Sequence Alignment

Contents

1 RNA-seq

Initially, we should go from reads to transcripts. Note that a gene might also have different transcripts.

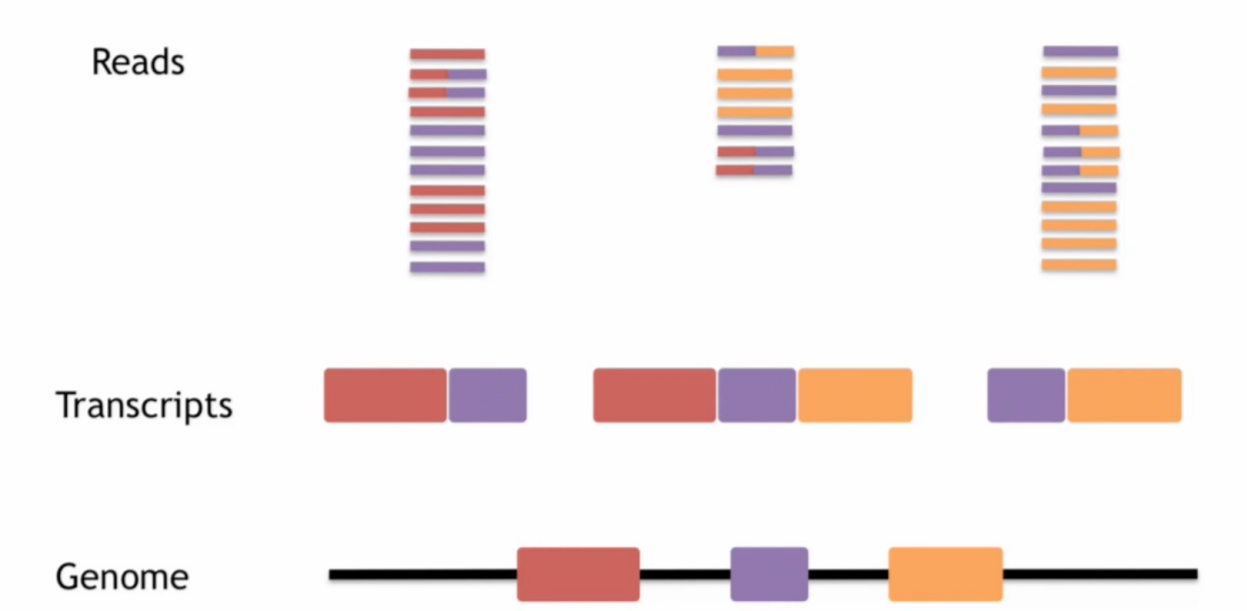

Figure 1: 2-1

Approaches to transcript building:

- Assembly approaches (i.e., starts with FASTQ): Trinity, Velvet/Oases, etc.
- Alignment approaches (to a reference genome): (i.e., starts with SAM/BAM): Cufflinks, Scripture, etc.

1.1 Check reads (FASTQ)

FASTA vs. FASTQ:

- FASTA files: the most common standard for storing reference or consensus sequence data. FASTA only stores sequences.
- FASTQ files: the most common format for storing raw sequence data. FASTQ stores both sequence and associated sequence quality values.

Looking at the reads in a FASTQ file:

```
head Data/SRR1039508.fastq
```

```
## @SRR1039508.12376817 HWI-ST177:290:C0TECACXX:1:2103:1747:96727/1
## CCGCCGCCCGCAGCTGCGGCCCGGGAGCGCGGCGCCCTCCCACTTGTGGGTTTGTGCACTCCC
## +
## DEIIIIDIII8D@D8B3A;'@A453>'',')0:::855<(29<?AAA(+500<(89>AAA>??
## @SRR1039508.11626500 HWI-ST177:290:C0TECACXX:1:2101:10492:130982/1
## ACCCTTACTAACATTAACGAAAATAACCCCACCCTACTAAACCCCATTAAACGCCTGAGAGCC
## +
## HJJJIJJJIIJJJIGH*:CDHJGEHGGHJIGIJGGHIJIJEEHHFEHEFFFFFDDDDC,-;AB
## @SRR1039508.12982779 HWI-ST177:290:C0TECACXX:1:2104:8918:162135/1
## CGAAAGGCTCACTCTCAAGCACGACTAAGAGCCTTCTGAGCCCAGCGACTTCTGAAGGGCCCC
```

1.2 Alignment with STAR

1- Prepare the reference genome FASTA file (only chromosome 1 for this example). Take a peak:

```
head -n 1000 Reference/Homo_sapiens.GRCh38.dna.chromosome.1.fa | tail -20 | less -S
```

```
## TCCTAGAAGCTTTCCAAAGTCATCAGTGTTTCCTAAGAAGGCAGAGAAATCAAACACATG
## GTCTTTTCCTCCAGACAAGCTCCTTTGGGTCATCAGGATTTCTTCAACAATAAAATGTAA
## TAATTCCAAATGTTTGTAACAGAATGGGTAGGACTTTCTTCACTTATTTAAATACTCCCT
## TTTTTATGCAACTGAGTTTTCATCAACAAGTACAAGCTTGTGAAGGAGTACTTTAAAATG
## CAATTTCTCTCTATTTTTGTGGGGGCTAATATTTTATTTCTCATATTGACAATTTATTAT
## GCTGTTTTTAAAAAGTTCATTCATCAAGTATTTCTTGAGCTTTTTCTATGAGACAGGCAC
## TGTTTTAGGCAAGTAATTATGCACTGAACAATGCAAAAAGTTTCCCTGCACTCATGGACT
## TTAATTTTACATTTATGAAAAGCTACAAATATTAGAATAAGTAAAATACTGCCTGGAGGC
## TAAAGCATATTTTGATCACTTATTCCCTAATTCTTTTAGAAGAGAACTCACCTGTCGGTT
## AGCTGAACCACTGCCAGTGATATCCAACTATACATTCAATCCCACCATACCTCATTATCA
## CACCTATTCACTCACAAGCTTAAACTCTTAACTTTTCTCCACATATCAGTGACTATTTCC
## TACAGCTTTTCTTTTACTTTCCATGTTTGCAGTGACAATATACATAAACAGTGTATGAAA
## ACTCAAGTAAAATCTACTCTCTCAGGTGTTCATAATGTATCAATGTATATTGCTTTAAGC
## CTGAAGGTAACCTAAGTAAAGATGTACCATGTTCCACCAATGCTTCTTTTGATCATCATT
## TTATCCTGTTTTTTCTTTAGGATTCTTTCTTATTCCTTCCCCTGACCCTTCTTTTATTCT
## CCAAATTTCTTTCCAATTCATCTTTGTTCTTCCCTTTCCTTTTTACTCTCTTTAAACATT
## CTATGGACTCTGCCTCCTTCACACTGATATTGAACGCCCATAGTTTCATATTTTGGATTG
## CGATTGTTTTATTTTAAAATGGCAAATGTTCATGTTATAAAGAGAATTTTTCAGTCTTTA
## GACTAATAGGTTCATGTAGTTTGGGATTTTCCTCTTTAAGAAAATTAATTATCACTCACA
## CTCCAAGACAAACACCATTTCAGTAGCAATATGAATTTCAGTAGTAATAGGAATCTCCAA
```

2- Prepare the reference genome annotation (gtf) file (only chromosome 1 again). Take a peak:

```
head -n 1 Reference/Homo_sapiens.GRCh38.107.chrom1.gtf | less -S
```

```
## 1    ensembl_havana  gene    1471765 1497848 .   +   .   gene_id "ENSG00000160072"; gene_version "20
```

3- Run the code below to generate the genome index:

```
STAR --runThreadN 12 \
--runMode genomeGenerate \
--genomeSAindexNbases 12 \
--genomeDir Output \
--genomeFastaFiles Reference/Homo_sapiens.GRCh38.dna.chromosome.1.fa \
--sjdbGTFfile Reference/Homo_sapiens.GRCh38.107.chrom1.gtf \
--sjdbOverhang 62
```

```
##  STAR --runThreadN 12 --runMode genomeGenerate --genomeSAindexNbases 12 --genomeDir Output --genomeFa
##  STAR version: 2.7.10a_alpha_220207   compiled:  :/Users/travis/build/alexdobin/travis-tests/STARcomp
## Sep 17 17:47:10 ..... started STAR run
## Sep 17 17:47:10 ... starting to generate Genome files
## Sep 17 17:47:17 ..... processing annotations GTF
## Sep 17 17:47:20 ... starting to sort Suffix Array. This may take a long time...
## Sep 17 17:47:24 ... sorting Suffix Array chunks and saving them to disk...
## Sep 17 17:52:03 ... loading chunks from disk, packing SA...
## Sep 17 17:52:10 ... finished generating suffix array
## Sep 17 17:52:10 ... generating Suffix Array index
## Sep 17 17:52:32 ... completed Suffix Array index
## Sep 17 17:52:32 ..... inserting junctions into the genome indices
## Sep 17 17:53:02 ... writing Genome to disk ...
## Sep 17 17:53:02 ... writing Suffix Array to disk ...
## Sep 17 17:53:06 ... writing SAindex to disk
## Sep 17 17:53:06 ..... finished successfully
```

4- Run the code below to map the reads to the indexed genome:

```
STAR --runThreadN 12 \
--genomeDir Output \
--readFilesIn Data/SRR1039508.fastq

mv Aligned.out.sam Output/Aligned.out.sam
mv Log.final.out Output/Log.final.out
mv Log.progress.out Output/Log.progress.out
mv SJ.out.tab Output/SJ.out.tab
mv Log.out Output/Log.out
```

```
##  STAR --runThreadN 12 --genomeDir Output --readFilesIn Data/SRR1039508.fastq
##  STAR version: 2.7.10a_alpha_220207   compiled:  :/Users/travis/build/alexdobin/travis-tests/STARcomp
## Sep 17 17:53:06 ..... started STAR run
## Sep 17 17:53:06 ..... loading genome
## Sep 17 17:53:10 ..... started mapping
## Sep 17 17:53:44 ..... finished mapping
## Sep 17 17:53:45 ..... finished successfully
```

Check the log file:

```
cat Output/Log.final.out
```

```
##                                 Started job on |    Sep 17 17:53:06
##                              Started mapping on |    Sep 17 17:53:10
##                                     Finished on |    Sep 17 17:53:45
##        Mapping speed, Million of reads per hour |    10.29
##
##                            Number of input reads |    100000
##                        Average input read length |    63
##                                     UNIQUE READS:
##                    Uniquely mapped reads number |    18909
##                         Uniquely mapped reads % |    18.91%
##                           Average mapped length |    62.30
##                        Number of splices: Total |    1958
##             Number of splices: Annotated (sjdb) |    1867
##                        Number of splices: GT/AG |    1923
##                        Number of splices: GC/AG |    22
##                        Number of splices: AT/AC |    1
##                Number of splices: Non-canonical |    12
##                       Mismatch rate per base, % |    2.71%
##                          Deletion rate per base |    0.02%
##                         Deletion average length |    1.75
##                         Insertion rate per base |    0.03%
##                        Insertion average length |    1.72
##                              MULTI-MAPPING READS:
##         Number of reads mapped to multiple loci |    2046
##              % of reads mapped to multiple loci |    2.05%
##         Number of reads mapped to too many loci |    59
##              % of reads mapped to too many loci |    0.06%
##                                   UNMAPPED READS:
##   Number of reads unmapped: too many mismatches |    0
##        % of reads unmapped: too many mismatches |    0.00%
##             Number of reads unmapped: too short |    78952
##                  % of reads unmapped: too short |    78.95%
##                Number of reads unmapped: other |    34
##                     % of reads unmapped: other |    0.03%
##                                 CHIMERIC READS:
##                       Number of chimeric reads |    0
##                            % of chimeric reads |    0.00%
```

Check some lines of the output:

```
head -n 40 Output/Aligned.out.sam | tail -1 | less -S
```

```
## SRR1039508.15492987  0   1   28982304    255 1S62M   *   0   0   CACTTGTTCTTGCCTGTCTTGTATCTCTCTTCAAA
```

Finally, it's better to convert SAM files to BAM:

```
samtools view -S -b Output/Aligned.out.sam > Output/Aligned.out.bam
```

Let's import it now and check its data:

```
library(Rsamtools)
bam <- scanBam("Output/Aligned.out.bam")
```

Field	Type	Brief description
QNAME	String	Query template name
FLAG	Int	bitwise flag
RNAME	String	Reference sequence name
POS	Int	1-based leftmost mapping position
MAPQ	Int	Mapping quality
CIGAR	String	CIGAR string
RNEXT	String	Reference name of the mate/next read
PNEXT	Int	Position of the mate/next read
TLEN	Int	Observed template length
SEQ	String	Segment sequence
QUAL	String	ASCII of Phred-scaled base quality

QNAME: Reads/segments having identical QNAME are regarded to come from the same template. A QNAME '*' indicates the information is unavailable. **FLAG:**

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

Figure 2: 2-2

RNAME: Reference sequence name of the alignment. **POS:** 1-based leftmost mapping POSition of the first CIGAR operation that "consumes" a reference base. The first base in a reference sequence has coordinate 1. POS is set as 0 for an unmapped read without coordinate. **MAPQ:** Mapping quality. A value 255 indicates that the mapping quality is not available. **CIGAR:**

RNEXT: Reference sequence name of the primary alignment of the next read in the template. For the last read, the next read is the first read in the template. **PNEXT:** 1-based Position of the primary alignment of the next read in the template. Set as 0 when the information is unavailable. This field equals POS at the primary line of the next read. **TLEN:** For primary reads where the primary alignments of all reads in the template are mapped to the same reference sequence, the absolute value of TLEN equals the distance between the mapped end of the template and the mapped start of the template **SEQ:** If not a '*', the length of the sequence must equal the sum of lengths of M/I/S/=/X operations in CIGAR. An '=' denotes the base is identical to the reference base. **QUAL:** Base quality.

Let's check the data for our example:

Op	BAM	Description	Consumes query	Consumes reference
M	0	alignment match (can be a sequence match or mismatch)	yes	yes
I	1	insertion to the reference	yes	no
D	2	deletion from the reference	no	yes
N	3	skipped region from the reference	no	yes
S	4	soft clipping (clipped sequences present in SEQ)	yes	no
H	5	hard clipping (clipped sequences NOT present in SEQ)	no	no
P	6	padding (silent deletion from padded reference)	no	no
=	7	sequence match	yes	yes
X	8	sequence mismatch	yes	yes

Figure 3: 2-3

```
bam[[1]]$qname[1:4]
```

```
## [1] "SRR1039508.12982779" "SRR1039508.15992012" "SRR1039508.7773969"
## [4] "SRR1039508.18072752"
```

```
bam[[1]]$flag[1:4]
```

```
## [1] 16 16  0  0
```

```
bam[[1]]$strand[1:4]
```

```
## [1] - - + +
## Levels: + - *
```

```
bam[[1]]$rname[1:4]
```

```
## [1] 1 1 1 1
## Levels: 1
```

```
bam[[1]]$pos[1:4]
```

```
## [1] 104153271 180189598  37544602  55154177
```

```
bam[[1]]$cigar[1:4]
```

```
## [1] "61M2S" "63M"   "63M"   "63M"
```

```
bam[[1]]$seq[1:4]
```

```
## DNAStringSet object of length 4:
##     width seq
## [1]    63 GGGGCCCTTCAGAAGTCGCTGGGCTCAGAAGGCTCTTAGTCGTGCTTGAGAGTGAGCCTTTCG
## [2]    63 ATGAATGGCTCAAGAGGCAGAAGAGAAATAAAATTCCCTACAGTTTCTTTAAAACTGCCCTGG
## [3]    63 AACAGTGCTTGGACGGAACCCGGCGCTCGTTCCCCACCCCGGACGGCCGCCCATAGGTAGCCC
## [4]    63 GCATATGCATCACTAAGGATTGTCAGGTGCTCCTCCAAGGCCTGCTGAATAAGGCTACTGGGC
```

```
bam[[1]]$qual[1:4]
```

```
## PhredQuality object of length 4:
##     width seq
## [1]    63 DDDDDEEEFFFFFFHIIHIHC=JJJJJJJJJJJJJJJIGIJIJJJJJJJJJJJJJJJJJH
## [2]    63 HHHJJJJJJJJJJJJJJJJJJJJJJJJIJJJJJJJJJGHJJJJJJJJJJJJJIJJJJJJJH
## [3]    63 D9BFGIICCGI?F+<??FHCGFHDO;@;A=H6?C?B############################
## [4]    63 HJJJJJJJJJJJJJJJJHJIJIJJJJFHIJJJJIJIJJJJJJJJIJJJJJJJJJIIJJJJJI
```

At last, let's delete the output files:

```
rm Output/Aligned.out.bam
rm Output/Aligned.out.sam
rm Output/chrLength.txt
rm Output/chrName.txt
rm Output/chrNameLength.txt
rm Output/chrStart.txt
rm Output/exonGeTrInfo.tab
rm Output/exonInfo.tab
rm Output/geneInfo.tab
rm Output/Genome
rm Output/genomeParameters.txt
rm Output/Log.final.out
rm Output/Log.out
rm Output/Log.progress.out
rm Output/SA
rm Output/SAindex
rm Output/SJ.out.tab
rm Output/sjdbInfo.txt
rm Output/sjdbList.fromGTF.out.tab
rm Output/sjdbList.out.tab
rm Output/transcriptInfo.tab
```