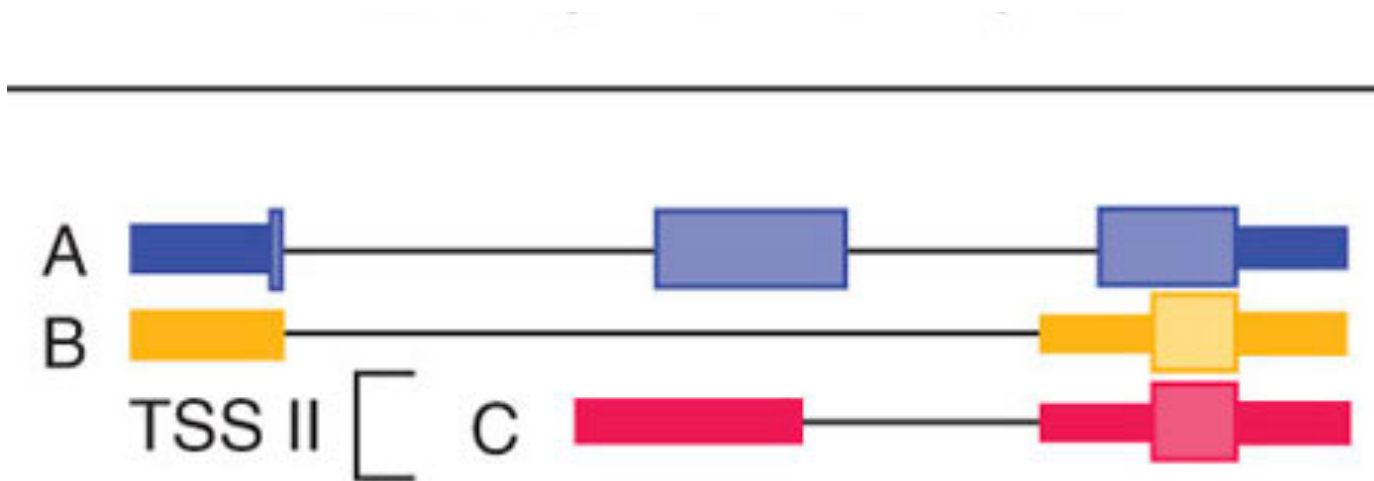


Tuxedo Pipeline for Novel Transcript Discovery

Dhivya Arasappan

What is a gene? What is a transcript?

A gene can have multiple transcripts!



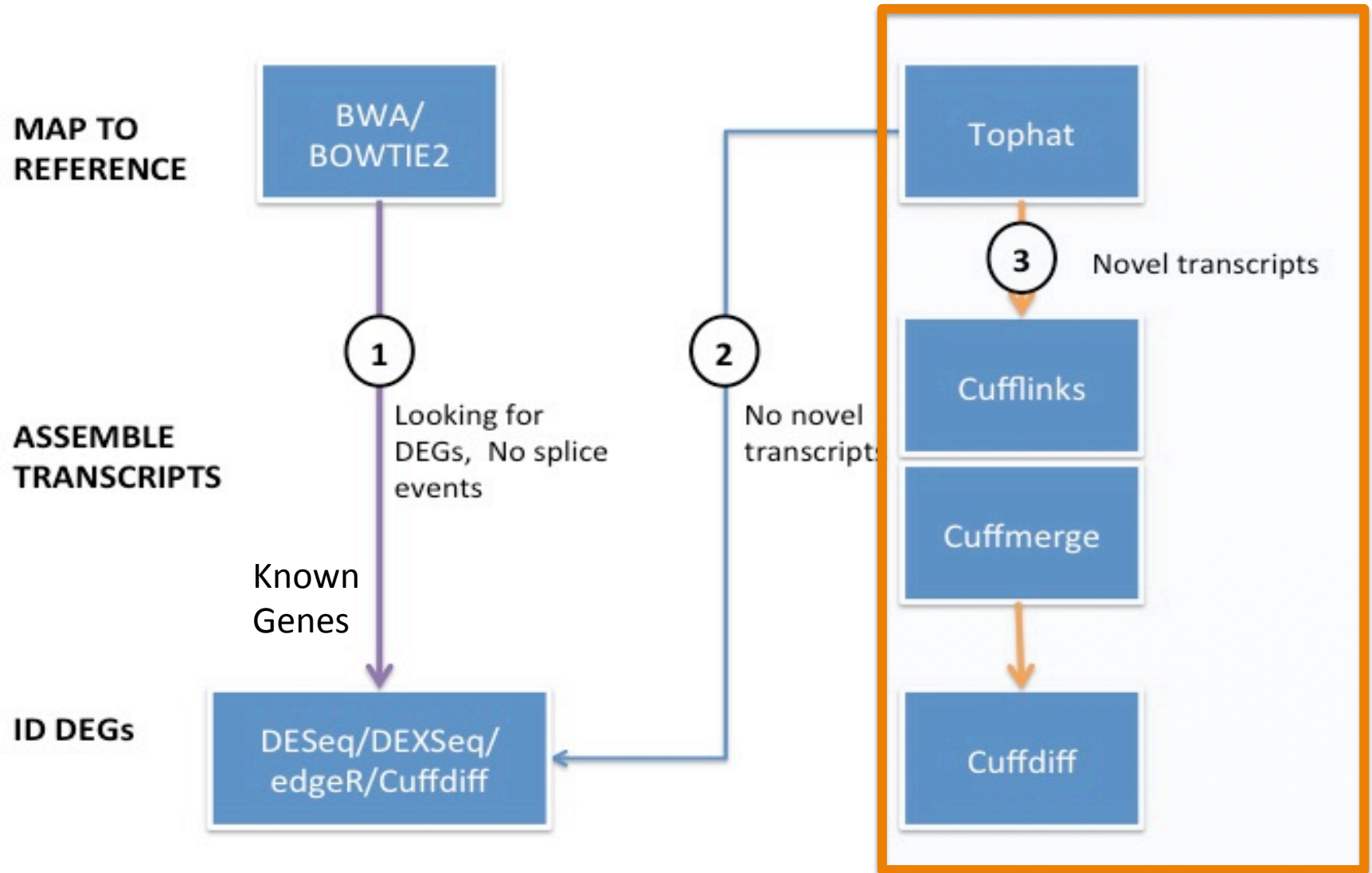
- We want to identify all these transcripts, whether annotated or not.

What? Why? How?

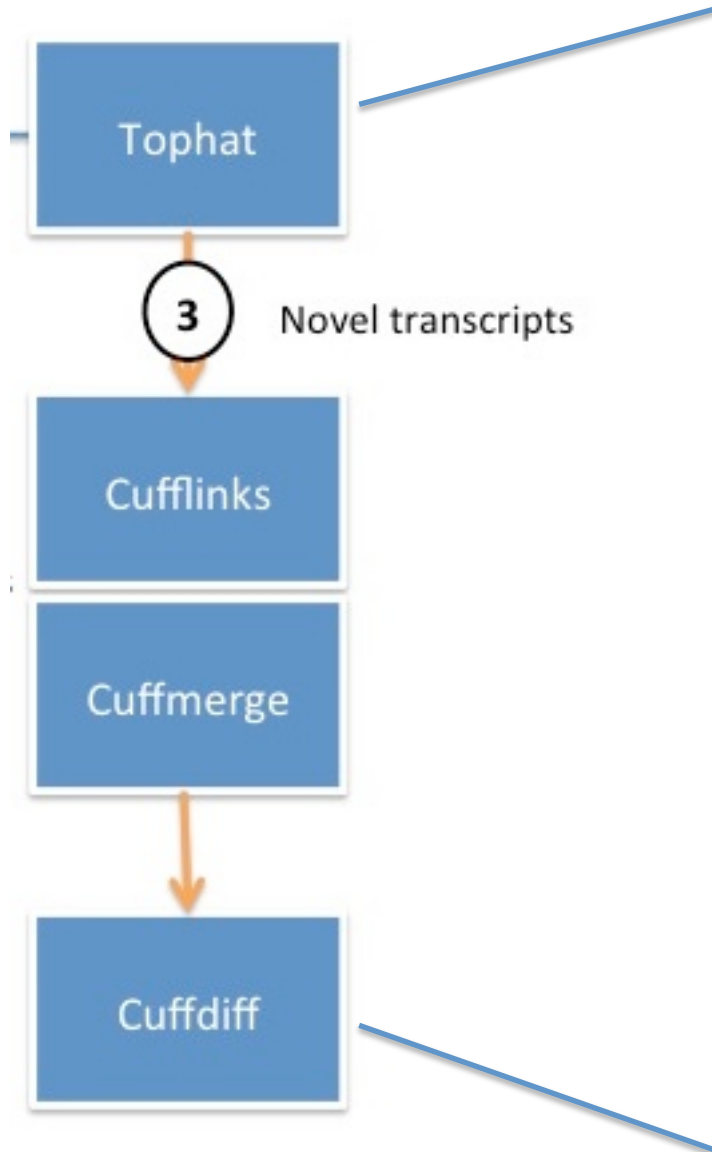
- What are we trying to do?
 - Novel gene and transcript discovery!!
- Why?
 - If our sample condition has novel transcripts not found in annotated transcriptome, that tells us something about what may make our sample unique.
 - If across two conditions, novel transcripts are differentially expressed, those transcripts may be the underlying reasons for the biological differences between our two conditions.
- How?
 - Using the tuxedo pipeline.

Back to the big picture...

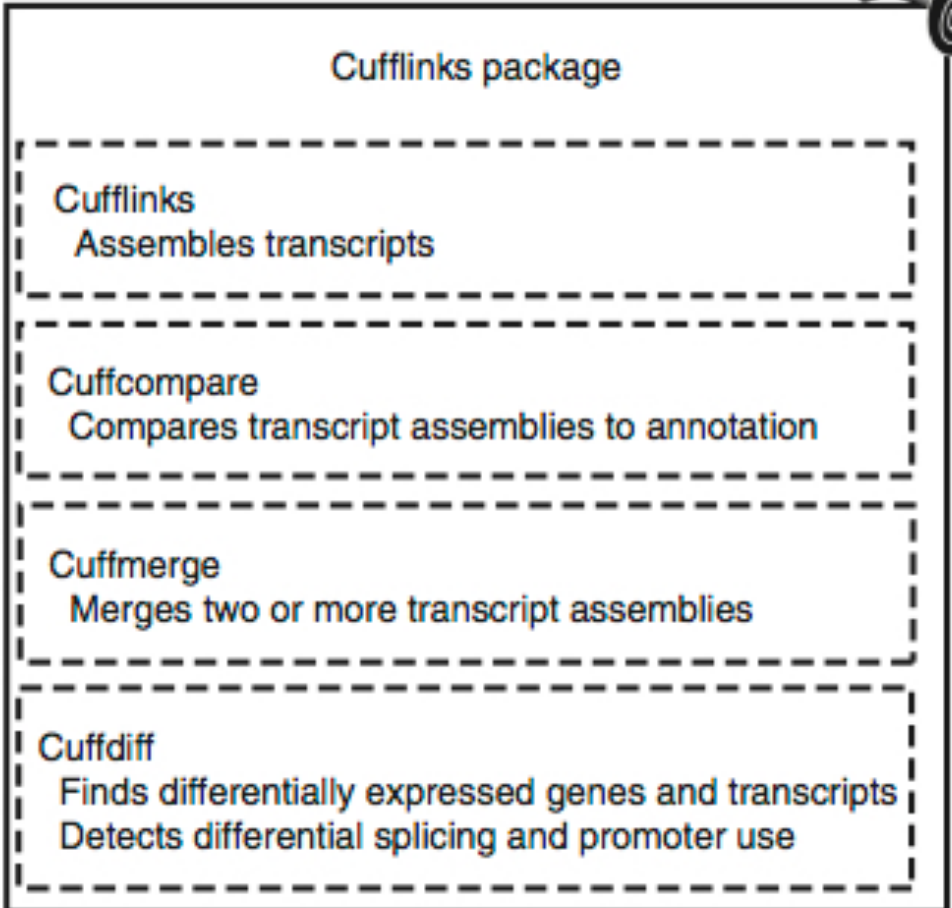
RNA-SEQ ANALYSIS PIPELINES



TUXEDO PIPELINE

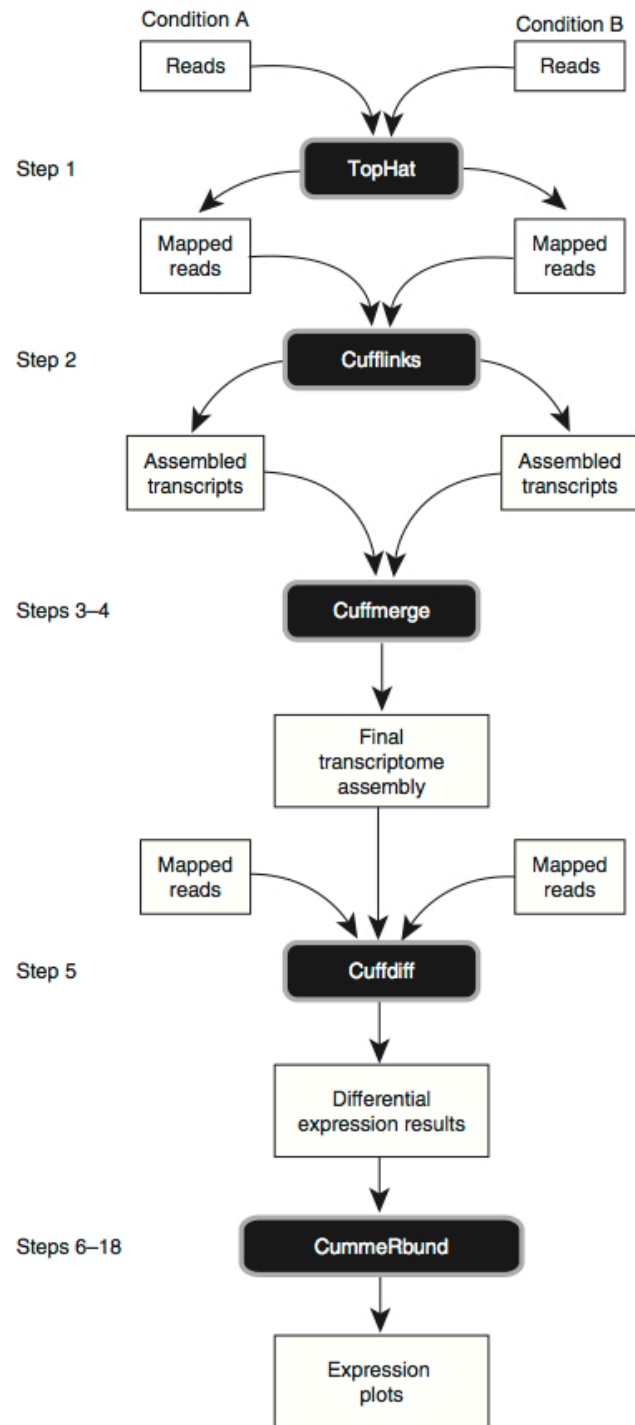


TopHat
Aligns RNA-Seq reads to the genome using Bowtie
Discovers splice sites



Cufflinks package

- Cufflinks
Assembles transcripts
- Cuffcompare
Compares transcript assemblies to annotation
- Cuffmerge
Merges two or more transcript assemblies
- Cuffdiff
Finds differentially expressed genes and transcripts
Detects differential splicing and promoter use



The pipeline is sequential.

Output of one step becomes input of next step.

Figure from:
Trapnell et al, Nature protocols, 2012.

Of course, Tuxedo Pipeline can be run without looking for novel events

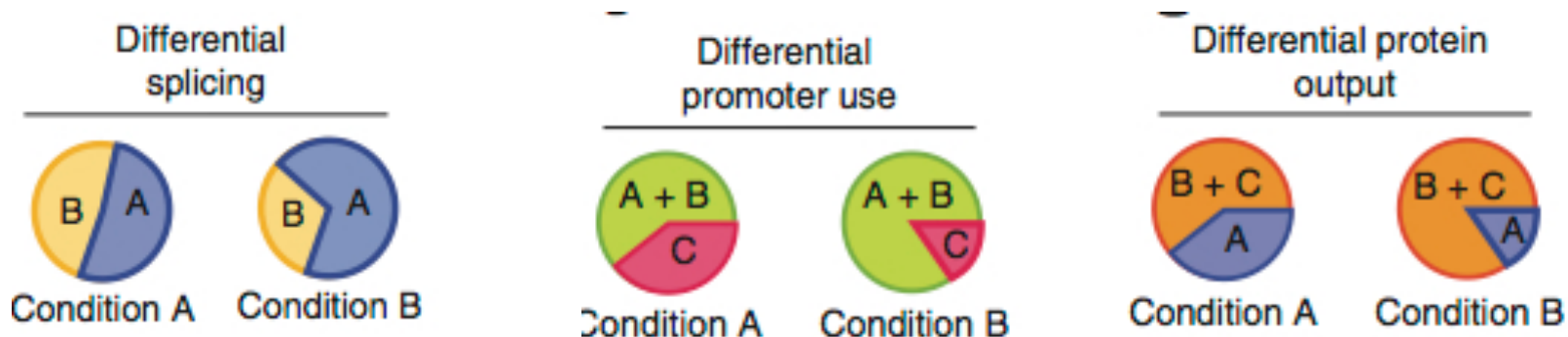
- NO NOVEL JUNCTIONS: Simple differential gene expression analysis against a set of known transcripts.
 - User provides a gff/gtf file containing annotated features. Quantify only the annotated features and id DEGs.
- NOVEL JUNCTIONS ALSO: In addition to known transcripts, novel transcripts should be explored.
 - User provides a gff/gtf file containing annotated features. But you also allow the search for novel variants as well. Both annotated and novel variants are quantified and DEGs are identified.
- ONLY NOVEL/DE NOVO JUNCTIONS: No gff/gtf file is provided. Using just the read data and the genome reference, construct *de novo* transcripts, quantify them and id DEGs.

What do we get at the end of running this pipeline?

A view of how the transcriptome is different between condition C1 and condition C2

- Both in terms of annotated genes and transcripts.
- And novel genes and transcripts

Differential gene expression and so much more...



STEP 1: TOPHAT

What does Tophat do?

Tophat maps your data to your reference in a transcriptome-aware manner, that will also identify junctions. We've already looked at how you can turn on and off its ability to identify novel junctions.

--no-novel-juncs	Only look for reads across junctions indicated in the supplied GFF file.
-G <GTF/GFF3 file>	Supply TopHat with a set of gene model annotations and/or known transcripts, as a GTF 2.2 or GFF3 formatted file.

STEP 2: CUFFLINKS

- What does cufflinks do?
 - **TRANSCRIPT ASSEMBLY** (also referred to as transcriptome reconstruction)
 - Let's see first what that entails.

Why transcript assembly?

Transcript assembly = assembly of mapped reads into transcriptional units.

Why?

- Define a precise map of all transcripts expressed in a sample.
- How does our transcriptome look in comparison to the known transcriptome?
- Look for novel transcripts between conditions/samples.
- Look for differences in expression for these novel transcripts between conditions/samples.

Why is transcript assembly hard?

Difficult to tell which read came from which transcript

- - Many Short reads, many transcripts!
- Transcripts are expressed in different amounts. So, coverage of reads can be vastly different.
- Reads can come from mature mRNA (exons only) and precursor RNA (containing partial introns).

Table 1 | Selected list of RNA-seq analysis programs

Class	Category	Package	Notes	Uses	Input
Read mapping					
Unspliced aligners ^a	Seed methods	Short-read mapping package (SHRiMP) ⁴¹ Stampy ³⁹	Smith-Waterman extension Probabilistic model	Aligning reads to a reference transcriptome	Reads and reference transcriptome
	Burrows-Wheeler transform methods	Bowtie ⁴³ BWA ⁴⁴	Incorporates quality scores		
Spliced aligners	Exon-first methods	MapSplice ⁵² SpliceMap ⁵⁰ TopHat ⁵¹	Works with multiple unspliced aligners Uses Bowtie alignments	Aligning reads to a reference genome. Allows for the identification of novel splice junctions	Reads and reference genome
	Seed-extend methods	GSNAP ⁵³ QPALMA ⁵⁴	Can use SNP databases Smith-Waterman for large gaps		
Transcriptome reconstruction					
Genome-guided reconstruction	Exon identification	G.Mor.Se	Assembles exons	Identifying novel transcripts using a known reference genome	Alignments to reference genome
	Genome-guided assembly	Scripture ²⁸ Cufflinks ²⁹	Reports all isoforms Reports a minimal set of isoforms		
Genome-independent reconstruction	Genome-independent assembly	Velvet ⁶¹ TransABySS ⁵⁶	Reports all isoforms	Identifying novel genes and transcript isoforms without a known reference genome	Reads
Expression quantification					
Expression quantification	Gene quantification	Alexa-seq ⁴⁷	Quantifies using differentially included exons	Quantifying gene expression	Reads and transcript models
		Enhanced read analysis of gene expression (ERANGE) ²⁰ Normalization by expected uniquely mappable area (NEUMA) ⁸²	Quantifies using union of exons Quantifies using unique reads		
	Isoform quantification	Cufflinks ²⁹ MISO ³³ RNA-seq by expectation maximization (RSEM) ⁶⁹	Maximum likelihood estimation of relative isoform expression	Quantifying transcript isoform expression levels	Read alignments to isoforms
Differential expression		Cuffdiff ²⁹ DegSeq ⁷⁹ EdgeR ⁷⁷	Uses isoform levels in analysis Uses a normal distribution	Identifying differentially expressed genes or transcript isoforms	Read alignments and transcript models
		Differential Expression analysis of count data (DESeq) ⁷⁸ Myrna ⁷⁵	Cloud-based permutation method		

Figure :
Garber et al, Nature Methods, 2011

Most commonly used, if you have a genome.

Less resource-intensive

We'll call this coverage islands method

Transcriptome reconstruction				
Genome-guided reconstruction	Exon identification	G.Mor.Se	Assembles exons	Identifying novel transcripts using a known reference genome
	Genome-guided assembly	Scripture ²⁸ Cufflinks ²⁹	Reports all isoforms Reports a minimal set of isoforms	
Genome-independent reconstruction	Genome-independent assembly	Velvet ⁶¹ TransABySS ⁵⁶	Reports all isoforms	Identifying novel genes and transcript isoforms without a known reference genome

If you don't have a genome.

If you believe your sample has major rearrangements

More CPU and RAM intensive

Figure :
Garber et al, Nature Methods, 2011

Genome guided transcript assembly

Different assembly methods

- **Coverage islands**

- ID putative exons by looking for coverage islands.
- Older method, were meant for shorter read lengths.
- G.MorSe

- **Exon first approach**

- Directly uses mappings of spliced reads to reconstruct transcriptome.
- Uses graph topology.
- **Cufflinks (part of tuxedo suite)**, scripture

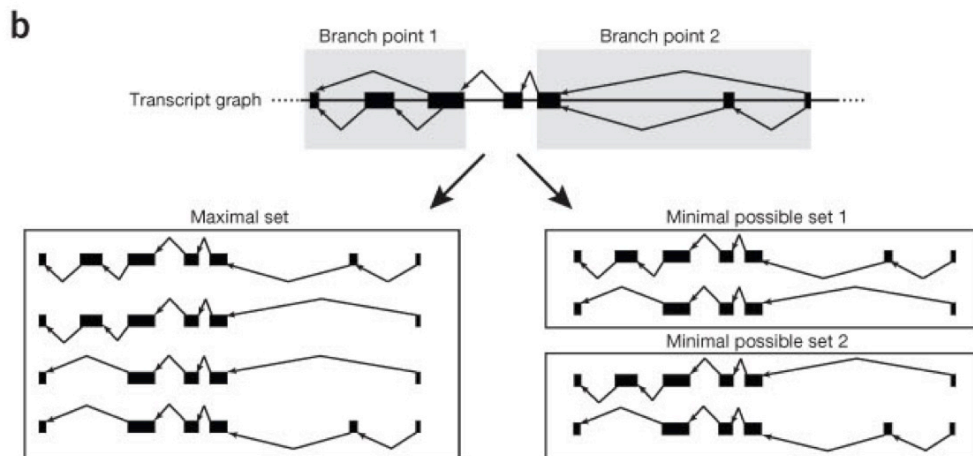
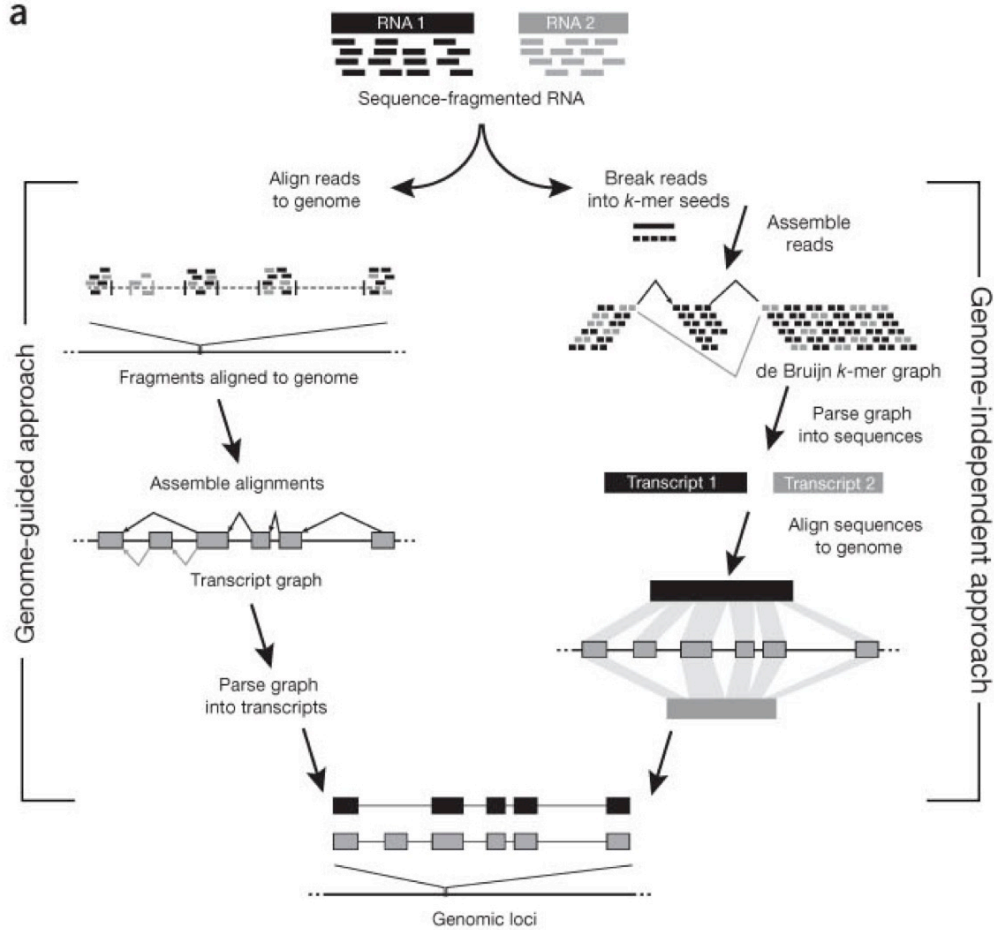
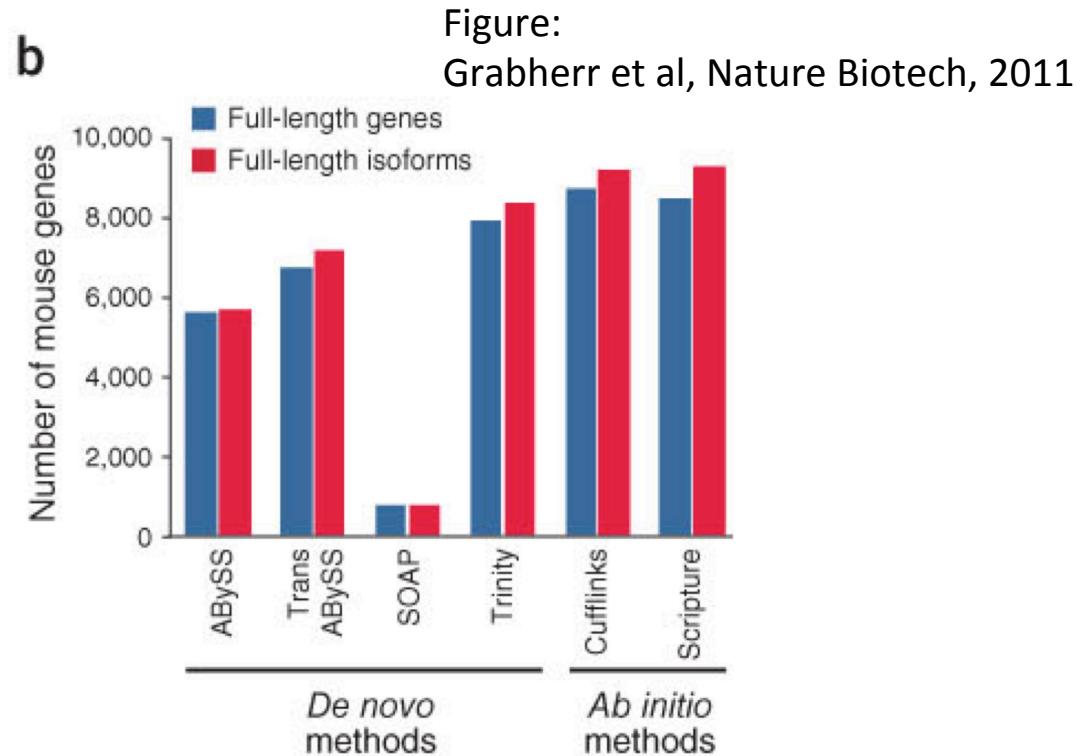
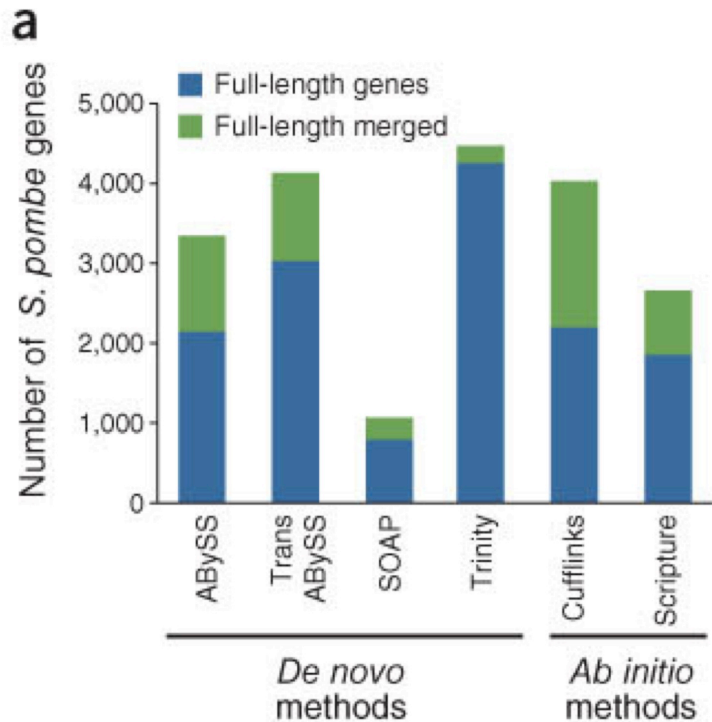


Figure :
Garber et al, Nature Methods, 2011

How do these tools compare?



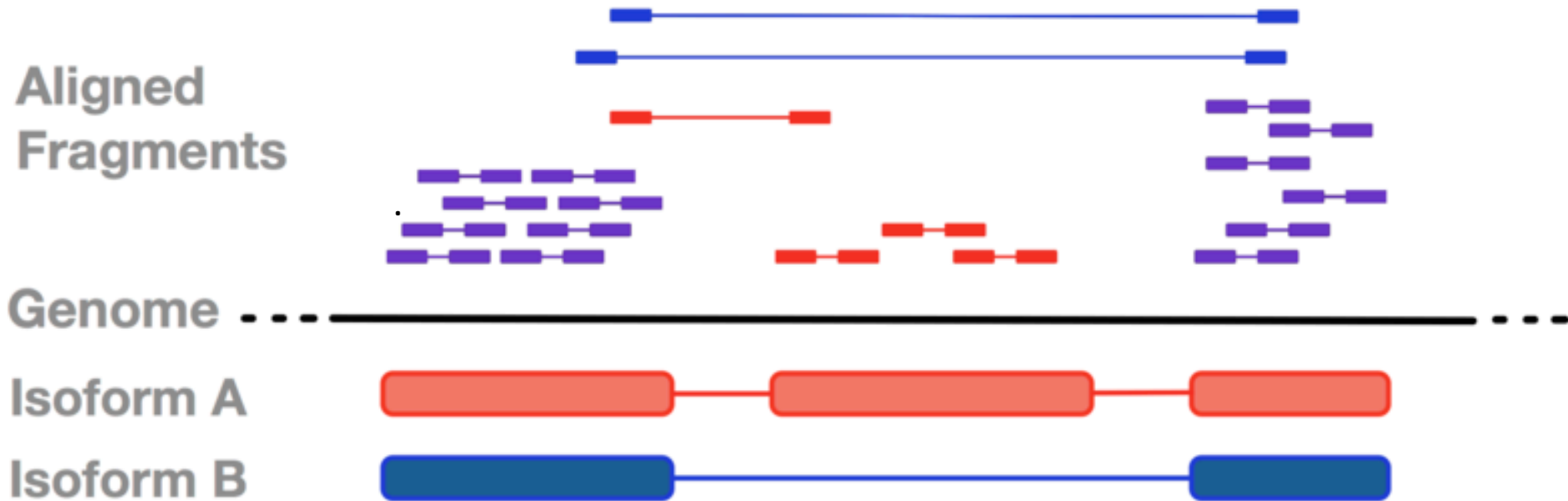
Program combination	vs. orthology annotation		vs. EST annotation	
	Base-level accuracy (%) ¹	Confirmed junctions (%) ¹	Base-level accuracy (%) ¹	Confirmed junctions (%) ¹
TopHat + Cufflinks	83.9	75.8	68.9	63.0
GSNAP + Cufflinks	79.4	71.2	65.7	58.4
GSNAP + Cufflinks (subsample ²)	80.3	72.7	60.2	66.3
TopHat + Scripture	70.3	67.9	60.8	62.5

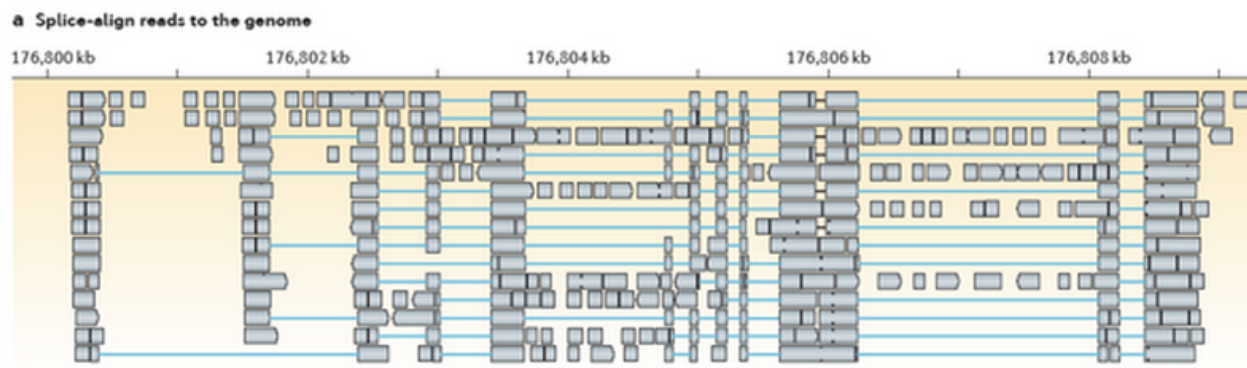
Figure:
Palmieri et al,
PLOS One, 2012

¹Base level accuracy and percentage of confirmed junctions with different combinations of mapper and assembler on the sample ps94 males compared to the orthology annotation and the EST annotation (²based on 48 M reads).

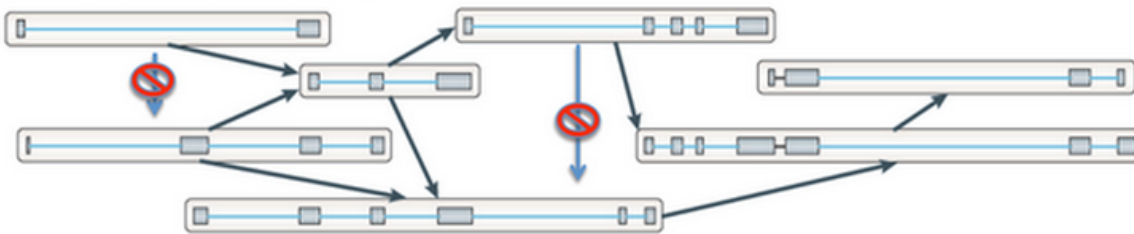
How does Cufflinks do transcript assembly

Exon first method!

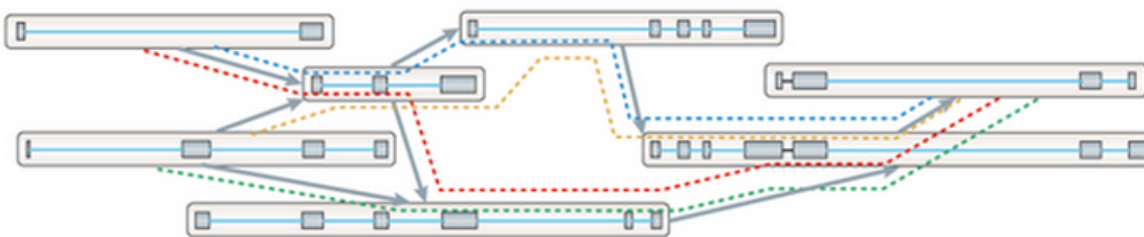




b Build a graph representing alternative splicing events



c Traverse the graph to assemble variants



d Assembled isoforms

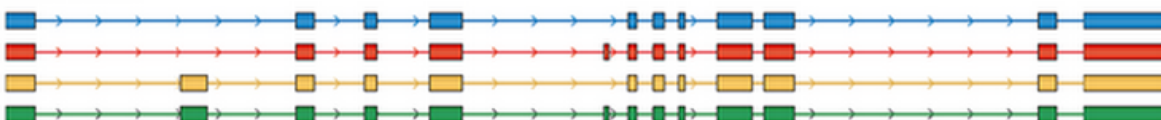
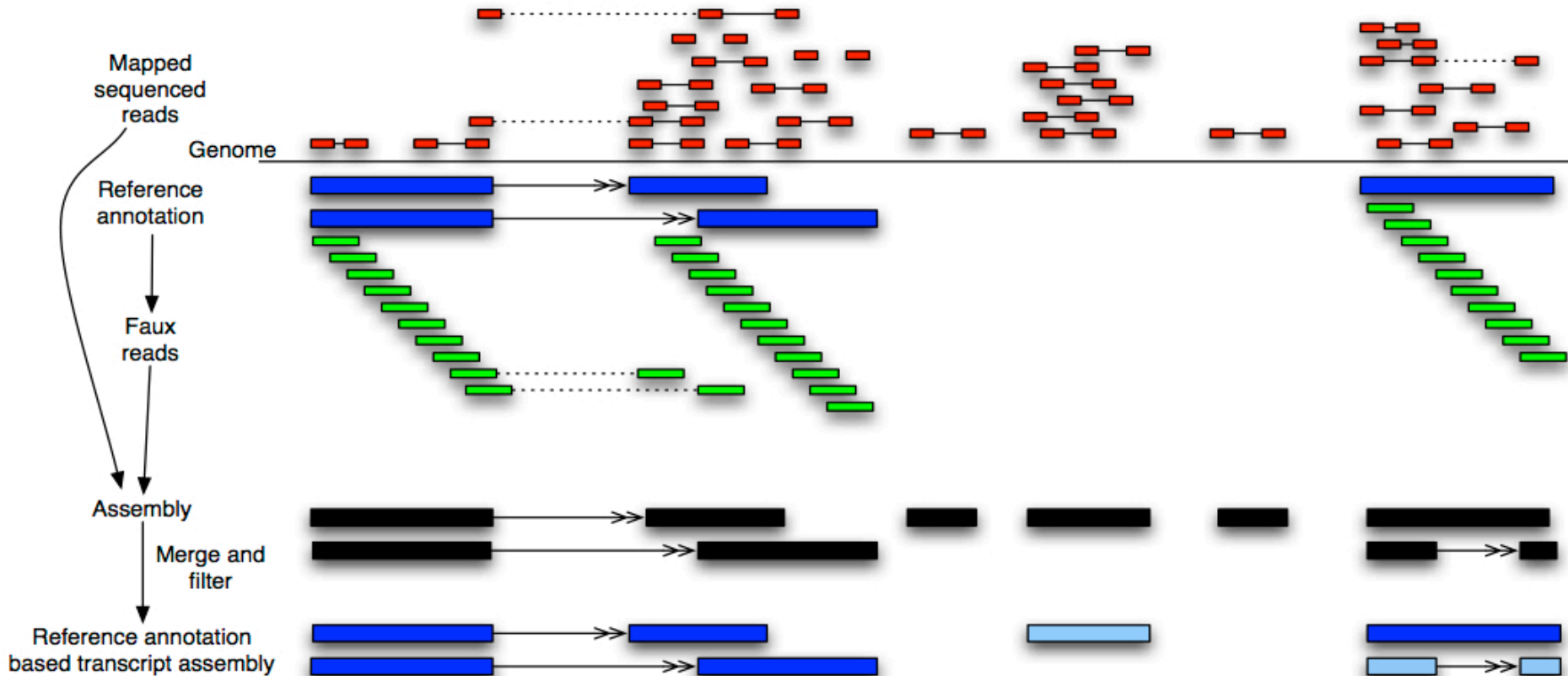


Figure :
http://sourceforge.net/projects/trinityrnaseq/files/misc/RNASEQ_WORKSHOP/rnaseq_workshop_slides.pdf

RABT

- Reference annotation based transcript assembly (RABT)
 - Uses existing annotation to guide assembly of transcripts.



After assembly

- Calculates abundance for these assembled transcripts.
- Normalized using FPKM (Fragments Per Kilobase of Exon Per Million) (variation of RPKM)
 - RPKM normalizes for **transcript length variations** and **sequencing depth**.
 - $RPKM = (\text{No. of Mapped reads} * 10^9) / (\text{length of transcript} * \text{total no. of reads})$
 - FPKM just exchanges reads with fragments.

General syntax for cufflinks command

```
cufflinks [options] <accepted_hits.bam>
```

Some of the important options:

- p/--num-threads

- G/--GTF (quantify only annotated transcripts)

- g/--GTF-guide (both annotated and novel transcripts)

- b/--frag-bias-correct

- u/--multi-read-correct

General syntax for cufflinks command

-b/--frag-bias-correct

When quantifying abundance, corrects for sequence-specific bias at the ends of reads by 'learning' from the data.

-u/--multi-read-correct

By default, if a read maps to 2 genes it will count as 50% (half a read) towards each gene.

With this flag, it handles this question in a more fine-tuned manner.

Let's look at some results from a cufflinks transcript assembly

- Input:
 - Tophat mapped results (bam files)
 - Transcriptome annotation (genes.gtf)
- Let's **SWITCH TO THE WIKI** for instructions on looking at these results...

STEP 3: CUFFCOMPARE

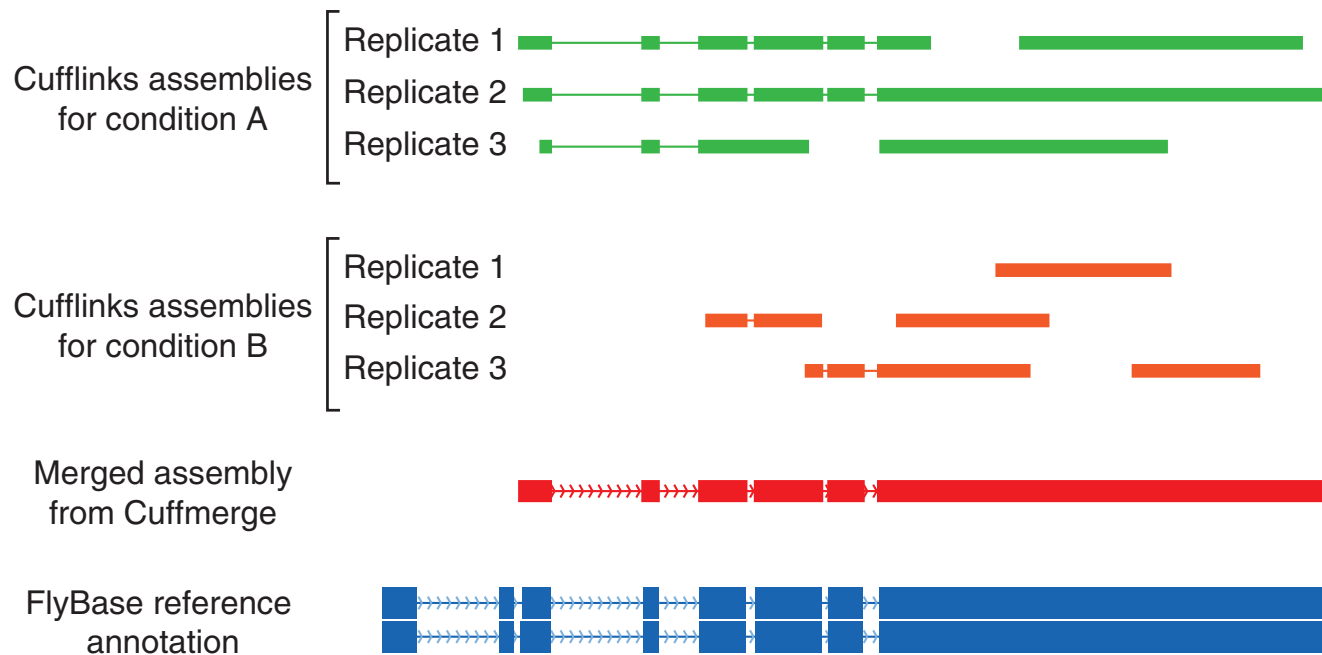
- Compares your assembled transcripts to a reference annotation.
- Input:
 - cufflinks transcripts.gtf files
 - Annotation file (genes.gtf)
- Output:
 - *.tracking file- outputs which assembled transcripts match transcripts in the annotation and how they match.

STEP 3: CUFFCOMPARE

- SWITCH TO THE WIKI for instructions on viewing these results

STEP 4: CUFFMERGE

- Cuffmerge is used to merge all the transcripts that cufflinks assembled into one file.



STEP 4: CUFFMERGE

- Input: All cufflinks assembly files (in gtf format)
- Output: merged.gtf
 - Your very own gtf file, containing all the transcripts found in your samples (both novel and otherwise).
- **SWITCH TO THE WIKI** for instructions on viewing these results

STEP 5: CUFFDIFF

- Calculates differential expression!
- Input:
 - Our newly created merged.gtf file or A gtf file we downloaded (genes.gtf)
 - Mapped bam files
- Counts the number of fragments(reads) generated by each isoform to obtain isoform-level expression.
- Calculates difference in isoform-level expression among conditions.
- If the chance of seeing this difference is small enough under the chosen statistical model, it is deemed significantly differentially expressed.

Other differential expression tools vs cuffdiff

Others	Cuffdiff
Raw count method for assigning counts to genes	Isoform deconvolution method for assigning counts to genes
Count the reads mapping to exons of each gene/normalization factor = expression for gene	Count the reads that map to each isoform of the gene/normalization factor = expression for gene
If all isoforms of the gene are up/down, works fine	If all isoforms of the gene are up/down, works fine
If some isoforms of the gene are up and some are down, inaccurate results	If some isoforms of the gene are up and some are down, works fine

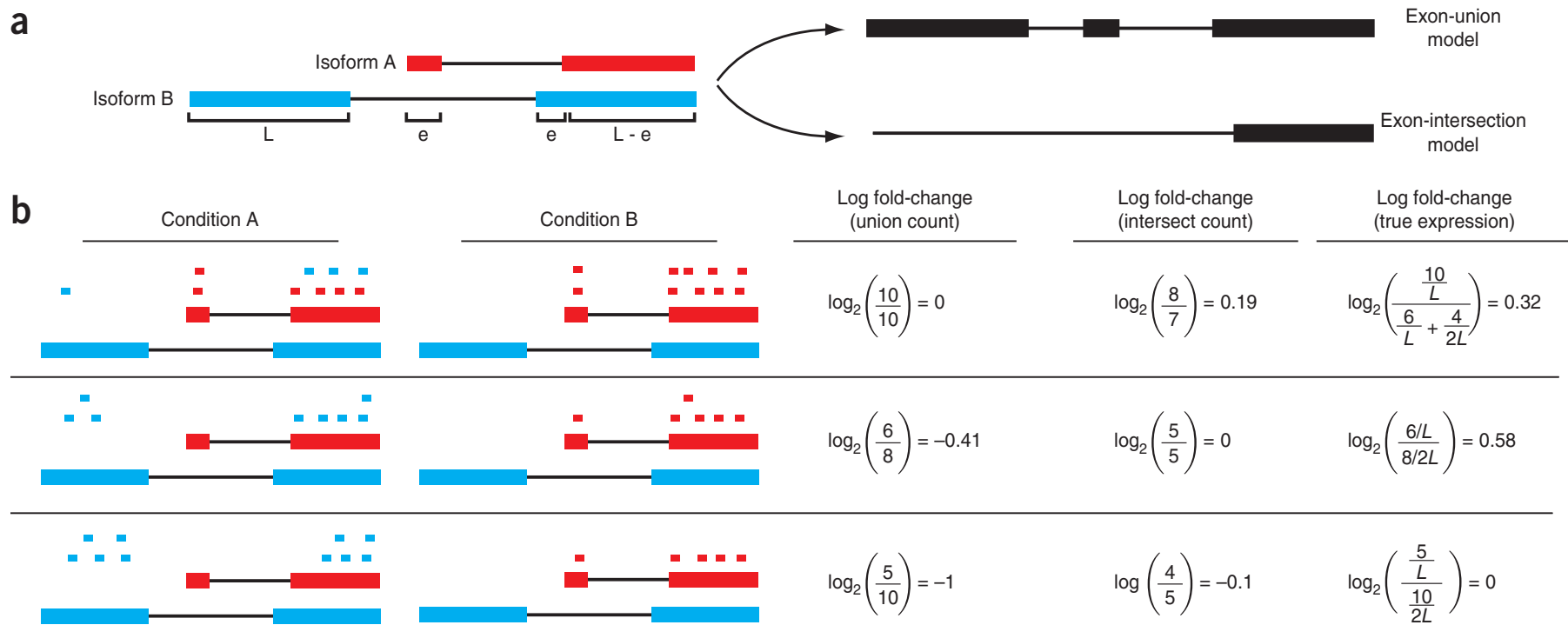
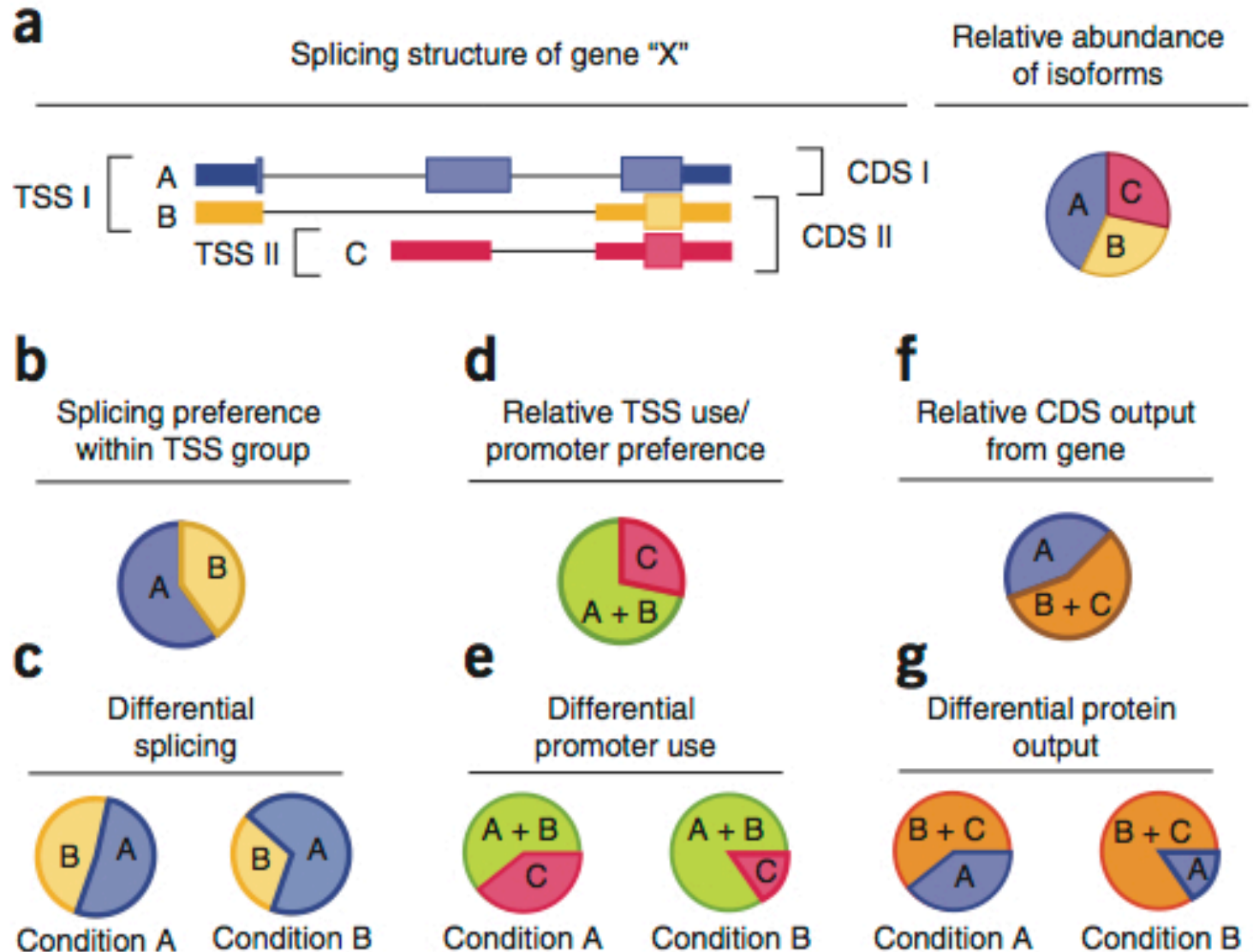


Figure 1 Changes in fragment count for a gene does not necessarily equal a change in expression. **(a)** Simple read-counting schemes sum the fragments incident on a gene's exons. The exon-union model counts reads falling on any of a gene's exons, whereas the exon-intersection model counts only reads on constitutive exons. **(b)** Both of the exon-union and exon-intersection counting schemes may incorrectly estimate a change in expression in genes with multiple isoforms. The true expression is estimated by the sum of the length-normalized isoform read counts. The discrepancy between a change in the union or intersection count and a change in gene expression is driven by a change in the abundance of the isoforms with respect to one another. In the top row, the gene generates the same number of reads in conditions A and B, but in condition B, all of the reads come from the shorter of the two isoforms, and thus the true expression for the gene is higher in condition B. The intersection count scheme underestimates the true change in gene expression, and the union scheme fails to detect the change entirely. In the middle row, the intersection count fails to detect a change driven by a shift in the dominant isoform for the gene. The union scheme detects a shift in the wrong direction. In the bottom row, the gene's expression is constant, but the isoforms undergo a complete

STEP 5: CUFFDIFF

Figure from: Differential analysis of gene regulation at transcript resolution with rNA-seq, Trapnell et al, Nature Biotechnology, 2013



STEP 5: CUFFDIFF

- SWITCH TO THE WIKI for instructions on viewing these results

Limitations of the Tuxedo Pipeline

- A Reference is needed.
- Not quick.
 - Step 1, align the RNA-seq reads to the genome: ~6 h
 - Steps 2–4, assemble expressed genes and transcripts: ~6 h
 - Step 5, identify differentially expressed genes and transcripts: ~6 h

If you don't have a genome:

- De novo transcriptome assembly using trinity.
- Map your data to this to calculate gene expression changes.

DESeq/edgeR output vs Tuxedo pipeline output

- Yesterday we generated differential expressed genes too. So, why the big fuss?
 - They were all from annotated genes. So, they all has flybase ids.
 - Now our output has genes with ids ‘CUFF...’ - they are novel.
 - In addition to differential gene expression, we also have results for differential regulation.
 - We also have results telling us where our novel transcripts are with respect to the annotated ones.