یابی CLIP ریبونکلئوزید قابل فعال‌سازی نوری تقویت شده (توالی‌یابی PAR-CLIP)	Photoactivatable ribonucleoside-enhanced CLIP (PAR-CLIPSeq)
توالی کوتاه سید	Short seed sequence
توالی‌یابی مبتنی بر سنتز	Sequencing by synthesis
توالی‌یابی نسل جدید (NGS)	Next Generation Sequencing (NGS)

توان آماری	Statistical power
تورم صفر	Zero inflation
توزیع پواسون	Poisson distribution
توزیع دوجمله‌ای منفی	Nagative binomial distribution
توزیع گاما - پواسون	Gamma-Poisson distribution
توفیق	Hit
تیز	Sharp

ج

جابجایی	Translocation
جاوا	Java
جداسازی (استخراج)	Isolation
جدا کننده	Delimiter
جدا کننده‌ی تَب	Tab separated
جدا کننده‌ی فاصله	Space separated
جدول هَش	Hash table
جذب فلورسنت	Fuorescent absorbance
جریان	Stream
جزء	Thread

چ

چارچوب داده‌ها	Data frame
چرخه‌ی تکثیر پینگ‌پنگ	Ping-pong amplifcation cycle
چندشکلی تک نوکلئوتیدی (SNP)	Single Nucleotide Polymorphism (SNP)
چند عضوی	Multiplex
چند مسیری	Multithreated

ح

حاشیه‌نگاری	Annotation

حباب	Bubble
حداکثرسازی امید ریاضی	Expectation Maximum (EM)
حداکثر طول قابل مکان‌یابی	Maximum mappable length
حلقه بازخورد	Feed-forward loop

خ

خوانش خُرد شده	Splite-read
خوانش سراسری	Read-through
خط فرمان	Command line

د

دامنه‌ی توالی	Sequence range
دایسر	Dicer
دائره‌المعارف کیوتو در زمینه‌ی ژن‌ها و ژنوم‌ها (KEGG)	Kyoto Encyclopedia of Genes and Genomes (KEGG)
در محل	*in situ*
دروشا	Drosha
دستور	Command
دستورالعمل	Protocol
دسته‌ای	Batch
دسته	Bundle
دوربین مجهز به شارژ	Chargecoupled camera

ر

رابط کاربری	Interface
رده‌ی سلولی	Cell line
رسوب ایمنی (ایمیونوپرسیپیتاسیون)	Immunoprecipitation
رمزگذاری	Encoding
رمزگر	Coding
RNA تعدیل کننده‌ی RNA کوچک (moRNA)	MicroRNA off-set RNA (moRNA)

RNA تقویت کننده (eRNA)	Enhancer RNA (eRNA)
RNA درون‌زاد رقابتی (ceRNA)	Competitive Endogenous RNA (ceRNA)
RNA سرکوبگر برون‌زاد (exo-siRNA)	Exogenous silencing RNA (endo-siRNA)
RNA سرکوبگر درون‌زاد (endo-siRNA)	Endogenous silencing RNA (endo-siRNA)
RNA کوچک	Small RNA
RNA کوچک بخش‌بندی شده/انتخاب شده بر مبنای اندازه	Size-selected/fractionated small RNA
RNA کوچک مرتبط با پروموتور	Promoter-associated small RNA
RNA مرتبط با پیوی (piRNA)	Piwi-associated RNA (piRNA)
RNA مرتبط با محل آغاز رونویسی (TSSa-RNA)	Transcription start site-associated RNA (TSSa-RNA)
RNA نارمزگر بلند	Long non-coding RNA
RNA نارمزگر کوچک	Small noncoding RNA
RNA هستکی کوچک (snoRNA)	Small nucleolar RNAs (snoRNA)
RNA هسته‌ای کوچک (snRNA)	Small nuclear/nucleolar RNA (snRNA)
RNA هسته‌ای ناهمگون	Heteronuclear RNA
روش	Method
روش پالیندروم	Palindrome approach
روش پنجره‌ی لغزان	Sliding window approach
روش مجموع اجرا	Running sum method
روش وابستگی شدید	Intensity-dependent manner
رونوشت	Transcript
رونوشت‌بردار معکوس	Reverse transcriptase
رونویسی معکوس	Reverse transcription
ریزآرایه	Microarray

ز

زنجیره‌ی مسافر	Passenger strand
زیربرآورد	Underestimate
زیست‌زایی	Biogenesis

ژ

ژن اختصاصی PCR	PCR-specific gene
ژنوم مرجع	Reference genome
ژن‌های تلفیقی	Fusion genes

س

ساختار سنجاق‌سری	Hairpin
سخت‌گیری	Strictness
سرتیتر	Header
سلول جریان	Flow cell
سنجه	Measurement
snoRNA مشتق از RNA (sdRNA)	snoRNA-derived RNA (sdRNA)
سودویوریدیلاسیون	Pseudouridylation

ش

شبکه آرشیو جامع R (CRAN)	Comprehensive R Archive Network (CRAN)
شبه ژن	Pseudogene
شکاف	Gap
شمارش	Count
شماره‌ی دسترسی	Accession number
شناساگر	Identifier

ط

طول خوانش هدف	Target read length
طول قابل مکان‌یابی	Mappable length
طول موثر	Effective length

ع

عدم تطابق	Mismatch

علامت	Legend
عملگر	Operator
عمومی	Universal
عنصر قابل انتقال	Transposable element

ف

فرابیان، فزون‌بیان	Over-expressed
فراداده	Metadata
فرمت اسناد ترابرپذیر (PDF)	Portable Document Format (PDF)
فرمت رنگ افزایشی	Additive color format
فرمت رنگ کاهشی	Subtractive color format
فرمت فایل تصویر نشانمند (TIFF)	Tagged Image File Format (TIFF)
فرونمایی	Underpresentation
فزون‌تکثیری	Overamplification
فزون‌نمایی	Overrepresentation
فناوری‌های نانوپور	Nanopore technologies
فنودیتا	Phenodata
فوق پراکنده	Overdispersed
فوق تصحیح	Overcorrection
فولد	Fold

ق

قانون مور	Moor's law
قوس	Arc

ک

کانتیگ	Contig
کاوشگر	Probe
کاهش بیان	Downregulated

کاهش همتابی	Deconvolution
کُد مستند	Evidence code
کمپلکس ریزپردازنده	Microprocessor complex
کمی‌سازی	Quantifying
کنترل کیفیت	Quality control
کیبِت فلورومتر	Quibitfluorometer

گ

گرافیک پایه	Base graphics
گرافیک شبکه‌ای	Grid graphics
گرافیک‌های شبکه‌ای ترابرپذیر (PNG)	Portable Network Graphics (PNG)
گردش کار	Workflow
جی‌رنجز	Granges
گروه مشترک خبرگان عکاسی (JPEG)	Joint Photographic Experts Group (JPEG)
گره	Node
گستره‌ی ژنومی	Genome-wide
گیرایی	Clip

ل

لاگ	Log
لینوکس	Linux

م

ماتریس طرح	Design matrix
ماتریس مقایسه	Contrast matrix
ماشین بردار پشتیبان (SVM)	Support Vector Machine (SVM)
مبداء	Proximal
متصل کردن	Ligate
متغیر شاخص	Indicator variable

متن باز	Open source
متیلاسیون	Methylation
مجاور	Flanking
مجموعه‌ی آموزش	Training set
محل آغاز رونویسی (TSS)	Transcription Start Site (TSS)
مدخل	Entry
مدل جایگزین	Alternative model
مدل خطی تعمیم یافته (GLM)	Generalized Linear Model (GLM)
مدل صفر	Null model
مدل مارکوف مخفی (HMM)	Hidden Markov Model (HMM)
مدل نمونه‌گیری ژنی	Gene-sampling model
مدل نمونه‌گیری موضوعی	Subject-sampling model
مرتب کننده	Classifier
مرز، لبه	Edge
مسطح شدن	Flatten
مسیر	Pipeline
مصورسازی	Visualization
مضاعف‌شدگی	Duplicate
مطالعه‌ی پیوستگی گستره‌ی ژنوم (GWAS)	Genome-Wide Association Study (GWAS)
مطالعه‌ی مورد - شاهد	Case-control study
مقدار p	p-value
مناطق بین ژنی	Intergenic region
منطبق	Concordant
منطقه‌ی ناترجمان (UTR)	Untranslated Region (UTR)
موج‌بَر حالت صفر (ZMW)	Zeromode waveguide (ZMW)
مولکول راهنما	Guide molecule
مویین الکتروفورز	Capillary electrophoresis
مُهره	Bead
مُهره‌ی تثبیت کننده برگشت‌پذیر فاز جامد (SPRI)	Solid-Phase Reversible Immobilization (SPRI) bead

میان جادهی	Interleave
میرترون	Mirtron
miRNA اولیه	pri-miRNA
miRNA پیش‌ساز	pre-miRNA
میکروبیوم	Microbiome
میکروفلوییدی	Microfluidics

ن

نانودراپ	Nanodrop
نرخ غلط‌یابی بنجامینی و هوکبرگ (BH)	Benjamini and Hochberg's false discovery rate (BH)
نرمال کردن	Normalize
نقشه‌ی حرارتی	Heatmap
نقطه‌ی انفصال تلفیقی	Fusion break point
نقطه‌های موجود در هر اینچ (DPI)	Dots Per Inch (DPI)
نمودار آتشفشانی	Volcano plot
نمودار پراکنش	Scatter plot
نمودار دوبران	de Bruijn graph
نمودار دوتایی	Biplot
نمودار ستونی	Box plot
نمودار شناسایی	Exploratory plot
نمودار میله‌ای	Bar plot
نشان	Tag
نشانه‌ی توالی بیان شده (EST)	Expressed Sequence Tag (EST)
نشانه، نما	Flag
نقشه‌ی خود سازمان‌ده (SOM)	Self-Organizing Map (SOM)
نمایه	Index
نوار کروموزومی	Cytoband
نوآیند	Nascent
نوک	Tip

نووا	Nova
نیمبلِگِن	Nimblegen

و

واکنش زنجیره‌ای پلیمراز (PCR)	Polymerase Chain Reaction (PCR)
ورساخته	Artifact
وضعیت	Mode
وضعیت خوانش تکی	Single read mode
وضعیت خوانش جفت انتهایی (هر دو انتها)	Paired-end read mode
وضوح تک نوکلئوتیدی در همبرسازی UV و رسوب ایمنی (توالی‌یابی iCLIP)	Individual-Nucleotide Resolution UV Crosslinking and Immunoprecipitation (iCLIPSeq)
وکتور گرافیک	Vector graphics
ویژگی	Attribute

هـ

هستی‌شناسی ژن (GO)	Gene Ontology (GO)
همپوشان جفتی	Pairwise overlap
همپوشانی - طرح‌بندی - اجماع (OLC)	Overlap-Layout-Consensus (OLC)
همردیف‌ساز پیرایش شده	Spliced aligner
همردیفی	Alignment
همگرایی	Converge
همگردانی	Compile
همگن	Homogeneous
هیبرید، آمیخته	Hybrid
هیستوگرام (بافت‌نگار)	Histogram

ی

یگانگی	Uniqueness
یونیکس	Unix

فهرست موضوعی

ب

ژ

س

ش

ع

ف

ک

گ

ل

م

RNA-seq
Data Analysis
A Practical Approach

By:
Eija Korpelainen, Jarno Tuimala,
Panu Somervuo, Mikael Huss, Garry Wong

Translated by:

Farjad Rafeie
Habibollah Samizadeh Lahiji

RNA-seq Data Analysis

A Practical Approach

By:

Eija Korpelainen
Jarno Tuimala
Panu Somervuo
Mikael Huss
Garry Wong

Translated by:

Farjad Rafeie
Habibollah Samizadeh Lahiji

ISBN: 978-622-6861-00-7