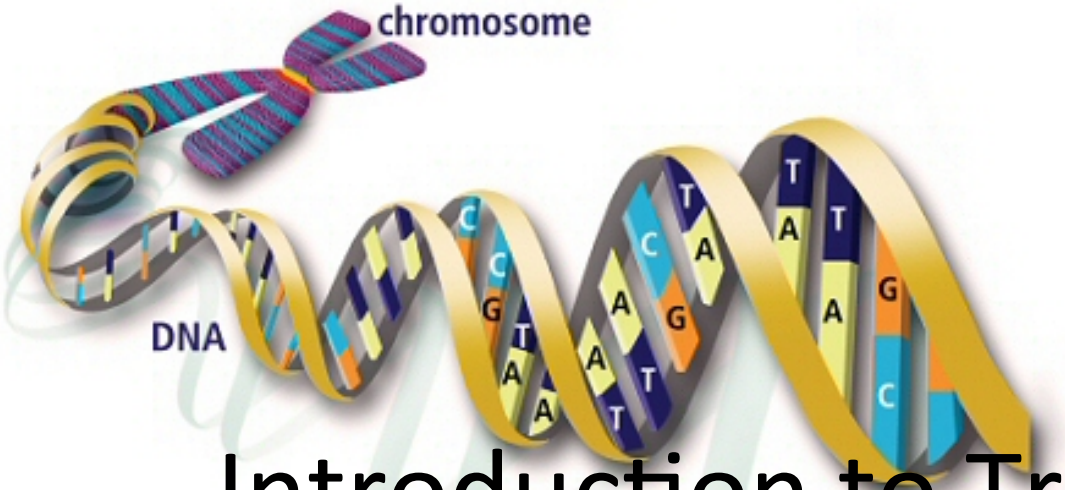


Introduction to Transcriptome Assembly

Panu Somervuo

University of Helsinki

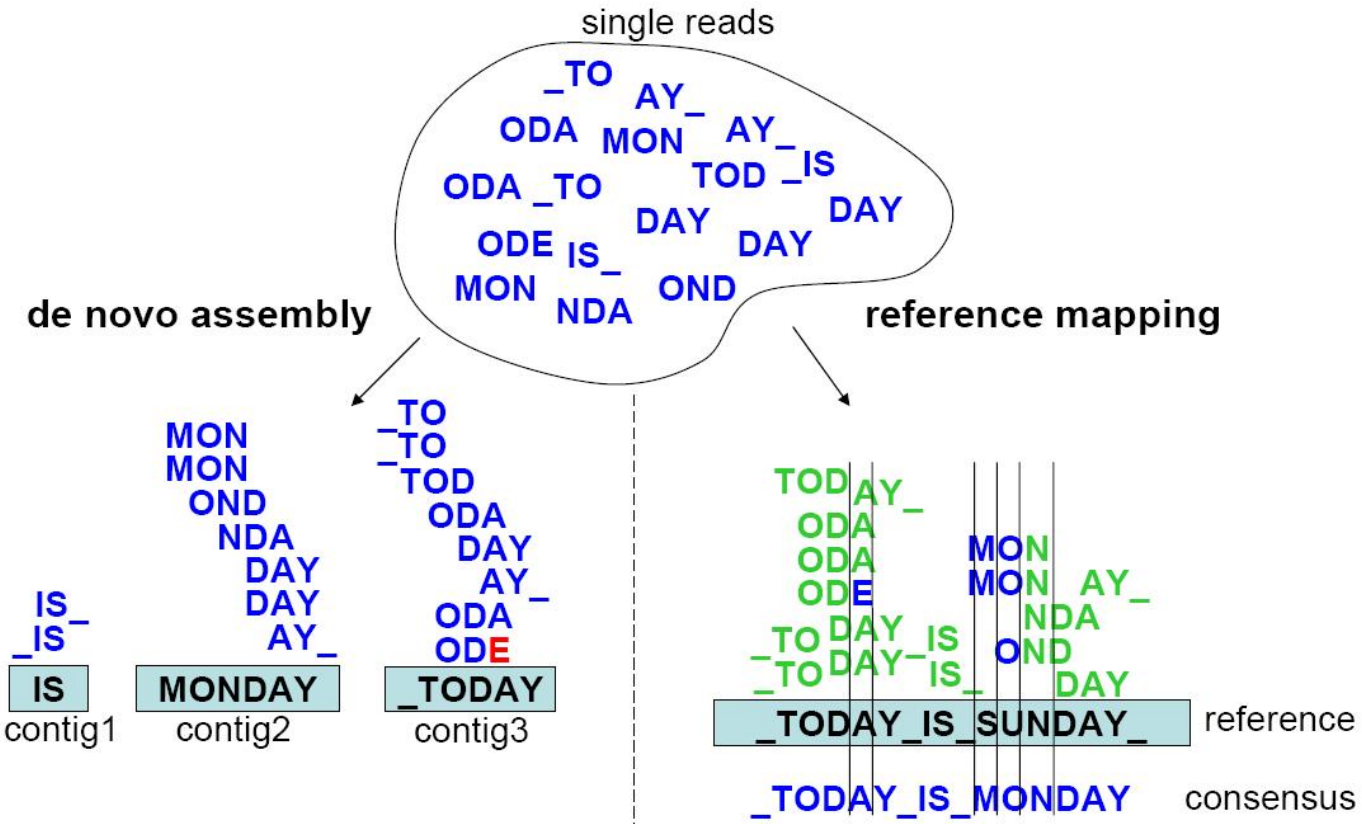
Holm Group

Why transcriptome assembly?

- Finding new transcripts

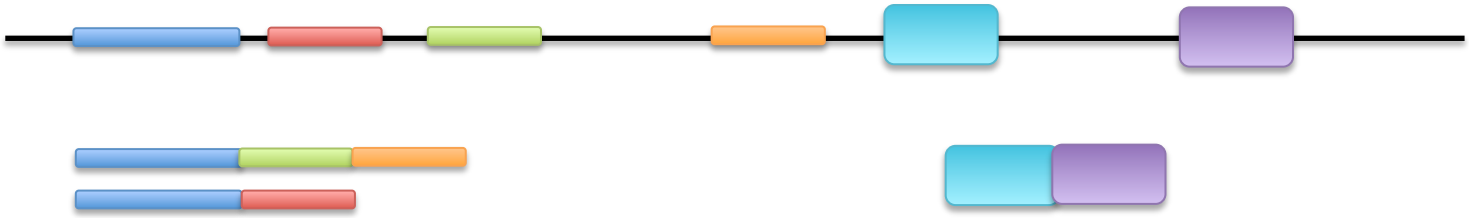
- Only exons: shorter than full genome
  - coding regions: target of interest?
  - transcriptome assembly easier than full genome assembly?

How: mapping vs de novo



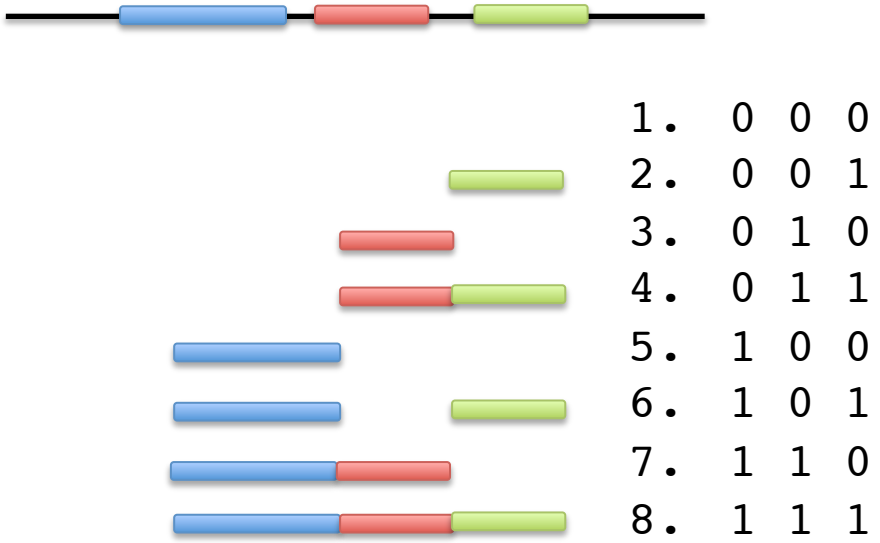
Differences between genome and transcriptome assembly

- Genome assembly:
  - uniform coverage
  - linear sequence
- Transcriptome assembly:
  - abundance differences
  - alternative splicing
  - need for segmentation



Combinatorial complexity

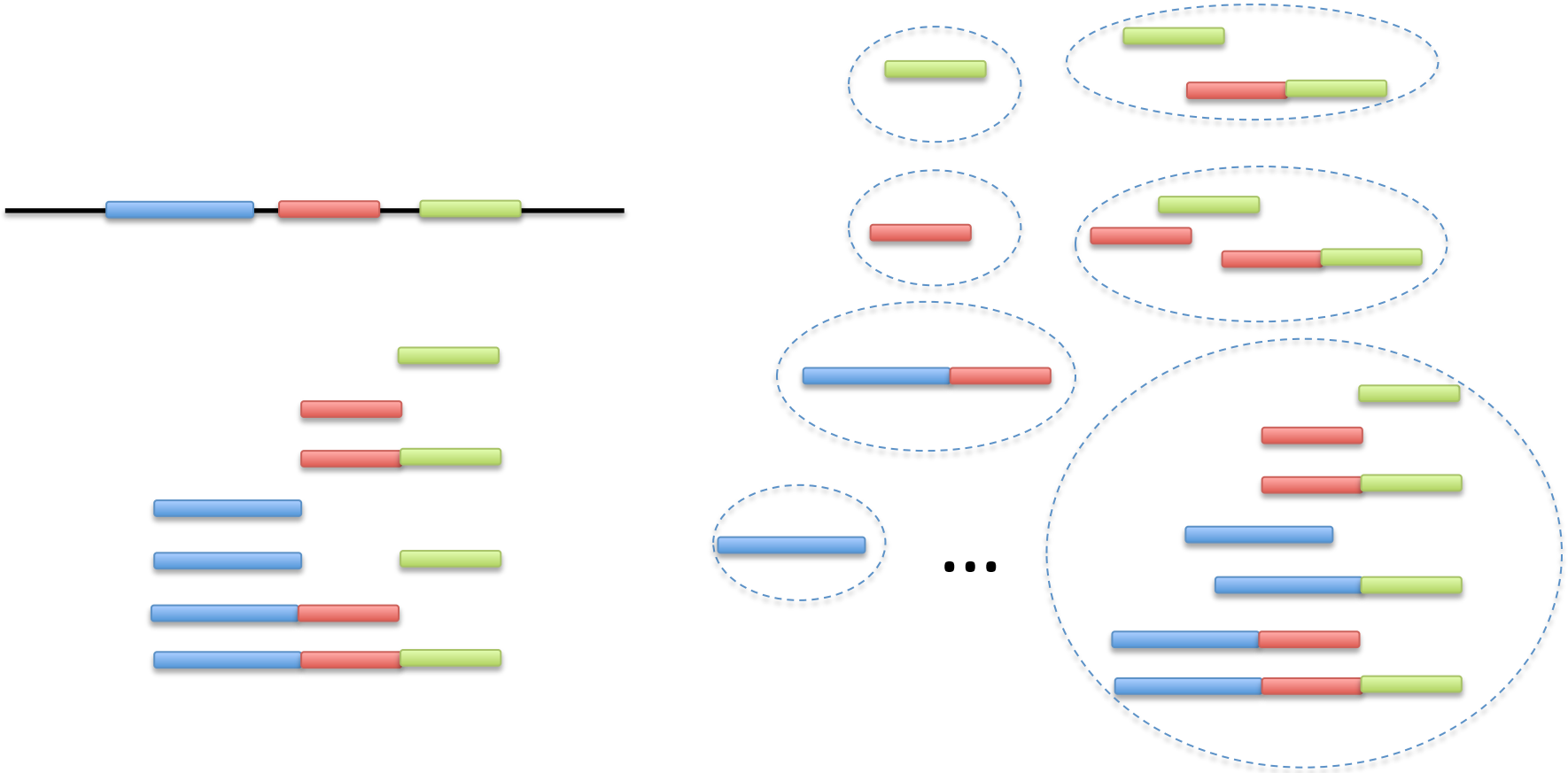
- Exon usage
- Isomer usage



1. 0 0 0
2. 0 0 1
3. 0 1 0
4. 0 1 1
5. 1 0 0
6. 1 0 1
7. 1 1 0
8. 1 1 1

N exons: 2^{N-1} possible isomers

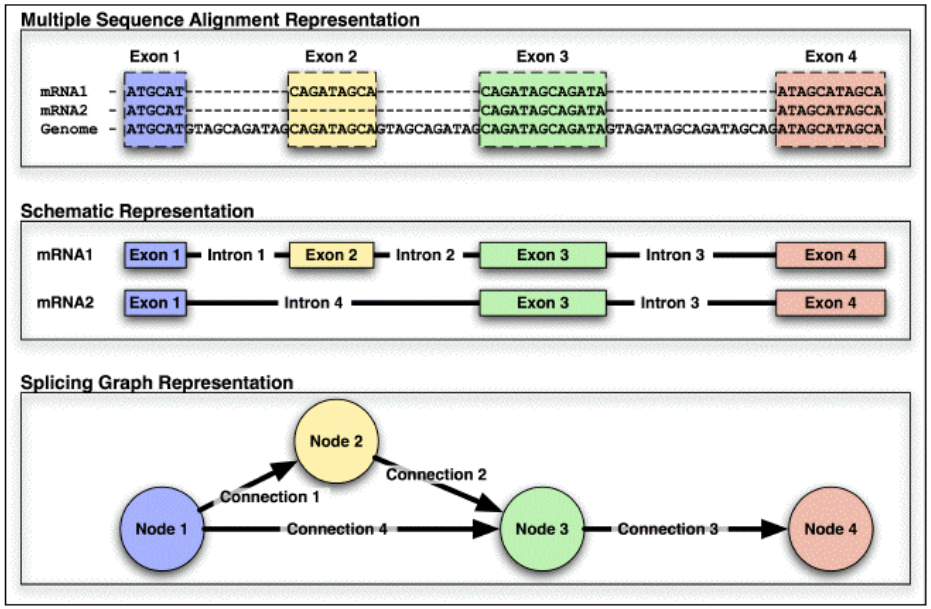
Combinatorial complexity

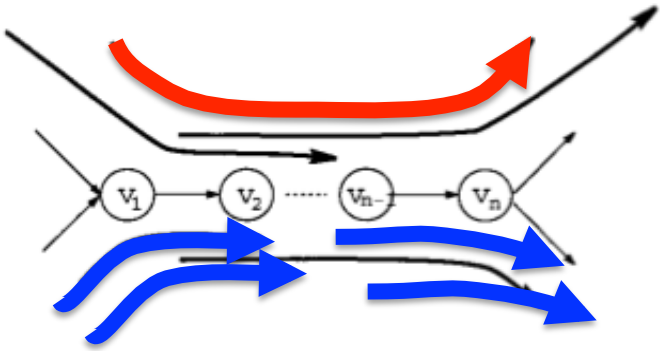


N exons: 2^N-1 possible isomers

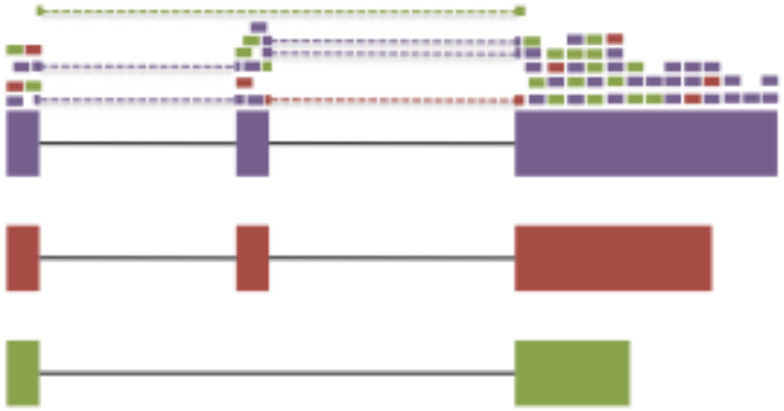
$2^{2^N-1}-1$ possible isomer sets

Transcript reconstruction, approach 1: Mapping against exon graph





Read mapping decomposition

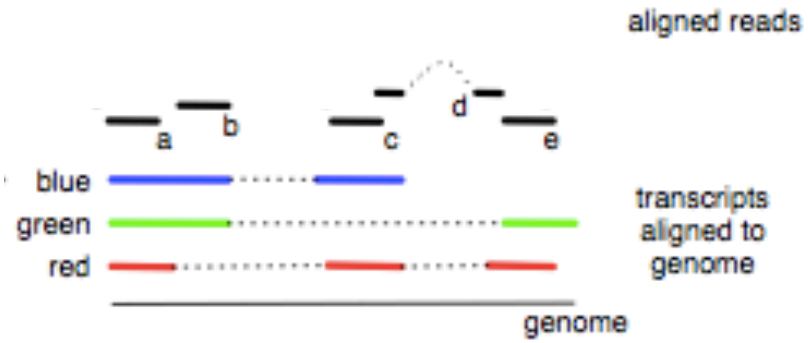


If we knew the origin of the reads we could compute each isoform's expression.
The gene's expression would be the sum of the expression of all its isoforms.

$$E = RPKM_1 + RPKM_2 + RPKM_3$$

From Garber 2011

Abundance estimation: using EM algorithm



Compatibility matrix

$$\mathbf{Y} = \begin{array}{c} \\ \texttt{red} \\ \texttt{green} \\ \texttt{blue} \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

Task: maximize likelihood function

$$\mathcal{L}(\rho) = (\rho_{\texttt{red}} + \rho_{\texttt{blue}})(\rho_{\texttt{red}} + \rho_{\texttt{green}})(\rho_{\texttt{blue}} + \rho_{\texttt{green}})\rho_{\texttt{red}}$$

abundance

From Pachter 2011

3150–3160 *Nucleic Acids Research, 2006, Vol. 34, No. 10*
doi:10.1093/nar/gkl396

An expectation-maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs

Yi Xing[1,*], Tianwei Yu[2,3], Ying Nian Wu[2], Meenakshi Roy[1], Joseph Kim[1] and Christopher Lee[1,*]

[1]Molecular Biology Institute, Center for Computational Biology, Departmer
University of California, Los Angeles, USA, [2]Department of Statistics, Uni
USA and [3]Dental Research Institute, School of Dentistry, University of Ca

Multinomial model with uncommitted categorization. Suppose there are K possible isoforms for a gene. Let's denote them by $I_1, I_2, \ldots, I_K$. For each sequence observation, the probability that it is generated by isoform I_k is p_k, where $k = 1, \ldots, K$ and $p_1 + p_2 + \ldots + p_K = 1$. This probability model is called multinomial model.

We denote $\theta = (p_1, \ldots, p_K)$. The log likelihood function of the multinomial model with uncommitted categorization is

$$l(\theta \mid \mathbf{Y}) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} y_{i,k} p_k \right).$$

The maximum likelihood estimates (MLE) of θ, $\hat{\theta} = \arg\max_\theta l(\theta \mid \mathbf{Y})$, cannot be obtained in closed-form.

EM: soft categorization and fractional counts. The EM algorithm (35) can be used to compute the maximum likelihood estimates (MLE) of the category probabilities $\theta = (p_k, k = 1, \ldots, K)$ from the observed data $\mathbf{Y}$. The EM algorithm is an iterative algorithm. In describing the algorithm, we add subscript $\theta^{(t)}$ to the relevant quantities. For instance, $\theta_{(t)}$ is the parameter value computed after t th iteration. The algorithm starts from an initial guess $\theta^{(0)} = (p_k^{(0)}, k = 1, \ldots, K)$. For instance, $p_k^{(0)} = 1/K$. Each iteration is a mapping from $\theta^{(t)}$ to $\theta^{(t+1)}$, which is accomplished via the following two steps:

E-step:

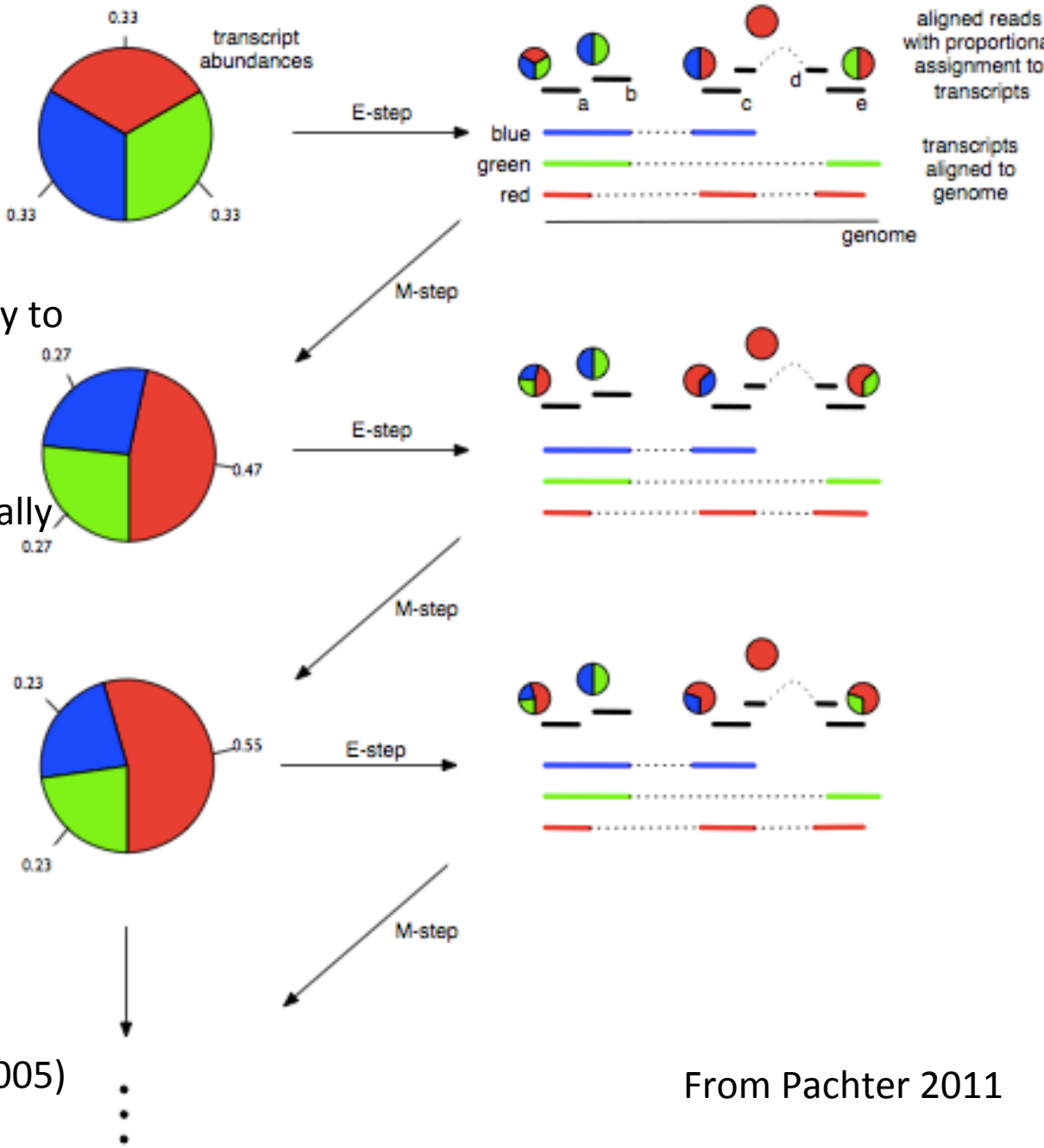
$$z_{i,k}^{(t+1)} = E[z_{i,k} \mid Y_i, \theta^{(t)}] = \Pr(z_{i,k} = 1 \mid Y_i, \theta^{(t)})$$
$$= \frac{y_{i,k} p_k^{(t)}}{\sum_{k=1}^{K} y_{i,k} p_k^{(t)}}, \forall i, k.$$

M-step: Let $n_k^{(t+1)} = \sum_{i=1}^{N} z_{i,k}^{(t+1)}, \quad \forall k,$

$$p_k^{(t+1)} = \frac{n_k^{(t+1)}}{N}, \quad \forall k.$$

E-step: assign reads proportionally to transcripts

M-step: recalculate isoform abundances based on proportionally assigned read counts

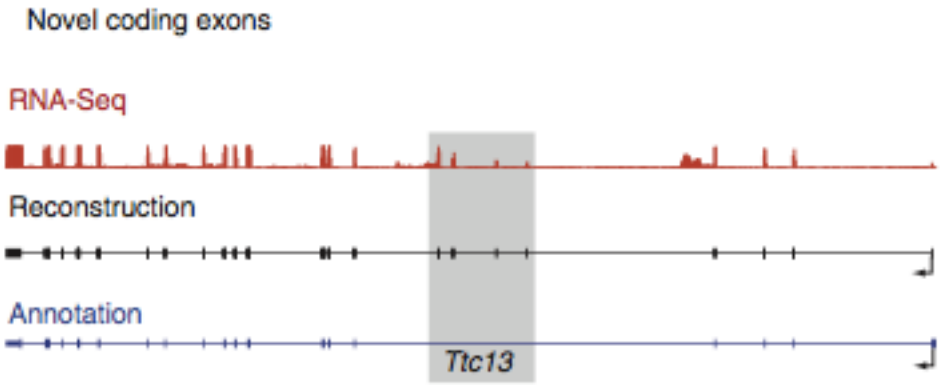


Note:
EM➔regularized EM (Li et al. 2005)
Or ML ➔Bayes (e.g. Cufflinks)

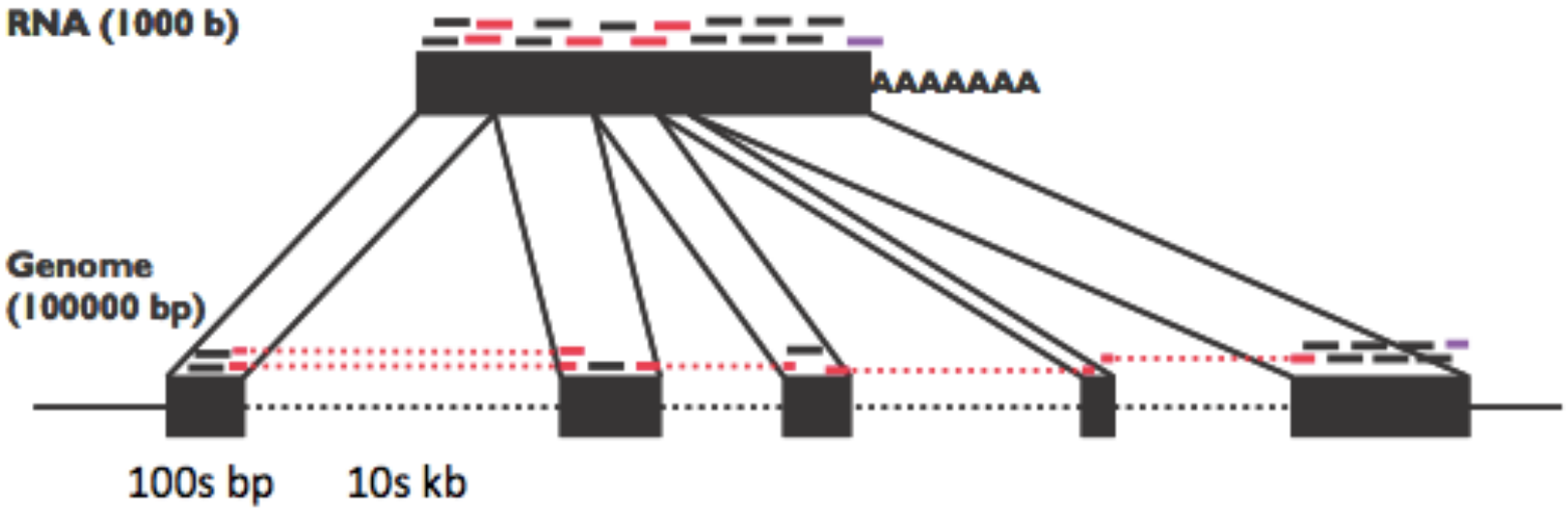
From Pachter 2011

Transcript reconstruction, approach 2: Mapping against reference genome

1. Segmentation problem
2. Isomer resolving problem



Split-read & paired-end read information

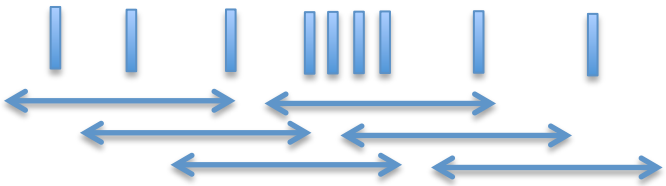


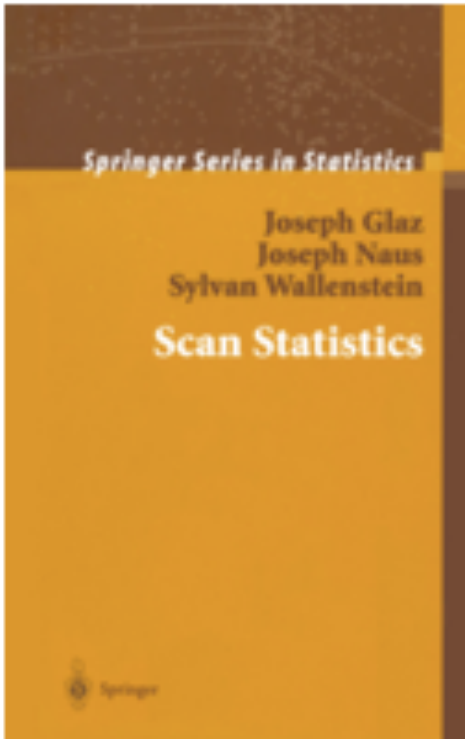
From Garber 2011

Segmentation: moving window analysis

- Oxford dictionary of statistics: **Scan statistic**
  "A statistic used to identify clusters of events in space or time. A window of length or radius *h* traverses the time series or spatial region. The value of the scan statistic is the maximum number of events that fall within this window."

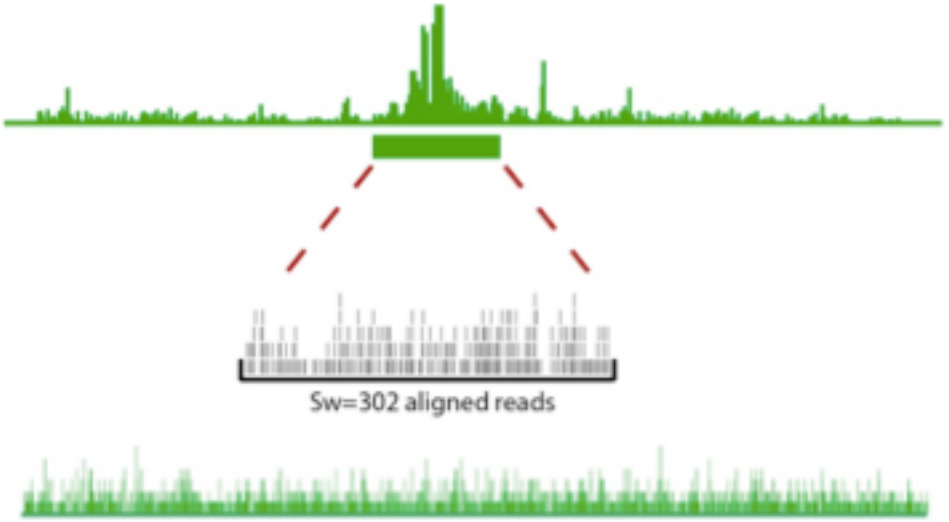
- Joseph Naus 1963 PhD thesis

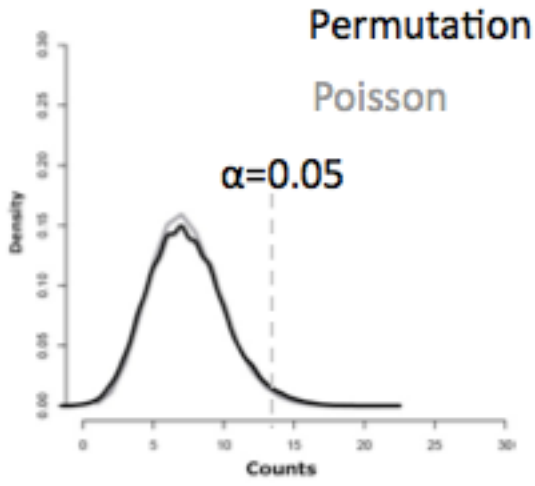




Scan distribution

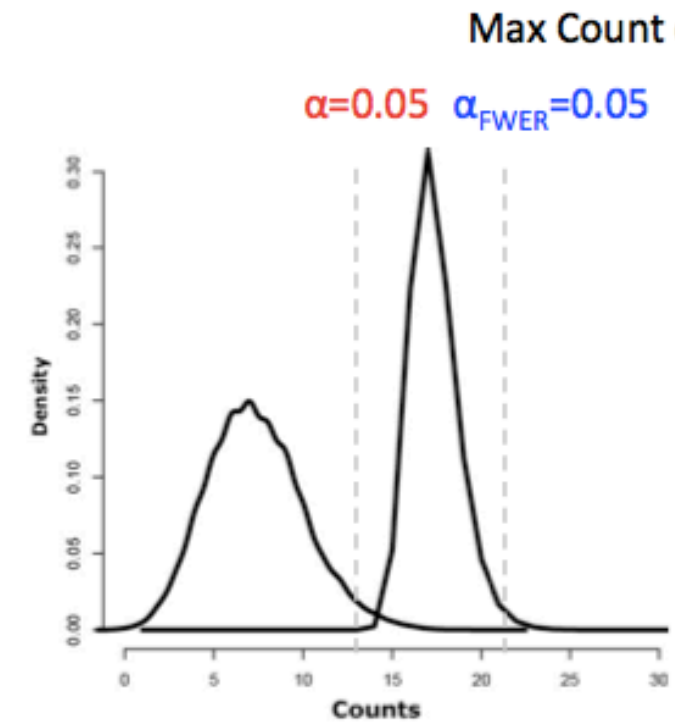
- Significance of event
- Segment “signal” from background
- Window size?
- Background noise?





From Garber 2011

Scan distribution



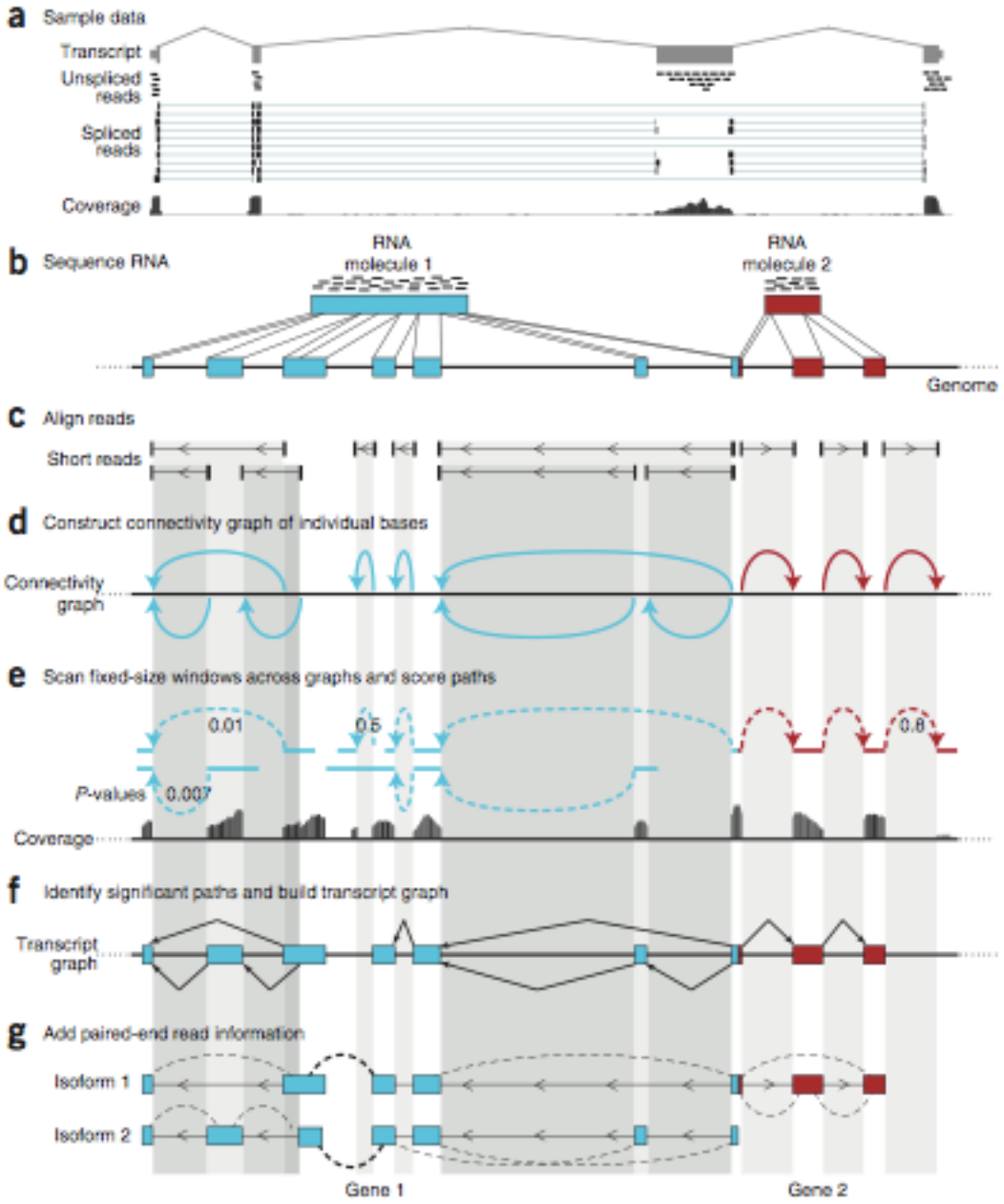
Count distribution (Poisson)

Given a region of size w and an observed read count n. What is the probability that one or more of the $3x10^9$ regions of size w has read count >= n under the null distribution?

We could go back to our permutations and compute an FWER: **max of the genome-wide distributions of same sized region)→**
but really really really slow!!!

From Garber 2011

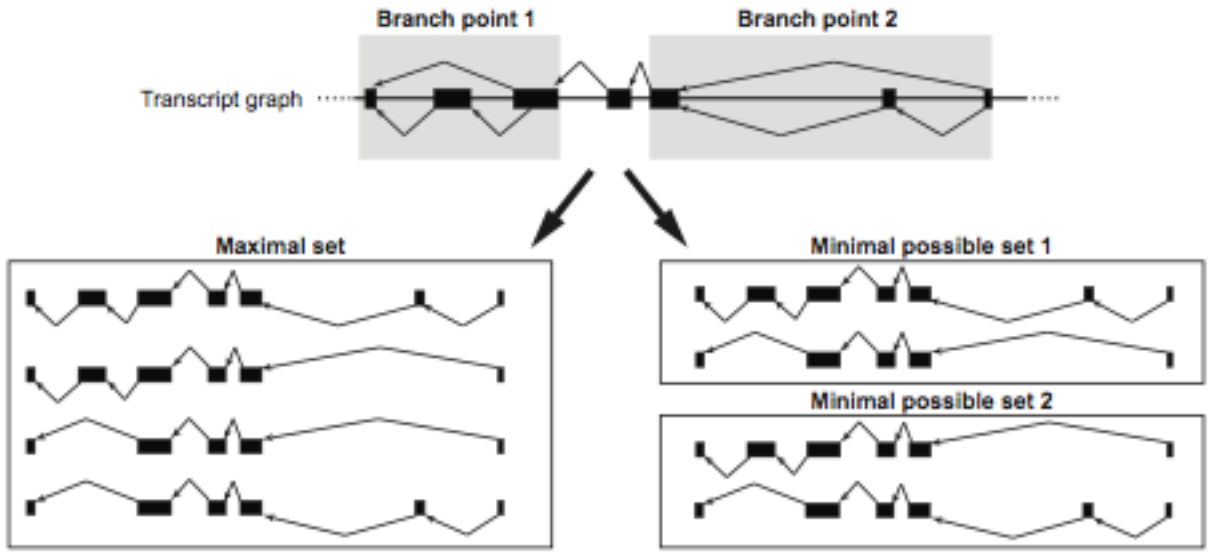
Software: Scripture



From Guttman et al. 2010

Segmentation done. Which isomer candidates are real?

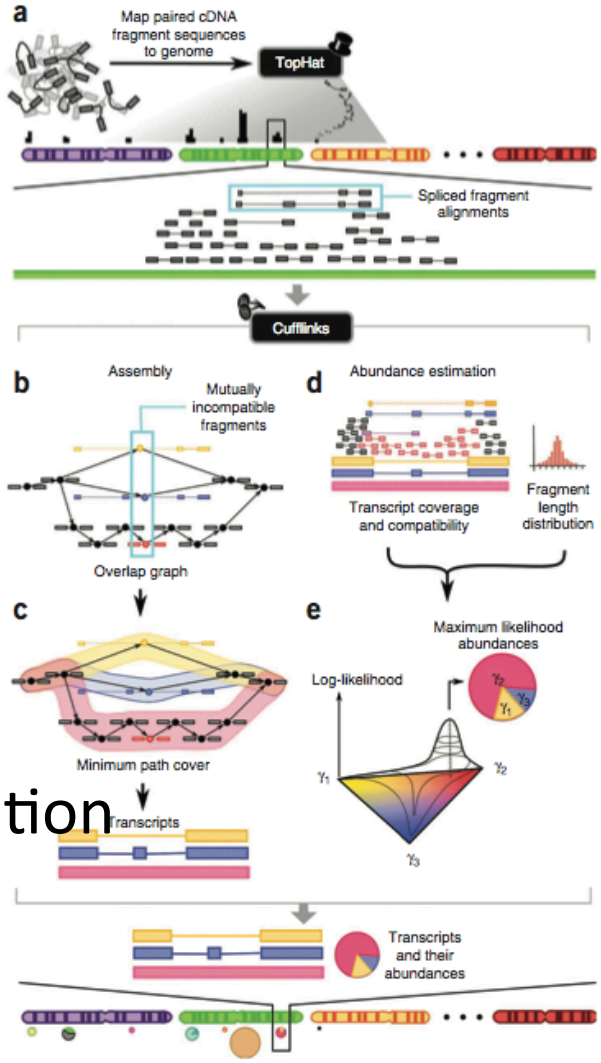
- Report all possible isomers
- Report minimum number of isomers to explain the data
- Utilize abundance information to select isomers



From Garber 2011

Software

- Scripture (2010)
  - Report all (significant) transcripts
- Cufflinks (2010)
  - 1) transcript reconstruction
  - 2) abundance estimation
- IsoLasso (2011), iReckon (2013)
  - "Simultaneous" transcript reconstruction & abundance estimation



1. Isoform identification
2. Re-mapping of reads
3. Isomer selection & abundance estimation (regularized EM)

In practice

```
> bowtie2-build chr18.fa chr18
> tophat2 -r 50 -p 4 -o top2 chr18 chr18_1.fq chr18_2.fq
```

```
> cufflinks –p 4 –o outdir top2/accepted_hits.bam

-rw------- 1 somervuo somervuo  50K Jul 15 10:43 genes.fpkm_tracking
-rw------- 1 somervuo somervuo  67K Jul 15 10:43 isoforms.fpkm_tracking
-rw------- 1 somervuo somervuo    0 Jul 15 10:42 skipped.gtf
-rw------- 1 somervuo somervuo 898K Jul 15 10:43 transcripts.gtf
```

```
> java –jar ScriptureVersion2.0.jar –task reconstruct -alignment top2/
accepted_hits.bam –genome chr18.fa –out out –strand unstranded –chr 18

-rw------- 1 somervuo somervuo  80K Jul  8 15:13 out.connected.bed
-rw------- 1 somervuo somervuo 250K Jul  8 14:09 out.pairedCounts.txt
-rw------- 1 somervuo somervuo 229K Jul  8 14:09 out.pairedGenes.bed
-rw------- 1 somervuo somervuo 104K Jul  8 14:09 out.scripture.paths.bed
```

De novo transcriptome assembly

- ESTs
  - Clustering
  - Assembly

Protocol for Assembly of ESTs and Transcripts

http://compbio.dfci.harvard.edu/tgi/definitions.html

Preparation of EST data

- Sequences were extracted from dbEST and were subjected to quality control screening (vector, E. coli, polyA, T, or CT removal, minimum length = 100 bp, < 3% N).

Preparation of transcript (ET) database

- All sequences from the appropriate divisions of GenBank (including RefSeq) were extracted.
- Non-coding sequences were discarded and cDNAs and coding sequences from genomic entries were saved.
- Sequences and related information (e.g. PubMed links) are stored in the qcGene database (qcGene).

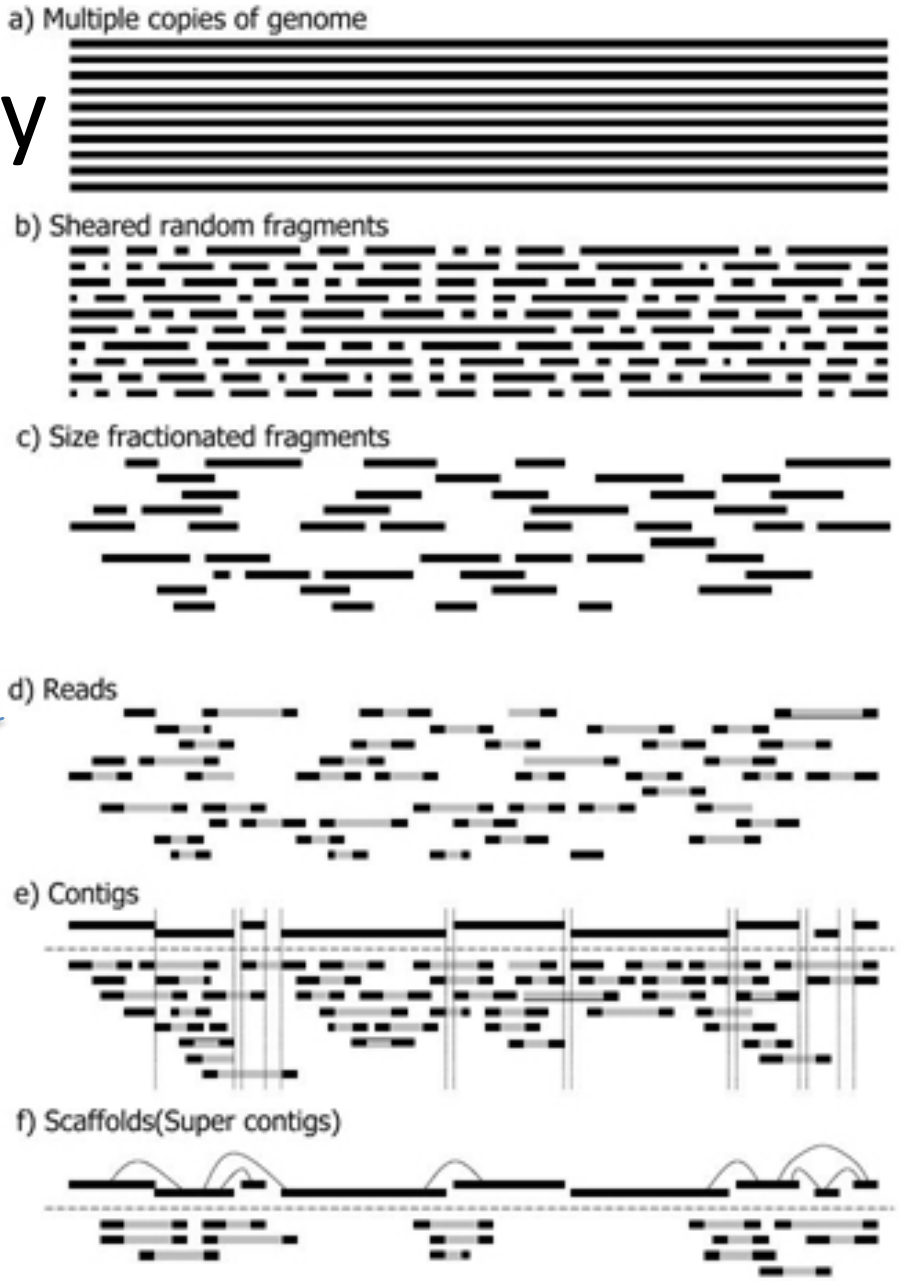
Assembly

- Cleaned EST sequences and non-redundant transcript (ET) sequences were combined.
- Using the Paracel Transcript Assembler Program, sequences were assembled into contigs. TCs are consensus sequences based on two or more ESTs (and possibly an ET) that overlap for at least 40 bases with at least 94% sequence identity. These strict criteria help minimize the creation of chimeric contigs. These contigs are assigned a TC (Tentative Consensus) number. TCs may comprise ESTs derived from different tissues.
- The best hits for TC's were assigned by searching the TC set against a non-redundant amino acid database(nraa) using BLAT. The top five hits based on score were selected and displayed for each TC.

Caveats

- TCs are only as good as the ESTs underlying them; there may be unspliced or chimeric ESTs and thus TCs
- There is still redundancy in the TC set because sequences must match end to end and at a certain percent identity to be combined
- Directionality of the TCs should not be assumed
- Not all TCs contain protein-coding regions

De novo assembly



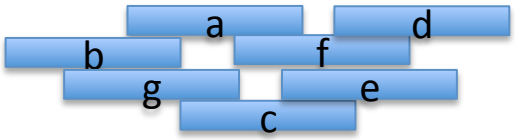
Sequencing errors
Repeats
Short reads

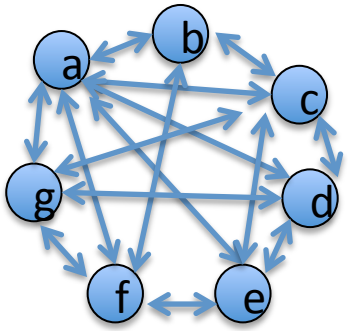
Shortest superstring problem

- Given a set of strings, find a shortest string that contains all of them
- Similar to Traveling Salesman problem (NP)

  → Have to use heuristic approaches, e.g.:
- Greedy algorithm (performance guarantee 2)
  - Merge a pair of strings with maximum overlap until only one string remains

Solution can be twice as bad as the optimal solution





S(a,b)

OLC

- Overlap: identify all potentially overlapping pairs of reads (define minmatch & score)
- Layout:
  - compute score for each pairwise alignment
  - merge reads into contigs
- Compute Consensus sequence for each contig from multiple alignment of reads (majority/ quality weighted voting of each base)

Algorithmica (1995) 13: 7–51

Algorithmica

Combinatorial Algorithms for DNA Sequence Assembly[1]

J. D. Kececioglu[2] and E. W. Myers[3]

Abstract. The trend toward very large DNA sequencing projects, such as those being undertaken as part of the Human Genome Program, necessitates the development of efficient and precise algorithms for assembling a long DNA sequence from the fragments obtained by shotgun sequencing or other methods. The sequence reconstruction problem that we take as our formulation of DNA sequence assembly is a variation of the shortest common superstring problem, complicated by the presence of sequencing errors and reverse complements of fragments. Since the simpler superstring problem is NP-hard, any efficient reconstruction procedure must resort to heuristics. In this paper, however, a four-phase approach based on rigorous design criteria is presented, and has been found to be very accurate in practice. Our method is robust in the sense that it can accommodate high sequencing error rates, and list a series of alternate solutions in the event that several appear equally good. Moreover, it uses a limited form of multiple sequence alignment to detect, and often correct, errors in the data. Our combined
sampled at err

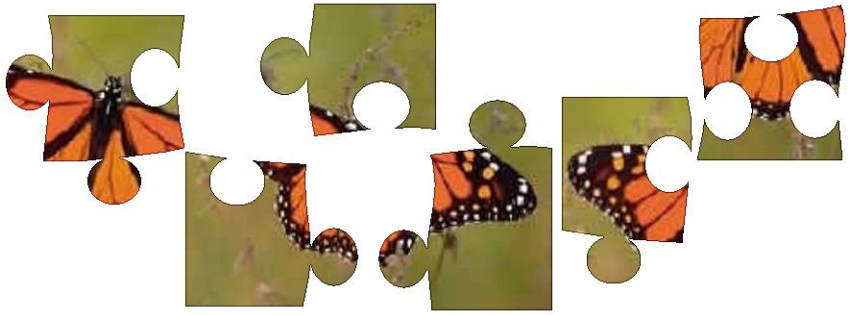
Key Words.
Fragment asse

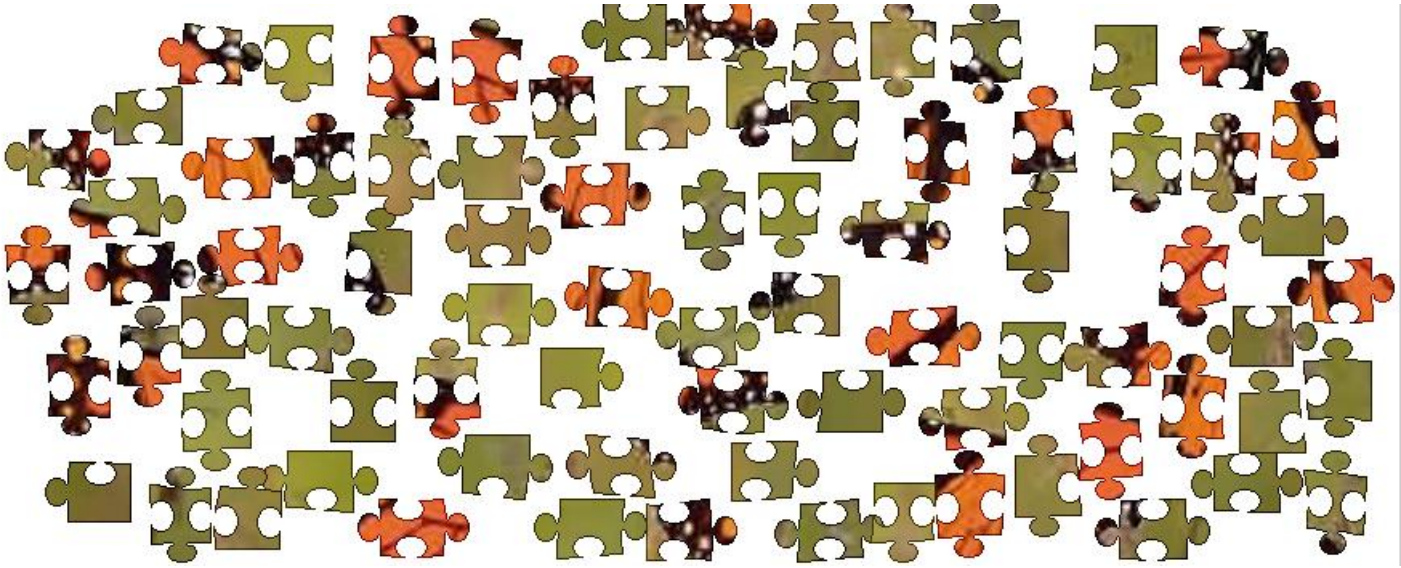
- 3 NP-complete subproblems
- Exact algorithm
- Approximation algorithm

Overview. Our algorithm proceeds in four phases consisting of the following combinatorial problems:

1. Constructing a graph of approximate overlaps between pairs of fragments.
2. Assigning an orientation to the fragments, i.e., choosing the forward or reverse complement sequence for each fragment.
3. Selecting a set of overlaps that induce a consistent layout of the oriented fragments.
4. Merging the selected overlaps into a multiple sequence alignment, and voting on a consensus.

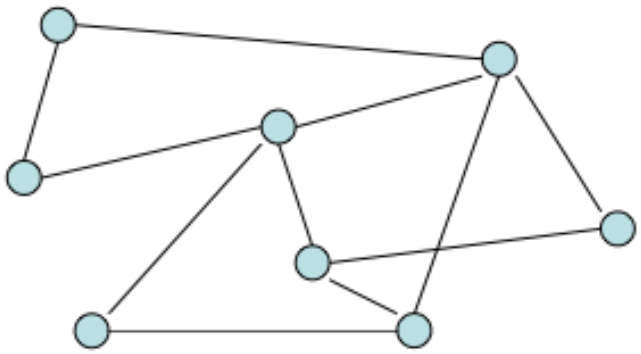
Long vs short reads





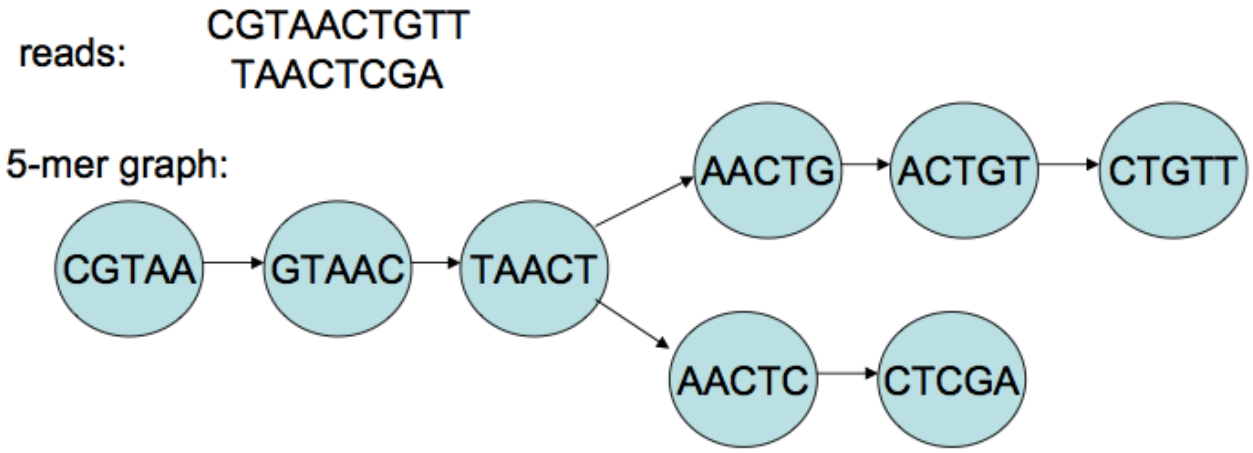
Graph problems

- Eulerian cycle problem
  - find a cycle in a graph that visits every edge exactly once
  - linear time algorithm
- Hamiltonian cycle problem
  - find a cycle in a graph that visits every node exactly once
  - NP-complete



Assembler based on deBruijn graph

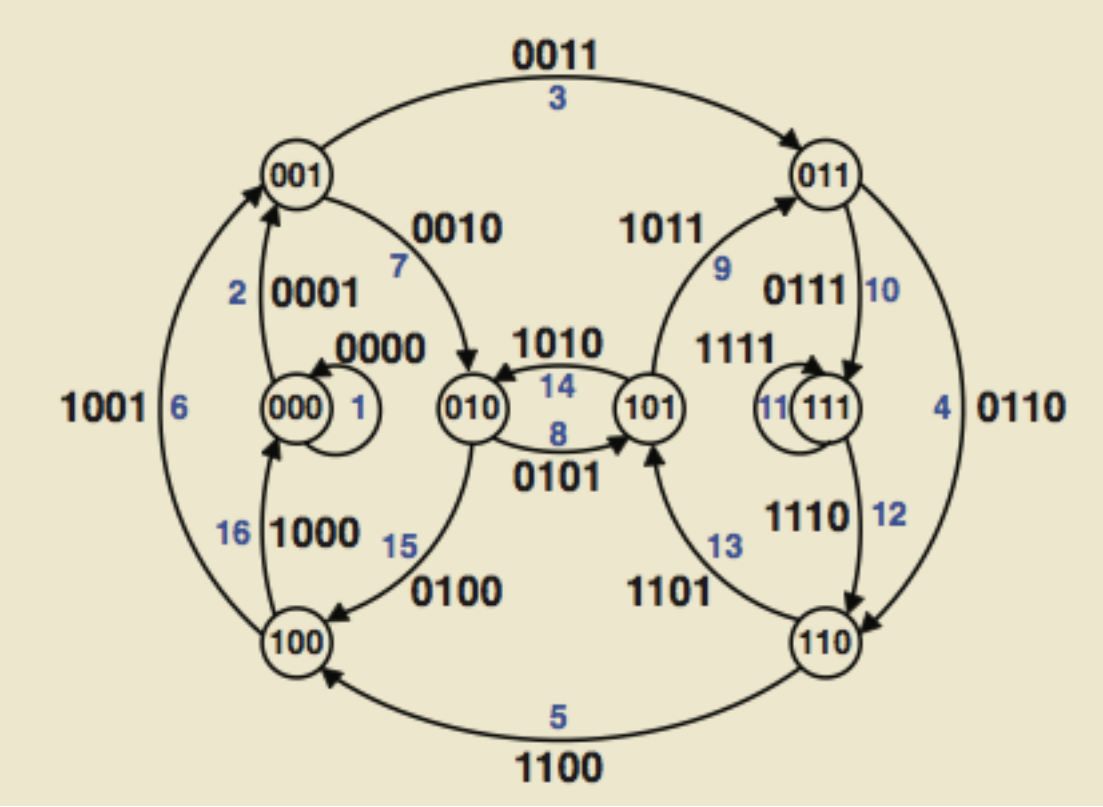
- Pevzner, Tang, Waterman: An Eulerian path approach to DNA fragment assembly, PNAS 2001
- nodes are k-mers instead of full-length reads
  - small k
    →dense graph
  - large k
    →sparse graph

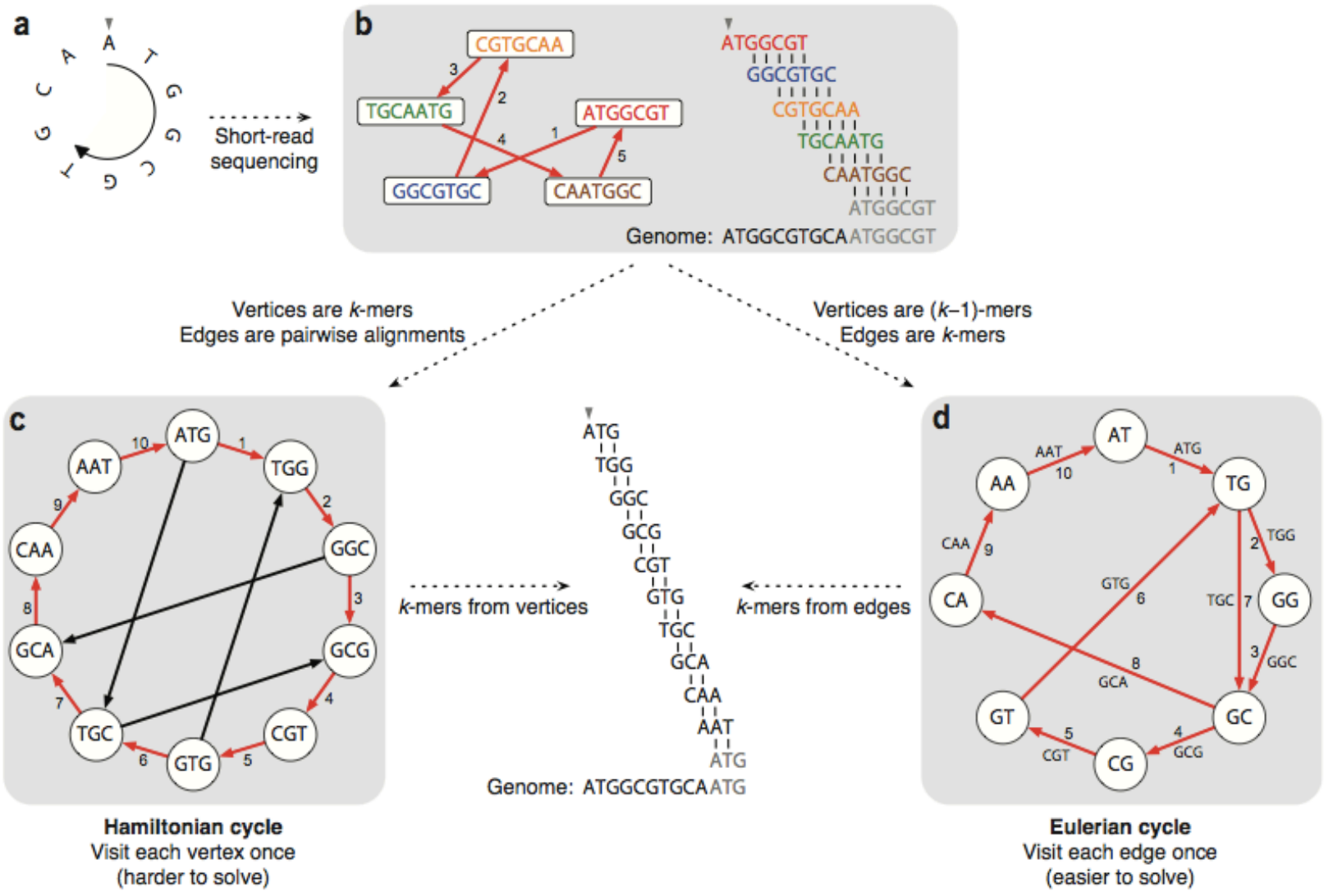


deBruijn graph

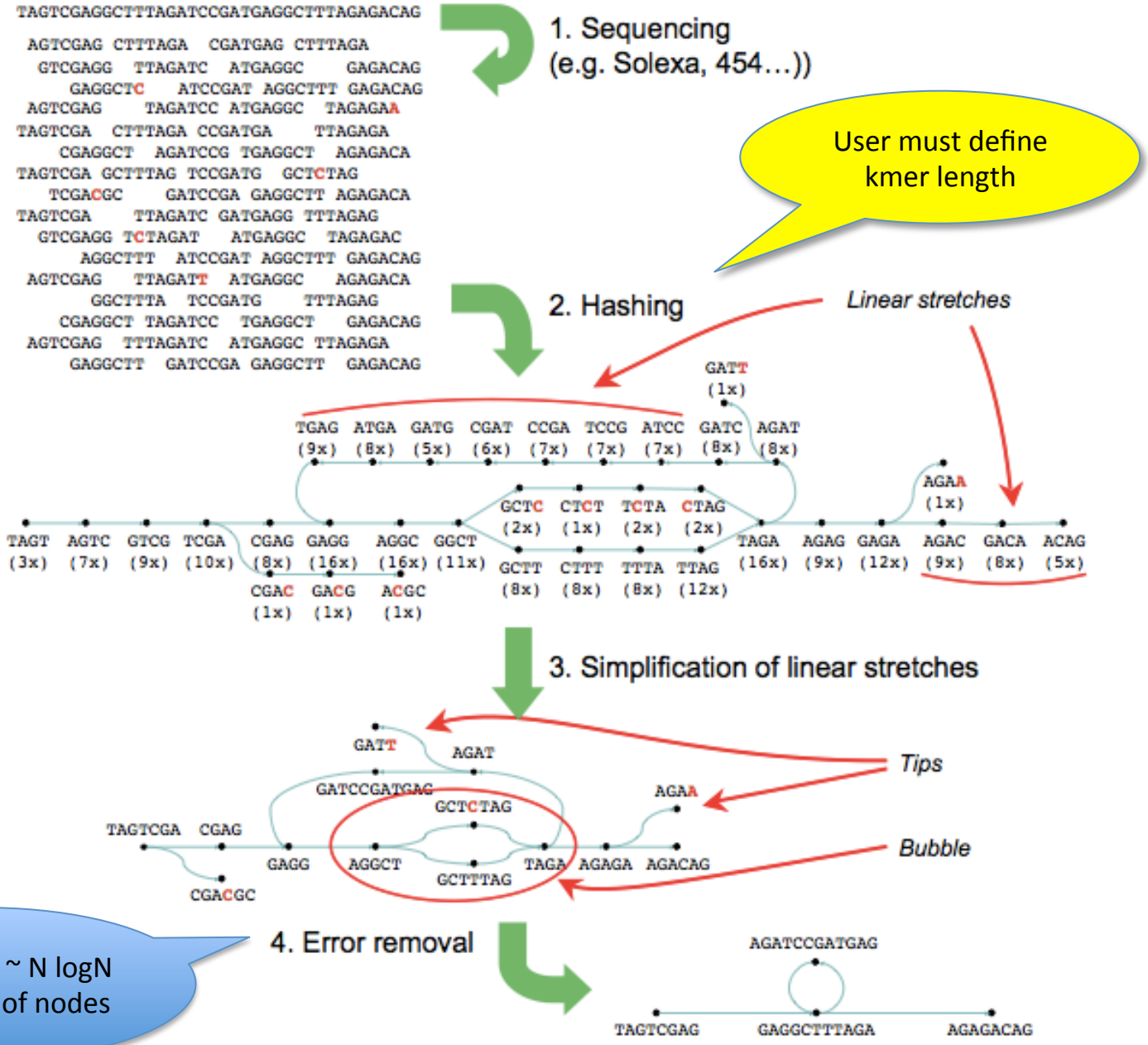
Nicolaas de Bruijn 1946 “superstring problem”: find a shortest circular ‘superstring’ that contains all possible ‘substrings’ of length k (k-mers) over a given alphabet.

k=4





From Compeau et al. 2011



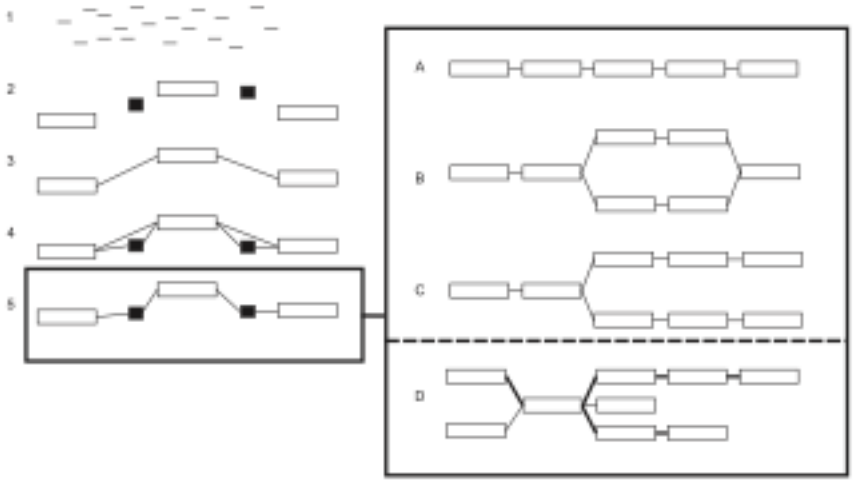
Velvet: Zerbino & Birney 2008

BIOINFORMATICS ORIGINAL PAPER Vol. 28 no. 8 2012, pages 1086–1092
doi:10.1093/bioinformatics/bts094

Sequence analysis Advance Access publication February 24, 2012

Oases: robust *de novo* RNA-seq assembly across the dynamic range of expression levels

- Utilizes velvet deBruijn graph
- Assembly with multiple k-mers, merging



Trinity

- De novo assembly:
  1. Inchworm: initial assembly
  2. Chrysalis: clustering
  3. Butterfly: solving isomers
- Actively developed http://trinityrnaseq.sourceforge.net/
- Guidelines
  - Abundances: RSEM
  - Genome-guided assembly

Nature Biotechnology **29**, 644–652 (2011)

Full-length transcriptome assembly from RNA-Seq data without a reference genome

Manfred G Grabherr, Brian J Haas, Moran Yassour, Joshua Z Levin, Dawn A Thompson, Ido Amit, Xian Adiconis, Lin Fan, Raktima Raychowdhury, Qiandong Zeng, Zehua Chen, Evan Mauceli, Nir Hacohen, Andreas Gnirke, Nicholas Rhind, Federica di Palma, Bruce W Birren, Chad Nusbaum, Kerstin Lindblad-Toh, Nir Friedman & Aviv Regev

1494 | VOL.8 NO.8 | 2013 | NATURE PROTOCOLS

PROTOCOL

***De novo* transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis**

Brian J Haas[1,21], Alexie Papanicolaou[2,21], Moran Yassour[1,3], Manfred Grabherr[4], Philip D Blood[5], Joshua Bowden[6], Matthew Brian Couger[7], David Eccles[8], Bo Li[9], Matthias Lieber[10], Matthew D MacManes[11], Michael Ott[2], Joshua Orvis[12], Nathalie Pochet[1,13], Francesco Strozzi[14], Nathan Weeks[15], Rick Westerman[16], Thomas William[17], Colin N Dewey[9,18], Robert Henschel[19], Richard D LeDuc[19], Nir Friedman[3] & Aviv Regev[1,20]

In practice

```
> velveth vdir 25 –fastq –shortPaired chr18_12.fq
> velvetg vdir –ins_length 200 –read_trkg yes
> oases vdir –ins_length 200 –min_trans_lgth 200

-rw------- 1 somervuo somervuo  25M Jul 16 11:56 Graph2
-rw------- 1 somervuo somervuo  11M Jul 16 11:59 LastGraph
-rw------- 1 somervuo somervuo 1.2K Jul 16 11:59 Log
-rw------- 1 somervuo somervuo 5.5M Jul 16 11:56 PreGraph
-rw------- 1 somervuo somervuo  34M Jul 16 11:55 Roadmaps
-rw------- 1 somervuo somervuo  84M Jul 16 11:55 Sequences
-rw------- 1 somervuo somervuo 1.3M Jul 16 11:59 contig-ordering.txt
-rw------- 1 somervuo somervuo 2.6M Jul 16 11:56 contigs.fa
-rw------- 1 somervuo somervuo 253K Jul 16 11:59 stats.txt
-rw------- 1 somervuo somervuo 1.6M Jul 16 11:59 transcripts.fa


> python oases_pipeline.py -m 19 -M 29 -o odir -d " -fastq -
shortPaired chr18_12.fq " -p " -ins_length 200 –min_trans_lgth 200"
```

In practice

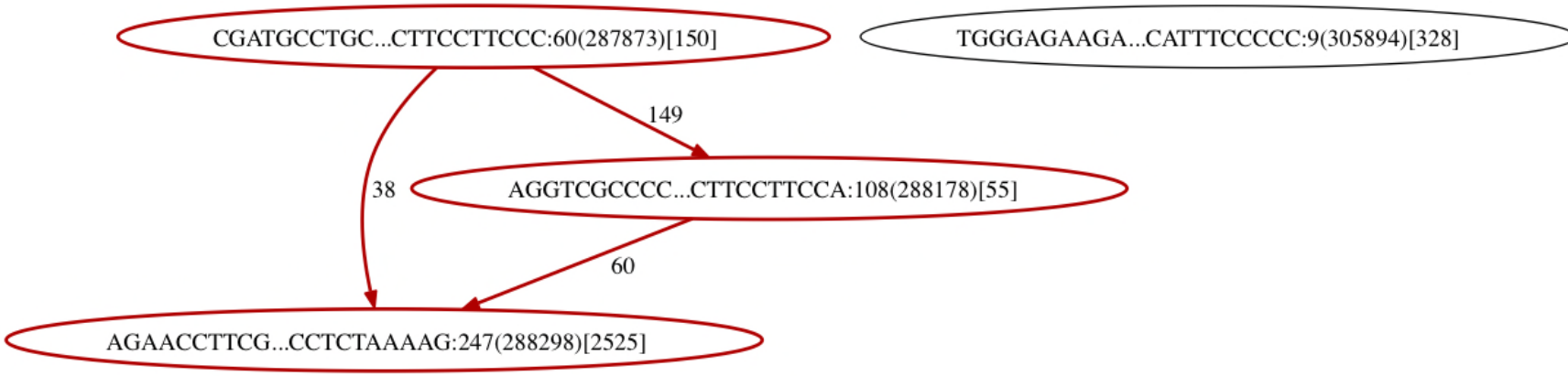
```
> Trinity.pl --seqType fq --JM 10G --left chr18_1.fq --right chr18_2.fq --CPU 4
--bfly_opts "-V 5"
```

```
-rw------- 1 somervuo somervuo 2.2M Dec 18 15:13 Trinity.fasta
-rw------- 1 somervuo somervuo  583 Dec 18 15:13 Trinity.timing
-rw------- 1 somervuo somervuo  78M Dec 18 14:56 both.fa
-rw------- 1 somervuo somervuo    7 Dec 18 14:56 both.fa.read_count
-rw------- 1 somervuo somervuo 159M Dec 18 14:59 bowtie.nameSorted.sam
-rw------- 1 somervuo somervuo    0 Dec 18 14:59 bowtie.nameSorted.sam.finished
-rw------- 1 somervuo somervuo    0 Dec 18 14:59 bowtie.out.finished
drwx------ 3 somervuo somervuo 4.0K Dec 18 15:04 chrysalis
-rw------- 1 somervuo somervuo 3.6M Dec 18 14:58 inchworm.K25.L25.DS.fa
-rw------- 1 somervuo somervuo    0 Dec 18 14:58 inchworm.K25.L25.DS.fa.finished
-rw------- 1 somervuo somervuo    8 Dec 18 14:58 inchworm.kmer_count
-rw------- 1 somervuo somervuo 148K Dec 18 14:59 iworm_scaffolds.txt
-rw------- 1 somervuo somervuo    0 Dec 18 14:59 iworm_scaffolds.txt.finished
-rw------- 1 somervuo somervuo    0 Dec 18 14:57 jellyfish.1.finished
-rw------- 1 somervuo somervuo 125M Dec 18 14:57 jellyfish.kmers.fa
-rw------- 1 somervuo somervuo  13M Dec 18 14:59 scaffolding_entries.sam
-rw------- 1 somervuo somervuo 6.3M Dec 18 14:59 target.1.ebwt
-rw------- 1 somervuo somervuo 279K Dec 18 14:59 target.2.ebwt
-rw------- 1 somervuo somervuo 170K Dec 18 14:58 target.3.ebwt
-rw------- 1 somervuo somervuo 557K Dec 18 14:58 target.4.ebwt
lrwxrwxrwx 1 somervuo somervuo   73 Dec 18 14:58 target.fa -> /…/inchworm.K25.L25.DS.fa
-rw------- 1 somervuo somervuo    0 Dec 18 14:59 target.fa.finished
-rw------- 1 somervuo somervuo 6.3M Dec 18 14:59 target.rev.1.ebwt
-rw------- 1 somervuo somervuo 279K Dec 18 14:59 target.rev.2.ebwt
```

directory chrysalis:

```
drwx------ 4 somervuo somervuo 4.0K Dec 18 15:02 Component_bins
-rw------- 1 somervuo somervuo    0 Dec 18 15:00 GraphFromIwormFasta.finished
-rw------- 1 somervuo somervuo 2.1M Dec 18 15:00 GraphFromIwormFasta.out
-rw------- 1 somervuo somervuo 1.5M Dec 18 15:00 bundled_iworm_contigs.fasta
-rw------- 1 somervuo somervuo  57M Dec 18 15:02 bundled_iworm_contigs.fasta.deBruijn
-rw------- 1 somervuo somervuo    0 Dec 18 15:00 bundled_iworm_contigs.fasta.finished
-rw------- 1 somervuo somervuo 507K Dec 18 15:03 butterfly_commands
-rw------- 1 somervuo somervuo 507K Dec 18 15:13 butterfly_commands.completed
-rw------- 1 somervuo somervuo    0 Dec 18 15:02 chrysalis.finished
-rw------- 1 somervuo somervuo 138K Dec 18 15:03 component_base_listing.txt
-rw------- 1 somervuo somervuo    0 Dec 18 15:03 file_partitioning.ok
-rw------- 1 somervuo somervuo 643K Dec 18 15:03 quantifyGraph_commands
-rw------- 1 somervuo somervuo 643K Dec 18 15:04 quantifyGraph_commands.completed
-rw------- 1 somervuo somervuo    0 Dec 18 15:04 quantifyGraph_commands.run.finished
-rw------- 1 somervuo somervuo    7 Dec 18 15:02 rcts.out
-rw------- 1 somervuo somervuo    0 Dec 18 15:02 readsToComponents.finished
-rw------- 1 somervuo somervuo  79M Dec 18 15:02 readsToComponents.out.sort
-rw------- 1 somervuo somervuo    0 Dec 18 15:02 readsToComponents.out.sort.finished
```

```
-rw------- 1 s s 5.9K Dec 18 16:49 c420.graph.allProbPaths.fasta
-rw------- 1 s s 134K Dec 18 16:23 c420.graph.out
-rw------- 1 s s 1.8M Dec 18 16:23 c420.graph.reads
-rw------- 1 s s  492 Dec 18 16:49 c420.graph_finalCompsWOloops.L.dot
-rw------- 1 s s  492 Dec 18 16:49 c420.graph_withLoops.J.dot
```



Read error correction

CCCCTTTCTATACAGCCCTATCTCATATAGCTACTATTCCGACATATTCT

CCCCTTTCTATACAGCCCTATCTCATATAGCTACTATTCCGACATATTCT

CCCCTTTCTATACAGCCCTATCTCATATAGCTACTATTCCGACATATTCT

CCCCTTTCTATACAGCCCTATCTCATATAGCTACTATTCCGACATATTCT

CCCCTTTCTATACAGCCCTATCTCATATAGCTACTATTCCGACATATTCT

CCCCTTTCTATACAGCCCTATCTTATATAGCTACTATTCCGACATATTCT

CCCCTTTCTATACAGCCCTATCTCATATAGCTACTATTCCGACATATTCT

SEquencing Error CorrEction in Rna-seq data (SEECER)

Nucleic Acids Research, 2013, 1–11
doi:10.1093/nar/gkt215

Probabilistic error correction for RNA sequencing

Hai-Son Le[1], Marcel H. Schulz[2], Brenna M. McCauley[3], Veronica F. Hinman[3] and Ziv Bar-Joseph[1,2,]

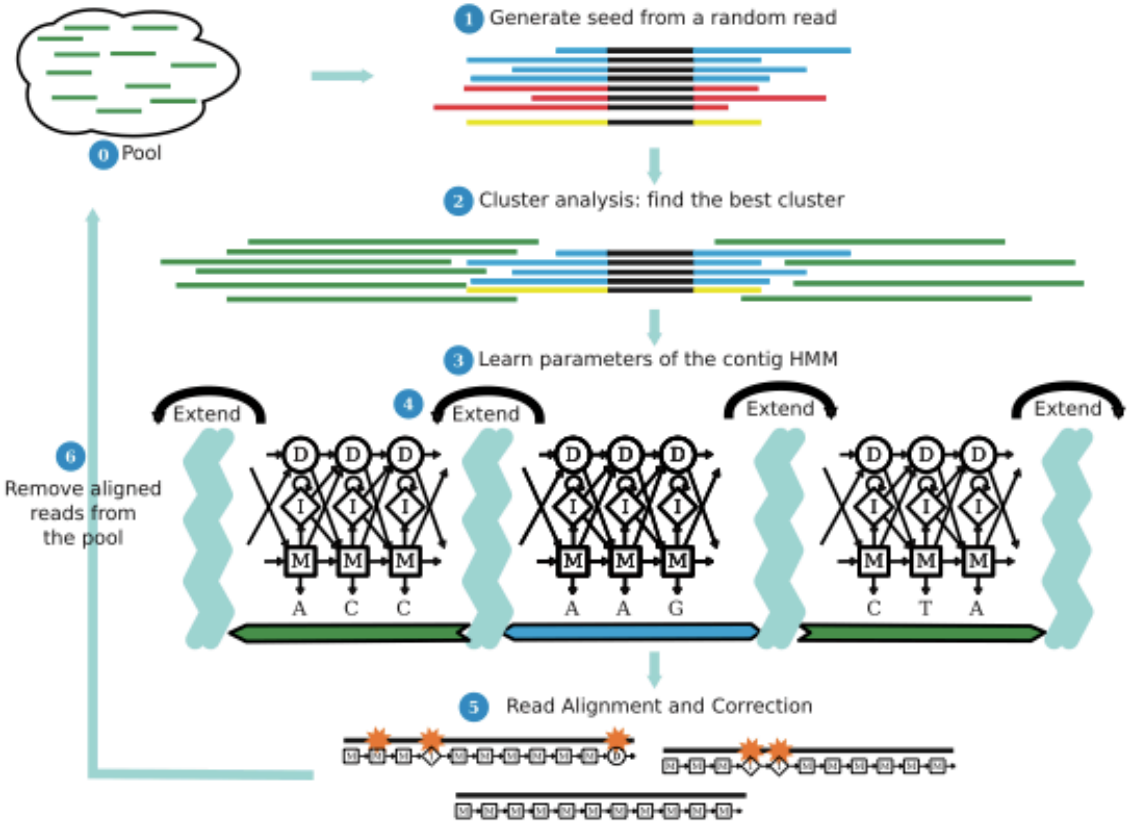


Figure 1. An overview of SEECER. Step 1: We select a random read that has not yet been assigned to any contig HMM. Next, we extract all reads with at least k consecutive nucleotides that overlap with the selected read. Step 2: We cluster all reads and then select the most coherent subset as the initial set of the contig HMM. Step 3: We learn an initial HMM using the alignment specified by the k-mer matches of selected reads. Step 4: We use the consensus sequence defined by the contig HMM to extract additional reads from our unassigned set. These additional reads are used to extend the HMM in both directions. Step 5: When no more reads can be found to extend the HMM, we determine for each of the reads that were used to construct the HMM the likelihood of being generated by this contig HMM. For those with a likelihood above a certain threshold, we use the HMM consensus to correct errors. Step 6: We remove the reads that are assigned or corrected from the unassigned pool. See 'Materials and Methods' section for complete details.

Which transcript reconstruction method is the best?

- Which implementation is best?
  - Algorithm, parameters
  - Thresholds, k-mer coverage, branching
  - Length vs accuracy

- Mapping
- De novo

Comparing methods

Assessment of transcript reconstruction methods for RNA-seq

Tamara Steijger[1], Josep F Abril[2,11], Pär G Engström[1,10,11], Felix Kokocinski[3,11], The RGASP Consortium[4], Tim J Hubbard[3], Roderic Guigó[5,6], Jennifer Harrow[3] & Paul Bertone[1,7–9]

NATURE METHODS | VOL.10 NO.12 | DECEMBER 2013 | 1177

Table 1 Metric trends and consistency

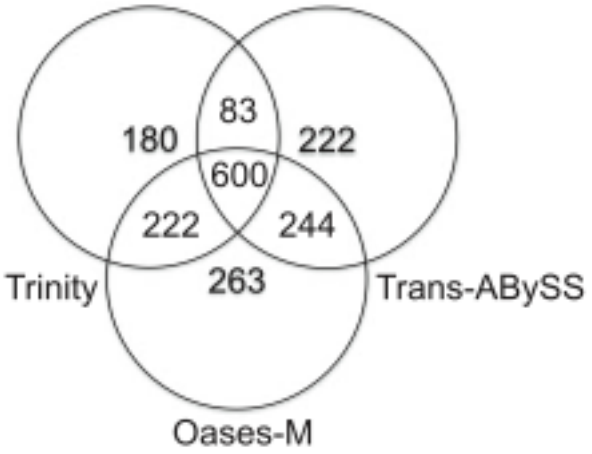
Metric	Trend by sequencing depth	Consistent over sequencing depths	Trend by read length	Consistent over read lengths	Fully consistent metric
Contig count	↗	✓	↘	✓	✗
% of reads used in contigs	↗	✓	↗	✓	✓
BP in contigs	↗	✓	↗	✓	✓
% BP in contigs	↗	✓	↗	✓	✓
Average contig coverage	↗	✗	↗	✗	✗
Average unigene coverage	↗	✓	↗	✓	✓
Contig read count COV	⌒↘	✓	↘	✓	✗
Unigene read count COV	⌒↘	✓	↘	✓	✗
Average contig length	↗	✗	↗	✗	✗
Average unigene length	↗	✓	↗	✓	✓
Contig N50 length	↗	✗	↗	✗	✗
Unigene N50 length	↗	✓	↗	✓	✓
Unique annotations in singletons	⌒↘	✓	↘	✓	✗
Unique annotations in contigs	↗	✓	↗	✓	✓
Unique annotations in unigenes	↗	✗	⌒↗	✓	✗
Average contig OHR	↗	✗	↗	✗	✗
Average unigene OHR	↗	✓	↗	✓	✓
Contig RBH count	↗	✓	↗	✓	✓
Unigene RBH count	↗	✓	↗	✓	✓
% of Annotated contigs with RBHs	⌒↗	✗	↗	✗	✗
% of Annotated unigenes with RBHs	⌒↗	✓	↗	✗	✗
Average contig CF	↘	✓	↘	✓	✓
Average unigene CF	↘	✗	↘	✗	✗
Unique reverse annotations in contigs	↗	✓	↗	✓	✓
Unique reverse annotations in unigenes	↗	✓	↗	✓	✓

Trends for *de novo* transcriptome assembly metrics as sequencing depth and read length are varied. Straight lines indicate monotonically increasing or decreasing trends, while curved lines indicate non-monotonic trends (either increasing then decreasing, or decreasing then increasing). Metrics are considered consistent if they consistently ranked perfectly assemblies as better; fully consistent metrics are consistent metrics with similar monotonic trends by sequencing depth and read length for perfect assemblies. Annotation-based metrics shown are computed in comparison to *B. mori* proteins. See Additional file 1 for full results.

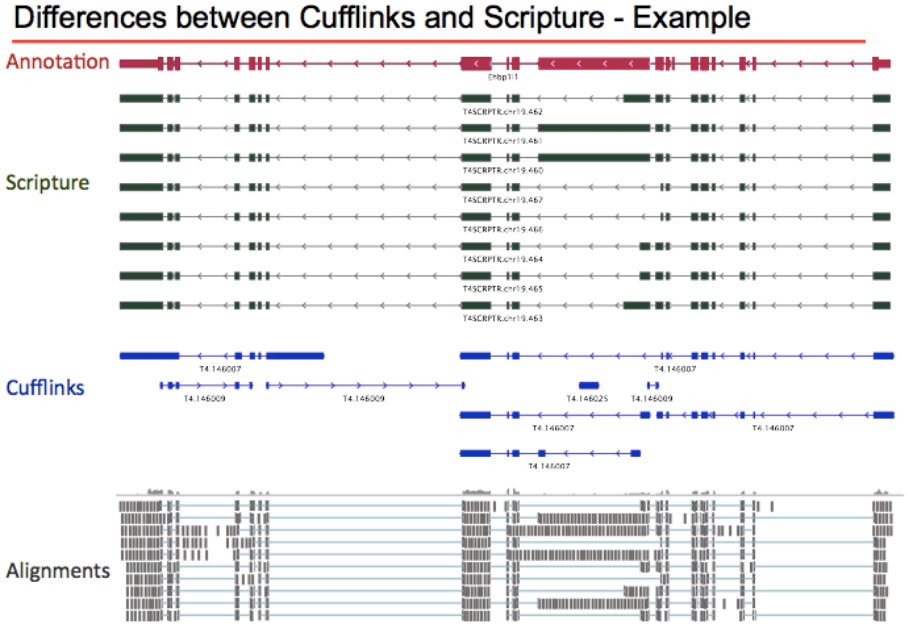
Comparing methods…

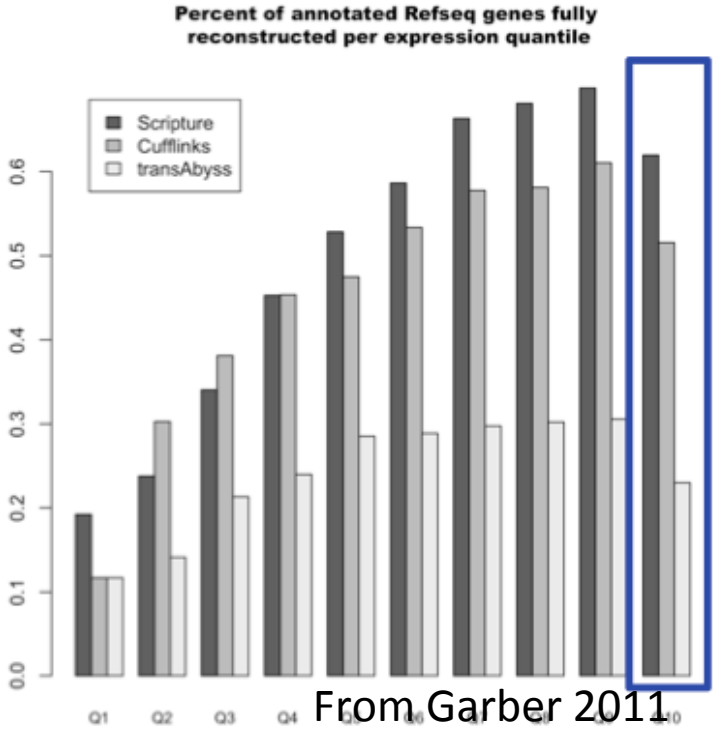
Table 2. Overall comparison of the different RNA-seq assembly methods on human and mouse datasets

Data	Method	Tfrags > 100 bp	Sens. (%)	Spec. (%)	Full lgth	80% lgth
Human	Oases-M	174 469	21.44	92.35	1463	11 169
	tABySS	100 127	19.65	92.16	1358	10 992
	Trinity	76 232	19.99	88.63	953	7129
Mouse	Oases-M	175 914	30.83	89.08	1324	9880
	tABySS	174 744	30.66	92.79	1149	9376
	Trinity	92 810	31.57	87.14	1085	7028
	Cufflinks	63 207	48.13	75.29	4369	21 222



From Schultz et al.2012





From Garber 2011

Platforms

- 454
- Illumina
- SOLiD
- IonTorrent/Proton
- PacBio

RNA-Seq library preparation

- Normalization
- Strand specificity

Example

Leveraging skewed transcript abundance by RNA-Seq to increase the genomic depth of the tree of life

Chris Todd Hittinger[a,b], Mark Johnston[a,b], John T. Tossberg[c], and Antonis Rokas[c,1]

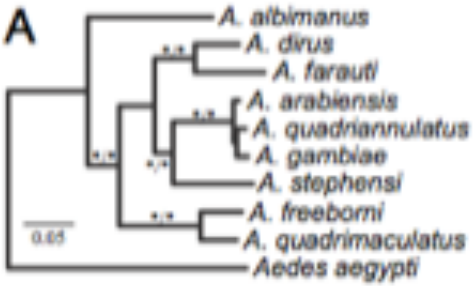
[a]Department of Biochemistry and Molecular Genetics, University of Colorado Denver Health Sciences Center, Aurora, CO 80045; [b]Center for Genome Sciences, Department of Genetics, Washington University School of Medicine, St. Louis, MO 63108; and [c]Department of Biological Sciences, Vanderbilt University, Nashville, TN 37235

Edited by Sean B. Carroll, University of Wisconsin, Madison, WI, and approved December 10, 2009 (received for review September 13, 2009)

- 10 mosquito species
- Non-normalized RNA-Seq libraries
- 13M 36bp Illumina reads per species
- Denovo assembly
  - Velvet: kmers 17, 19, 21, 23, 25, 27, and 29
- Phylogenic tree construction (RAXML, MrBayes)
  - Orthologs: reciprocal best- BLAST-hit matches between reference and single contigs of all nine *Anopheles* species

n sequences from the short-read lengths produced by next-eration DNA sequencing technologies is challenging (28). rd, and most important, the grossly uneven abundance of nscripts [varying over five orders of magnitude (29)] means that n light sequence coverage should provide in-depth sampling of w hundred loci simply by sequencing transcripts in proportion heir representation in the library (9, 17, 19, 30). Moreover, nly expressed genes are typically involved in housekeeping and rgy functions and therefore tend to be well-conserved (29, 30), ding to the expectation that orthologous genes can be effi-ntly sampled across species (9, 17, 19, 30).

Ve tested the idea that RNA-Seq (31) of non-normalized nscriptomes could rce of orthologou sequenced the eloped a novel m p sequence read hundreds of tho rices were compe tained very few rences made fro sitive to biological pr this approach sugges



Recent advances in statistical phylogenetics, information technology, and molecular biology make it feasible to assemble a

EVOLUTION

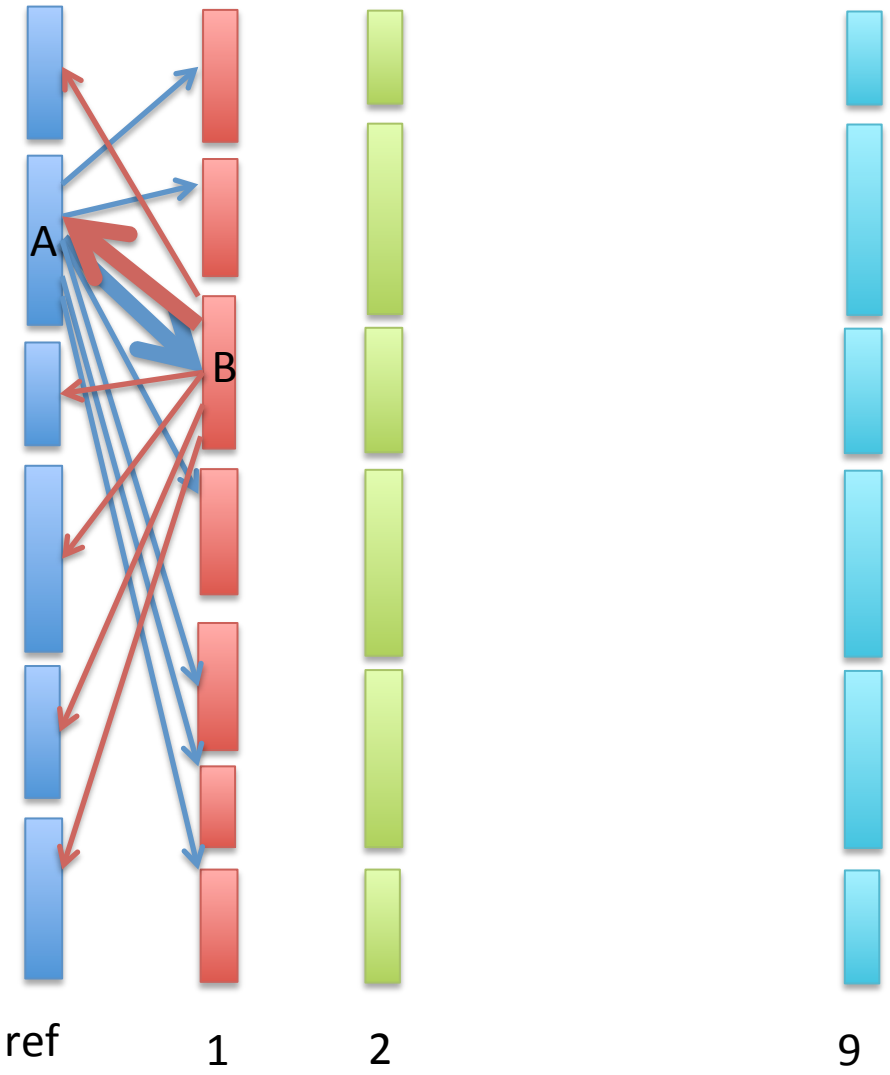
www.pnas.org/cgi/doi/10.1073/pnas.0910449107

Goal to find orthologous sequences: why RNA-Seq?

1. only 7% of the *Anopheles gambiae* genome codes for proteins
2. transcriptomes contain few simple-sequence regions and repetitive elements
3. uneven abundance of transcripts [varying over five orders of magnitude] means that even light sequence coverage should provide in-depth sampling of a few hundred loci
4. highly expressed genes are typically involved in housekeeping and energy functions and therefore tend to be well-conserved, leading to the expectation that orthologous genes can be efficiently sampled across species

Data matrix

reciprocal BLAST



Example: Dungbeetle, 8 species

Trinity assembly

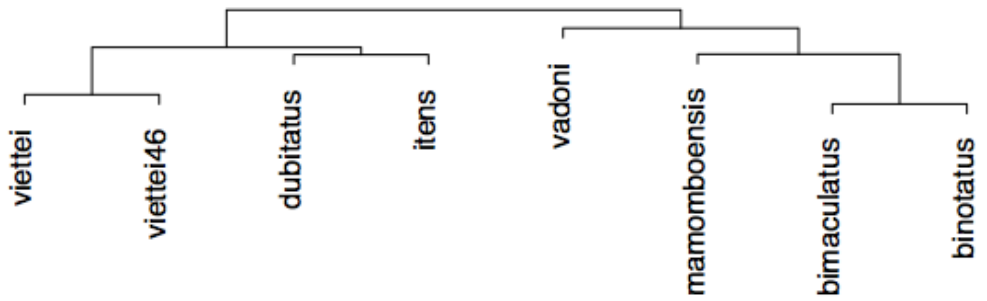
species	non-empty graphs	contigs	length: sum	min	max	N50
bimaculatus :	44208	73357	54398613	201	13577	1308
binotatus :	34509	71843	47086800	201	14386	1020
dubitatus :	70153	148960	77120739	201	16022	657
itens :	35530	58027	44741528	201	18201	1440
mamomboensis:	29301	41479	30622569	201	14393	1223
vadoni :	68092	150436	75159079	201	13967	613
viettei :	76352	131742	82123256	201	22170	996
viettei46 :	48734	86633	58982768	201	15280	1182

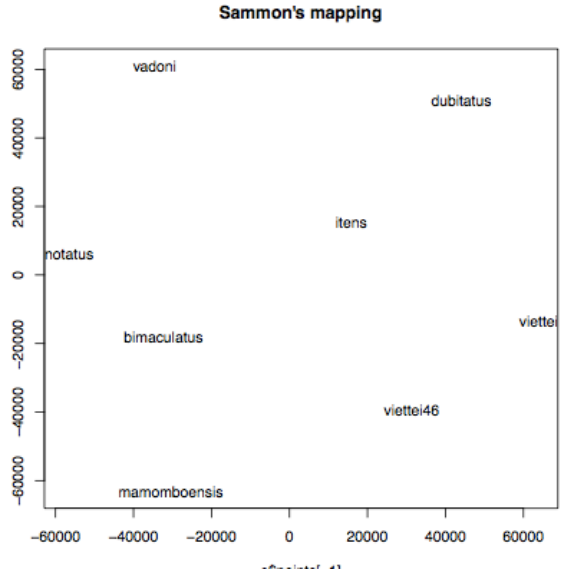
single sequence (seq1) per component

species	contigs	sumlength	all%	min	max	N50
bimaculatus	44678	33780803	62.1	201	13577	1358
binotatus	34908	25815360	54.8	201	14386	1228
dubitatus	71321	38930669	50.5	201	16022	736
itens	35836	28915349	64.6	201	18201	1528
mamomboensis	29501	22938250	74.9	201	14393	1298
vadoni	69060	37687554	50.1	201	13967	728
viettei	77469	48412226	59.0	201	22170	1013
viettei46	49493	35185008	59.7	201	15280	1303

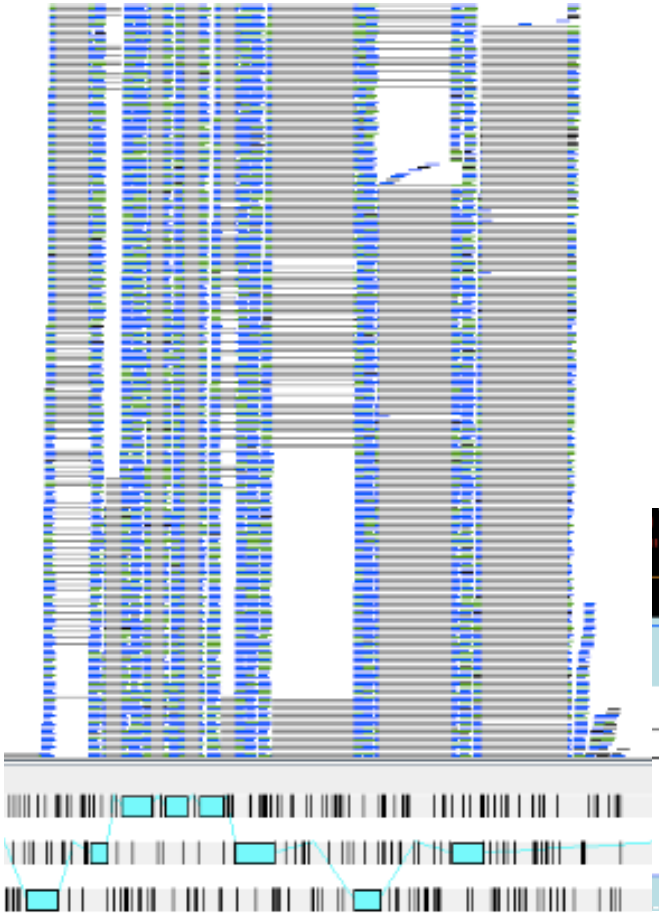
Example: Dungbeetle, 8 species

1. BLASTX (id >=30%) against 16502 uniprot Tribolium protein sequences
2. select best contigs covering uniprot sequences (dynamic programming)
3. contig alignment against uniprot sequences (exonerate -m proten2dna)
4. realignment (compute consensus and contig alignment against it)
5. calculate differences in positions where all 8 species cover reference



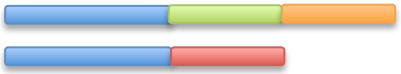


Example: Melitaea cinxia
RNA-Seq evidence for gene models

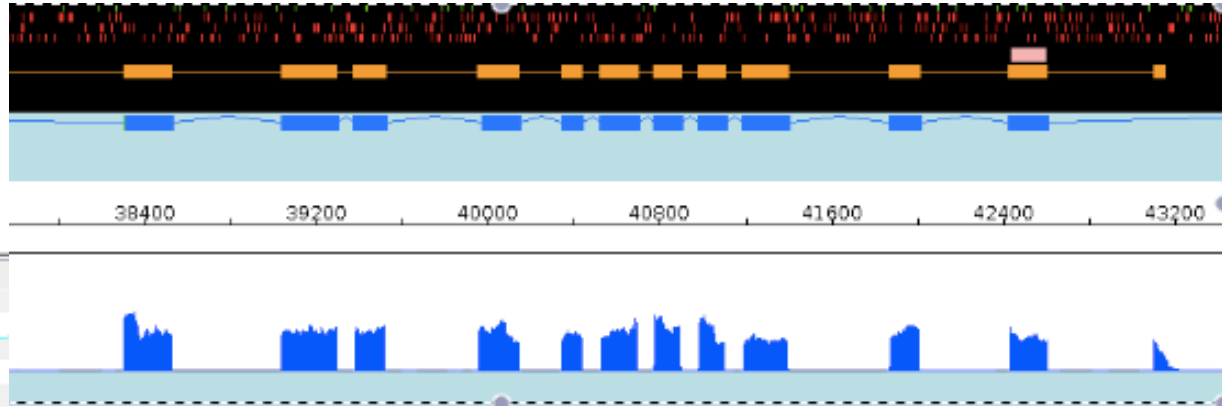


Mapping (Bowtie +Tophat + Cufflinks)

Denovo transcriptome assembly (Trinity)



- 186M Illumina 2*76nt
- 28Gbp →duplicate removal, quality filtering →15Gbp



Summary: transcriptome assembly

- Tasks
  - Segmentation
  - Isomer finding
  - (abundance estimation)
- Approaches
  - Mapping
  - De novo (OLC, deBruijn graph)
- Validation
  - Known transcripts

To go further

- Similarities between metagenomics, haplotype inference, and transcriptomics (Trapnell)

- J. Laserson, V. Jojic, and D. Koller (2011). "**Genovo: De Novo Assembly For Metagenomes**." *Journal of Computational Biology*, 18(3), 429-43.