Analysis of Next Generation Sequencing Data
Alignment of NGS data

Luce Skrabanek

5 February, 2019

1 Running STAR

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

```
spack load star@2.6.1a
```

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.

```
STAR --runMode genomeGenerate \
     --runThreadN 8 \
     --genomeDir sacCer3_STARindex \
     --genomeFastaFiles sacCer3.fa \
     --sjdbGTFfile sacCer3.sgd.gtf \
     --sjdbOverhang 99
```

Each sample can now be aligned to this index.

```
STAR --runMode alignReads \
     --runThreadN 8 \
     --genomeDir referenceGenomes/sacCer3_STARindex \
     --readFilesIn gierlinski/fastq/ERR458878.fastq.gz \
     --readFilesCommand zcat \
     --outFileNamePrefix gierlinski/alignments/ERR458878. \
     --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

```
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.

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

Samples can now be mapped against the genome using:

```
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 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`.

```
spack load samtools@1.9%gcc@6.3.0

samtools view -b ERR458493.bwa.sam -o ERR458493.bwa.bam
rm ERR458493.bwa.sam
samtools sort ERR458493.bwa.bam -o ERR458493.bwa.sorted.bam

samtools view -h ERR458493.Aligned.sortedByCoord.out.bam
```

Samtools also contains a few basic QC tools:

```
samtools stats ERR458493.Aligned.sortedByCoord.out.bam > ERR458493.stats
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.

```
spack load -r py-multiqc
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.

```bash
#! /bin/bash

# Usage: fastq2bam.bash <fastq_dir> <fastqc_dir> <alignment_dir>

# Check that we have our command line argument(s)
arg_count=$#
if [ $arg_count -lt 3 ]; then
  echo "Not enough command line arguments. Exiting ..."
  echo "Usage: fastq2bam.bash <fastq_dir> <fastqc_dir> <alignment_dir>"
  exit
fi

# Read arguments from command line
# Could check here if these directories exist!
fastq_dir=$1
fastqc_dir=$2
alignment_dir=$3

# Check that we have the files we need to pluck out the sample IDs and URLs
if [ ! -r ERP004763_sample_mapping.tsv ]; then
  echo "Cannot find file with sample IDs (expecting ERP004763_sample_mapping.tsv)"
  echo "Exiting ... "
  exit
fi
if [ ! -r PRJEB5348.txt ]; then
  echo "Cannot find file with sample URLs (expecting PRJEB5348.txt)"
  echo "Exiting ... "
  exit
fi

# Load packages that we will need
spack load fastqc
spack load star@2.6.1a
spack load samtools@1.9%gcc@6.3.0

# Extract the sample IDs for WT replicate 1
wt1=$(cat ERP004763_sample_mapping.tsv | egrep "WT" | egrep "\b1$" | cut -f 1)

# Extract the sample IDs for SNF2 replicate 1
snf1=$(cat ERP004763_sample_mapping.tsv | egrep "SNF2" | egrep "\b1$" | cut -f 1)

# Process each sample:
#    - download the fastq.gz from ENA
#    - run FastQC on the sample
#    - run a basic QC on the FastQC output
#    - align each sample using STAR
#    - count all uniquely mapping reads
for sample in `echo $wt1 $snf1`; do
  echo "------------------------------"
  echo "Now processing sample ${sample}"

  # Download the fastq.gz files associated with those IDs
  # Get the URLs for those samples from the data file PRJEB5348.txt
  # Only download them if we don't already have them
  # The -P option for wget allows you to specify a download directory
  if [ ! -r ${fastq_dir}/${sample}.fastq.gz ]; then
    url=$(egrep $sample PRJEB5348.txt | cut -f 11)
    wget -P ${fastq_dir} $url
  fi

  # Run FastQC, if not already present
  if [ ! -d ${fastqc_dir}/${sample}_fastqc ]; then
    fastqc ${fastq_dir}/${sample}.fastq.gz --extract --outdir $fastqc_dir
  fi

  # Some basic QC on the FastQC result
  egrep "Total Sequences"  ${fastqc_dir}/${sample}_fastqc/fastqc_data.txt
  egrep "Adapter Content"  ${fastqc_dir}/${sample}_fastqc/summary.txt
  egrep "(FAIL|WARN)"      ${fastqc_dir}/${sample}_fastqc/summary.txt
  echo

  # Run STAR, if result not already present (should really check if genome directory exists)
  if [ ! -r ${alignment_dir}/${sample}.Aligned.sortedByCoord.out.bam ]; then
    STAR --runMode alignReads \
         --genomeDir ~luce/angsd/referenceGenomes/sacCer3_STARindex \
         --readFilesIn ${fastq_dir}/${sample}.fastq.gz \
         --readFilesCommand zcat \
         --outFileNamePrefix ${alignment_dir}/${sample}. \
         --outSAMtype BAM SortedByCoordinate
  fi

  # Index the BAM file
  samtools index ${alignment_dir}/${sample}.Aligned.sortedByCoord.out.bam

  # How many uniquely mapped reads were there?
  echo "Number of uniquely mapped reads: " $(samtools view -c -q10 ${alignment_dir}/${sample}.Aligned.sortedByCoord.out.bam)

  echo
done
```