دانشکدهٔ ریاضی و آمار دانشگاه اصفهان

پروژه کارشناسی

استاد راهنما: دکتر فاطمه منصوری

راهنمای تحلیل ژن‌های دارای بیان متفاوت در تکنولوژی RNA-Seq

نویسنده: مریم رضائی

ویرایش دوم

نیم‌سال سوم ۱۴۰۲-۱۴۰۱

فهرست مطالب

در مطالعات وابسته به بیان ژن‌ها، دو تکنولوژی اصلی ساختار نمونه مورد نیاز مطالعه را تشکیل می‌دهند: Micorarray و RNA-Seq. این دو تکنولوژی نمای کاملی از الگوهای بیان ژن‌ها فراهم می‌کنند و این امکان را به محققان می‌دهند که با تشخیص ژن‌های با بیان متفاوت (DEGs) به بررسی بهتر رشد، بیماری‌ها، واکنش به دارو و تجویزات شخصی‌سازی شده پزشکی بپردازند. اما کار با هر یک نیازمند ابزارها، فایل‌ها و مراحل خاص خود است. به علت بروزتر بودن تکنولوژی Micorarray و وجود انواع پکیج برای تحلیل دادهٔ آن، این راهنما یک چهارچوب گام به گام برای تحلیل داده‌های RNA-Seq شامل پیش‌پردازش داده، کنترل کیفیت و شناسایی ژن‌های با بیان متفاوت، ارائه می‌دهد.

پیش از استفاده ابزارهای شرح داده شده، نیازمند انتخاب مطالعاتی هستیم که بیان ژن در نمونه‌های بیماری یا مصرف دارو را جمع‌آوری کرده‌اند؛ آنگاه با دانلود داده می‌توانیم ابتدا با پردازش آن و به دست آوردن تعداد بیان ژن‌ها و سپس مقایسه آن با نمونه سالم یا پیشین، میزان تفاوت بیان ژن را محاسبه کنیم. دانلود داده‌های Micorarray یا RNA-Seq مطالعه مرجع، از پایگاه داده‌های متوالی‌های نوکلئوتید قابل انجام است. در صورتی که با این پایگاه داده‌ها و شیوه کار با آن‌ها آشنا نیستید، به پیوست آ مراجعه کنید.

نکته

لازم به ذکر است که استفاده از این ابزارها نیازمند سیستم‌عامل لینوکسی و حافظه قابل توجهی می‌باشد و توصیه می‌شود محاسبات با استفاده از یک سرور لینوکسی انجام گیرد. با این حال، در انتخاب ابزارها، مواردی ذکر شده‌اند که حافظه کمتر نیاز داشته و بر روی دستگاه لینوکسی نیز قابل انجام باشند.

۱ تحلیل داده‌های RNA-Seq

برای تحلیل این داده‌ها با فرض اینکه داده‌های نمونه‌ها از پیش آماده و توالی‌یابی شده‌اند، به ترتیب مراحل زیر مورد توجه هستند:

۱. **تهیه داده‌های بیان ژن:** پسوند این فایل‌ها `fastq`. می‌باشد و دانلود از پایگاه داده‌های توضیح داده شده در پیوست آ ممکن است.

۲. **پیش‌پردازش داده‌ها:** در این مرحله با انواع ابزارهای لینوکس، داده‌ها را بررسی و تمیز می‌کنیم. مسیر کلی مورد نیاز و ابزارهای شرح داده شده در این راهنما به شرح زیر هستند:

(آ) **سنجش کیفیت کنترل نمونه:** با ابزار FastQC، نمونه را تحلیل و فایل HTML خروجی حاوی نتیجه انواع تست‌ها را دریافت می‌کنیم. در صورت کیفیت بالای نمونه، به مرحله (ج) می‌رویم. در غیر این صورت ادامه می‌دهیم.

(ب) **فیلتر و برش نمونه:** بر اساس تحلیل‌های مرحله پیشین، با ابزار Trimmomatic بخش‌های دارای کیفیت کم یا Adapterهای موجود را حذف می‌کنیم؛ خروجی نمونه‌های برش یافته با کیفیت مناسبند. بر اساس تحقیقات، حذف Adapterها لازم نیست و حتی می‌تواند به نمونه آسیب بزند.

۳. **شمارش ژن داده‌ها:** این مرحله نیز با انواع ابزارها که همگی بر لینوکس قابل اجرا هستند ممکن است و در آن برای شمارش خوانش ژن‌ها در نمونه‌ها گام‌های زیر را طی می‌کنیم:

(آ) **ساخت یا دانلود اندیس ژن‌ها:** لازم است فایل ژن‌های اندیس‌گذاری شده جاندار مورد نظر را با پسوند `fa.` تهیه کنیم. برای این کار بایستی یا فایل‌های آماده را دانلود کنیم، یا با دانلود خود داده‌های موقعیت ژن‌ها، با ابزار HISAT2-build از HISAT2

آن‌ها را بسازیم که بسیار زمان‌بر است.

(ب) **هم‌راستا کردن نمونه‌ها با ژن مرجع:** نمونه‌ها و فایل‌های اندیس‌گذاری شده را به ابزار HISAT2 وارد کرده و فایل SAM یا BAM (فشرده‌تر) را دریافت می‌کنیم که در آن اینکه هر خوانش مربوط به کدام کروموزم است تشخیص داده شده است. در صورت نیاز می‌توان با ابزار SAMtools این فایل‌ها را بر اساس موقعیت مرتب کرد (یعنی خوانش‌های کروموزم ۱ را ابتدا قرار دهد، سپس کروموزم ۲ و ...) یا به هم تبدیل کرد.

(ج) **شمارش خوانش‌های هر ژن:** پس از تهیه فایل موقعیت ژن‌ها با پسوند `gtf.` یا `gff.`، فایل SAM به دست آمده از مرحله قبل را با ابزار HTSeq.count از HTSeq تحلیل می‌کنیم تا تعداد خوانش هر ژن در نمونه موجود را به دست آوریم.

۴. **یافتن ژن‌های دارای بیان متفاوت:** پس از به دست آوردن شمارش ژن‌ها، با استفاده از زبان R در گام‌های زیر به تحلیل DEG می‌پردازیم:

(آ) **ساخت ماتریس تفاوت:** ابتدا لازم است تمامی نمونه‌ها را در ماتریسی که سطرهای آن نشانگر ژن و ستون‌های آن نشانگر تعداد در هر نمونه هستند قرار دهیم.

(ب) **تحلیل تفاوت بیان ژن‌ها:** با ساخت ماتریس، لازم است برای ستون‌ها برچسب مربوط به گروهشان (برای مثال، سالم یا بیمار) قرار دهیم. سپس دو گروه را پایه تحلیل تفاوت بیان انجام دهیم تا ماتریس جدید تفاوت بیان هر ژن را پیدا کنیم.

در ادامه به شرح گام‌های هر یک از مراحل مذکور تحلیل داده می‌پردازیم. اما در ابتدا، لازم است ابزارهای مورد استفاده در مراحل پیش رو را بر سیستم‌عامل لینکوسی خود نصب کنیم.

۰.۱ نصب ابزارها

برای نصب ابزارها بر سیستم‌عامل لینوکسی خود، با اتصال به اینترنت ابتدا در ترمینال ابزار دانلود و نصب بسته‌ها را آپدیت کرده و یک دایرکتوری برای نرم‌افزارهای مورد نیاز می‌سازیم و به آن می‌رویم:

```
$ sudo apt-get update
$ cd
$ mkdir apps
$ cd apps
```

در ادامه، شیوه نصب تمام ابزارهای مورد نیاز ذکر شده‌اند؛ هر ابزاری را که از پیش ندارید نصب کنید و در صورت برخورد با ارور حین نصب، اگر بسته‌ای که به آن وابسته هستند را ندارید، آن را نیز نصب کنید. لازم به ذکر است که راهنماهای نصب برای اوبونتو و دبین می‌باشند و در صورتی که لینوکس شما فدورا است، کافیست نام بسته متناظر را در اینترنت جستجو کنید.

- **ابزار FastQC:** دستور زیر را در ترمینال سیستم وارد کنید:

```
$ sudo apt-get -y install fastqc
```

- **ابزار Trimmomatic:** دستورات زیر را به ترتیب در ترمینال سیستم وارد کنید تا فایل کد را دانلود کرده، از فشردگی خارج کنید و نام دایرکتوری آن را برای آینده ساده‌تر کنید:

```
$ wget http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/Trimmomatic-0.39.zip
$ unzip Trimmomatic-0.39.zip
$ mv Trimmomatic-0.39 TRM39
```

- **ابزار HISAT2:** دستور زیر را در ترمینال سیستم وارد کنید:

```
$ sudo apt-get -y install hisat2
```

- **ابزار SAMtools:** در صورت تمایل به نصب مستقیم بسته از سورس، دستورات زیر را در ترمینال سیستم وارد کنید و در مرحله ششم تمامی دستورات لیست شده برای شما در ترمینال را کپی و اجرا کنید:

```
$ wget https://github.com/samtools/samtools/releases/download/1.18/samtools-1.18.tar.bz2
$ tar -vxjf samtools-1.18.tar.bz2
$ cd samtools-1.18
$ ./configure --prefix=/apps/samtools-1.18/install
$ make
$ sudo make install
$ export PATH=/apps/samtools-1.18/install/bin:$PATH
```

در غیر این صورت و علاقه به استفاده از سیستم مدیریت بسته conda با دستورات زیر آن را نصب کرده و بسته را با آن دریافت و خودکار نصب کنید:

```
$ wget https://repo.anaconda.com/miniconda/Miniconda2-latest-Linux-x86_64.sh
$ bash Miniconda2-latest-Linux-x86_64.sh
$ rm Miniconda2-latest-Linux-x86_64.sh
$ source ~/.bashrc
$ conda install -c bioconda samtools bcftools
```

- **ابزار HTSeq:** دستور زیر را در ترمینال سیستم وارد کنید تا خود pip نیازمندی‌ها را برطرف کند:

```
$ sudo pip install HTSeq
```

- **زبان R و ابزار RStudio:** ابتدا با راهنمای سایت زیر بسته به سیستم‌عامل خود R را نصب کنید:

cran.rstudio.com/

سپس باز بسته به سیستم‌عامل خود، از لینک زیر راهنمای دانلود مربوطه را دنبال کنید:

posit.co/download/rstudio-desktop/

- **ابزار DESeq2:** ابتدا کدهای زیر را در ترمینال سیستم خود وارد کنید:

```
$ sudo apt-get install libxml2-dev
$ sudo apt-get install r-cran-xml
$ sudo apt-get install libcurl4-openssl-dev
```

سپس در RStudio کد زیر را اجرا کنید:

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("DESeq2")
```

در صورت برخورد به هرگونه ارور در حین نصب از راهنماهای موجود در اینترنت کمک بگیرید. در نهایت با نصب ابزارهای مورد نیاز می‌توانید مراحل پردازش داده نمونه‌ها را در ادامه دنبال کنید.

۱.۱ تهیه داده‌های بیان ژن

شیوه دانلود داده‌های نمونه RNA-Seq از مراحل تشریح شده در پیوست آ پیروی می‌کند. تنها نکته قابل توجه این است که داده‌ها می‌توانند به دو نوع قابل دسترسی باشند: داده‌های خام و داده‌های پردازش شده.

۱.۱.۱ داده‌های خام RNA-Seq

داده‌های خام بیان ژن تهیه شده توسط تکنولوژی RNA-Seq به فایل FASTQ شناخته می‌شود و یک فایل متنی با پسوند `.fastq` است. این فایل با جدا کردن RNA از سلول و تقسیم آن به توالی‌های کوچکتر و سپس نمونه برداری (خواندن) سر توالی‌ها تولید می‌شوند و در نتیجه در فایل تولیدی این روش، به ازای هر خوانش، به ترتیب چهار مورد زیر ثبت شده‌اند:

۱. اطلاعاتی در مورد شناسه توالی خوانده شده
۲. چیستی خود توالی خوانده شده (پایه‌های A، C، T، G، N که نماد نوکلئیک اسیدهایی هستند که ژن را تشکیل می‌دهند)
۳. جدا کننده +
۴. نمره کیفیت پایه‌ها (کدگذاری شده با Phred +33 که اعداد آن با کاراکترهای ASCLII داده شده‌اند)

برای مثال، متن زیر یک خوانش در یک فایل متنی `.fastq` است:

```
@ML-P2-14:9000H003HG:1:111:02:17290:1073 1:N:0:TVVTGAGC+GCGATCTA
TTTGGTAACAGCATGAATTAATTCTAGCCACTAAAACTCTATGAACATCTTGTGAAGGTTTCAGATAGAGCCTGAAGTACACAGAGAACAATTCTTAAAAAA
+
AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE<AEEEEEEE
```

لازم به ذکر است که شیوه خواندن توالی‌ها می‌تواند به دو گونه single-read و paired-end باشد، یعنی توالی تنها از یک سر توالی بریده شده به سوی دیگر یا، برای دقت بالاتر و اطلاعات بیشتر درمورد جایگاه توالی، از هر دو سمت (جهت اصل و برعکس آن) نمونه برداری شود. این امر در زمان دانلود فایل‌ها مشخص است؛ فایل‌های خوانش single-read دارای نام‌های مجزا هستند اما فایل‌های paired-end با R1 و R2 در نام خود به هم مربوط شده‌اند که هر فایل خوانش یک سر توالی را ثبت کرده است و بایستی دو فایل با هم پردازش شوند.

نکته

در صورتی که فایل خام مطالعه خود را دانلود کرده‌اید، لازم است بخش ۲.۱ و ۳.۱ را برای پردازش و شمارش آن و بخش ۴.۱ را برای یافتن ژن‌های دارای بیان متفاوت آن مطالعه کنید.

۲.۱.۱ داده‌های پردازش شده RNA-Seq

داده‌های پردازش شده نمونه حاصل از تکنولوژی RNA-Seq در بعضی مواقع توسط محققیق اصلی بارگزاری شده و در صفحه فراداده مطالعه در پایگاه منتخب به صورت یک فایل متنی txt. یا جدول قابل دانلود قابل دسترسی هستند. این فایل‌ها به طور معمول دارای یک یا چند ستون شناسایی ژن (نام، آیدی و یا دیگر موارد) و یک ستون برای تعداد شمرده شده از بیان آن ژن در نمونه می‌باشند. برای مثال، چند سطر ابتدایی فایل پردازش شده مطالعه GSE68391 به شکل زیر می‌باشند:

ID	GeneName	GeneType	Count
1500011B03Rik.5.n	1500011B03Rik	protein_coding	136
2410002O22Rik.13.n	2410002O22Rik	protein_coding	988
2810055G20Rik.16.p	2810055G20Rik	protein_coding	1
2810453I06Rik.5.p	2810453I06Rik	protein_coding	354
4922502B01Rik.8.n	4922502B01Rik	protein_coding	0
4930523C07Rik.1.p	4930523C07Rik	protein_coding	533
4930556M19Rik.15.p	4930556M19Rik	lincRNA	7
5730522E02Rik.11.n	5730522E02Rik	protein_coding	2
6030419C18Rik.9.p	6030419C18Rik	protein_coding	7
A530058N18Rik.2.p	A530058N18Rik	lincRNA	0
AU015836.X.p	AU015836	protein_coding	0
Acer2.4.p	Acer2	protein_coding	9
Adora3.3.p	Adora3	protein_coding	13
Alkbh1.12.n	Alkbh1	protein_coding	1245

```
Als2cr11.1.n        Als2cr11        protein_coding 1
```

در این صورت بایستی نمونه حالت سالم (گروه دوم) را تهیه کنیم و سپس با ساخت ماتریس تفاوت‌ها، عملیات یافتن ژن‌های دارای بیان متفاوت را انجام دهیم. همچنین امکان دارد سه ستون نام ژن، شمارش در گروه اول نمونه‌ها و شمارش در گروه دوم نمونه‌ها را داشته باشیم که در این صورت بایستی تنها تحلیل تفاوت بیان ژن را اعمال کنیم.

نکته

در صورتی که فایل پردازش شده حاوی شمارش ژن‌های مطالعه خود را دانلود کرده‌اید، تنها لازم است بخش ۴.۱ را برای یافتن ژن‌های دارای بیان متفاوت آن مطالعه کنید.

۲.۱ پیش‌پردازش داده‌ها

در صورتی که داده‌های خام RNA-Seq با پسوند `fastq.` در اختیار دارید، لازم است برای به دست آوردن بیان ژن در آن‌ها، داده‌های خود را پردازش کنید. برای اینکار، در ادامه فرض می‌کنیم داده‌ها در پوشه `proj` در `home` سیستم عامل لینوکسی شما قرار دارند. بنابراین با فرض اینکه داده‌های مطالعه منتخب شما در آدرس زیر قرار دارند پیش می‌رویم:

```
proj/study_num/
```

با باز کردن ترمینال سیستم‌عامل و دستورات زیر به آدرس مذکور می‌رویم و محتوی آن را مشاهده می‌کنیم:

```
$ cd ~/proj/study_num
$ ls
```

بایستی حاصل دو دستور مذکور، نمایش نام محتوای پوشه شما حاوی فایل‌های فشرده `.fastq.gz` مطالعه باشد. حال در این دایرکتوری، به انجام مراحل می‌پردازیم. دقت کنید که پیش از انجام گام‌های پیش‌رو، لازم است تمامی ابزارهای بخش ۰.۱ را نصب کرده باشید.

۱.۲.۱ سنجش کیفیت کنترل نمونه

برای بررسی کیفیت نمونه‌ها و دریافت گزارش انواع خصوصه آن‌ها، از ابزار FastQC استفاده می‌کنیم. این ابزار، با دریافت فایل‌های فشرده `.fastq.gz`، فایل‌های اصلی `.fastq`، یا فایل‌های `.sam` یا `.bam` هم راستا شده با ژن‌ها، گزارشی شامل وضعیت و نمره نمونه‌ها از انواع منظرها تهیه می‌کند. علاوه بر لیست کردن تعداد خوانش‌ها و کیفیت کدگذاری آن‌ها، FastQC اطلاعاتی را در مورد کیفیت و محتوای پایه‌ها، طول خوانش‌ها، و محتوای k-mer (قطعات چند حرفی) نشان می‌دهد، وجود پایه‌های مبهم، دنباله‌های بیش‌تر از حد معمول و تکراری را گزارش می‌کند و نتایج تست و معیارها را به تصویر می‌کشد و قضاوت می‌کند.

با اجرای دستور زیر در دایرکتوری ذکر شده، می‌توانید یک فایل نمونه با نام `reads.fastq.gz` را کیفیت‌سنجی کنید:

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

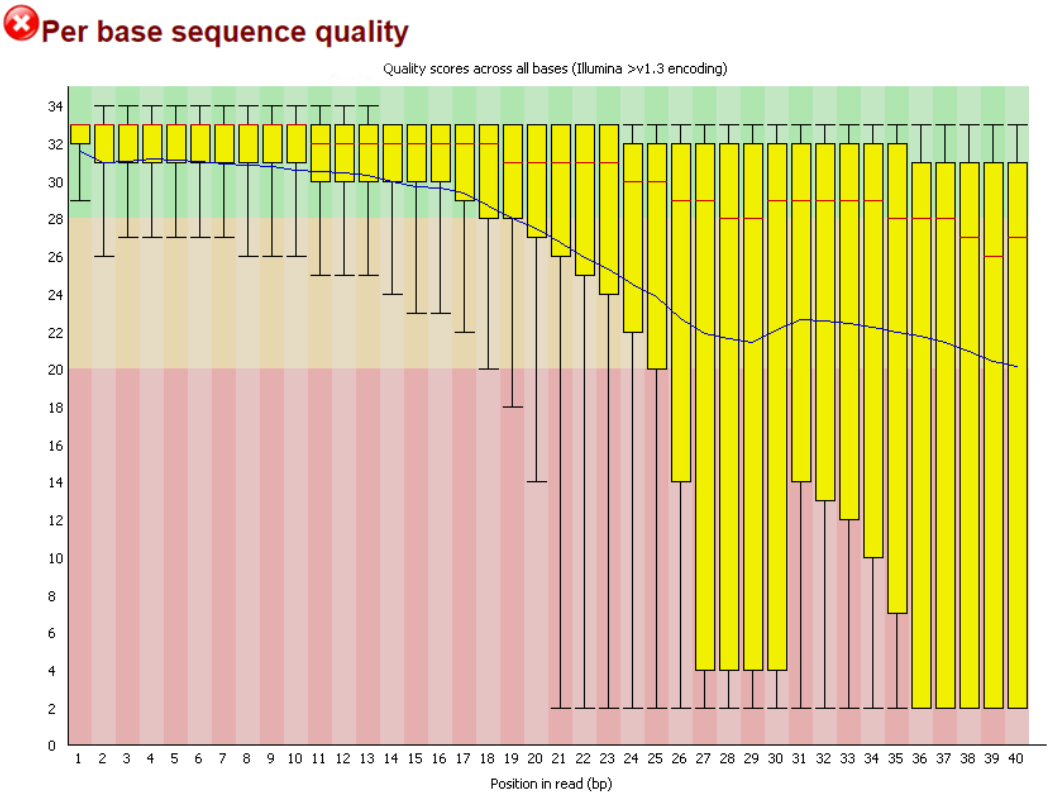
پس از اجرای کامل ابزار بر فایل منتخب شما، ابزار یک پوشه به نام `reads_fastqc` جهت ذخیره‌سازی فایل‌های نتیجه ایجاد می‌کند و نتایج را در فایل `fastqc_report.html` در دایرکتوری حاضر به تصویر می‌کشد.

در صورتی که قصد استفاده خودکار ابزار بر تمام فایل‌های `.fastq.gz` داخل پوشه مذکور را دارید، می‌توانید با استفاده از دستورات زیر، پوشه‌ای برای تمام فایل‌های نتیجه بسازید، به آن پوشه بروید و تمام فایل‌های نمونه پوشه قبل را کنترل کیفیت کرده و در پوشه جدید بریزید:

```
$ mkdir QC
$ cd QC
$ for FILE in ~/proj/study_num/*.fastq.gz; do fastqc $FILE; done
# doesn't save to QC; what should i do?
```

برای توضیح کامل نمودارهای حاصل، به این لینک مراجعه کنید. اما برای معرفی مهم‌ترین این نمودارها موارد زیر قابل توجه هستند:

۱. **تست کیفیت توالی مبنی بر پایه‌ها:** در نمودار Per base sequence quality محور x نشانگر جایگاه در طول هر توالی خوانده شده و محور y نشانگر کیفیت پایه‌های داخل آن جایگاه بین خوانش‌های متفاوت می‌باشد. نمونه‌ای از این نمودار که تست را نگذارنده است در شکل ۱ قابل مشاهده است.

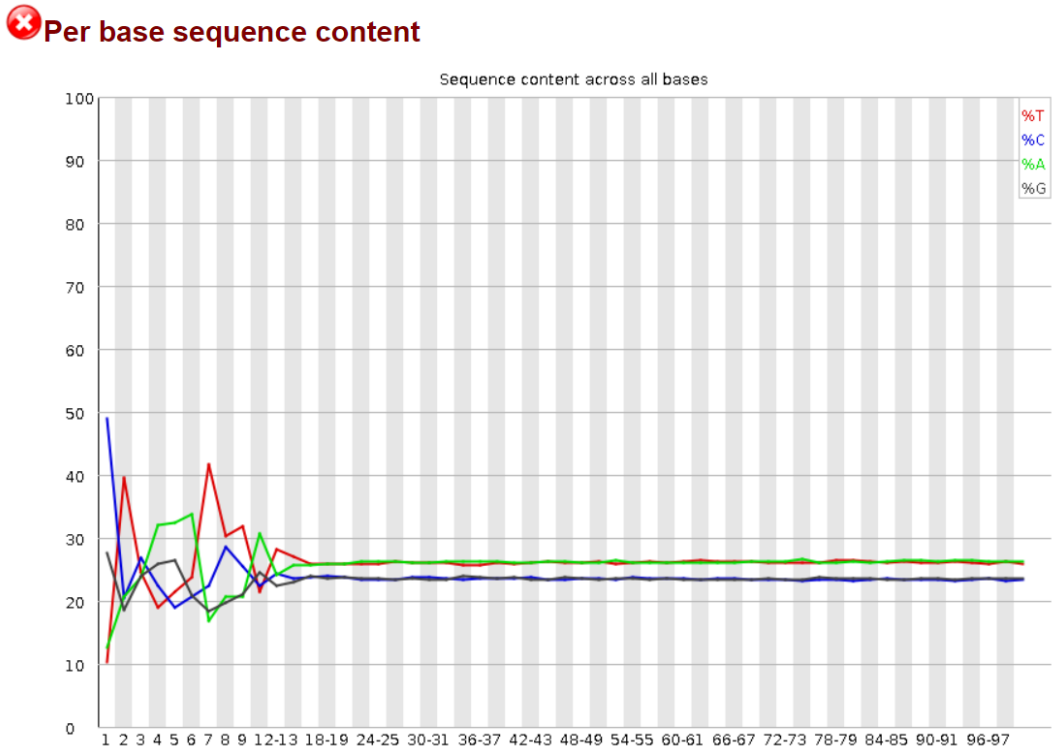


شکل ۱: کیفیت توالی مبنی بر پایه‌ها

در این نمودار boxplot مینیمم، ماکسیمم و میانگین کیفیت پایه‌های موجود در هر جایگاه را مشاهده می‌کنیم و می‌توانیم تصمیم بگیریم

که حرف‌هایی که کیفیت آن‌ها زیر مقدار مشخصی است را حذف کنیم.

۲. **تست محتوای توالی مبنی بر پایه‌ها:** در نمودار Per base sequence content محور x نشانگر جایگاه در طول هر توالی خوانده شده و محور y نشانگر درصد وجود هر پایه T، C، A و G در آن جایگاه خوانش‌ها می‌باشد. نمونه‌ای از این نمودار که تست را نگذرانده است در شکل ۲ قابل مشاهده است.



شکل ۲: محتوای توالی مبنی بر پایه‌ها

در این نمودار، توضیع ایده‌آل زمانیست که به علت تصادفی بودن توالی‌ها، هر چهار رنگ خط حدوداً همشکل و صاف باشند. اما همانطور که مشاهده می‌شود، به طور معمول سر و ته خوانش‌ها ناموزون می‌باشند و لازم است در مرحله بعد بریده شوند. در این مثال، می‌توان ۱۷ حرف اول نمونه را برید و طول خوانش‌ها را به ۸۰ رساند که همچنان طول خوبیست.

لازم به ذکر است که در بخش Overrepresented sequences می‌توان توالی‌های پرتکرار که ممکن است شامل ژن زیاد بیان شده، یا آلودگی یا Adapterها (نیرونوکلئوتیدهای کوتاهی که برای لیگاسیون و اتصال بهتر استفاده می‌شوند) باشند. امکان تشخیص این آداپتورها از گزارش و حذف آن‌ها در مرحله بعد موجود است، اما مطالعات نشان داده‌اند که امکان دارد حذف آداپتورها باعث آسیب به نمونه و کاهش تشخیص ژن‌ها شود و به طور خودکار آداپتورهای موجود در هم‌راستاسازی نمونه با ژن‌های مرجع کنار گذاشته می‌شوند.

در ادامه، در صورت نیاز به حذف بخش‌هایی از نمونه، به بخش ۲.۲.۱ می‌رویم. در غیر این صورت، به بخش ۳.۱ پرش می‌کنیم.

۲.۲.۱ فیلتر و برش نمونه

ابزار Trimmomatic از بهترین ابزارها برای فیلتر و برش نمونه بر اساس کیفیت است که قابلیت کار با خوانش‌های paired-end را نیز دارد. در ورودی ابزار، فایل‌های fastq. و یا فشرده آن‌ها به صورت fastq.gz. دریافت شده و فایل مربوطه خروجی تولید می‌شود. دقت کنید بسته به single-read بودن یا paired-end بودن نمونه‌ها، شیوه برش و فیلترینگ متفاوت است و در حالت دوم لازم است نمونه‌ها با هم تغییر کنند.

برای راهنمای کامل ابزار می‌توانید به این لینک مراجعه کنید. در ادامه، هر دو نوع استفاده از ابزار را با شرح چند قابلیت آن بررسی می‌کنیم.

پیش از بررسی دستور ابزار، در صورتی که به دایرکتوری جدید QC رفته بودید، دستور زیر را اجرا کنید تا یک پوشه به عقب برگردید:

```
$ cd ..
```

سپس با استفاده از دستورات زیر، دایرکتوری جدیدی برای ذخیره نمونه‌های فیلتر و برش شده بسازید، به داخل آن بروید و برای دسترسی به فایل‌های پوشه قبل از آن‌ها در این پوشه یک لینک به فایل‌ها بسازید (مقصد این پوشه با نقطه انتها تعیین شده است):

```
$ mkdir FLTR
$ cd FLTR
$ ln -s ../*.fastq.gz .
```

در ادامه در نظر داریم که برای اجرای ابزار که به صورت فایل jar. است، java -jar را برای اجرای کد و آدرس /apps/TRM39/ و نام فایل trimmomatic-39.0.jar را برای اشاره به خود کد نیاز داریم. پس از آن با SE یا PE تعیین می‌کنیم که خوانش single-read یا paired-end است. در ادامه، دو حالت زیر را داریم:

- **نمونه single-read:** در صورتی که فایل ما نام reads.fastq.gz داشته باشد، با دستور زیر می‌توانیم ۱۷ حرف اول تمام خوانش‌های آن را حذف کنیم، از ابتدا و انتهای نتیجه حروفی که کیفیت زیر ۲۶ دارند را ببریم، سپس خوانش‌های حاصلی را که طول آن‌ها پس اعمال پیشین زیر ۵۰ شد را به دور بی‌اندازیم:

```
$ java -jar ~/apps/TRM39/trimmomatic-0.39.jar SE reads.fastq.gz reads_trmd.fastq.gz
    HEADCROP:17 LEADING:26 TRAILING:26 MINLEN:50
```

در نتیجه دستور بالا، خروجی اعمال در فایل reads_trmed.fastq.gz ذخیره می‌شوند. در صورتی که بیش از یک فایل single-read در پوشه داشتید می‌توانید با دستور زیر، با حلقه و به ترتیب دستور را بر هر فایل اجرا کنید:

```
$ for FILE in *.fastq.gz; do java -jar ~/apps/TRM39/trimmomatic-0.39.jar SE $FILE
    $(basename -s .fastq.gz $FILE)_trmed.fastq.gz HEADCROP:17 LEADING:26 TRAILING:26
    MINLEN:50; done
```

با دستور basename -s نام بدون پسوند fastq.gz. جدا شده و در ادامه به آن پسوند _trmed.fastq.gz اضافه می‌شود.

- **نمونه paired-end**: در صورتی که فایل‌های ما reads_1.fastq.gz و reads_2.fastq.gz باشند، با استفاده از دستور زیر می‌توانیم آن‌ها را به ابزار بدهیم، ۱۷ حرف اول تمام خوانش‌های آن‌ها را حذف کنیم، از ابتدا و انتهای نتیجه حروفی که کیفیت زیر ۲۶ دارند را ببریم، سپس خوانش‌های حاصلی را که طول آن‌ها زیر ۵۰ شد را به دور بی‌اندازیم:

```
$ java -jar ~/apps/TRM39/trimmomatic-0.39.jar PE reads_1.fastq.gz reads_2.fastq.gz
    -baseout reads.fastq.gz HEADCROP:17 LEADING:26 TRAILING:26 MINLEN:50
```

در دستور بالا، خروجی با پایه اسمی reads.fastq.gz در چهار فایل تولید می‌شود: فایل‌های reads_1P.fastq.gz و reads_2P.fastq.gz که با هم همخوان بریده شده‌اند و همچنان نیز به صورت paired-end با هم قابل استفاده می‌باشند و فایل‌های reads_1U.fastq.gz و reads_2U.fastq.gz که در آن‌ها خوانش متناظر در فایل دیگر حذف شده و حاوی نمونه‌هایی هستند که به صورت single-read عمل می‌کنند.

در صورتی که تعداد زیادی جفت فایل با الگوی اسمی مذکور داشتید و می‌خواستید دستور را به صورت حلقه‌ای به ترتیب بر فایل‌ها اعمال کنید، از دستور زیر استفاده کنید:

```
$ for FILE in *_1.fastq.gz; do java -jar ~/apps/TRM39/trimmomatic-0.39.jar PE $FILE
    ${FILE/_1/_2} -baseout ${FILE/_1} HEADCROP:17 LEADING:26 TRAILING:26 MINLEN:50;
    done
```

در این دستور، با FILE/_1/_2 پسوند 1_ را از انتهای نام فایل حذف می‌کنیم و با 2_ جایگزین می‌کنیم. بقیه بخش‌های دستور به شکل قبل تعیین و اجرا می‌شود.

در نهایت در دایرکتوری /proj/study_num/FLTR/ فایل‌های نمونه فیلتر و بریده شده به علاوه لینک به فایل‌هاای اصلی موجودند. با دستور زیر، لینک‌ها را پاک کنید:

```
$ find maxdepth 1 -type l -delete
#doesn't work
```

پس از این مرحله، با فایل‌های fastq. جدید به جای نمونه‌های اصلی پیش رفته و در مراحل بعد کار کنید.

۳.۱ شمارش ژن داده‌ها

در این گام لازم است با هم‌راستاسازی نمونه‌ها با ژن‌های اصلی (کشف اینکه قطعات چه ژن‌هایی در خوانش ما موجود بودند) و سپس شمارش تعداد خوانش مربوط به هر ژن، به فایل پردازش‌شده نمونه برسیم تا تحلیل DEG را بر آن انجام دهیم.

در این گام، برای هم‌راستاسازی از ابزار HISAT2 و برای شمارش خوانش هر ژن از HTSeq استفاده می‌کنیم. هم‌راستاسازهای دیگری مانند Bowtie و TopHat نیز پر استفاده هستند که اولی به علت ساخت اولیه آن برای DNA و دومی به علت بر پایه Bowtie بودن دقت بالایی ندارند. هم‌راستاسازهای مناسب دیگری مانند STAR و BBMap نیز موجودند که برای RAM بیش از ۲۸ گیگابایت، STAR و برای کمتر، HISAT2 پیشنهاد می‌شود. به جای HTSeq، ابزارهای دیگری مانند cufflinks نیز موجود و پر استفاده هستند.

۱.۳.۱ ساخت یا دانلود اندیس ژن‌ها

پیش از هم‌راستا کردن نمونه‌ها با ژن‌های اصلی، لازم است برای ژن‌ها به ترتیب اندیس گذاشته باشیم تا بتوانیم در مراحل بعد از الگوریتم‌های سریعتر هم‌راستا کردن با فایل ژن‌ها استفاده کنیم. در این راستا لازم است یا با تهیه ژن‌های انسان آن‌ها را اندیس‌گذاری کنیم (که عملی بسیار زمان‌بر است)، یا فایل‌های اندیس‌گذاری آماده مخصوص ابزار هم‌راستاسازی خود را برای جاندار مورد نظر دانلود کنیم.

- **دانلود فایل اندیس‌گذاری شده:** فایل ژن اندیس‌گذاری شده بسیاری جانداران برای HISAT2 در آدرس زیر موجود است:

daehwankimlab.github.io/hisat2/download/

در صورت وجود جاندار مورد نظر شما، بهتر است فایل آن را دانلود کنید. لازم به ذکر است که برای انسان که گزینه‌های متعددی برای دانلود موجود است، می‌توانید با توجه به روش تحلیل خود مطالعه در صورت وجود، ورژن مربوطه را دانلود کنید. اما به طور معمول، GRCh38 گزینه مناسبی می‌باشد.

با فرض اینکه نام فایل دانلودی شما `refgenome.idx.tar.gz` باشد و در دایرکتوری `/proj/genome.idx/` موجود باشد، با دستور زیر از هر جایی به آن دایرکتوری رفته و آن را از فشردگی خارج می‌کنیم:

```
$ cd ~/proj/genome.idx
$ tar -xvzf refgenome.idx.tar.gz
```

این عمل دایرکتوری جدیدی خواهد ساخت که در آن چند فایل با نام‌های `genome.1.ht2` با شماره‌های متفاوت موجود خواهند بود. با دستور `ls` می‌توانیم نام این پوشه جدید در دایرکتوری حال حاضر را مشاهده کنیم. فرض می‌کنیم این فایل‌ها در دایرکتوری `refgenome` اضافه شده‌اند، پس برای راحتی ادامه کار، با دستور زیر آن‌ها را به `/proj/genome.idx/` جابه‌جا می‌کنیم و پوشه را حذف می‌کنیم:

```
$ cd ~/proj/genome.idx
$ mv refgenome/* .
$ rmdir refgenome
```

- **اندیس‌گذاری فایل ژن‌ها:** در صورتی که فایل اندیس‌گذاری شده جاندار شما موجود نباشد، می‌توانید با دانلود فایل `.fa` یا `.fasta` ژن‌های مربوطه از یکی از سایت‌های زیر، از آن برای تهیه فایل اندیس‌گذاری شده با HISAT2 استفاده کنید:
  - **Ensembl:** ensembl.org/info/data/ftp/index.html/

    download `dna.primary_assembly`
  - **UCSC:** hgdownload.soe.ucsc.edu/downloads.html/

با فرض اینکه فایل دانلودی شما نام `refgenome.fa` در دایرکتوری `/proj/genome.idx/` باشد، با دستورات زیر از هر جایی به آن دایرکتوری می‌رویم و سپس به ساخت فایل‌های اندیس‌گذاری شده می‌پردازیم:

```
$ cd ~/proj/genome.idx
$ hisat2-build refgenome.fa genome
```

پس از چند ساعت اجرا، چندین فایل با پیشوند نام genome و پسوند ht2. ساخته خواهند شد.

۲.۳.۱ هم‌راستا کردن نمونه‌ها با ژن مرجع

با ابزار HISAT2 و با استفاده از فایل‌های فیلتر شده با پسوند fastq.gz. داخل دایرکتوری /proj/study_num/FLTR/، فایل‌های هم‌راستا شده با ژن‌ها را تولید می‌کنیم؛ این فایل‌ها دارای پسوند sam. یا به صورت فشرده‌تر، bam. می‌باشند. می‌توان پس از این مرحله، فایل‌های خروجی را با ابزار SAMtools مرتب یا به هم تبدیل کرد.

دقت می‌کنیم که در ابزار HISAT2، چگونگی نمونه (یعنی single-read یا paired-end بودن) در هم‌راستا کردن بسیار مهم است زیرا بایستی خوانش‌هایی که برای یک توالی هستند با هم برای بهبود کیفیت هم‌راستاسازی در نظر گرفته شوند. در این راستا، مجدداً دو حالت single-read و paired-end را جدا بررسی می‌کنیم.

نکته

برای راهنمای کامل کار با SAMtools و قابلیت‌های آن، به این لینک و برای آشنایی با تمام گزینه‌های HISAT2 به این لینک مراجعه کنید. در ادامه چند قابلیت پر استفاده این ابزارها ذکر شده‌اند.

پیش از انجام هم‌راستاسازی و اعمال مربوطه، لازم است برای جدا کردن فایل‌های حاصل، دایرکتوری جدیدی در پوشه داده‌های مطالعه بسازیم و لینک فایل‌های نمونه را در آن ایجاد کنیم. برای این کار از دستورات زیر استفاده می‌کنیم:

```
$ cd ~/proj/study_num
$ mkdir ALGND
$ cd ALGND
$ ln -s ../FLTR/*.fastq.gz .
```

در ادامه، بایستی تنها محتوای /proj/genome.idx/ که با genome شروع می‌شوند، فایل‌های ht2. اندیس‌گذاری شده باشند.

- **نمونه single-read**: اگر فرض کنیم فایل‌های یگانه نمونه‌ها ما دارای نام‌های reads0X_trmed.fastq.gz باشند که X اعداد ۱ تا ۴ است، در دایرکتوری حال حاضر (/FLTR/) دستور زیر را اجرا می‌کنیم که در آن پس از x- آدرس و ابتدای نام فایل‌های اندیس‌گذاری شده، پس از U- (تعیین کننده یگانه بودن خوانش‌ها) نام فایل‌های نمونه در دایرکتوری حال حاضر جدا شده با ویرگول، پس از S- نام فایل SAM خروجی، و پس از p- تعداد هسته پردازنده‌های مورد استفاده در محاسبه را قرار می‌دهیم:

```
$ hisat2 -x ../../genome.idx/genome -U reads01_trmed.fastq.gz,
    reads02_trmed.fastq.gz, reads03_trmed.fastq.gz, reads04_trmed.fastq.gz -S
    study_num.sam -p 8
```

خروجی دستور مذکور، یک فایل study_num.sam است که تمام خوانش‌های چهار فایل یگانه را با مرجع هم‌راستا کرده است (یعنی به ازای هر خوانش ذکر کرده است که با چه بازه اندیسی از ژن‌ها همخوانی داشته است). در صورتی که گروه نمونه‌ها با همدیگر متفاوت

باشد و بخواهیم نتیجه هم‌راستاسازی آن‌ها را از هم جدا نگهداریم که برای مرحله بعد، شمارش ژن در هر نمونه مجزا باشد، می‌توانیم با حلقه زیر، تک تک نمونه‌ها را به دستور وارد کنیم تا برای هر یک، فایل reads0X.sam متناظر تولید شود:

```
$ for FILE in *_trmed.fastq.gz; do hisat2 -x ../../genome.idx/genome -U $FILE -S
    ${FILE/_trmed*/.sam} -p 8
```

اگر تمایل داشته باشیم فایل فشرده BAM از فایل خروجی نمونه reads01.sam تولید کنیم و آن را برای بهبود محاسبات مرتب کنیم، با دستورات زیر می‌توانیم تبدیل را انجام دهیم و فایل reads0X.bam را بسازیم، آن را بر اساس جایگاه مرتب کرده و فایل reads0X.sorted.bam را بسازیم، سپس آن را اندیس گذاری کنیم:

```
$ samtools view -bS -o reads01.bam reads01.sam
$ samtools sort reads0X.bam -o reads01.sorted.bam
$ samtools index reads01.sorted.bam
```

لازم به ذکر است که با دستور n- پس از sort می‌توان، فایل را بر اساس نام ژن‌ها مرتب کرد که این سبک خودکار ورودی HTSeq می‌باشد اما امکان اندیس‌گذاری ندارد.

نکته جالب این است که ابزار SAMtools قابلیت استفاده بر فایل‌های در جریان را دارد و می‌توان با قابلیت دستورات stream لینوکس برای کم کردن فایل‌های پر حجم و متعدد ذخیره شده، پیش از ذخیره اعمال SAMtools را تک تک با دستور بر فایل زیر اجرا کنیم:

```
$ hisat2 -q -p 8 -x ../../genome.idx/genome -U reads01_trmed.fastq.gz,
    reads02_trmed.fastq.gz, reads03_trmed.fastq.gz, reads04_trmed.fastq.gz -S - |
    samtools view -bS - | samtools sort - | samtools index - -o reads01-04.sorted.bam
```

در این دستور، q- به ساکت کردن و چاپ نکردن چیزی توسط عمل اشاره می‌کند و قرار دادن - به جای نام ورودی و خروجی، اجازه می‌دهد داده میان دستورات جدا شده با | جریان پیدا کند.

- **نمونه paired-end**: اگر فایل‌های موجود دوگانه با نام‌های reads01_1P.fastq.gz و reads01_2P.fastq.gz باشند، لازم است دستور را به صورت زیر تغییر دهیم که در آن پس از x- همچنان آدرس و ابتدای نام فایل‌های اندیس‌گذاری شده، پس از 1- و 2- (تعیین کننده دوگانه بودن خوانش‌ها) نام دو فایل نمونه دوگانه در دایرکتوری حال حاضر، پس از S- نام فایل SAM خروجی، و پس از p- تعداد هسته پردازنده‌های مورد استفاده در محاسبه را قرار می‌دهیم:

```
$ hisat2 -x ../../genome.idx/genome -1 reads01_1P.fastq.gz -2 reads01_2P.fastq.gz -S
    study_num.sam -p 8
```

در حالتی که چندین فایل دوگانه داشته باشیم (برای مثال فایل‌های reads02_1P.fastq.gz و reads02_2P.fastq.gz نیز در پوشه موجود باشند)، می‌توانیم با استفاده از حلقه، دستور را به شکل زیر تغییر دهیم تا اعمال را تک تک بر هر جفت نمونه اجرا کند و فایل reads0X_P.sam متناظر برای خوانش جفتی را بسازد:

```
$ for FILE in *_1P.fastq.gz; do hisat2 -x ../../genome.idx/genome -1 $FILE -2
    ${FILE/_1/_2} -S ${FILE/_1*/_P.sam} -p 8; done
```

لازم به ذکر است که از آنجا که فیلتر و برش باعث شکستن نمونه‌های دوگانه به چهار فایل شده بود، امکان دارد هم فایل‌های دوگانه با الگوی اسمی reads0X_YP.fastq.gz و هم فایل‌های یگانه با الگوی اسمی reads0X_YU.fastq.gz داشته باشیم. در این صورت، طبق دو حالت بالا موارد یگانه و دو گانه را جدا اجرا می‌کنیم و فایل‌های reads0X_U.sam از هر جفت نمونه یگانه شده (یعنی reads01_1U.fastq.gz و reads01_2U.fastq.gz) و فایل‌های reads0X_P.sam را از نمونه‌های دو گانه موجود (یعنی برای مثال reads01_1P.fastq.gz و reads01_2P.fastq.gz) می‌سازیم. حال با اجرای دستور زیر همه را به bam. تبدیل کرده و مثل قبل مرتب می‌کنیم:

```
$ for FILE in *.sam; do samtools view -bS -o - $FILE | samtools sort - -o
    ${FILE/.sam/.sorted.bam}; done
```

اگر در نتیجه این دستور فایل‌های reads01_U.sorted.bam و reads01_P.sorted.bam تولید شده باشند، از آنجا که مربوط به یک نمونه هستند می‌توانیم آن دو را در یک فایل bam. قرار دهیم تا فایل هم‌راستا شده نمونه کامل را با نام reads01.bam به دست آوریم:

```
$ samtools merge -o reads01.bam reads01_P.bam reads01_P.bam
```

در هر یک از این مراحل می‌توانیم فایل‌های تولید شده اضافه مراحل را حذف کنیم و تنها فایل نهایی هر نمونه را به مرحله بعد ببریم.

۳.۳.۱ شمارش خوانش‌های هر ژن

در آخرین مرحله پردازش داده‌ها برای یافتن بیان هر ژن، لازم است فایل sam. یا bam. حاوی نگاشت خوانش‌ها به ژن‌ها همراه با فایل gtf. یا gff حاشیه‌نویسی شده[1] ژن‌های اصلی جاندار حاوی جایگاه خصوصه‌ها را به ابزاری مانند HTSeq-count دهیم تا خوانش‌های هر ژن را به کمک موقعیت آن‌ها به دست آوریم.

برای اینکار ابتدا لازم است فایل gtf. را از یکی از لینک‌های زیر برای جاندار مورد نظر و ژن‌های منظور (یا تمام ژن‌ها) دانلود کنیم:

- **Ensembl:** ensembl.org/info/data/ftp/index.html/
- **UCSC:** genome.ucsc.edu/cgi-bin/hgTables/

سپس فایل فشرده gz. را که فرض می‌کنیم در دایرکتوری /proj/genome.anot/ قرار دارد، از حالت فشرده خارج می‌کنیم:

```
$ cd ~/proj/genome.anot
$ gunzip genome.anot.gtf.gz
```

[1] Annotated

در آخر با دستور زیر به دایرکتوری جدیدی برای شمارش‌ها رفته و با اشاره به آدرس فایل هم‌راستا شده نمونه reads01.bam و فایل ژن‌های حاشیه‌نویسی شده، خوانش‌ها را می‌شماریم:

```
$ cd ~/proj/study_num
$ mkdir CNT
$ cd CNT
$ htseq-count -f bam -r pos ../ALGND/reads01.bam ../../genome.anot/genome.anot.gtf >
    reads01.count -q
```

در دستور بالا پس از f- نوع فایل (bam یا sam) و پس از r- مرتب شدن بر اساس جایگاه (در صورتی که بر اساس جایگاه مرتب نکرده بودید r- pos را قرار ندهید) را تعیین می‌کنیم و سپس آدرس فایل bam. یا sam. حاصل مطالعه و آدرس فایل gtf. یا gff. ژن‌ها را به ترتیب قرار می‌دهیم. همچنین با > تعیین می‌کنیم که فایل متنی خروجی را با نام reads01.count ذخیره کند.

در صورت داشتن چند فایل و نمونه، با حلقه زیر پس از ساخت دایرکتوری جدید و ایجاده لینک به فایل‌ها، دستور را چندبار تکرار می‌کنیم سپس لینک‌ها را حذف می‌کنیم:

```
$ cd ~/proj/study_num
$ mkdir CNT
$ cd CNT
$ ln -s ~/proj/study_num/ALGND/*.bam .
$ for FILE in *.bam; do htseq-count -f bam -r pos FILE ../../genome.anot/genome.anot.gtf >
    ${FILE/.bam/.count} -q; done
$ find maxdepth 1 -type l -delete
```

۴.۱ یافتن ژن‌های دارای بیان متفاوت

در این مرحله با فایل‌های متنی با پسوند count. شروع می‌کنیم که هر یک شمارش ژن‌ها همراه با نام ژن در هر نمونه را ثبت کرده‌اند. برای یافتن DEGها لازم است ماتریس تفاوت را شکل دهیم و سپس بر آن تحلیل تفاوت بیان ژن را با ابزار منتخب (اینجا DESeq2) انجام دهیم.

پیش از ادامه، با بازکردن RStudio، خطوط زیر را در ابتدای فایل خود قرار می‌دهیم تا کتابخانه مورد نیاز را لود کرده و تنظیمات فایل را برای کار با رشته‌ها و خواندن و نوشتن فایل‌ها تصحیح کنیم:

```
library(DESeq2)
options(stringAsFactor=F)
setwd("~/proj/study_num/CNT/")
```

کد بالا را اجرا می‌کنیم تا کتابخانه‌ها اضافه شوند. حال در ادامه، دو مرحله اصلی تحلیل را شرح می‌دهیم.

۱.۴.۱ ساخت ماتریس تفاوت

با فرض اینکه فایل‌های متنی شمارش ژن‌ها با پسوند count . برای هر نمونه حاوی دو ستون نام ژن و تعداد آن باشند، با قطعه کد زیر می‌توانیم فایل‌ها را در یک ماتریس به گونه‌ای قرار دهیم که هر سطر با عنوان نام یک ژن و هر ستون با نام نمونه که بر فایل مربوطه آن بوده است، عدد شمارنده را نگه دارد:

```
files <- list.files(".", "*.count")
cntMatrix <- lapply(files, read.delim, header=F, comment.char="_")
cntMatrix <- do.call(cbind, cntMatrix)
rownames(cntMatrix) <- cntMatrix[,1]
cntMatrix <- cntMatrix[,-seq(1, ncol(cntMatrix), 2)]
colnames(cntMatrix) <- sub(".count", "", files)
```

۲.۴.۱ تحلیل تفاوت بیان ژن‌ها

پس از تولید ماتریس شمارش ژن‌ها در هر نمونه با نام cntMatrix، لازم است از اطلاعات مطالعه گروه هر نمونه را تعیین کنیم. اگر فرض کنیم مطالعه ۶ نمونه داشته باشد که به ترتیب نام، سه نمونه اول دارای برچسب YRI، دو نمونه بعد برچسب GBR، و نمونه آخر مجددا برچسب YRI داشته باشند، با قطعه کد زیر برچسب‌ها را به ترتیب به ستون‌ها به عنوان گروهشان داده، تحلیل DEG را بر ماتریس بر اساس گروه تعیین شده انجام می‌دهیم و پس از نرمال‌سازی pvalue حاصل، دو ستون pvalue اصلی و padj ماتریس با نام سطر و ستون‌های آن را در فایلی ذخیره می‌کنیم:

```
grp <- factor(c(rep("YRI",3), rep("GBR",2), "YRI"))
colData <- data.frame(group=grp, type="paired-end")
dataset <- DESeqDataSetFromMatrix(cntMatrix, colData, design=~group)
datasetDE <- DESeq(dataset)
cntNorm <- log2(1+counts(datasetDE, normalized=T))
diffMatrix <- data.frame(results(datasetDE, c("group", "YRI", "GBR")))
diffMatrix$padj <- p.adjust(diffMatrix$pvalue, method="BH")
diffMatrix <- diffMatrix[order(diffMatrix$padj),]
write.table(diffMatrix, file="study_num.DE.txt", quote=F, sep="\tab")
```

آ دانلود داده‌های بیان ژن

داده‌های نمونه خوانش توالی برای بیان ژن‌ها به صورت رایگان از پایگاه داده‌های متفاوت توالی‌های نوکلئوتید شامل NCBI و EBI و DDBJ قابل دانلود هستند که دو مورد اول از پر استفاده‌ترین‌ها می‌باشند. این پایگاه داده‌ها اطلاعات مطالعات را در پایگاه داده فرادادهٔ خود (GEO برای NCBI و ArrayExpress برای EBI) و داده‌های خام را در آرشیو خوانش توالی خود (SRA برای NCBI و ENA برای EBI) ذخیره می‌کنند.

برای دسترسی به این موارد می‌توانید از لینک‌های زیر استفاده کنید:

- **NCBI GEO:** ncbi.nlm.nih.gov/geo/
- **NCBI SRA:** ncbi.nlm.nih.gov/sra/
- **EBI ArrayExpress:** ebi.ac.uk/biostudies/arrayexpress/
- **EBI ENA:** ebi.ac.uk/ena/browser/home/

در این دو پایگاه داده، مطالعات با اعداد دسترسی[۲] با فرمت خاص خود قابل مشاهده هستند. در GEO اعداد دسترسی دارای پیشوندهای GSE (ثبت شده توسط ارئه دهنده) و GDS (باز تنظیم و گردآوری شده توسط تیم GEO) و در ArrayExpress اعداد دسترسی دارای پیشوندهای E-MEXP- و E-TABM- (قدیمی)، E-MTAB- (جدید) و E-GEOD- (وارد شده از GEO) می‌باشند. برای مثال یک مطالعه با عدد دسترسی GSE60939 در GEO، دارای عدد دسترسی E-GEOD-60939 در ArrayExpress می‌باشد.

با جستجوی عدد دسترسی خود در پایگاه فراداده مربوطه می‌توانید اطلاعات مطالعه از جمله جاندار مورد بررسی، وضعیت، چگونگی بررسی، لیست نمونه‌ها و حتی چگونگی پردازش داده‌ها را مشاهده کنید. به طور معمول در صفحه فراداده مطالعه، داده‌های پردازش شده متنی و یا لینک‌هایی به داده‌های خام از جنس تکنولوژی مربوطه موجود است. برای راهنمای جزئی‌تر کار با GEO از این لینک کمک بگیرید.

نکته

در صورت عدم وجود لینک به داده خام مطالعه، می‌توانید عدد دسترسی داده نمونه‌ها که در صفحه فراداده مطالعه ذکر شده است را در آرشیو خوانش توالی پایگاه داده مربوطه، جستجو کنید. برای نمونه تکی، این عدد دارای پیشوند GSM، SRX، SRR، SRS و SAMN (منبع NCBI) یا ERR، ERX، ERS و SAMEA (منبع EBI) بوده و برای تمام نمونه‌های مطالعه، دارای پیشوند PRJNA و SRP (منبع NCBI) یا PRJEB و ERP (منبع EBI) می‌باشد. از آنجا که پایگاه داده‌های مذکور با هم همگام می‌شوند، هر یک از این اعداد دسترسی به نمونه‌ها، در هر دو پایگاه داده کار می‌کنند.

[2] Accession Number

ب منابع

1. Korpelainen, E., Tuimala, J., Somervuo, P., Huss, M., and Wong, G. (2014). RNA-Seq data analysis. In Chapman and Hall/CRC eBooks.

   https://doi.org/10.1201/b17457

2. Sharifi Zarchi, Ali. Advanced Bioinformatics Course. Maktabkhooneh.

   https://maktabkhooneh.org/course/بیوانفورماتیک_پیشرفته-mk375/