RNA-seq: From reads to counts

Bernard Pereira & Oscar Rueda

July 25, 2015

Contents

1 Introduction

Once we are confident in the quality of our sequencing data, we can proceed to align the reads within a FASTQ file to a reference sequence. Mapping reads onto a reference within a reasonable period of time is nontrivial from a computational perspective, and there is a lot of work that is focused on developing and improving algorithms for this task. In the case of RNA-seq data, the problem is complicated by the fact that we need to align non-contiguous reads (non-contiguous in terms of a reference genome), and we therefore need to use a splice-aware aligner.

1.1 The Data

The data used here comes from a study on esophageal squamous cell carcinoma, in which three patients (sample ids: 16, 18 and 19) had their tumours sequenced along with matched normal tissue[1]. Once the RNA-seq data was available for the samples, the authors performed differential expression analysis to look for genes whose expression was deregulated in the tumours. The data was downloaded from GEO (GSE29968).

2 Alignment

As alignment is a time-consuming process, we will be aligning the reads from just one chromsome (Chr22) of the sample 16N. We will start off with raw FASTQ files, and use Bowtie2/TopHat2[2, 3] to align the data before counting the reads that have mapped to all the genes.

Broadly speaking, there are two main steps to aligning RNA-seq data with TopHat2:

1. Align the reads to a reference genome (using Bowtie2).

2. Identifying splice junctions and mapping reads to these junctions.

Tophat2 consequently does not require a reference annotation.

2.1 Initial Steps

Ideally, when we receive raw data, we should ensure that it has the expected quality and that there have been no unexpected errors. We can run FASTQC, as previously decribed, to explore some features of the data.

```
cd Day2
fastqc 16N_reads.fq
```

2.2 Bowtie 2

To begin, TopHat2 uses Bowtie2 to align reads to the genome. Alignment with Bowtie2 requires an indexed genome, which is represented in a collection of files suffixed with '.bt2'. For the purpose of this practical, the index has already been created. However, we need to tell TopHat2 where to find this index when it requires it for alignment. One way to do this is to set up an **environment variable**, which TopHat2 can use to locate the files it requires. TopHat2 will look in the directory specified by the environment variable 'BOWTIE2_INDEXES'.

```
ls ../ref_data/bowtie
cp ../ref_data/chr22.fa ../ref_data/bowtie/
bowtie2-build #Command to build index

export BOWTIE2_INDEXES="$HOME/cruk-bioinf-sschool/ref_data/bowtie/"
echo $BOWTIE2_INDEXES
```

2.3 Alignment

When aligning with TopHat2, we need to specify the basename of the genome index.

```
mkdir Tophat2_Alignment
tophat2 --output-dir Tophat2_Alignment chr22 16N_reads.fq
```

While our reads are aligning, we can have a look at some of the many options that TopHat2 provides: https://ccb.jhu.edu/software/tophat/manual.shtml. These can be altered depending on the analysis in question, although for the most part, the default parameters work well.

2.4 Viewing Data

TopHat2 automatically generates a BAM file, although we need to index this when the alignment is complete.

```
samtools index Tophat2_Alignment/accepted_hits.bam
samtools flagstat Tophat2_Alignment/accepted_hits.bam
samtools flagstat Tophat2_Alignment/unmapped.bam
```

We can now open this BAM file in IGV and have a look at our alignments. Go to the gene *TPST2* to see what spliced alignments look like. We can also load the file Tophat2_Alignment/junctions.bed to view the splice junctions that TopHat2 identified.

Looking only at one chromosome may not be that exciting, so we can also have a browse through some of the files in

```
ls bam/
samtools flagstat bam/16N_aligned.bam
samtools view -H 16N_aligned.bam
```

on IGV. The authors of the original publication identified *PTK6* as a tumour suppressor; can we see downregulation of the gene on the genome browser?

3 Counting

After alignment is complete, we need to count the number of reads that have mapped to the features of interest. This is not necessarily a trivial task, and the method used will depend on the question being asked. For example, we may want to quantify the numbers of reads mapping to exons, isoforms or junctions. We may also be interested in quantifying novel isoform, or even novel genes (most likely in other species).

3.1 Counting with Cufflinks

Cufflinks[4] - which, as you might have guessed from its name, is part of the Bowtie/Tophat suite - is one tool that can be used for assembling and quantifying *transcripts* from reads aligned to a reference genome. While it is possible to provide an annnotation file to guide the quantification, we will not be doing so now.

```
cufflinks -h

cufflinks Tophat2_Alignment/accepted_hits.bam > Tophat2_Alignment/cufflinksResult
#This will take ~7-8 minutes
```

Cufflinks produces three files: isoforms.fpkm_tracking, genes.fpkm_tracking and transcripts.gtf. The last of these stores information about the transcripts that Cufflinks has assembled, while the other two provides count estimates for the genes and isoforms identified from this set of transcripts (fpkm = fragments per kilo million). We can learn more about the GTF format from the Ensembl website: http://www.ensembl.org/info/website/upload/gff.html.

```
ls Tophat2_Alignment

less transcripts.gtf
less genes.fpkm\_tracking
less isoforms.fpkm\_tracking
```

3.2 Counting with Rsubread

As you might imagine, running Cufflinks for all our samples to assemble transcripts would take a while, In our case, however, we are only interested in looking for differential expression in genes that we have already defined. Believe it or not, we can do this in Bioconductor – so let's move back to our trusty R! We will use the package *Rsubread*[5] to count the reads mapping to genes in the human genome.

```
library(Rsubread)
filesToCount <- dir("bam", pattern=".bam$", full.names=T)
```

Rsubread has a number of inbuild annotations that we can make use of. Once again, there are a number of options to play around with, although we will keep things simple for now and generate a count matrix to use in the practical on differential expression with *edgeR*.

```
tmp <- featureCounts(filesToCount, annot.inbuilt="hg19", ignoreDup=F)
save(tmp, file="../Day3/countMatrix.RData")
```

References

[1] Ma, S. et al. (2012) *Identification of PTK6, via RNA Sequencing Analysis, as a Suppressor of Esophageal Squamous Cell Carcinoma*, Gastroenterology, 143 (3) 675-686.

[2] Langmead, B. & Salzberg, S. (2012) *Fast gapped-read alignment with Bowtie2*, Nature Methods, 9:357-359.

[3] Kim, D. et al. (2013) *TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions*, Genome Biology, 14:R36.

[4] Trapnell, C. et al. (2010) *Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation*, Nature Biotech. 28, 511-515.

[5] Liao, Y., Smyth, GK. & Shi, W. (2013) *The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote*, Nucleic Acids Res. 41, e108.