

Analysis of Next Generation Sequencing Data

Alignment of NGS data

Luce Skrabanek

11 February, 2020

1 Running STAR

STAR is a fast and accurate splice-aware aligner. To use STAR on our systems:

```
1 spack load star@2.7.0e
```

Before running STAR to align your sample to a genome, you must first create the genome index, which will create the suffix array, and related indices. Note that the directory where you will store the index (`--genomeDir`) must already exist. You only need to do this once per combination of genome/annotation file. We have done this already for you. The genome sequence was downloaded from the UCSC Genome Browser (<http://hgdownload.soe.ucsc.edu/goldenPath/sacCer3/bigZips/sacCer3.2bit>), and the annotation was downloaded via UCSC's Table Browser, selecting SGD genes, and using the GTF format.

```
1 STAR --runMode genomeGenerate \  
2     --runThreadN 1 \  
3     --genomeDir sacCer3_STARindex \  
4     --genomeFastaFiles sacCer3.fa \  
5     --sjdbGTFfile sacCer3.sgd.gtf \  
6     --sjdbOverhang 99
```

Each sample can now be aligned to this index. Make sure you have requested enough memory to run this command!

```
1 STAR --runMode alignReads \  
2     --runThreadN 1 \  
3     --genomeDir referenceGenomes/sacCer3_STARindex \  
4     --readFilesIn gierlinski/fastq/ERR458493.fastq.gz \  
5     --readFilesCommand zcat \  
6     --outFileNamePrefix gierlinski/alignments/ERR458493. \  
7     --outSAMtype BAM SortedByCoordinate
```

If you are writing your results to a directory, that directory must already exist. STAR has many options, and you will likely have to tweak some parameters to best suit your analysis.

Some commonly modified parameters include:

1. `--outFilterMultimapNmax` : max number of multiple alignments allowed for a read: if exceeded, the read is considered unmapped
2. `--alignIntronMin` : minimum intron length
3. `--alignIntronMax` : maximum intron length
4. `--outSAMattributes` : specifies which information to include in the optional SAM attribute field. Can include any of: NH HI NM MD AS nM jM jI XS

The `SJ.out.tab` file contains the list of novel splice junctions identified by STAR. These splice junctions can be used as input to further STAR runs (e.g., other samples in the same study) with the `--sjdbFileChrStartEnd` option. The authors recommend to run all samples in a study through STAR once, to get novel junctions for each sample, and then do a second pass, incorporating information on all the new junctions for all samples, again.

2 Running BWA

To access BWA on our systems, use

```
1 spack load bwa@0.7.15%gcc@6.3.0
```

The BWA `sacCer3` genome was indexed with the following command. As for STAR, this only needs to be done per genome.

```
1 bwa index -p sacCer3_BWAindex/sacCer3 sacCer3.fa
```

Samples can now be mapped against the genome using:

```
1 bwa mem referenceGenomes/sacCer3_BWAindex/sacCer3 gierlinski/fastq/
  ERR458493.fastq.gz > gierlinski/alignments/ERR458493.bwa.sam
```

Note that BWA outputs a SAM file. It is strongly recommended to convert this to a BAM file (see below).

Some common options for `bwa mem` include:

1. `-M` : mark shorter split reads as secondary [make Picard-compatible]
2. `-h 100` : output up to 100 alternative alignments, if their scores are >80% of the max score
3. `-a` : if there are alternative alignments, don't output the CIGAR string of the alternates in the OPT field, instead output each as a separate alignment (gives more information, including alignment score for alternates)
4. `-L 50,50` : penalizes 5'- and 3'-clipping (encourages alignments to just end, rather than be clipped)
5. `-O 7` : increase gap open penalty slightly, but not so much as to prevent including appropriate gaps

3 Exploring SAM files with samtools

The most commonly used tool to access, view, sort and manipulate the SAM/BAM files that contain the aligned reads is `samtools`.

```
1 spack load samtools@1.9%gcc@6.3.0
2
3 samtools view -b ERR458493.bwa.sam -o ERR458493.bwa.bam
4 rm ERR458493.bwa.sam
5 samtools sort ERR458493.bwa.bam -o ERR458493.bwa.sorted.bam
6
7 samtools view -b ERR458493.bwa.sam | samtools sort -o ERR458493.bwa.
  sorted.bam
8
9 samtools view -h ERR458493.Aligned.sortedByCoord.out.bam
```

Samtools also contains a few basic QC tools:

```
1 samtools stats ERR458493.Aligned.sortedByCoord.out.bam > ERR458493.
  stats
2 samtools flagstat ERR458493.Aligned.sortedByCoord.out.bam > ERR458493.
  flagstats
```

4 MultiQC

MultiQC is a handy tool that can be used to aggregate and visualize all the descriptive and QC information about a set of samples. MultiQC searches a given directory for analysis logs from a wide variety of NGS tools (73 at last count), and compiles them into a single HTML report.

We have run FastQC, TrimGalore, STAR and samtools flagstat on our samples, and MultiQC will recognize all of these.

```
1 spack load -r py-multiqc
2 multiqc -n gierlinski.multiqc.html .
```

5 A real script

Let's write a script that will:

1. generate a list of WT_1 and SNF2_1 replicates
2. download the files from ENA or SRA [check to see if the files were already downloaded]
3. run FastQC on each file
4. run a basic QC on the output from FastQC
5. map each sample to the genome using STAR [maybe clean up after STAR - remove `..STARtmp` directories]
6. use samtools to pluck out reads that were uniquely mapped

Remember that a script always begins with a shebang line, indicating the shell that should be used to run the commands.

To make your script executable, use the `chmod +x` command.

```

1  #!/bin/bash
2
3  # Usage: fastq2bam.bash <fastq_dir> <fastqc_dir> <alignment_dir>
4
5  # Check that we have our command line argument(s)
6  arg_count=$#
7  if [ $arg_count -lt 3 ]; then
8      echo "Not enough command line arguments. Exiting ..."
9      echo "Usage: fastq2bam.bash <fastq_dir> <fastqc_dir> <alignment_dir>"
10     exit
11 fi
12
13 # Read arguments from command line
14 # Could check here if these directories exist!
15 fastq_dir=$1
16 fastqc_dir=$2
17 alignment_dir=$3
18
19 # Check that we have the files we need to pluck out the sample IDs and URLs
20 if [ ! -r ERP004763_sample_mapping.tsv ]; then
21     echo "Cannot find file with sample IDs (expecting ERP004763_sample_mapping.tsv)"
22     echo "Exiting ..."
23     exit
24 fi
25 if [ ! -r PRJEB5348.txt ]; then
26     echo "Cannot find file with sample URLs (expecting PRJEB5348.txt)"
27     echo "Exiting ..."
28     exit
29 fi
30
31 # Load packages that we will need
32 spack load fastqc
33 spack load star@2.7.0e
34 spack load samtools@1.9%gcc@6.3.0
35
36 # Extract the sample IDs for WT replicate 1
37 wt1=$(cat ERP004763_sample_mapping.tsv | egrep "WT" | egrep "\b1$" | cut -f 1)
38
39 # Extract the sample IDs for SNF2 replicate 1
40 snf1=$(cat ERP004763_sample_mapping.tsv | egrep "SNF2" | egrep "\b1$" | cut -f 1)
41
42 # Process each sample:
43 #   - download the fastq.gz from ENA
44 #   - run FastQC on the sample
45 #   - run a basic QC on the FastQC output
46 #   - align each sample using STAR
47 #   - count all uniquely mapping reads
48 for sample in `echo $wt1 $snf1`; do
49     echo "-----"
50     echo "Now processing sample ${sample}"
51
52     # Download the fastq.gz files associated with those IDs
53     # Get the URLs for those samples from the data file PRJEB5348.txt
54     # Only download them if we don't already have them
55     # The -P option for wget allows you to specify a download directory
56     if [ ! -r ${fastq_dir}/${sample}.fastq.gz ]; then
57         url=$(egrep $sample PRJEB5348.txt | cut -f 7)
58         wget -P ${fastq_dir} $url
59     fi
60
61     # Run FastQC, if not already present
62     if [ ! -d ${fastqc_dir}/${sample}_fastqc ]; then
63         fastqc ${fastq_dir}/${sample}.fastq.gz --extract --outdir ${fastqc_dir}
64     fi
65
66     # Some basic QC on the FastQC result
67     egrep "Total Sequences" ${fastqc_dir}/${sample}_fastqc/fastqc_data.txt
68     egrep "Adapter Content" ${fastqc_dir}/${sample}_fastqc/summary.txt
69     egrep "(FAIL|WARN)" ${fastqc_dir}/${sample}_fastqc/summary.txt
70     echo
71
72     # Run STAR, if result not already present (should really check if genome directory exists)
73     if [ ! -r ${alignment_dir}/${sample}.Aligned.sortedByCoord.out.bam ]; then
74         STAR --runMode alignReads \
75             --genomeDir /luce/angsd/referenceGenomes/sacCer3_STARindex \
76             --readFilesIn ${fastq_dir}/${sample}.fastq.gz \
77             --readFilesCommand zcat \
78             --outFileNamePrefix ${alignment_dir}/${sample}. \
79             --outSAMtype BAM SortedByCoordinate
80     fi
81
82     # Index the BAM file
83     samtools index ${alignment_dir}/${sample}.Aligned.sortedByCoord.out.bam
84
85     # How many uniquely mapped reads were there?
86     echo "Number of uniquely mapped reads: " $(samtools view -c -q10 ${alignment_dir}/${sample}.Aligned.sortedByCoord.out.bam)
87
88     echo
89 done

```