RNA-Seq Lab

BCB 5200 Introduction Bioinformatics I

Fall 2017

Tae-Hyuk (Ted) Ahn

Department of Computer Science
Program of Bioinformatics and Computational Biology
Saint Louis University

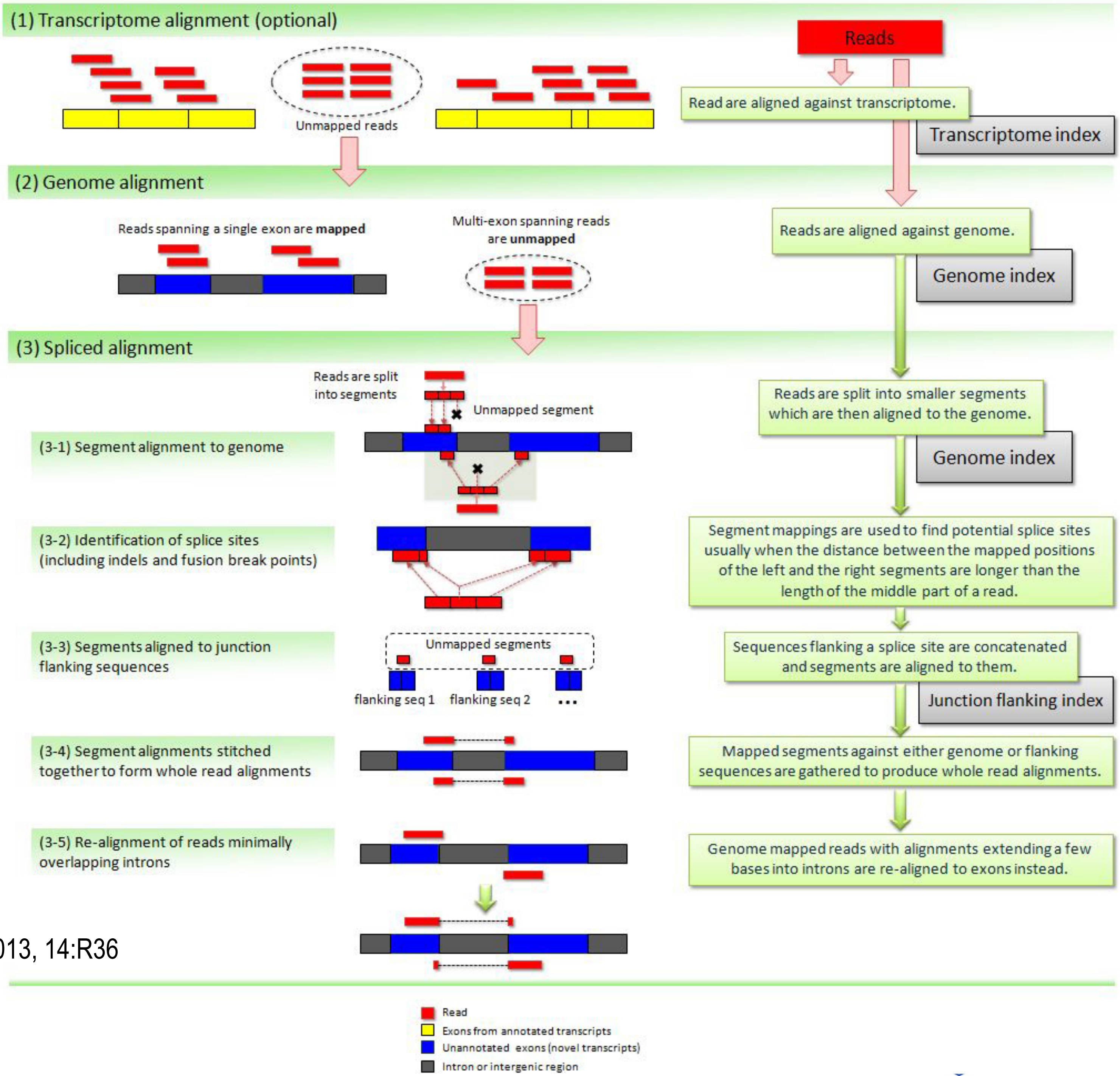
SAINT LOUIS UNIVERSITY™

— EST. 1818 —

How tophat works

- Two Steps
  1. Uses unspliced aligner Bowtie to map reads to reference genome
  2. For unmapped reads in step 1
     1. It detects potential splice sites for introns
     2. It uses these candidate splice sites to correctly align multiexon-spanning reads

Tophat 2



(1) Transcriptome alignment (optional)

(2) Genome alignment

(3) Spliced alignment

(3-1) Segment alignment to genome

(3-2) Identification of splice sites (including indels and fusion break points)

(3-3) Segments aligned to junction flanking sequences

(3-4) Segment alignments stitched together to form whole read alignments

(3-5) Re-alignment of reads minimally overlapping introns

Reads

Read are aligned against transcriptome.

Transcriptome index

Reads are aligned against genome.

Genome index

Reads are split into smaller segments which are then aligned to the genome.

Genome index

Segment mappings are used to find potential splice sites usually when the distance between the mapped positions of the left and the right segments are longer than the length of the middle part of a read.

Sequences flanking a splice site are concatenated and segments are aligned to them.

Junction flanking index

Mapped segments against either genome or flanking sequences are gathered to produce whole read alignments.

Genome mapped reads with alignments extending a few bases into introns are re-aligned to exons instead.

Unmapped reads
Reads spanning a single exon are **mapped**
Multi-exon spanning reads are **unmapped**
Reads are split into segments
Unmapped segment
Unmapped segments
flanking seq 1
flanking seq 2
...

🟥 Read
🟨 Exons from annotated transcripts
🟦 Unannotated exons (novel transcripts)
⬛ Intron or intergenic region

Kim et al. Genome Biology 2013, 14:R36

How tophat works: junctions

- From supplied annotation file (GFF) or list of junction coordinates
- Find splice junctions without a reference annotation:
  - By first mapping RNA-Seq reads to the genome, TopHat identifies potential exons
  - Using this initial mapping information, TopHat builds a database of possible splice junctions
  - then maps the reads against these junctions to confirm them

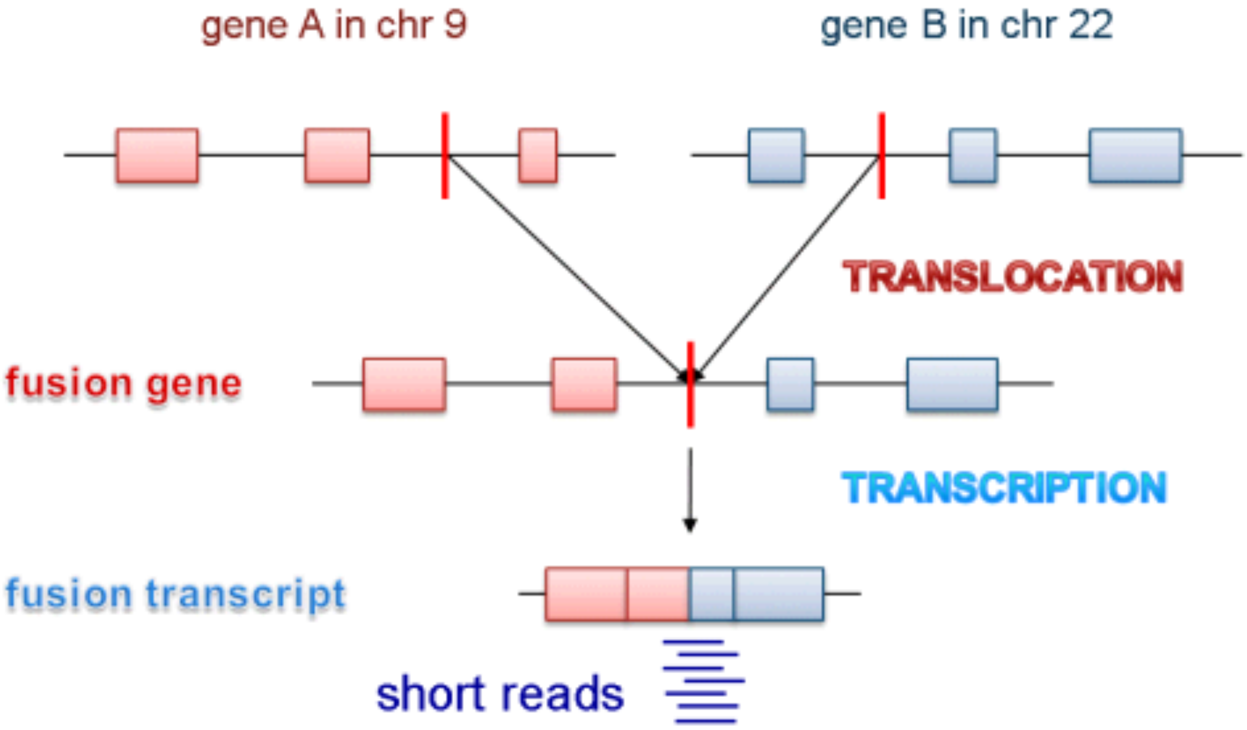
https://ccb.jhu.edu/software/tophat/manual.shtml

Tophat 1 vs Tophat 2

- TopHat2 can align longer reads (optimized for reads 75bp or longer)
- TopHat2 allowing for variable-length indels with respect to the reference genome.
- TopHat2 can align reads across fusion breaks

Kim et al. Genome Biology 2013, 14:R36

Gene fusion



Fusion genes are chimeric genes formed by two previously separated genes. They may be the products of chromosome structure changes such as insertion, deletion, inversion and translocation.

http://donglab.ecnu.edu.cn/databases/FusionCancer/

Tophat2 usage: input files

- Required input files
  - read file (fastq)
  - genome_index_base: indexed genome sequence
    - Download genome sequence
    - Generate genome index
- Optional input file
  - Genome annotation file (gff or gtf format)

Genome sequence file: where to get?

- Illumina iGenomes (recommended)
  - http://support.illumina.com/sequencing/sequencing_software/igenome.html
- Ensembl
  - http://ensemblgenomes.org/info/access/ft

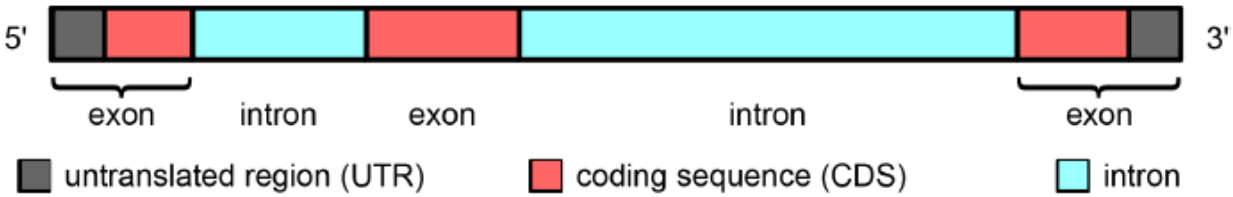
e!Ensembl

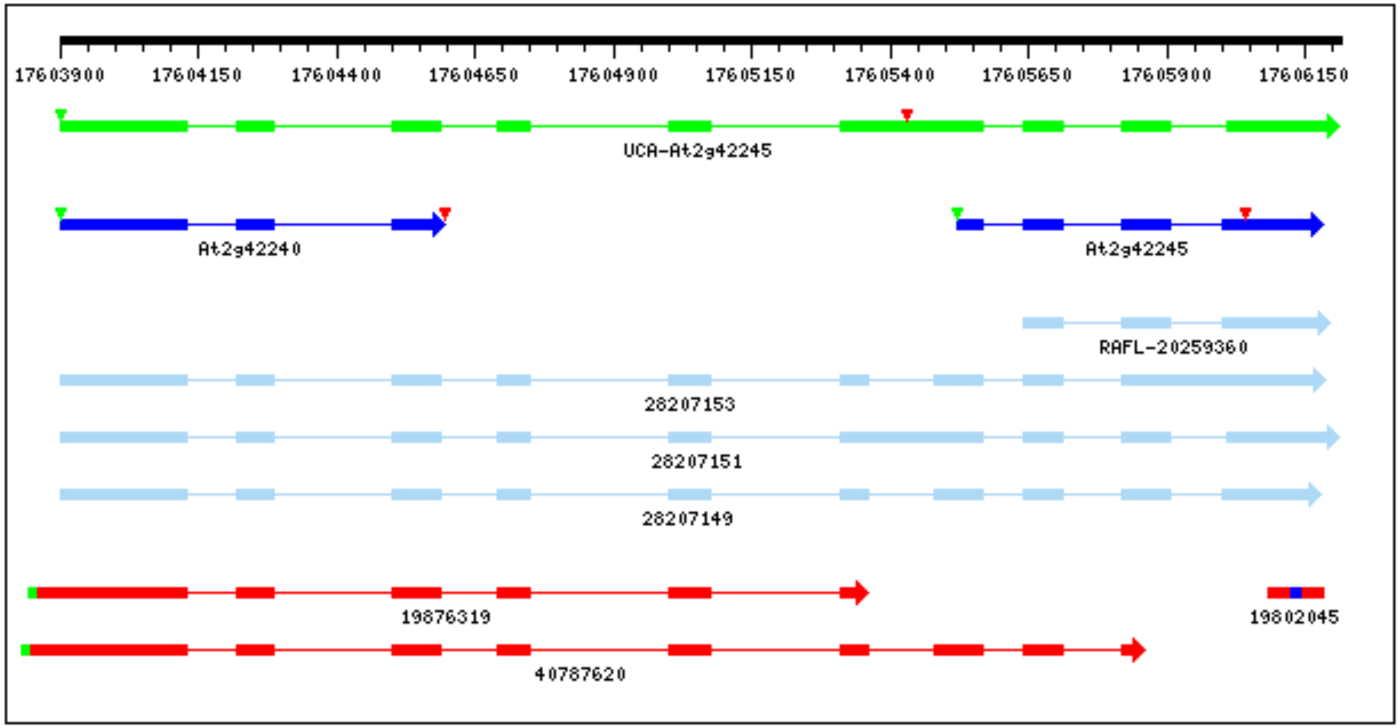
- NCBI genome
  - http://www.ncbi.nlm.nih.gov/genome/
- Organism specific databases/websites.

SGD Saccharomyces GENOME DATABASE

tair

Genome annotation file





GFF/GTF File Format - Definition

- The GFF (General Feature Format) format
  - one line per feature
  - each containing 9 columns of data
  - plus optional track definition lines.
- GFF has many versions (GFF, GFF2, GFF3)
- GTF (General Transfer Format) identical to GFF2.
- Tophat supports both GTF and GFF3 (mostly GTF)

GFF/GTF File Format - Definition

- The GFF (General Feature Format) format
  - one line per feature
  - each containing 9 columns of data
  - plus optional track definition lines.
- GFF has many versions (GFF, GFF2, GFF3)
- GTF (General Transfer Format) identical to GFF2.
- Tophat supports both GTF and GFF3

GTF/GTF2 format

9 columns:

```
<seqname> <source> <feature> <start> <end> <score> <strand>
<frame> [attributes] [comments]
```

- seqname - name of the chromosome or scaffold
- source – program or database that generated this feature.
- feature – Examples: "CDS", “gene", “transcript”, and "exon".
- start - The starting position of the feature in the sequence.
- end - The ending position of the feature (inclusive).
- score - A score between 0 and 1000.
- strand – ‘+ ‘(forward) or ‘-’ (reverse) or '.' (don't know/don't care).
- Frame – reading frame '0', '1' or '2'
- attribute – A semicolon-separated list of tag-value pairs, providing additional information about each feature.

Example of GTF2 format

```
AB000381 Twinscan  CDS          380   401   .   +   0  gene_id "001"; transcript_id "001.1";
AB000381 Twinscan  CDS          501   650   .   +   2  gene_id "001"; transcript_id "001.1";
AB000381 Twinscan  CDS          700   707   .   +   2  gene_id "001"; transcript_id "001.1";
AB000381 Twinscan  start_codon  380   382   .   +   0  gene_id "001"; transcript_id "001.1";
AB000381 Twinscan  stop_codon   708   710   .   +   0  gene_id "001"; transcript_id "001.1";
```

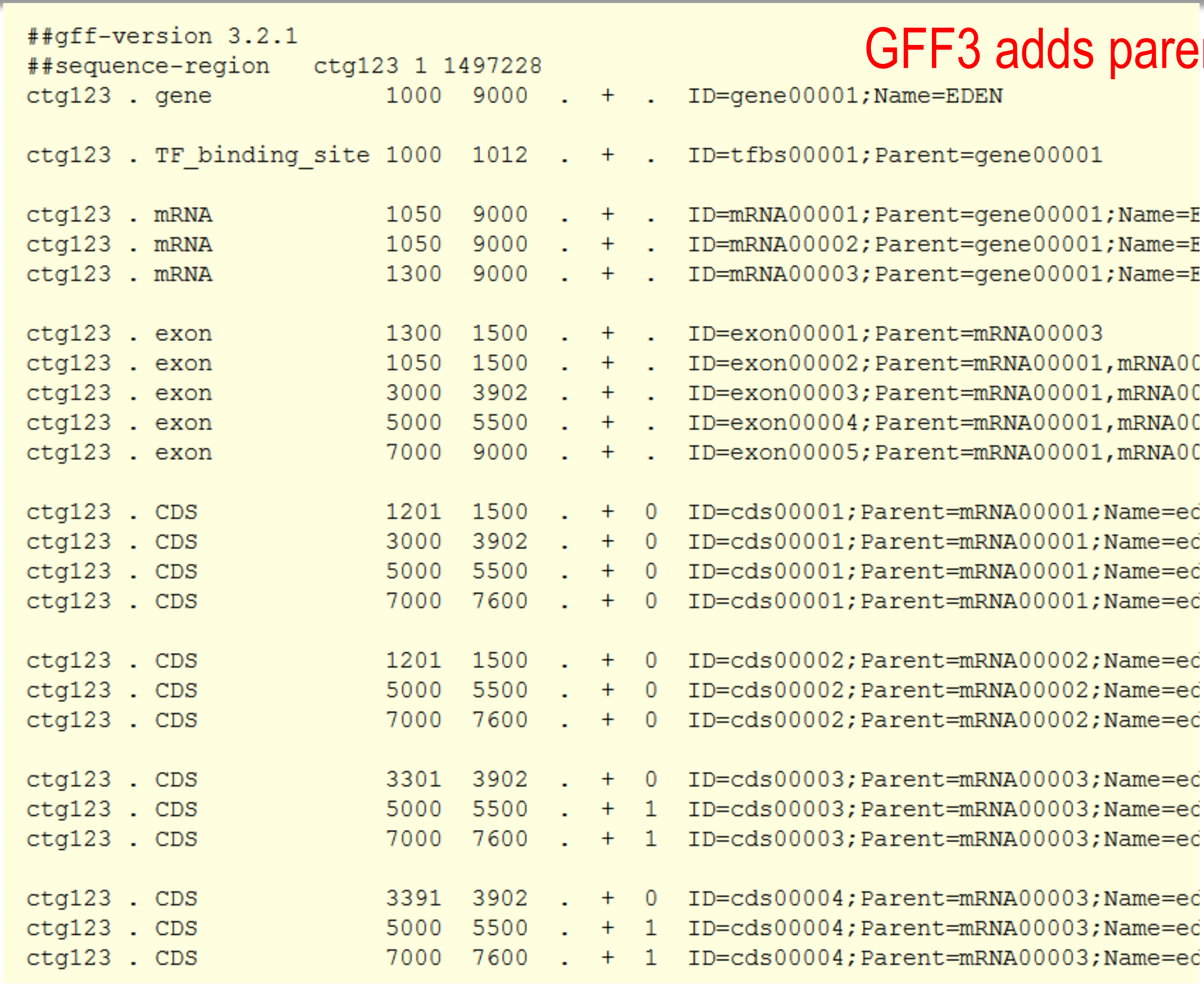
A simple example with 3 translated exons. Order of rows is not important.

Some annotation sources (e.g. Ensembl) add the gene_name attribute

```
gene_id "ENSBTAG00000020601"; transcript_id "ENSBTAT00000027448"; gene_name "ZNF366";
```

http://mblab.wustl.edu/GTF2.html

Generic Feature Format Version 3 (GFF3)



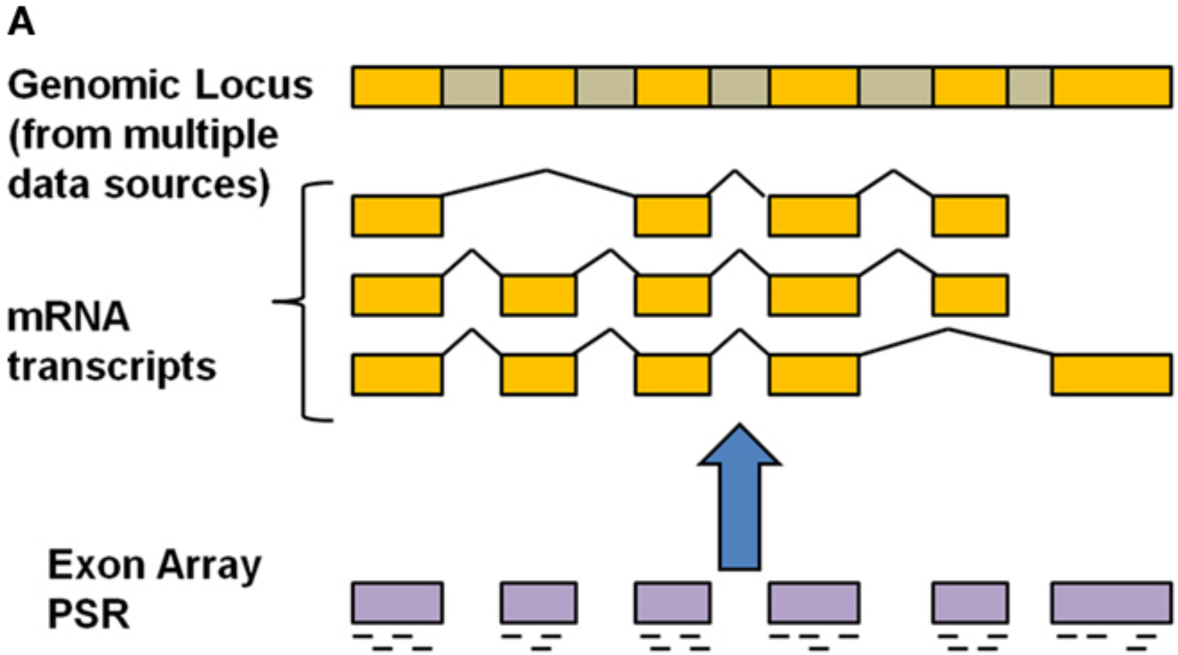
```
##gff-version 3.2.1
##sequence-region   ctg123 1 1497228
ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN

ctg123 . TF_binding_site 1000  1012  .  +  .  ID=tfbs00001;Parent=gene00001

ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00001;Parent=gene00001;Name=E
ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00002;Parent=gene00001;Name=E
ctg123 . mRNA            1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001;Name=E

ctg123 . exon            1300  1500  .  +  .  ID=exon00001;Parent=mRNA00003
ctg123 . exon            1050  1500  .  +  .  ID=exon00002;Parent=mRNA00001,mRNA00
ctg123 . exon            3000  3902  .  +  .  ID=exon00003;Parent=mRNA00001,mRNA00
ctg123 . exon            5000  5500  .  +  .  ID=exon00004;Parent=mRNA00001,mRNA00
ctg123 . exon            7000  9000  .  +  .  ID=exon00005;Parent=mRNA00001,mRNA00

ctg123 . CDS             1201  1500  .  +  0  ID=cds00001;Parent=mRNA00001;Name=ed
ctg123 . CDS             3000  3902  .  +  0  ID=cds00001;Parent=mRNA00001;Name=ed
ctg123 . CDS             5000  5500  .  +  0  ID=cds00001;Parent=mRNA00001;Name=ed
ctg123 . CDS             7000  7600  .  +  0  ID=cds00001;Parent=mRNA00001;Name=ed

ctg123 . CDS             1201  1500  .  +  0  ID=cds00002;Parent=mRNA00002;Name=ed
ctg123 . CDS             5000  5500  .  +  0  ID=cds00002;Parent=mRNA00002;Name=ed
ctg123 . CDS             7000  7600  .  +  0  ID=cds00002;Parent=mRNA00002;Name=ed

ctg123 . CDS             3301  3902  .  +  0  ID=cds00003;Parent=mRNA00003;Name=ed
ctg123 . CDS             5000  5500  .  +  1  ID=cds00003;Parent=mRNA00003;Name=ed
ctg123 . CDS             7000  7600  .  +  1  ID=cds00003;Parent=mRNA00003;Name=ed

ctg123 . CDS             3391  3902  .  +  0  ID=cds00004;Parent=mRNA00003;Name=ed
ctg123 . CDS             5000  5500  .  +  1  ID=cds00004;Parent=mRNA00003;Name=ed
ctg123 . CDS             7000  7600  .  +  1  ID=cds00004;Parent=mRNA00003;Name=ed
```

GFF3 adds parent feature

http://www.sequenceontology.org/gff3.shtml

GFF2 vs GFF3

- GFF2
  - two-level hierarchies *transcript* → *exon*
- GFF3
  - three-level hierarchy of *gene* → *transcript* → *exon*



http://gmod.org/wiki/GFF2

Conversion GFF3 To GTF

- Optional
- Use `gffread` (comes with the Cufflinks software suite)

```
$ gffread my.gff3 -T -o my.gtf
```

- See `gffread -h` for more information

Download Prepare GTF/GFF file

- Download it from Illumina's iGenomes project (for model species)
  - http://support.illumina.com/sequencing/sequencing_software/igenome.html

Or

- Download gff3 files from genome database,
  - Ensembl gnome
  - http://ensemblgenomes.org/

Tuxedo Genome Guided Transcriptome Assembly Workshop

- https://github.com/trinityrnaseq/RNASeq_Trinity_Tuxedo_Workshop/wiki/Tuxedo-Genome-Guided-Transcriptome-Assembly-Workshop
- The following details the steps involved in:
  - Aligning RNA-Seq reads to a genome using Tophat
  - Assembling transcript structures from read alignments using Cufflinks
  - Visualizing reads and transcript structures using IGV
  - Performing differential expression analysis using Cuffdiff
  - Expression analysis using CummeRbund

Building genome index using `bowtie2-build`

- Only has to be done once!
- `bowtie2-build` builds a Bowtie index from a set of DNA sequences.
- Generates 6 output files
  - `name.1.bt2,name.2.bt2, name.3.bt2, name.4.bt2`
  - `name.rev.1.bt2, name.rev.2.bt2.`
- First, prepare the 'genome.fa' file for tophat alignment:

Usage:

```
bowtie2-build [options]* <reference_in> <bt2_base>
```

```
$ mkdir RNASeq_lab
$ cd RNASeq_lab
$ cp -R /public/ahnt/courses/bcb5200/RNASeq_lab/* .
$ bowtie2-build GENOME_data/genome.fa genome
```

Align reads and assemble transcripts for sample Sp_ds

Align reads using tophat:

```
$ mkdir RNASeq_lab
$ cd RNASeq_lab
$ cp -R /public/ahnt/courses/bcb5200/RNASeq_lab/* .
$ bowtie2-build GENOME_data/genome.fa genome
```

Rename the alignment (bam) output file according to this sample name:

```
$ mv tophat.Sp_ds.dir/accepted_hits.bam tophat.Sp_ds.dir/
Sp_ds.bam
```

Index this bam file for later viewing using IGV:

```
$ samtools index tophat.Sp_ds.dir/Sp_ds.bam
```

Reconstruct transcripts for this sample using Cufflinks:

```
$ cufflinks --no-update-check --overlap-radius 1 \
             --library-type fr-firststrand \
             -o cufflinks.Sp_ds.dir tophat.Sp_ds.dir/Sp_ds.bam
```

Rename the cufflinks transcript structure output file according to this sample:

```
$ mv cufflinks.Sp_ds.dir/transcripts.gtf cufflinks.Sp_ds.dir/
Sp_ds.transcripts.gtf
```

Tophat2 options: example

- -p/--num-threads <int>
  - Use this many threads to align reads. The default is 1.
- -o/--output-dir <string>
  - The default is "./tophat_out".
- -G/--GTF <GTF/GFF3 file>
  - gene model annotations
- -r/--mate-inner-dist <int>
  - expected (mean) inner distance between mate pairs. The default is 50bp.
- -N/--read-mismatches
  - Final read alignments having more than these many mismatches are discarded. The default is 2.

TopHat output

- `accepted_hits.bam`
  - A list of read alignments in BAM format
- `align_summary.txt`
- `deletions.bed`
- `insertions.bed`
- `junctions.bed`
- `prep_reads.info`
- `unmapped.bam`
- Logs folder

Running Cufflinks

Run cufflinks from the command line as follows:

```
cufflinks [options] <aligned_reads.(sam/bam)>
```

- **Cufflinks Input Files**
  - Required:
    - SAM/BAM files as input: must be sorted by reference position
  - Optional
    - Genome sequence (fasta)
    - Genome annotation (gtf, gff)

Cufflinks General Options

```
-o/--output-dir                    write all output files to this directory
                [default:     ./ ]
-p/--num-threads                   number of threads used during analysis
                [default:      1 ]
-G/--GTF                           quantitate against reference transcript
                annotations
-g/--GTF-guide                     use reference transcript annotation to guide
                assembly
-b/--frag-bias-correct             use bias correction - reference fasta
                    required             [ default:   NULL ]
-u/--multi-read-correct            use 'rescue method' for multi-reads (more
                accurate)    [ default:  FALSE ]
```

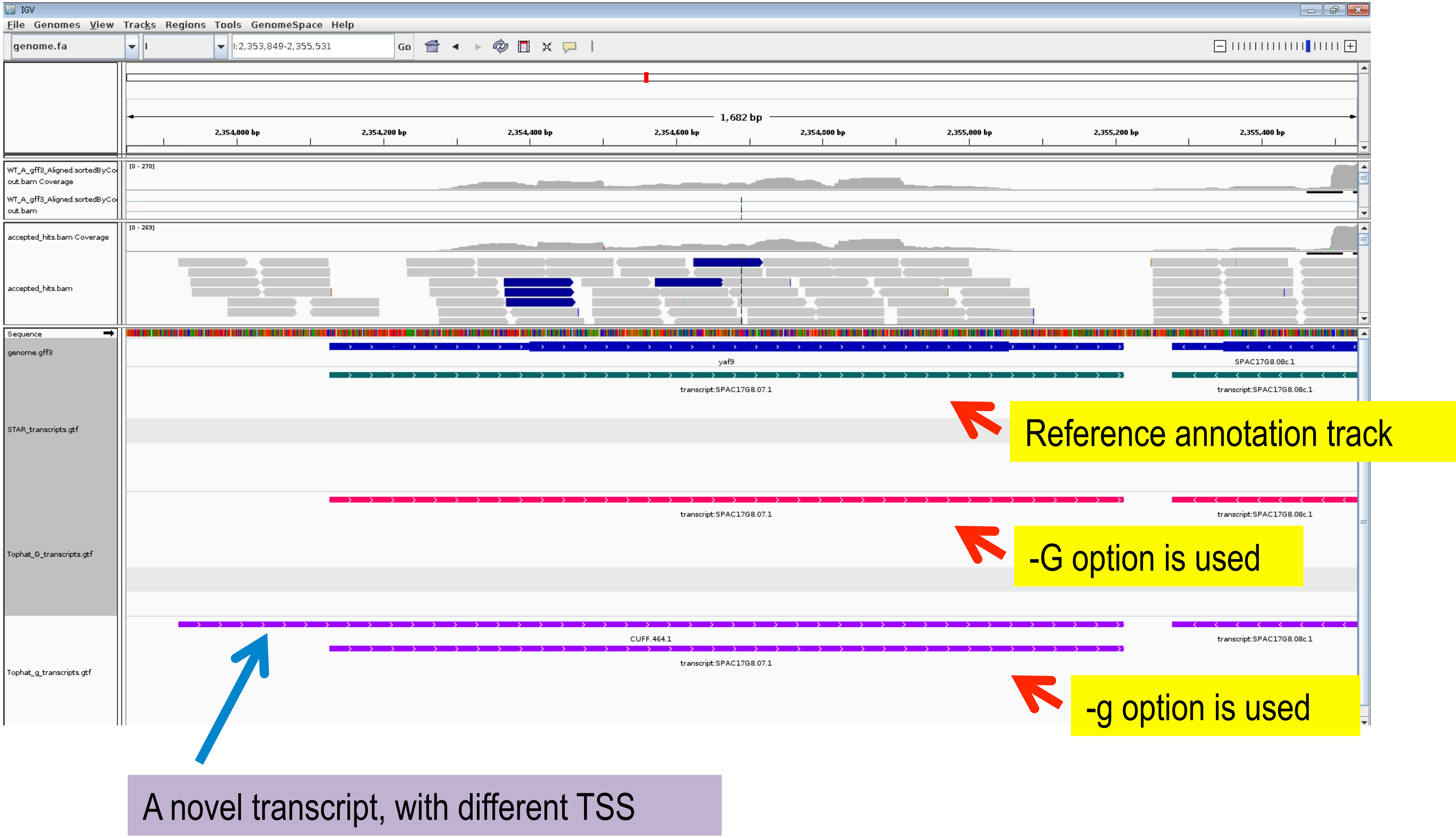
Example command

```
$ cufflinks -o cufflinks_out/ -p 8 -u -b ens/genome.fa -G ens/
genome.gff3 tophat_out/accepted_hits.bam
```

What to use: -g or -G

- -g
  - does reference guided assembly that will use the GTF as a base, but also look for novel transcripts.
  - output will include all reference transcripts and any novel genes and isoforms that are assembled
- -G
  - make cufflinks use only the annotated transcripts in the GTF
  - It will not assemble novel transcripts

The differences: -g or -G



Reference annotation track

-G option is used

-g option is used

A novel transcript, with different TSS

The differences: -g or -G

With -G

```
Missed exons:        0/12230 (   0.0%)
Novel exons:         0/12350 (   0.0%)
Missed introns:      0/5331  (   0.0%)
Novel introns:       0/5331  (   0.0%)
Missed loci:         0/6204  (   0.0%)
Novel loci:          0/6203  (   0.0%)
```

With -g

```
Missed exons:         0/12230   (   0.0%)
Novel exons:          118/12976 (   0.9%)
Missed introns:       0/5331        (   0.0%)
Novel introns:        77/5455   (   1.4%)
Missed loci:          0/6204        (   0.0%)
Novel loci:           84/6212   (   1.4%)
```

Cufflinks Output Files

- transcripts.gtf
  - This GTF file contains Cufflinks' assembled isoforms.
- genes.fpkm_tracking
  - This file contains the estimated isoform-level expression values (FPKM).
- isoforms.fpkm_tracking
  - This file contains the estimated gene-level expression values (FPKM).

Align reads and assemble transcripts for sample Sp_hs

```
$ tophat2 -I 1000 -i 20 --library-type fr-firststrand \
          -o tophat.Sp_hs.dir genome \
          RNASEQ_data/Sp_hs.left.fq.gz RNASEQ_data/Sp_hs.right.fq.gz

$ mv tophat.Sp_hs.dir/accepted_hits.bam tophat.Sp_hs.dir/Sp_hs.bam

$ samtools index tophat.Sp_hs.dir/Sp_hs.bam

$ cufflinks --no-update-check --overlap-radius 1 \
            --library-type fr-firststrand \
            -o cufflinks.Sp_hs.dir tophat.Sp_hs.dir/Sp_hs.bam

$ mv cufflinks.Sp_hs.dir/transcripts.gtf cufflinks.Sp_hs.dir/
Sp_hs.transcripts.gtf
```

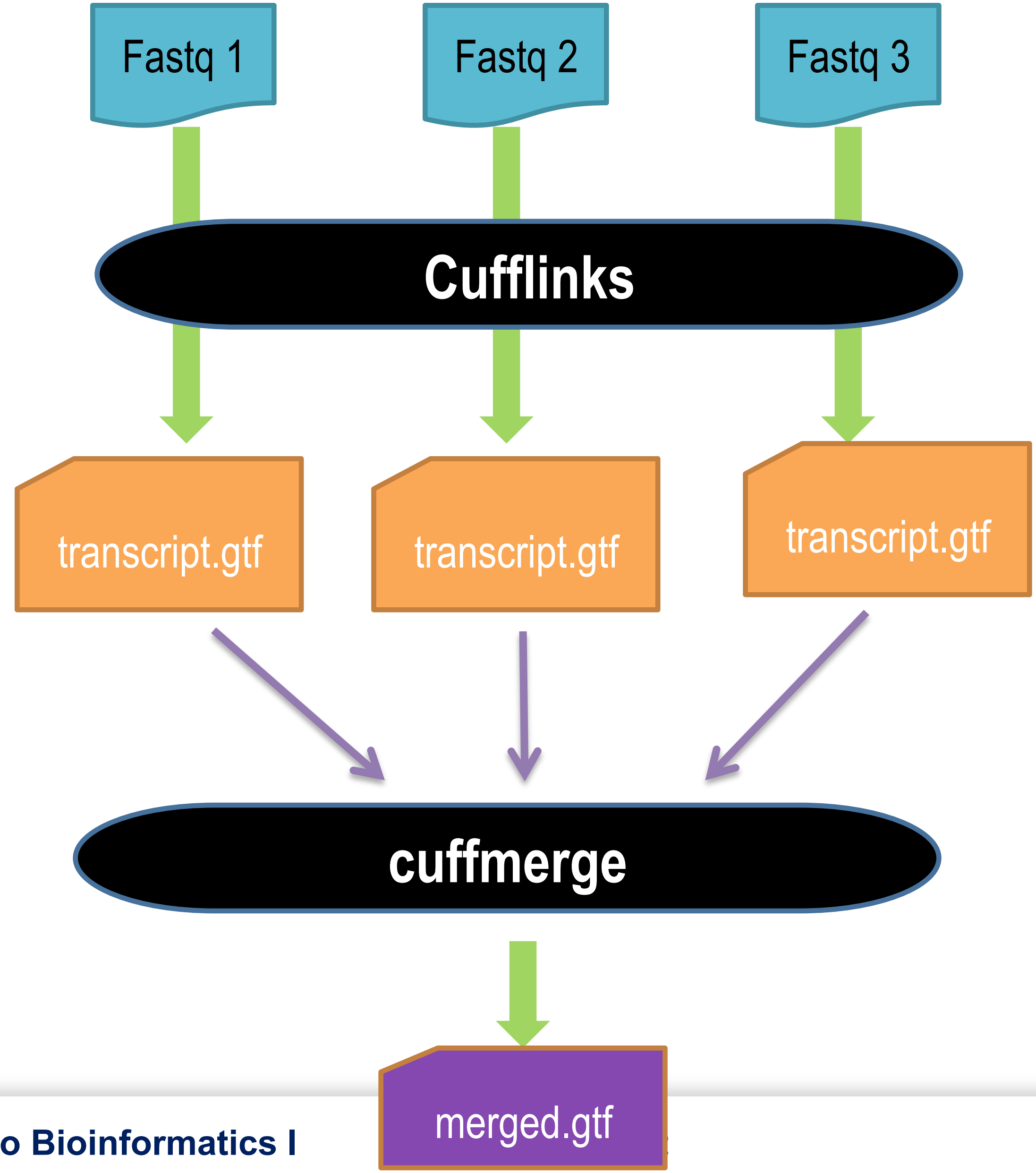
Align reads and assemble transcripts for sample Sp_log

```
 $ tophat2 -I 1000 -i 20 --library-type fr-firststrand \
           -o tophat.Sp_log.dir genome \
           RNASEQ_data/Sp_log.left.fq.gz RNASEQ_data/Sp_log.right.fq.gz

 $ mv tophat.Sp_log.dir/accepted_hits.bam tophat.Sp_log.dir/Sp_log.bam

 $ samtools index tophat.Sp_log.dir/Sp_log.bam

 $ cufflinks --no-update-check --overlap-radius 1 \
             --library-type fr-firststrand \
             -o cufflinks.Sp_log.dir tophat.Sp_log.dir/Sp_log.bam

 $ mv cufflinks.Sp_log.dir/transcripts.gtf cufflinks.Sp_log.dir/
Sp_log.transcripts.gtf
```

Align reads and assemble transcripts for sample Sp_log

```
 $ tophat2 -I 1000 -i 20 --library-type fr-firststrand \
           -o tophat.Sp_plat.dir genome \
           RNASEQ_data/Sp_plat.left.fq.gz RNASEQ_data/
Sp_plat.right.fq.gz

 $ mv tophat.Sp_plat.dir/accepted_hits.bam tophat.Sp_plat.dir/
Sp_plat.bam

 $ samtools index tophat.Sp_plat.dir/Sp_plat.bam

 $ cufflinks --no-update-check --overlap-radius 1 \
             --library-type fr-firststrand \
             -o cufflinks.Sp_plat.dir tophat.Sp_plat.dir/Sp_plat.bam

 $ mv cufflinks.Sp_plat.dir/transcripts.gtf cufflinks.Sp_plat.dir/
Sp_plat.transcripts.gtf
```

Merge separately assembled transcript structures into a cohesive set:



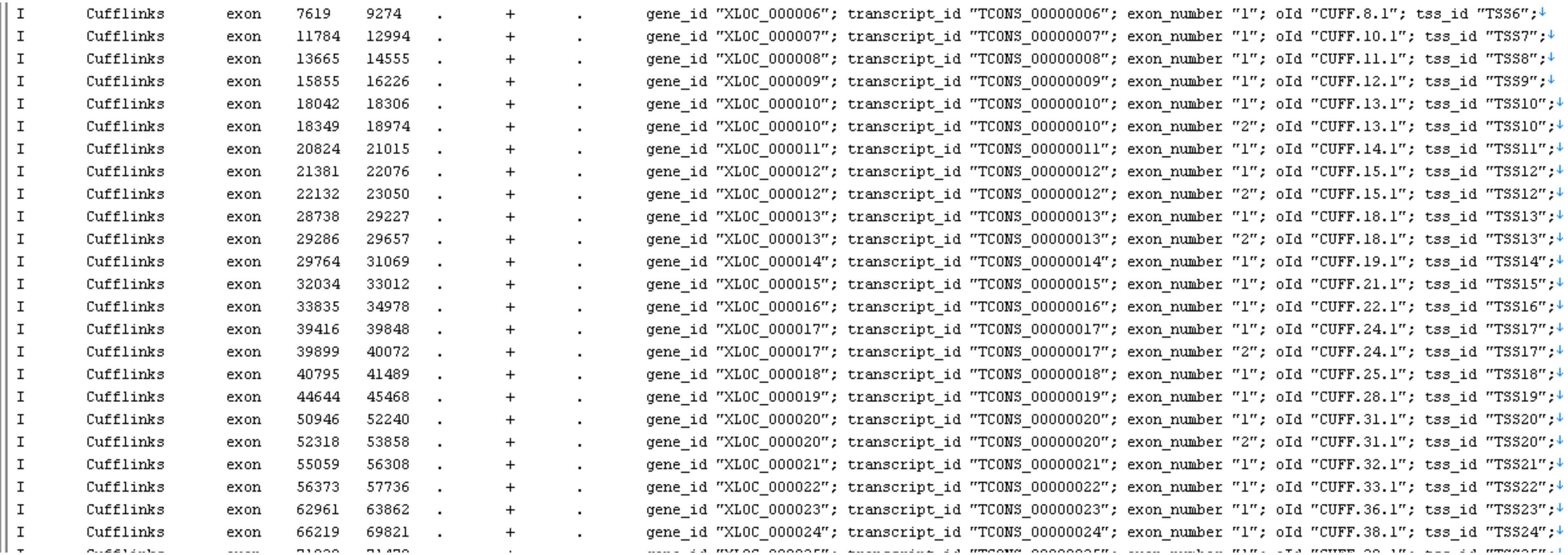
cuffmerge

```
$ echo cufflinks.Sp_ds.dir/Sp_ds.transcripts.gtf > assemblies.txt
$ echo cufflinks.Sp_hs.dir/Sp_hs.transcripts.gtf >> assemblies.txt
$ echo cufflinks.Sp_log.dir/Sp_log.transcripts.gtf >> assemblies.txt
$ echo cufflinks.Sp_plat.dir/Sp_plat.transcripts.gtf >> assemblies.txt

$ cuffmerge -s GENOME_data/genome.fa assemblies.txt
```

The merged set of transcripts should now exist as file 'merged_asm/merged.gtf'.

Cuffmerge output files

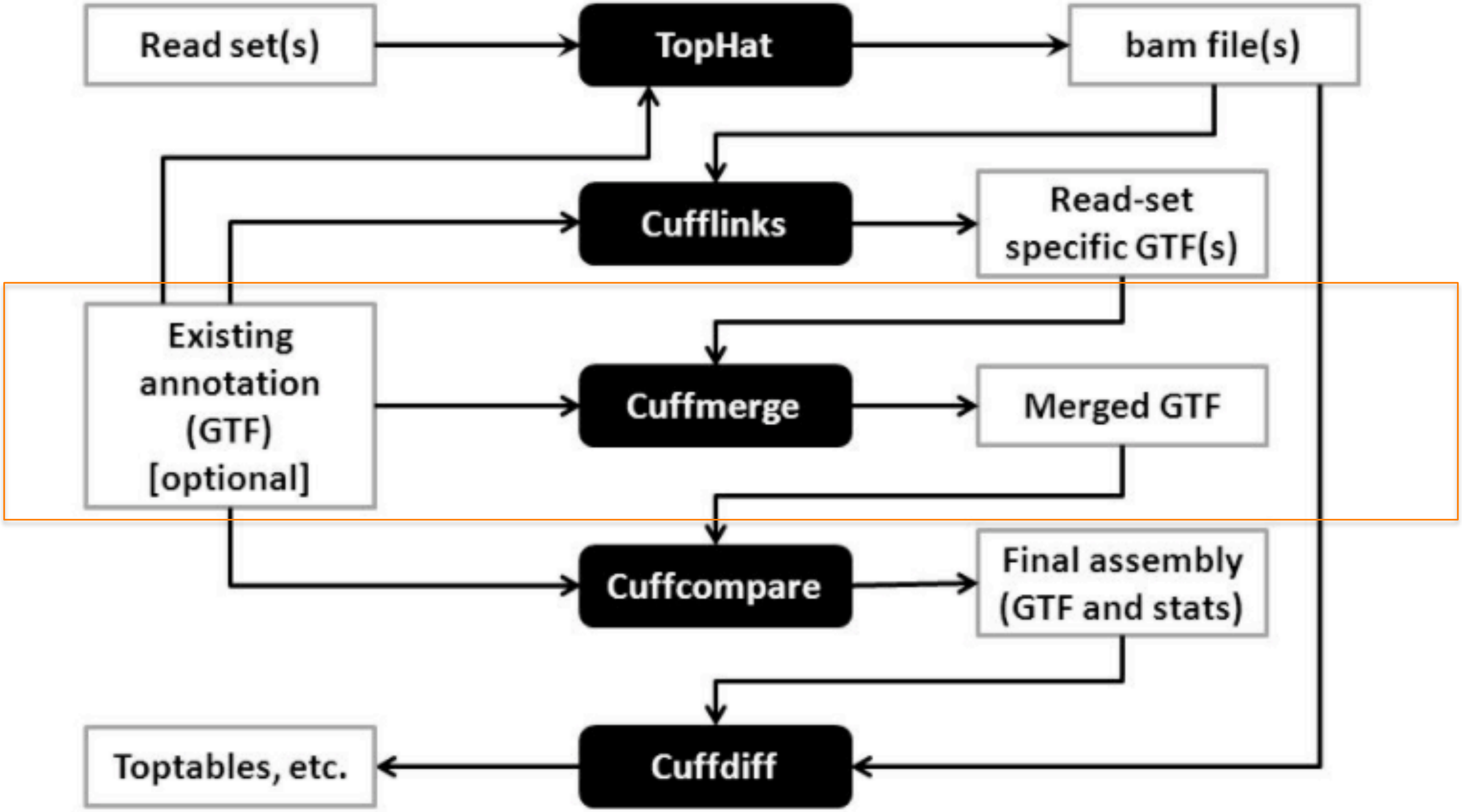
- Cuffmerge produces a GTF file, merged.gtf that merges together the input assemblies.



```
I   Cufflinks   exon   7619    9274    .   +   .   gene_id "XLOC_000006"; transcript_id "TCONS_00000006"; exon_number "1"; oId "CUFF.8.1"; tss_id "TSS6";
I   Cufflinks   exon   11784   12994   .   +   .   gene_id "XLOC_000007"; transcript_id "TCONS_00000007"; exon_number "1"; oId "CUFF.10.1"; tss_id "TSS7";
I   Cufflinks   exon   13665   14555   .   +   .   gene_id "XLOC_000008"; transcript_id "TCONS_00000008"; exon_number "1"; oId "CUFF.11.1"; tss_id "TSS8";
I   Cufflinks   exon   15855   16226   .   +   .   gene_id "XLOC_000009"; transcript_id "TCONS_00000009"; exon_number "1"; oId "CUFF.12.1"; tss_id "TSS9";
I   Cufflinks   exon   18042   18306   .   +   .   gene_id "XLOC_000010"; transcript_id "TCONS_00000010"; exon_number "1"; oId "CUFF.13.1"; tss_id "TSS10";
I   Cufflinks   exon   18349   18974   .   +   .   gene_id "XLOC_000010"; transcript_id "TCONS_00000010"; exon_number "2"; oId "CUFF.13.1"; tss_id "TSS10";
I   Cufflinks   exon   20824   21015   .   +   .   gene_id "XLOC_000011"; transcript_id "TCONS_00000011"; exon_number "1"; oId "CUFF.14.1"; tss_id "TSS11";
I   Cufflinks   exon   21381   22076   .   +   .   gene_id "XLOC_000012"; transcript_id "TCONS_00000012"; exon_number "1"; oId "CUFF.15.1"; tss_id "TSS12";
I   Cufflinks   exon   22132   23050   .   +   .   gene_id "XLOC_000012"; transcript_id "TCONS_00000012"; exon_number "2"; oId "CUFF.15.1"; tss_id "TSS12";
I   Cufflinks   exon   28738   29227   .   +   .   gene_id "XLOC_000013"; transcript_id "TCONS_00000013"; exon_number "1"; oId "CUFF.18.1"; tss_id "TSS13";
I   Cufflinks   exon   29286   29657   .   +   .   gene_id "XLOC_000013"; transcript_id "TCONS_00000013"; exon_number "2"; oId "CUFF.18.1"; tss_id "TSS13";
I   Cufflinks   exon   29764   31069   .   +   .   gene_id "XLOC_000014"; transcript_id "TCONS_00000014"; exon_number "1"; oId "CUFF.19.1"; tss_id "TSS14";
I   Cufflinks   exon   32034   33012   .   +   .   gene_id "XLOC_000015"; transcript_id "TCONS_00000015"; exon_number "1"; oId "CUFF.21.1"; tss_id "TSS15";
I   Cufflinks   exon   33835   34978   .   +   .   gene_id "XLOC_000016"; transcript_id "TCONS_00000016"; exon_number "1"; oId "CUFF.22.1"; tss_id "TSS16";
I   Cufflinks   exon   39416   39848   .   +   .   gene_id "XLOC_000017"; transcript_id "TCONS_00000017"; exon_number "1"; oId "CUFF.24.1"; tss_id "TSS17";
I   Cufflinks   exon   39899   40072   .   +   .   gene_id "XLOC_000017"; transcript_id "TCONS_00000017"; exon_number "2"; oId "CUFF.24.1"; tss_id "TSS17";
I   Cufflinks   exon   40795   41489   .   +   .   gene_id "XLOC_000018"; transcript_id "TCONS_00000018"; exon_number "1"; oId "CUFF.25.1"; tss_id "TSS18";
I   Cufflinks   exon   44644   45468   .   +   .   gene_id "XLOC_000019"; transcript_id "TCONS_00000019"; exon_number "1"; oId "CUFF.28.1"; tss_id "TSS19";
I   Cufflinks   exon   50946   52240   .   +   .   gene_id "XLOC_000020"; transcript_id "TCONS_00000020"; exon_number "1"; oId "CUFF.31.1"; tss_id "TSS20";
I   Cufflinks   exon   52318   53858   .   +   .   gene_id "XLOC_000020"; transcript_id "TCONS_00000020"; exon_number "2"; oId "CUFF.31.1"; tss_id "TSS20";
I   Cufflinks   exon   55059   56308   .   +   .   gene_id "XLOC_000021"; transcript_id "TCONS_00000021"; exon_number "1"; oId "CUFF.32.1"; tss_id "TSS21";
I   Cufflinks   exon   56373   57736   .   +   .   gene_id "XLOC_000022"; transcript_id "TCONS_00000022"; exon_number "1"; oId "CUFF.33.1"; tss_id "TSS22";
I   Cufflinks   exon   62961   63862   .   +   .   gene_id "XLOC_000023"; transcript_id "TCONS_00000023"; exon_number "1"; oId "CUFF.36.1"; tss_id "TSS23";
I   Cufflinks   exon   66219   69821   .   +   .   gene_id "XLOC_000024"; transcript_id "TCONS_00000024"; exon_number "1"; oId "CUFF.38.1"; tss_id "TSS24";
```

How do you find novel transcripts or transcripts that are different from reference annotation from the assembled transcript file?

Cuffcompare



cuffcompare

```
 $ cuffcompare merged_asm/merged.gtf -r GENOME_data/genes.gff3 -o 
cuffcmp
 $ mkdir cuffcompare
 $ mv cuffcmp.* ./cuffcompare/
```

The merged set of transcripts should now exist as file 'merged_asm/merged.gtf'.

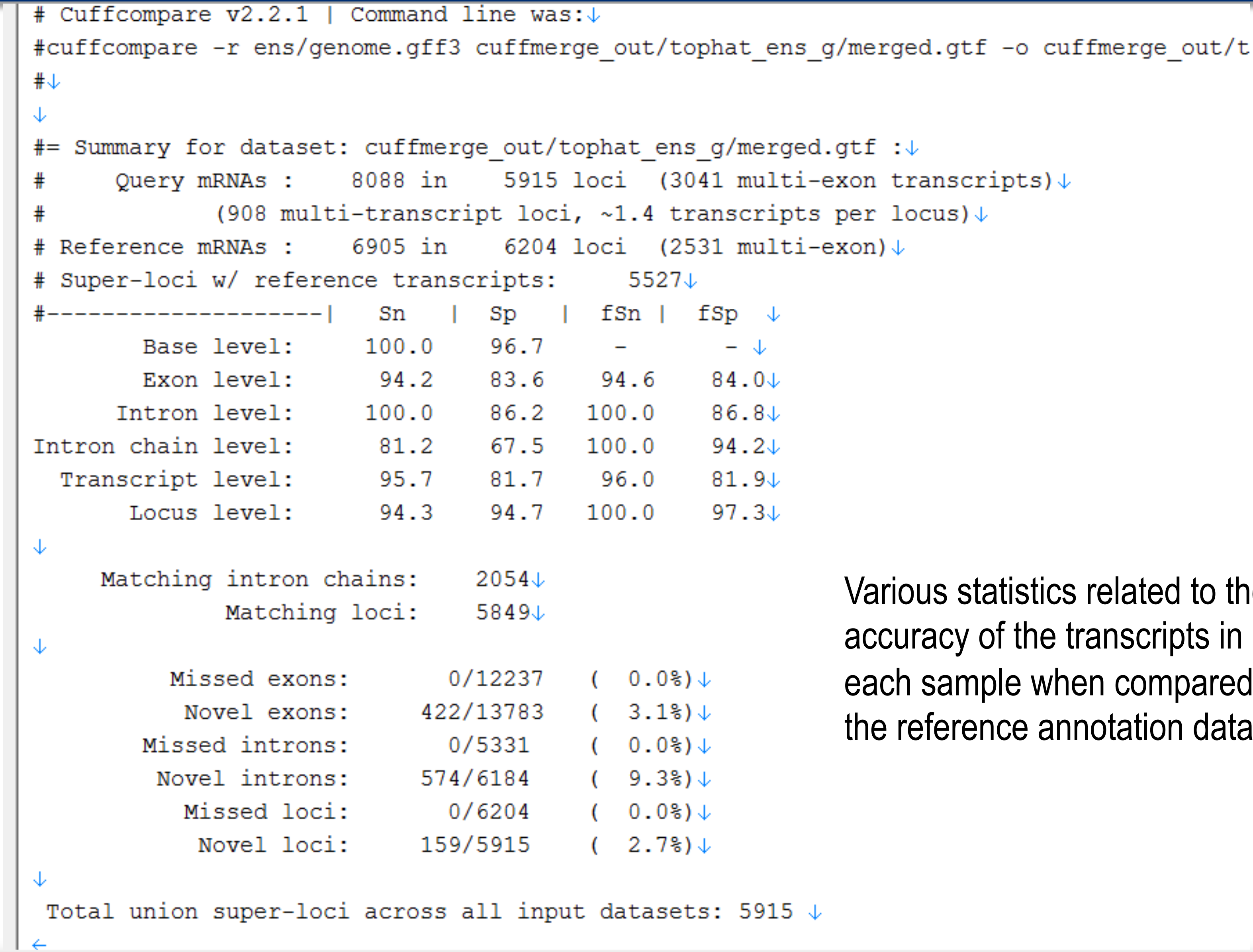
- Compare the assembled (and merged) transcripts to the reference genome.
- This is not necessary if you only used reference annotations (use –G option for cufflinks) since all samples will share the same transcripts.
- If you use –g option for cufflinks, or did not use annotation file for cufflinks, Cuffcompare help you to find differences in transcripts (for novel transcripts, different transcript boundaries, alternative TSS)

Cuffcompare output files

- <outprefix>.stats
  - Various statistics related to the accuracy of the transcripts in each sample when compared to the reference annotation data
- <outprefix>.combined.gtf
  - The “union” of all transfrags in all assemblies.
- <cuff_in>.refmap
  - This tab-delimited file lists, for each reference transcript, which Cufflinks transcripts either fully or partially match it
- <cuff_in>.tmap
  - This tab-delimited file lists the most closely matching reference transcript for each Cufflinks transcript
- <outprefix>.tracking
  - This file matches transcripts between samples

http://cole-trapnell-lab.github.io/cufflinks/cuffcompare/

<outprefix>.stats



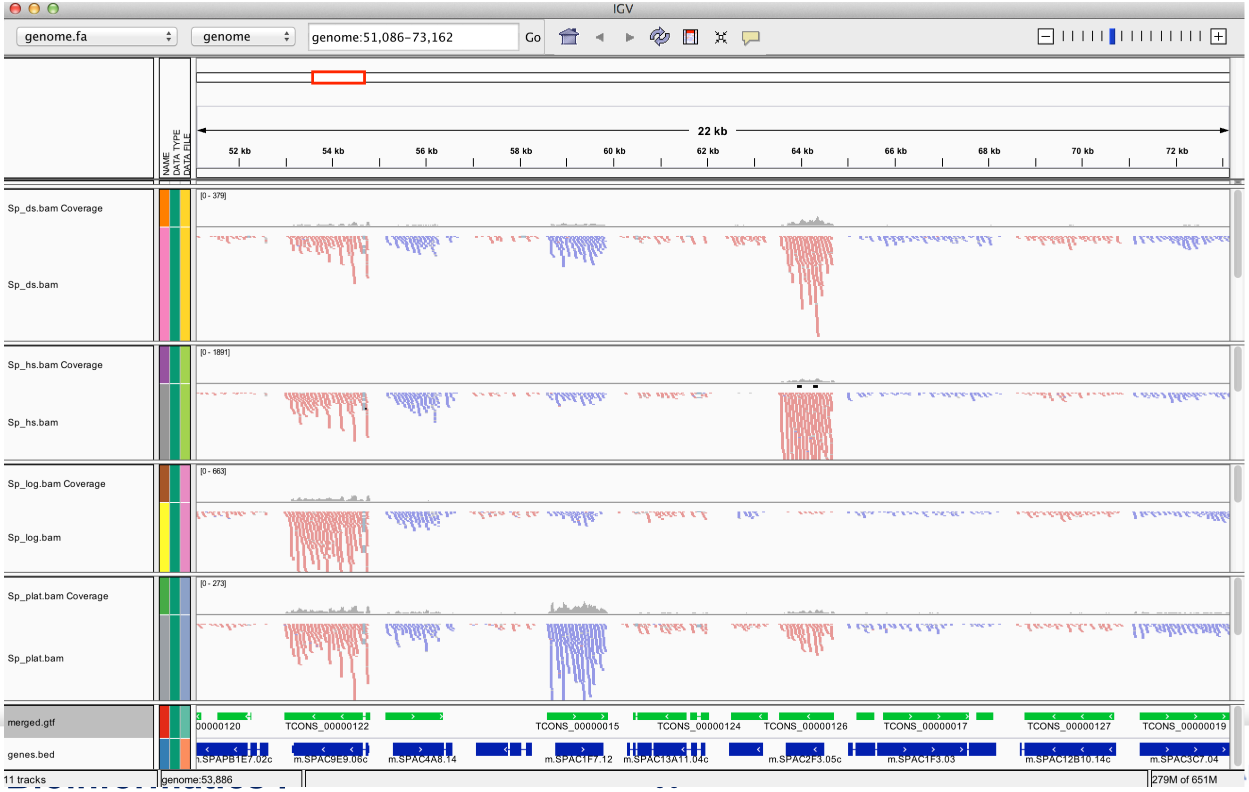
```
# Cuffcompare v2.2.1 | Command line was:↓
#cuffcompare -r ens/genome.gff3 cuffmerge_out/tophat_ens_g/merged.gtf -o cuffmerge_out/t
#↓
↓
#= Summary for dataset: cuffmerge_out/tophat_ens_g/merged.gtf :↓
#     Query mRNAs :    8088 in    5915 loci  (3041 multi-exon transcripts)↓
#            (908 multi-transcript loci, ~1.4 transcripts per locus)↓
# Reference mRNAs :    6905 in    6204 loci  (2531 multi-exon)↓
# Super-loci w/ reference transcripts:     5527↓
#--------------------|   Sn   |  Sp   |  fSn |  fSp  ↓
        Base level:    100.0    96.7     -       - ↓
        Exon level:     94.2    83.6    94.6    84.0↓
      Intron level:    100.0    86.2   100.0    86.8↓
Intron chain level:     81.2    67.5   100.0    94.2↓
  Transcript level:     95.7    81.7    96.0    81.9↓
       Locus level:     94.3    94.7   100.0    97.3↓
↓
     Matching intron chains:    2054↓
              Matching loci:    5849↓
↓
          Missed exons:       0/12237    (  0.0%)↓
           Novel exons:     422/13783    (  3.1%)↓
        Missed introns:       0/5331     (  0.0%)↓
         Novel introns:     574/6184     (  9.3%)↓
           Missed loci:       0/6204     (  0.0%)↓
            Novel loci:     159/5915     (  2.7%)↓
↓
 Total union super-loci across all input datasets: 5915 ↓
←
```

Various statistics related to the accuracy of the transcripts in each sample when compared to the reference annotation data

IGV

- View the reconstructed transcripts and the tophat alignments in IGV

```
 $ igv.sh -g `pwd`/GENOME_data/genome.fa \
  `pwd`/merged_asm/merged.gtf,`pwd`/GENOME_data/genes.bed,`pwd`/
tophat.Sp_ds.dir/Sp_ds.bam,`pwd`/tophat.Sp_hs.dir/Sp_hs.bam,`pwd`/
tophat.Sp_log.dir/Sp_log.bam,`pwd`/tophat.Sp_plat.dir/Sp_plat.bam
```



Identify differentially expressed transcripts using Cuffdiff

```
$ cuffdiff  --no-update-check --library-type fr-firststrand  \
            -o diff_out -b GENOME_data/genome.fa \
            -L Sp_ds,Sp_hs,Sp_log,Sp_plat \
            -u merged_asm/merged.gtf \
            tophat.Sp_ds.dir/Sp_ds.bam \
            tophat.Sp_hs.dir/Sp_hs.bam \
            tophat.Sp_log.dir/Sp_log.bam \
            tophat.Sp_plat.dir/Sp_plat.bam
```

Examine the output files generated in the diff_out/ directory.

Study transcript expression and analyze DE using CummeRbund

Homework: Follow the CummeRbund part and report the results.

- https://github.com/trinityrnaseq/RNASeq_Trinity_Tuxedo_Workshop/wiki/Tuxedo-Genome-Guided-Transcriptome-Assembly-Workshop