Published in final edited form as:
Curr Protoc Bioinformatics. ; 51: 11.14.1–11.14.19. doi:10.1002/0471250953.bi1114s51.

Mapping RNA-seq Reads with STAR

Alexander Dobin and **Thomas R. Gingeras**
Cold Spring Harbor Laboratory, Cold Spring Harbor NY 11746

Alexander Dobin: dobin@cshl.edu; Thomas R. Gingeras: gingeras@cshl.edu

Abstract

Mapping of large sets of high-throughput sequencing reads to a reference genome is one of the foundational steps in RNA-seq data analysis. The STAR software package performs this task with high levels of accuracy and speed. In addition to detecting annotated and novel splice junctions, STAR is capable of discovering more complex RNA sequence arrangements, such as chimeric and circular RNA. STAR can align spliced sequences of any length with moderate error rates providing scalability for emerging sequencing technologies. STAR generates output files that can be used for many downstream analyses such as transcript/gene expression quantification, differential gene expression, novel isoform reconstruction, signal visualization, and so forth. In this unit we describe computational protocols that produce various output files, use different RNA-seq datatypes, and utilize different mapping strategies. STAR is Open Source software that can be run on Unix, Linux or Mac OS X systems.

Keywords

sequence alignment; reads mapping; RNA-seq; transcriptome; spliced alignment; STAR

Introduction

Recent advances in the high-throughput sequencing have made sequencing of RNA transcripts (RNA-seq) an attractive tool for the studies of the transcriptome at single nucleotide resolution. One of the foundational steps in the RNA-seq data analysis is mapping (alignment) of the large sets of sequenced reads to a reference genome. This task presents more challenges than alignment of genomic DNA reads because RNA sequences are often spliced, i.e. derived from the non-contiguous regions of the genome.

The STAR (Dobin et al, 2013) software package enables highly accurate and ultra-fast alignment of RNA-seq reads to a reference genome. In addition to detecting of annotated and novel splice junctions, STAR is capable of discovering more complex RNA sequence

Internet Resources

https://github.com/alexdobin/STAR

GitHub STAR code repository: the best place to obtain the latest versions of the source code, executables and documentation.

https://groups.google.com/forum/#!forum/rna-star

STAR user discussion group: the best place to ask questions and obtain help from the authors and other users.

https://github.com/alexdobin/STAR/raw/master/doc/STARmanual.pdf

STAR manual containing detailed information on all options, output and formatting. This file is also a part of STAR source code distribution.

arrangements, such as chimeric and circular RNA. STAR can align spliced sequences of any length with moderate error rates providing scalability for emerging sequencing technologies. STAR generates output files that can be used for many downstream analyses such as transcript/gene expression quantification, differential gene expression, novel isoform reconstruction, signal visualization, and so forth.

In this unit we describe computational protocols that produce various output files, use different RNA-seq datatypes, and utilize different mapping strategies. Most of the options described in these Protocols can be combined to generate all the necessary output files in a single STAR job.

- The Basic Protocol describes the regular mapping job using a real RNA-seq dataset as an example.
- Alternate Protocol 1 shows how to generate the genome indices required for all mapping jobs.
- Alternate Protocol 2 describes a more advanced 2-pass mapping strategy for a more accurate spliced alignment to novel junctions.
- Alternate Protocol 3 explains how to output STAR alignments in unsorted and coordinate-sorted BAM formats.
- Alternate Protocols 4 and 5 show how to generate signal files which can be visualized in genome browsers for stranded (un-stranded) RNA-seq data.
- Alternate Protocol 6 explains how to make STAR detect and output chimeric (fusion) alignments.
- Alternate Protocol 7 describes how to output alignments in transcriptomic coordinates and use RSEM to quantify expression of transcripts and genes.
- Alternate Protocols 8 and 9 show how to use quantify transcripts and genes with Cufflinks using STAR alignments for stranded (un-stranded) RNA-seq data.
- Support Protocol 1 shows how to download and install STAR.
- Support Protocol 2 shows how to download pre-built genome indices.

Basic Protocol: Mapping RNA-seq reads to the reference genome

RNA-seq data from the Next Generation Sequencing platforms such as Illumina or Ion Torrent consists of millions of relatively short sequences (“reads”) representing fragments of the original RNA molecules. The Basic Protocol performs the most common analysis task - alignment of the reads to the references genome. These alignments serve as a basis for many types of downstream analysis, such as calculating transcript /gene expression, differential gene expression, detection of novel splice junctions and isoforms, signal visualization in the genomic browsers, and so forth.

The Basic Protocol describes mapping RNA-seq to the reference genome with only the essential options. The genome indices are assumed to have been already generated (see Alternate Protocol 1) or downloaded (see Support Protocol 4). There are multiple parameters that control the STAR execution, with the most important described in the Commentary/ Critical Parameters section, while a full list of options can be found in the STAR manual.

The Basic Protocol uses transcript/gene annotations in GTF format. The gene annotations allow STAR to identify and correctly map spliced alignments across known splice junctions. While it is possible to run the mapping jobs without annotations, it is not recommended. When gene annotations are not available, use the 2-pass mapping described in Alternate Protocol 2.

Necessary Resources

Hardware

- A computer with Unix, Linux or Mac OS X operating systems.
- RAM requirements: at least 10 x GenomeSize bytes. For instance, human genome of ~3 GigaBases will require ~30 GigaBytes of RAM. 32GB is recommended for human genome alignments.
- Sufficient free disk space (>100 GigaBytes) for storing output files.

STAR jobs can be run on multiple execution threads which significantly increases mapping throughput. The number of STAR threads is selected with --runThreadN <Ntrheads> in all Protocols. Typically, this Nthreads is chosen to be equal the number of physical processors (cores). If other processes are running in parallel with STAR, this number may need to be reduced. On some systems with efficient hyper-threading the mapping speed can be further increased by increasing Nthread to up to twice number of physical cores.

Software—STAR 2.4.1a was used in this example. The latest release of STAR software from https://github.com/alexdobin/STAR/releases is recommended for use in production.

Input files—Download an annotation GTF file and unzip it:

```
cd ~/star
wget ftp://ftp.ensembl.org/pub/release-79/gtf/homo_sapiens/
Homo_sapiens.GRCh38.79.gtf.gz
gunzip Homo_sapiens.GRCh38.79.gtf.gz
```

We will use RNA-seq data from the ENCODE project (105M 2x101 Illumina reads from GM12878 cell line, stranded "dUTP" protocol on total RNA). Downloaded two files ("read 1" and "read 2") into the ~/star directory:

```
cd ~/star
wget https://www.encodeproject.org/files/ENCFF001RFH/@@download/
ENCFF001RFH.fastq.gz -O
ENCFF001RFH.fastq.gz wget https://www.encodeproject.org/files/ENCFF001RFH/
```

```
@@download/ENCFF001RFG.fastq.gz -O
ENCFF001RFG.fastq.gz
```

Running a STAR mapping job: 1. Make a "run directory" for the "Basic" protocol and switch to it:

```
mkdir ~/star/basic
cd ~/star/basic
```

2. Map the gzipped FASTQ files located in the ~/star/ directory (see Input Files):

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR \
--runThreadN 12 --genomeDir ~/star/genome/ \
--sjdbGTFfile ~/star/Homo_sapiens.GRCh38.79.gtf --sjdbOverhang 100 \
--readFilesIn ~/star/ENCFF001RFH.fastq.gz ~/star/ENCFF001RFG.fastq.gz \
--readFilesCommand zcat
```

Options:

i. If the input FASTQ files have been previously uncompressed, remove --readFilesCommand zcat option.

ii. It is possible to run mapping jobs without annotations by removing --sjdbGTFfile~/star/Homo_sapiens.GRCh38.79.gtf --sjdbOverhang 100 options, however, this is not recommended. In the absence of annotations, use the 2-pass mapping described in Alternate Protocol 2.

iii. For paired-end data, the FASTQ files for read-1 and read-2 have to be specified in the --sjdbGTFfile option separated by a space, while for single-end data only one FASTQ file needs to be specified.

Note that the \ characters at the end of each line in the example are used to continue the command across multiple lines. [*Copy Editor: The chapter editor added this annotation; please inform the authors.]

3. While STAR is running, the status messages will be appearing on the screen, e.g.:

```
Mar 31 01:34:01 ..... Started STAR run
Mar 31 01:34:49 ..... Finished GTF processing
Mar 31 01:37:10 ..... Finished inserting 1st pass junctions into genome
Mar 31 01:37:10 ..... Started mapping
Mar 31 01:54:53 ..... Finished successfully
```

4. While STAR is running, the progress of the mapping job can be checked in the Log.progress.out file in the run directory. This file is updated every minute and shows the

number of reads that have been processed, and various mapping statistics. This is useful for initial quality control during the mapping job:

```
cat Log.progress.out
         Time Speed     Read  Read Mapped Mapped Mapped Mapped Unmapped
Unmapped Unmapped Unmapped
               M/hr    number length unique   length MMrate   multi    multi
+      MM     short      other
Mar 31 01:38:12 299.7   5161748   202  92.2%   201.0   0.3%   6.0%
0.1%     0.0%     1.7%      0.0%
Mar 31 01:39:12 356.2  12069587   202  92.2%   200.9   0.3%   6.0%
0.1%     0.0%     1.7%      0.0%
Mar 31 01:40:13 347.7  17674136   202  92.2%   200.9   0.3%   6.0%
0.1%     0.0%     1.7%      0.0%
Mar 31 01:41:14 345.2  23395592   202  92.2%   200.9   0.3%   6.0%
0.1%     0.0%     1.7%      0.0%
Mar 31 01:42:19 344.7  29583868   202  92.2%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:43:22 354.1  36589980   202  92.2%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:44:22 355.5  42661955   202  92.2%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:45:23 355.9  48733492   202  92.2%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:46:24 356.1  54805231   202  92.2%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:47:24 352.1  60059430   202  92.1%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:48:24 356.3  66715156   202  92.1%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:49:29 356.9  73254143   202  92.1%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:50:33 356.7  79559306   202  92.1%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:51:33 358.7  85981408   202  92.1%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:52:37 356.1  91703001   202  92.1%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:53:37 356.2  97657922   202  92.1%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
Mar 31 01:54:38 356.7 103831542   202  92.1%   200.9   0.3%   6.0%
0.1%     0.0%     1.8%      0.0%
ALL DONE!
```

5. After the job is finished, the following files will be generated in the run directory:

```
ls -sh1
 71G  Aligned.out.sam
 4.0K  Log.final.out
 50M  Log.out
 4.0K  Log.progress.out
8.5M  SJ.out.tab
```

Log.final.out contains the summary mapping statistics of the run. This file is described in more detail in the “Guidelines for Understanding Results / Mapping Statistics” section.

Log.out contains various run-time information and messages, and is typically used for debugging.

Aligned.out.sam is the main output file containing read alignments in the SAM format (Li et al., 2009). The SAM format specifications are described here: http://samtools.github.io/hts-specs/SAMv1.pdf. “Samtools” utilities http://www.htslib.org/doc/samtools-1.2.html provide means of manipulating the SAM files. Converting SAM into compressed binary BAM file with samtools is described in Support Protocol 1.

SJ.out.tab contains high confidence collapsed splice junctions in tab-delimited format (see STAR manual for details).

Alternate Protocol 1: Generating Genome Indices

This protocol describes generating the genome indices using the genome FASTA file as input. The genome indices are required for all type of mapping jobs. This step needs to be performed only once for each genome assembly. The resulting files are saved in a user-specified directory, and can be re-used for mapping different samples to the same genome.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Basic Protocol.

Input files—In the examples below we will be using ENSEMBL genome sequence and annotations. Download and unzip these files (~/ is the user home directory):

```
mkdir -p ~/star/genome
cd ~/star/genome
wget ftp://ftp.ensembl.org/pub/release-79/fasta/homo_sapiens/dna/
Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz
gunzip Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz
```

Generating Genome Indices: 1. Switch to genome directory where genome sequence FASTA file is stored (see Input Files).

```
cd ~/star/genome
```

2. Run STAR to generate genome indices specifying correct path to the genome FASTA and annotations GTF file:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR \
--runThreadN 12 --runMode genomeGenerate --genomeDir ./ \
--genomeFastaFiles ./Homo_sapiens.GRCh38.dna.primary_assembly.fa
```

3. While STAR is running, the status messages will be appearing on the screen (running time will vary depending on the reads lengths and number, genome size, number of threads etc.):

```
Mar 30 23:28:52 ..... Started STAR run
Mar 30 23:28:52 ... Starting to generate Genome files
Mar 30 23:29:56 ... starting to sort Suffix Array. This may take a long
time...
Mar 30 23:30:17 ... sorting Suffix Array chunks and saving them to disk...
Mar 30 23:46:51 ... loading chunks from disk, packing SA...
Mar 30 23:48:57 ... writing Suffix Array to disk ...
Mar 30 23:49:20 ... Finished generating suffix array
Mar 30 23:49:20 ... starting to generate Suffix Array index...
Mar 31 00:09:28 ... writing SAindex to disk
Mar 31 00:09:29 ..... Finished successfully
```

4. Inspect the generated files (sizes will vary depending on the genome size):

```
ls -sh1
4.0K chrLength.txt
4.0K chrNameLength.txt
4.0K chrName.txt
4.0K chrStart.txt
3.0G Genome
4.0K genomeParameters.txt
1.2G Homo_sapiens.GRCh38.79.gtf
3.0G Homo_sapiens.GRCh38.dna.primary_assembly.fa
 40K Log.out
 23G SA
1.5G SAindex
```

Alternate Protocol 2: Mapping RNA-seq reads with 2-pass procedure

In the 2-pass mapping job, STAR will map the reads twice. In the 1st pass, the novel junctions will be detected and inserted into the genome indices. In the 2nd pass, all reads will be re-mapped using annotated (from the GTF file) and novel (detected in the 1st pass) junctions. While this procedure doubles the run-time, it significantly increases sensitivity to novel splice junctions. In the absence of annotations, this option is strongly recommended.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Basic Protocol.

Input files—Same as in the Basic Protocol.

Running a 2-step mapping job: 1. Make a “run directory” for the “Basic” protocol and switch to it:

```
mkdir ~/star/alt_2step
cd ~/star/alt_2step
```

2. Map the gzipped FASTQ files located in the ~/star/ directory (see Input Files):

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR \
--runThreadN 12 --genomeDir ~/star/genome/ \
--sjdbGTFfile ~/star/Homo_sapiens.GRCh38.79.gtf --sjdbOverhang 100 \
--readFilesIn ~/star/ENCFF001RFH.fastq.gz ~/star/ENCFF001RFG.fastq.gz --
readFilesCommand zcat \
--twopassMode Basic
```

The --twopassMode Basic option on the command line activates the 2-pass procedure.

Option: it is possible to run mapping jobs without annotations (e.g. in case no annotations are available) by removing --sjdbGTFfile~/star/Homo_sapiens.GRCh38.79.gtf --sjdbOverhang 100 options.

The output files will be generated for the 2nd (final) pass only, as described in the Basic Protocol 4-5. Note that junctions detected in the 1st pass will be considered annotated in the 2nd pass.

Alternate Protocol 3: Mapping reads and generating unsorted and coordinate-sorted BAM files

Basic Protocol outputs alignments in the SAM format. In many downstream analyses the binary format BAM is utilized, often in the coordinate-sorted mode. BAM files contain the same information as the SAM files, but are much smaller in size due to compression and thus are more suitable for long range storage. While it is possible to convert SAM to BAM using samtools (see Support Protocol 4), it is a computationally intensive task. STAR allows for BAM conversion and coordinate-sorting while mapping, significantly reducing the time required for sorting/conversion.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Basic Protocol.

Input files—Same as in the Basic Protocol.

Running a mapping job with BAM output: 1. Make a “run directory” for the “Basic” protocol and switch to it:

```
mkdir ~/star/alt_bam
cd ~/star/alt_bam
```

2. Map the FASTQ files located in the ~/star/ directory (see Input Files) outputting unsorted and coordinate-sorted BAMs:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR \
--runThreadN 12 --genomeDir ~/star/genome/ \
--sjdbGTFfile ~/star/Homo_sapiens.GRCh38.79.gtf --sjdbOverhang 100 \
--readFilesIn ~/star/ENCFF001RFH.fastq.gz ~/star/ENCFF001RFG.fastq.gz --
readFilesCommand zcat \
--outSAMtype BAM SortedByCoordinate Unsorted
```

The --outSAMtype BAM Unsorted SortedByCoordinate option activates the output to unsorted and sorted BAM files.

Options:

> The Unsorted and SortedByCoordinate options can be used separately to generate just one type of the BAM file, e.g.:

```
--outSAMtype BAM Unsorted
--outSAMtype BAM SortedByCoordinate
```

3. All the output files are the same as in Basic Protocol 4-5, with exception of Aligned.out.sam. Instead of it STAR will generate unsorted and coordinate-sorted BAM files:

```
23G Aligned.out.bam
17G Aligned.sortedByCoord.out.bam
```

Alternate Protocol 4: Generating signal files for visualization on genome browsers for stranded RNA-seq data

The RNA-seq “signal” across the genome is calculated as the number of reads crossing (i.e. mapped to) each genomic position (nucleotide). This signal can be visualized in genomic browser such as UCSC genomic browser (http://genome.ucsc.edu/) or IGV browser (https://www.broadinstitute.org/igv/). STAR can calculate signal files starting from the coordinate-sorted BAM file Aligned.sortedByCoord.out.bam generated in the Alternative Protocol 3. This Protocol works with stranded RNA-seq data, such as Illumina stranded Tru-Seq protocol. For un-stranded RNA-seq data see the Alternate Protocol 5.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Basic Protocol.

Input files—Same as in the Basic Protocol.

Generating signal output from the coordinate-sroted BAM file: 1. Follow the Alternative Protocol 3 to produce the Aligned.sortedByCoord.out.bam file. If you do not require an unsorted BAM file, use --outSAMtype BAM SortedByCoordinate option to generate only the coordinate-sorted file.

2. Generate signal files:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR
--runMode inputAlignmentsFromBAM --inputBAMfile
Aligned.sortedByCoord.out.bam \
--outWigType bedGraph --outWigStrand Stranded
```

3. The output signal files:

```
776M Signal.UniqueMultiple.str1.out.bg
809M Signal.UniqueMultiple.str2.out.bg
743M Signal.Unique.str1.out.bg
780M Signal.Unique.str2.out.bg
```

The signal files are in the BedGraph format described here: http://genome.ucsc.edu/goldenpath/help/bedgraph.html

Two sets of files are generated:

.Unique. only includes signal from uniquely mapping reads

.UniqueMultiple. includes signal from both uniquely and multi-mapping reads.

The signal are generated separately for two genomic strands: *.str1.* and *.str2.*, which correspond to different genomic strands depending on the library preparation protocols. For the protocols in which the 1st read is on the opposite strand to the RNA molecule (such as Illumina stranded Tru-Seq), *.str1.* corresponds to the (−) strand and *.str2.* corresponds to the (+) strand.

Alternate Protocol 5: Generating signal files for visualization on genome browsers for un-stranded RNA-seq data

The RNA-seq “signal” across the genome is calculated as the number of reads crossing (i.e. mapped to) each genomic position (nucleotide). This signal can be visualized in genomic browser such as UCSC genomic browser (http://genome.ucsc.edu/) or IGV browser (https://www.broadinstitute.org/igv/). STAR can calculate signal files starting from the coordinate-sorted BAM file Aligned.sortedByCoord.out.bam generated in the Alternative Protocol 3. This Protocol works with un-stranded RNA-seq data, such as Illumina un-stranded Tru-Seq protocol. For stranded RNA-seq data see the Alternate Protocol 4.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Basic Protocol.

Input files—Same as in the Basic Protocol.

Generating signal output from the coordinate-sorted BAM file: 1. Follow the Alternate Protocol 3 to produce the Aligned.sortedByCoord.out.bam file. If you do not require unsorted BAM file, use --outSAMtype BAM SortedByCoordinate option to generate only the coordinate-sorted file.

2. Generate signal files:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR \
--runMode inputAlignmentsFromBAM --inputBAMfile
Aligned.sortedByCoord.out.bam \
--outWigType bedGraph --outWigStrand Untranded
```

3. The output signal files:

```
1.6G Signal.UniqueMultiple.str1.out.bg
1.5G Signal.Unique.str1.out.bg
```

The signal files are in the BedGraph format described here: http://genome.ucsc.edu/goldenpath/help/bedgraph.html

Two sets of files are generated:

.Unique. only includes signal from uniquely mapping reads

.UniqueMultiple. includes signal from both uniquely and multi-mapping reads.

This protocol generates signal collapsed (summed) on two genomic strands.

Alternate Protocol 6: Mapping RNA-seq reads and generating chimeric alignments to detect fusion transcripts and circular RNA

Standard STAR output includes only linear alignments to the genome, i.e. it does not include chimeric (fusion) alignments which include both reads split across chimeric junctions and paired-end reads with non-concordant (chimeric) mate alignments. This Protocol describes how to produce chimeric alignments and output them into separate files.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Basic Protocol.

Input files—Same as in the Basic Protocol.

Running a mapping job with chimeric output: 1. Make a “run directory” for the “Basic” protocol and switch to it:

```
mkdir ~/star/alt_chimeric
cd ~/star/alt_chimeric
```

2. Map the gzipped FASTQ files located in the ~/star/ directory (see Input Files) outputting :

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR \
--runThreadN 12 --genomeDir ~/star/genome/ \
--sjdbGTFfile ~/star/Homo_sapiens.GRCh38.79.gtf --sjdbOverhang 100 \
--readFilesIn ~/star/ENCFF001RFH.fastq.gz ~/star/ENCFF001RFG.fastq.gz --
readFilesCommand zcat \
--chimSegmentMin 20
```

The --chimSegmentMin <N> option with N>0 activates the chimeric output. N is the minimum allowed length for each of the two chimeric segments of a chimeric alignment.

3. All the output files are the same as in Basic Protocol 4-5, with addition of two chimeric files

```
280M Chimeric.out.sam
 36M Chimeric.out.junction
```

Chimeric.out.sam contains chimeric alignments only in the SAM format.

Chimeric.out.junction contains chimeric junctions in STAR-specific format that is described in detail in STAR manual.

Alternate Protocol 7: Mapping RNA-seq reads, generating output in transcriptomic coordinates and using RSEM to quantify expression of transcripts and genes

Quantifying expression of RNA transcripts and genes is one of the most important tasks in the analysis of RNA-seq data. RSEM (Li and Dewey, 2011) is a popular software package capable of quantifying annotated genes and transcripts using RNA-seq data. In this Protocol STAR outputs genomic alignments in transcriptomic coordinates, which are then used by RSEM to produce transcript/gene quantifications.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Basic Protocol, and RSEM 1.2.20. For downloading and installing instructions refer to RSEM web-sites: http://deweylab.biostat.wisc.edu/rsem/ or https://github.com/bli25wisc/RSEM. It is assumed that RSEM is installed in ~/star/code/RSEM-1.2.20

Input files—Same as in the Basic Protocol.

Running a mapping job with transcriptomic output: 1. Make a “run directory” for and switch to it:

```
mkdir ~/star/alt_rsem
cd ~/star/alt_rsem
```

2. Map the gzipped FASTQ files located in the ~/star/ directory (see Input Files) outputting unsorted and coordinate-sorted BAMs:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR \
--runThreadN 12 --genomeDir ~/star/genome/ \
--sjdbGTFfile ~/star/Homo_sapiens.GRCh38.79.gtf --sjdbOverhang 100 \
--readFilesIn ~/star/ENCFF001RFH.fastq.gz ~/star/ENCFF001RFG.fastq.gz --
readFilesCommand zcat \
--quantMode TranscriptomeSAM
```

The ---quantMode TranscriptomeSAM option activates the transcriptomic output.

3. All the output files are the same as in Basic Protocol 4-5 with addition to the BAM file in transcriptomic coordinates:

```
18G Aligned.toTranscriptome.out.bam
```

Note references in this file are annotated transcript sequences, in contrast to genomic SAM/BAM files in which references are genomic sequences (chromosomes).

4. Prepare the RSEM reference files. This step needs to be done only once for given genome assembly (FASTA) and annotations (GTF).

Make a directory to store RSEM reference files:

```
mkdir ~/star/rsem_ref
```

Switch to the RSEM source directory:

```
cd ~/star/code/RSEM-1.2.20
```

Run RSEM to prepare the reference file:

```
./rsem-prepare-reference --gtf ~/star/Homo_sapiens.GRCh38.79.gtf \
~/star/genome/Homo_sapiens.GRCh38.dna.primary_assembly.fa ./ref
```

5. Run RSEM quantification on the STAR transcriptomic BAM file:

```
rsem-calculate-expression --bam --no-bam-output -p 12 --paired-end --forward-
prob 0 \
~/star/alt_rsem/Aligned.toTranscriptome.out.bam ~/star/rsem_ref/ref ~/star/
alt_rsem/Quant \
>& ~/star/alt_rsem/rsem.log
```

--paired-end --forward-prob 0 options are applicable to paired stranded RNA-seq data such as Illumina stranded Tru-seq protocol. Refer to RSEM documentation for detailed description of RSEM parameters.

-p 12 defines the number of threads used by RSEM.

6. RSEM produces the following files with isoform (transcript) and gene expression levels (refer to RSEM documentation for detailed description of the RSEM output format):

```
13M Quant.isoforms.results
5.7M Quant.genes.results
```

Alternate Protocol 8: Mapping RNA-seq reads and running Cufflinks to assemble and quantify transcripts for stranded RNA-seq data

Cufflinks (Trapnell et al 2010) is a popular software package for assembly and quantification of transcript using RNA-seq data. In this protocol STAR outputs BAM file with coordinate-sorted alignments, which is then used by Cufflinks to assemble and quantify novel transcript structures. This Protocol works with stranded RNA-seq data, such as Illumina stranded Tru-Seq protocol. For un-stranded RNA-seq data see the Alternate Protocol 8.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Basic Protocol, plus Cufflinks 2.2.1. To download and install Cufflinks into ~/star/code/directory:

```
cd ~/star/code
wget http://cole-trapnell-lab.github.io/cufflinks/assets/downloads/
cufflinks-2.2.1.Linux_x86_64.tar.gz
tar xvfz cufflinks-2.2.1.Linux_x86_64.tar.gz
```

Input files—Same as in the Basic Protocol.

Generating the coordinate-sorted BAM file and running Cufflinks transcript assembly and quantification: 1. Follow Alternate Protocol 3 to produce the Aligned.sortedByCoord.out.bam file. If you do not require unsorted BAM file, use --outSAMtype BAM SortedByCoordinate option to generate only the coordinate-sorted file.

2. In the STAR run directory ~/star/alt_bam/, run the basic Cufflinks command:

```
cd ~/star/alt_bam
~/star/code/cufflinks-2.2.1.Linux_x86_64/cufflinks -p 12 --library-type fr-
firststrand \
Aligned.sortedByCoord.out.bam
```

-p 12 defines the number of threads used by Cufflinks.

--library-type fr-firststrand parameter has to be chosen according to the RNA-seq library protocol. Use fr-firststrand option for the protocols in which the 1st read is on the opposite strand to the RNA molecule (such as Illumina stranded Tru-Seq). Use fr-secondstrand for the protocols in which the 1st read is on the same strand as the RNA molecule. For un-stranded RNA-seq data use Alternative Protocol 8. For other library types, refer to Cufflinks documentation.

3. Cufflinks generates the following output:

```
89M transcripts.gtf
8.1M genes.fpkm_tracking
11M isoforms.fpkm_tracking
```

The first file contains transcripts assembled from the RNA-seq data, while the 2nd and 3rd files contain tables of gene and transcript-level expression. For a description of output formats, as well as advanced Cufflinks options, please refer to the Cufflinks documentation: http://cole-trapnell-lab.github.io/cufflinks/cufflinks/index.html

Alternate Protocol 9: Mapping RNA-seq reads and running Cufflinks to assemble and quantify transcripts for un-stranded RNA-seq data

Cufflinks (Trapnell et al 2010) is a popular software package for assembly and quantification of transcript using RNA-seq data. In this protocol STAR outputs BAM file with coordinate-sorted alignments, which is then used by Cufflinks to assemble and quantify novel transcript structures. This Protocol works with un-stranded RNA-seq data. For stranded RNA-seq data see the Alternate Protocol 7.

Necessary Resources

Hardware—Same as in the Basic Protocol.

Software—Same as in the Alternate Protocol 7.

Input files—Same as in the Basic Protocol.

Generating the coordinate-sorted BAM file and running Cufflinks transcript assembly and quantification: 1. Make a run directory and switch to it:

```
mkdir ~/star/alt_cuff-unstr
cd ~/star/alt_cuff-unstr
```

2. Map the FASTQ files located in the ~/star/directory (see Input Files) outputting coordinate-sorted BAM:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR\
--runThreadN 12 --genomeDir ~/star/genome/ \
--sjdbGTFfile ~/star/Homo_sapiens.GRCh38.79.gtf --sjdbOverhang 100 \
--readFilesIn ~/star/ENCFF001RFH.fastq.gz ~/star/ENCFF001RFG.fastq.gz --
readFilesCommand zcat \
--outSAMtype BAM SortedByCoordinate Unsorted \
--outSAMstrandField intronMotif
```

--outSAMstrandField intronMotif option adds an XS attribute to the spliced alignments in the BAM file, which is required by Cufflinks for unstranded RNA-seq data.

2. In the same directory run the basic Cufflinks command:

```
~/star/code/cufflinks-2.2.1.Linux_x86_64/cufflinks -p 12
Aligned.sortedByCoord.out.bam
```

-p 12 defines the number of threads used by Cufflinks.

3. Cufflinks output files are described in Alternative Protocol 7.

Support Protocol 1: Downloading and installing STAR

STAR source code and pre-compiled Linux and Mac OS X executables are distributed on GitHub: https://github.com/alexdobin/STAR/releases.

```
mkdir ~/star/code
cd ~/star/code
wget https://github.com/alexdobin/STAR/archive/STAR_2.4.0k.tar.gz
tar xvfz STAR_2.4.0k
```

The STAR executables:

Linux - dynamically linked:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64/STAR
```

Linux - statically linked:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64_static/STAR
```

Mac OS X:

```
~/star/code/STAR-STAR_2.4.0k/bin/Linux_x86_64_static/MacOSX_x86_64/STAR
```

Dynamically linked Linux executable is recommended for general use. The statically linked executable produces exactly the same results, and can be used if dynamic executable has problems with external libraries.

Support Protocol 2: Downloading pre-built genome indices

Pre-built genome indices for multiple species and genome assemblies can be downloaded from http://labshare.cshl.edu/shares/gingeraslab/www-data/dobin/STAR/STARgenomes/, e.g.

```
cd ~/star/
wget -r http://labshare.cshl.edu/shares/gingeraslab/www-data/dobin/STAR/
STARgenomes/ENSEMBL/homo_sapiens/ENSEMBL.homo_sapiens.GRCh38.release-79
```

These directories contain all the necessary genome indices and can be used directly as STAR --genomeDir directories, e.g. --genomeDir ~/star/ ENSEMBL.homo_sapiens.GRCh38.release-79/. Each directory also contains one or more annotations GTF file that can be used in the --sjdbGTFfile option, e.g. --sjdbGTFfile ~/star/ ENSEMBL.homo_sapiens.GRCh38.release-79/Homo_sapiens.GRCh38.79.gtf

Guidelines for Understanding Results

STAR outputs various mapping metrics the Log.final.out file which can be used for to evaluate both the mapping performance and the quality of the RNA-seq library. The Log.final.out file from the Basic Protocol example is as follows:

```
                                                 Started job on  |     Mar
31 01:34:01
                                        Started mapping on  |     Mar 31
01:37:10
                                                     Finished on  |     Mar
31 01:54:53
     Mapping speed, Million of reads per hour  |     355.90
                                   Number of input reads  |     105089150
                              Average input read length  |     202
                                            UNIQUE READS:
                    Uniquely mapped reads number  |     96821769
                             Uniquely mapped reads %  |     92.13%
                                 Average mapped length  |     200.92
                               Number of splices: Total  |     41726121
             Number of splices: Annotated (sjdb)  |     41092890
                           Number of splices: GT/AG  |     41206347
                           Number of splices: GC/AG  |     342550
                            Number of splices: AT/AC  |     42230
```

```
                 Number of splices: Non-canonical  |     134994
                             Mismatch rate per base, %  |     0.29%
                                    Deletion rate per base  |     0.02%
                                 Deletion average length  |     1.57
                                    Insertion rate per base  |     0.01%
                                 Insertion average length  |     1.48
                           MULTI-MAPPING READS:
     Number of reads mapped to multiple loci  |     6264908
               % of reads mapped to multiple loci  |     5.96%
   Number of reads mapped to too many loci  |     62660
             % of reads mapped to too many loci  |     0.06%
                                     UNMAPPED READS:
% of reads unmapped: too many mismatches  |     0.00%
                     % of reads unmapped: too short  |     1.82%
                           % of reads unmapped: other  |     0.02%
```

STAR always treats the "read-1" and "read-2" of the paired-end RNA-seq data as the ends of one paired-end read. Per standard RNA-seq library construction, read-1 and read-2 are the sequences of the ends of a fragment ("insert") of the original RNA molecule. By default, STAR does not allow unpaired alignments (i.e. those with only one read mapped), or non-concordantly mapped pairs (such as chimeric alignments), and these alignments are not counted in the summary statistics. The number of reads, read length and all other metrics refer to paired-end reads rather than separate read-1/ and read-2.

The most important metric is the "Uniquely mapped reads %", or mapping rate, which is defined as a proportion of uniquely mapped reads out of all input reads. For a very good library it exceeds 90%, and for good libraries it should be above 80%. Low mapping rates (<50%) are indicative of a problem with library preparations or data processing:

- *Insufficient depletion of ribosomal RNA.* Ribosomal RNAs comprise more than 90% of all RNA molecules in the cell. In a typical RNA-seq library preparation rRNAs are depleted either by ribo-depletion techniques (customary for total RNA libraries), by poly-A+ selection, or by poly-dT reverse transcription priming. If the rRNA depletion does not work sufficiently well, a substantial number of reads may be emanating from the rRNA transcripts. Most of the rRNA contain multiple highly sequence-similar paralogs, and hence RNA-seq reads will be mapped to multiple loci. Therefore, high percentage (>15%) of multi-mapping reads "% of reads mapped to multiple loci" is indicative of insufficient depletion of rRNA.
- *Poor sequencing quality.* The sequencing error rates can be estimated from the "Mismatch rate per base, %", "Deletion rate per base", "Insertion rate per base". These metrics are typically dominated by sequencing errors, but also include the genotype variants (i.e. sequence differences of individual under study from the consensus genome). For Illumina sequencing, presently the typical mismatch error rate are <0.5% and indel error rates <0.05%. Higher error rates may be indicative of the poor sequencing qualities. Another indicator of poor sequencing quality is a

significant reduction of “Average mapped length” with respect to “Average input read length”. Note that for paired-end data these quantities refer to the total paired-end read length. It is recommended that sequencing quality is assessed by plotting the distribution of “quality scores” from the FASTQ files. Poor sequencing quality is likely to result in increased number of unmapped reads.

- *Exogenous RNA/DNA contamination.* RNA/DNA from divergent species will not map to the reference genome, thus increasing the proportion of unmapped reads. If the sequencing quality of the reads is good (see above), the large “% of reads unmapped: too short” or “% of reads unmapped: other” (>15%) may be indicative of exogenous RNA/DNA contamination. In this case it is recommended to BLAST several of the unmapped reads to NCBI sequence database to identify possible sources of contamination.
- *Computational processing problems.* Usually, computational processing problems result in abortive runs and do not generate mapping statistics output (see the Troubleshooting section). However, in some cases the mapping job will complete successfully yielding extremely low mapping rate (<5%). Some of the typical mistakes include (i) using a wrong species genome indices directory; (ii) using the identical files as read-1 and read-2.

Commentary

Background Information

The key algorithm of STAR alignment strategy is the search for the maximum mappable length of a read, implemented as a speed-efficient suffix array search. Another important part of the algorithm is the split/search/extend algorithm driven by the sequencing quality scores, which allows a precise alignment of reads comprising one or more splice junctions, or a large number of sequencing errors. Some important advantages of the STAR mapping algorithm are briefly described below.

For *de novo* detection of the splice junctions, STAR does not require any previous knowledge of splice junctions’ loci and does not use *a priori* properties of the junctions. This unbiased splice junction mapping is imperative for discovery of novel (including non-canonical) splice junctions and isoforms, as well as other important RNA species such as inter-chromosomal chimeric RNAs. STAR is also capable of utilizing annotations to improve alignment accuracy to the annotated junctions. STAR can align reads of any length, working accurately and efficiently for both long and short RNA molecules. STAR can align reads containing any number of splice junctions, indels and/or mismatches, which is important as the length of the reads rapidly increases with advancements in sequencing technologies. STAR can deal with arbitrarily large intron lengths, which is important for detection of distal exons and chimeric RNA. STAR performs an “auto”-trimming of the poor quality read ends, which are a common occurrence as the read length is pushed to the limit. STAR can detect the non-templated poly-A tails, thus providing a means to determine the transcription termination site for poly-A+ mRNAs. STAR treats the paired-end sequencing in a most straightforward way by incorporating naturally the paired-end information into the mapping process.

Although the STAR algorithm is heuristic and non-exhaustive (i.e. it does not find *all* the possible alignments), it can recover almost all highly probable alignments.

Critical Parameters

The default values of all STAR parameters will work well for mammalian genomes and Illumina RNA-seq 75 to 200nt. The critical parameters described in Table 1 below may need to be tweaked for other species and datatypes, or to achieve required mapping performance. For the paired-end read, STAR always considers the read-1 and read-2 as part of one paired-end read, e.g. the number of mismatched is calculated (and controlled) as a sum from both read-ends (mates); read length is calculated as sum of the lengths of read 1 and 2.

Troubleshooting

Below we describe several categories of frequent computational issues.

STAR parameter/option errors—It is important to carefully check the command line syntax. Copy-pasting parameter names and allowed values from the manual is highly recommended to avoid misspelling. Typical parameter/option errors include misspelling of parameter name or allowed value; duplicate parameter usage; using forbidden values or combinations of values. Usually STAR will catch parameter errors at the beginning of the run and exit with an error/solution message, e.g.:

```
EXITING because of fatal PARAMETERS error: unrecoginzed parameter name
“runThreadn” in input
“Command-Line-Initial”
SOLUTION: use correct parameter name (check the manual)
```

Input file errors—STAR expects the input files to follow standard formatting: FASTA for the reference genome sequence, GTF for annotations, FASTQ or FASTA for input read files. In some cases STAR can identify the problems with the files and exit with an error/solution message, but in many cases formatting problems may result in crashes or nonsense output. For paired-end alignments, the order of the reads in the FASTQ files has to be exactly the same. While this is true for “raw” FASTQ files coming out of the sequencing instruments, this ordering may be broken by pre-mapping processing, such as adapter trimming. Adapter trimming cannot leave reads with 0 sequence length, and has to discard both read 1 and 2 if one of the reads was trimmed to 0 length.

Computing resources problem—Insufficient RAM (“system” or “physical” memory) is the most common source of run-time crashes. In some cases STAR will catch the exception, and exit with a error/solution message. In other cases STAR will exit with an OS message “what(): std::bad_alloc”, or can be simply killed by OS. The required RAM is ~10*GenomeSize bytes for the genome indices, plus per-thread buffer memory (150MB per thread, but this can be reduced with --limitIObufferSize option).

Another common resource problem is the amount of disk space for storing output files. Depending on the number and length of reads, the output file can take 10 to 100 GB of disk space, especially in the uncompressed SAM format. It is important to have sufficient disk space allocated for output files before starting the mapping jobs.

Disk I/O bandwidth (i.e. read/write speed) may become a bottleneck for mapping throughput especially with a large number of threads used. In particular, network file systems with slow interconnects may significantly reduce mapping speed. In these cases it may be more practical run STAR with output on local disk, and only copy the results after the run completes without any problems and yields satisfactory mapping results.

Literature Cited

Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR. STAR: ultrafast universal RNA-seq aligner. Bioinformatics. 2013; 29:15–21. [PubMed: 23104886]

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009; 25:2078–2079. [PubMed: 19505943]

Li B, Dewey CN. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. BMC Bioinformatics. 2011; 12:323. [PubMed: 21816040]

Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, Pachter L. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nature Biotechnology. 2010; 28:511–515.

Table 1

Parameter	Description and allowed values
Filtering of output alignments	
--outFilterType	type of filtering. *Normal [default]:* standard filtering using only current alignment *BySJout:* keep only those reads that contain junctions that passed filtering into SJ.out.tab
--outFilterMultimapScoreRange	the score range below the maximum score for multimapping alignments *[default=1]*
--outFilterMultimapNmax	read alignments will be output only if the read maps is less or equal than this value, otherwise no alignments will be output *[default=10]*
--outFilterMismatchNmax	alignment will be output only if it has <= mismatches than this value *[default=10]*
--outFilterMismatchNoverLmax	same as *--outFilterMismatchNmax*, but normalized to the *mapped* length is than this value *[default=0.3]*
--outFilterMismatchNoverReadLmax	same as *--outFilterMismatchNmax*, but normalized to the full *read* length *[default=1]*
--outFilterScoreMin	alignment will be output only if its alignment score is greater or equal than this value *[default=0]*
--outFilterScoreMinOverLread	same as *--outFilterScoreMin*, but normalized to read length *[default=0.66]*
--outFilterMatchNmin	alignment will be output only if the number of matched bases is greater or equal than this value *[default=0]*
--outFilterMatchNminOverLread	same as *--outFilterMatchNminOverLread*, but normalized to the read length *[default=.066]*
Search sensitivity	
--seedSearchStartLmax	defines the search start point through the read - the read is split into pieces no longer than this value *[default=50]*
Spliced alignments	
--alignIntronMin	minimum intron size: genomic gap is considered intron if its length greater or equal than this value, otherwise it is considered deletion *[default=21]*
--alignIntronMax	maximum intron size. *[default=0]* => max intron size will be determined by (2ˆwinBinNbits)*winAnchorDistNbins
--alignMatesGapMax	maximum gap between two mates, *[default=0]* => max intron gap will be determined by (2ˆwinBinNbits)*winAnchorDistNbins
--alignSJoverhangMin	minimum overhang for spliced alignments *[default=5]*
--alignSJDBoverhangMin	minimum mapped length for a read mate that is spliced *[default=5]*