

Tuesday – part 3

Ab initio transcriptome assembly

Michał Szcześniak, PhD

Faculty of Biology, Adam Mickiewicz University, Poznań
ideas4biology Ltd.

Pipelines

1. FASTQ → QC and filtering → mapping → ***ab initio* assembly**
2. FASTQ → QC and filtering → (mapping) → expression estimation → differential expression analysis
3. FASTQ → QC and filtering → *de novo* assembly

Preparing the data

We already have the BAM file

ERR990413.sorted.bam

Downloading chr 22 – we already have it

wget

[ftp://ftp.ensembl.org/pub/release-88/fasta/homo_sapiens/dna/
Homo_sapiens.GRCh38.dna.chromosome.22.fa.gz](ftp://ftp.ensembl.org/pub/release-88/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.dna.chromosome.22.fa.gz)

gunzip Homo_sapiens.GRCh38.dna.chromosome.22.fa.gz

Downloading genome annotations – we already have it

wget ftp://ftp.ensembl.org/pub/release-88/gtf/homo_sapiens/
Homo_sapiens.GRCh38.88.gtf.gz

gunzip Homo_sapiens.GRCh38.88.gtf.gz

Transcriptome assembly

Why StringTie?

<https://ccb.jhu.edu/software/stringtie/index.shtml?t=example>

Ab initio assembly

```
stringtie/stringtie ERR990413.sorted.bam -o ERR990413.gtf -p 1 -G  
Homo_sapiens.GRCh38.88.gtf -A abundance.txt --rf
```

- p 1: numer of threads
- G: reference annotations
- o: output file name
- A: write expression estimations to this file
- rf: strandedness (here: *fr-firststrand*)

Cuffcompare

Comparing the assembly with known annotations

```
cuffcompare -r Homo_sapiens.GRCh38.88.gtf -R -o ERR990413 -C -G ERR990413.gtf
```

-r: a file with reference annotations

-R: consider only the reference transcripts that overlap any of the input transfrags

-o: a prefix for output files

-C: include the "contained" transcripts in the .combined.gtf file

-G: input GTF file

Statistics for ERR990413.combined.gtf:

```
python counts_stringtie.py ERR990413.combined.gtf
```

```
Genes: 415
```

```
Transcripts: 930
```

```
c 51
```

```
i 26
```

```
j 18
```

```
o 8
```

```
p 5
```

```
u 47
```

```
x 1
```

```
= 774
```

Cuffcompare class codes

Priority	Code	Description
1	=	Complete match of intron chain
2	c	Contained
3	j	Potentially novel isoform (fragment): at least one splice junction is shared with a reference transcript
4	e	Single exon transfrag overlapping a reference exon and at least 10 bp of a reference intron, indicating a possible pre-mRNA fragment.
5	i	A transfrag falling entirely within a reference intron
6	o	Generic exonic overlap with a reference transcript
7	p	Possible polymerase run-on fragment (within 2Kbases of a reference transcript)
8	r	Repeat. Currently determined by looking at the soft-masked reference sequence and applied to transcripts where at least 50% of the bases are lower case
9	u	Unknown, intergenic transcript
10	x	Exonic overlap with reference on the opposite strand
11	s	An intron of the transfrag overlaps a reference intron on the opposite strand (likely due to read mapping errors)
12	.	(.tracking file only, indicates multiple classifications)

Visualizing in IGV

Let's prepare a GTF file for only chr 22

```
python extract_from_GTF.py Homo_sapiens.GRCh38.88.gtf 22 chr22.gtf
```

Load the files: chr22.gtf and ERR990413.gtf

Look into the region **chr22:17,137,180-17,139,784**

