Assembly and Alignment

Workshop on Comparative Genomics

İnanç Birol, Shaun Jackman

Smithsonian Institution – 5 October 2011

CANADA'S MICHAEL SMITH GENOME SCIENCES CENTRE

BC Cancer Agency CARE & RESEARCH

Assembly and Alignment

Workshop on Comparative Genomics

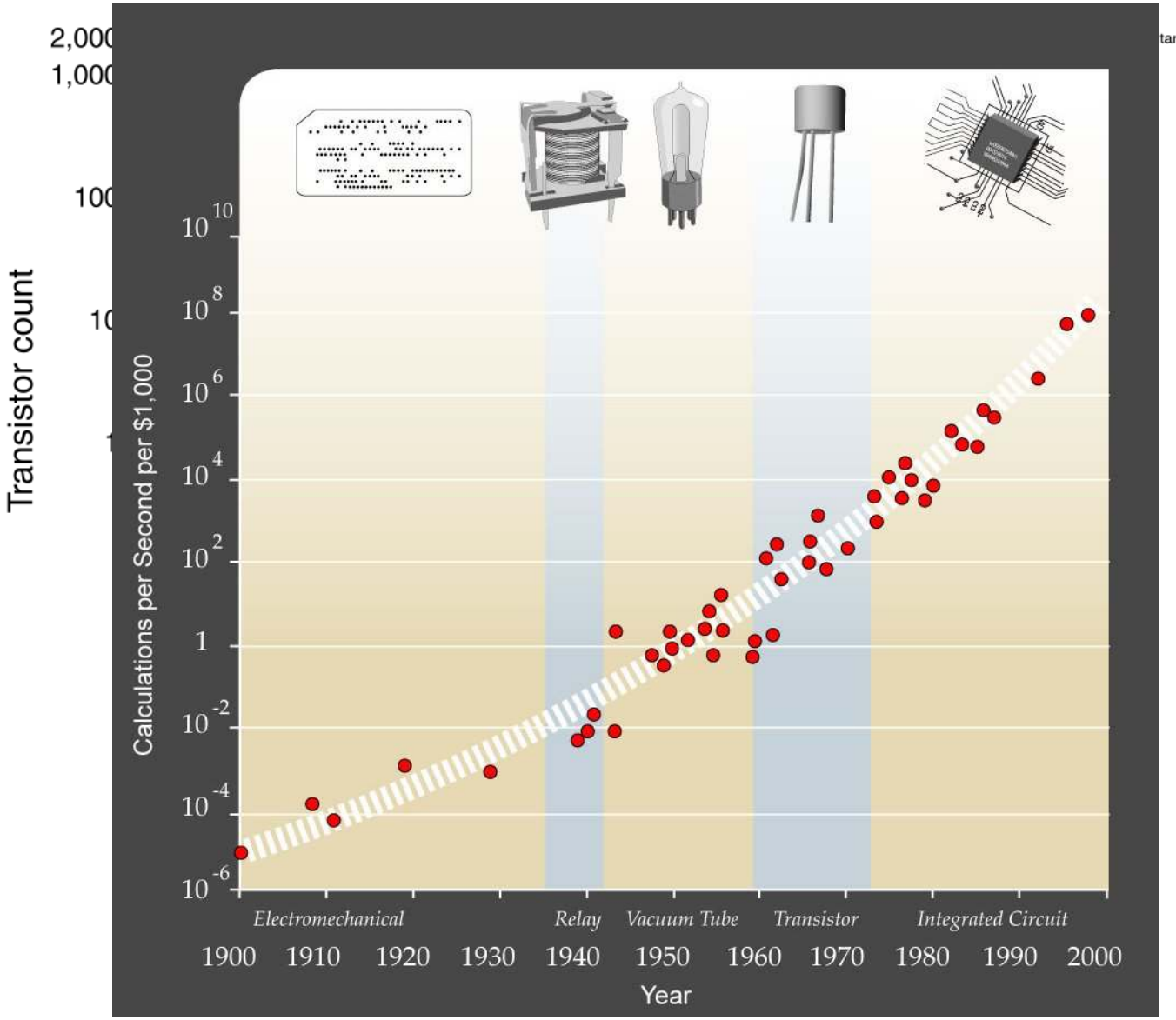
İnanç Birol, Shaun Jackman

Smithsonian Institution – 5 October 2011

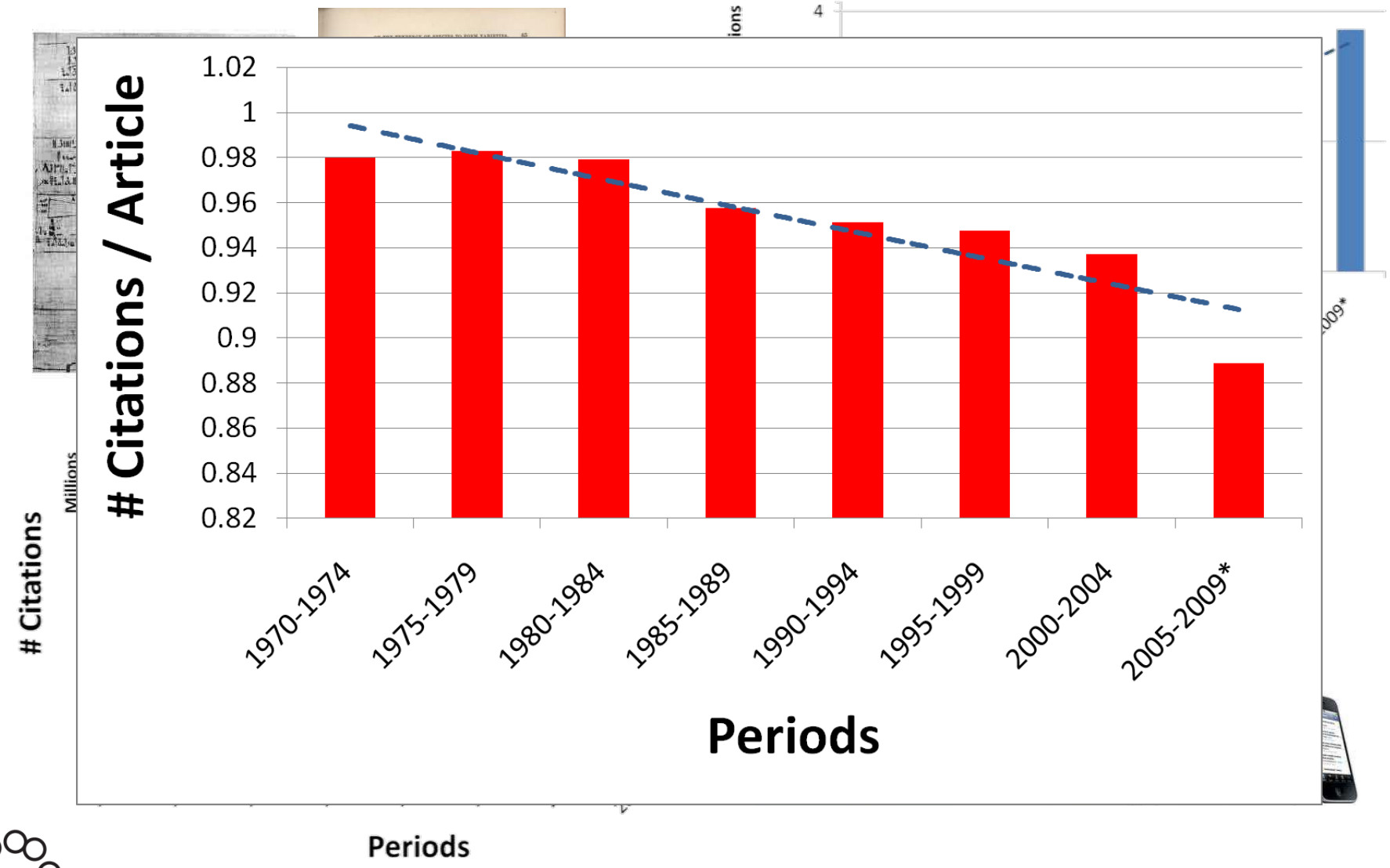
CANADA'S MICHAEL SMITH GENOME SCIENCES CENTRE

BC Cancer Agency CARE & RESEARCH

Moore's Law

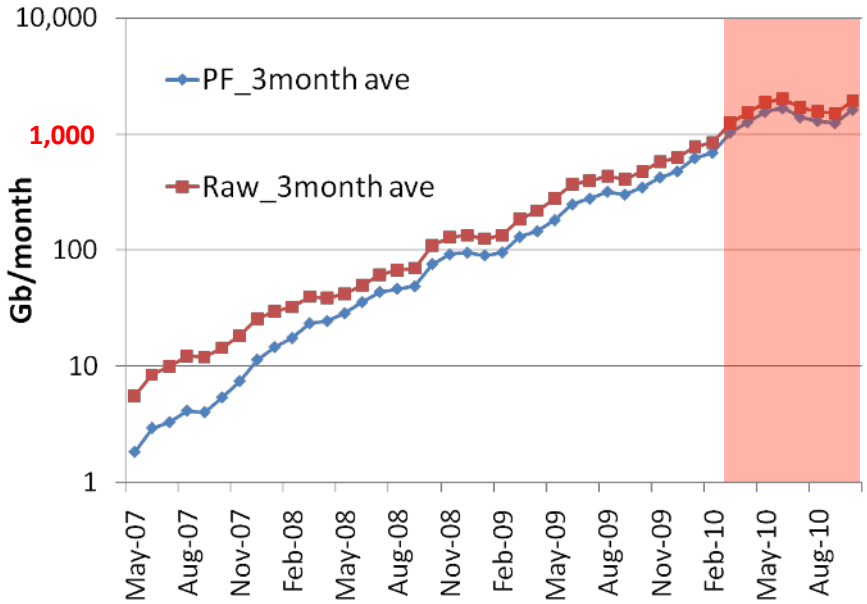

Transistor count
Calculations per Second per $1,000
10^{10}
10^{8}
10^{6}
10^{4}
10^{2}
1
10^{-2}
10^{-4}
10^{-6}
Electromechanical
Relay
Vacuum Tube
Transistor
Integrated Circuit
1900 1910 1920 1930 1940 1950 1960 1970 1980 1990 2000
Year

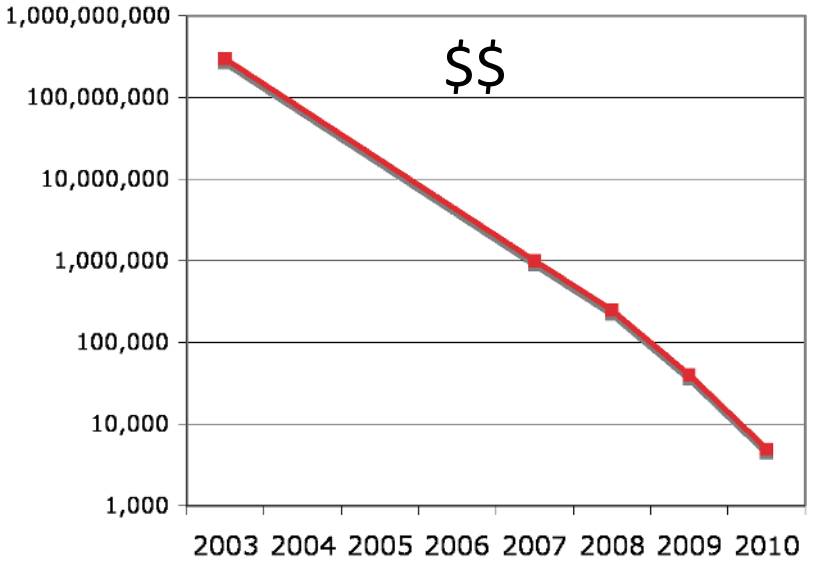
Growth of Knowledge



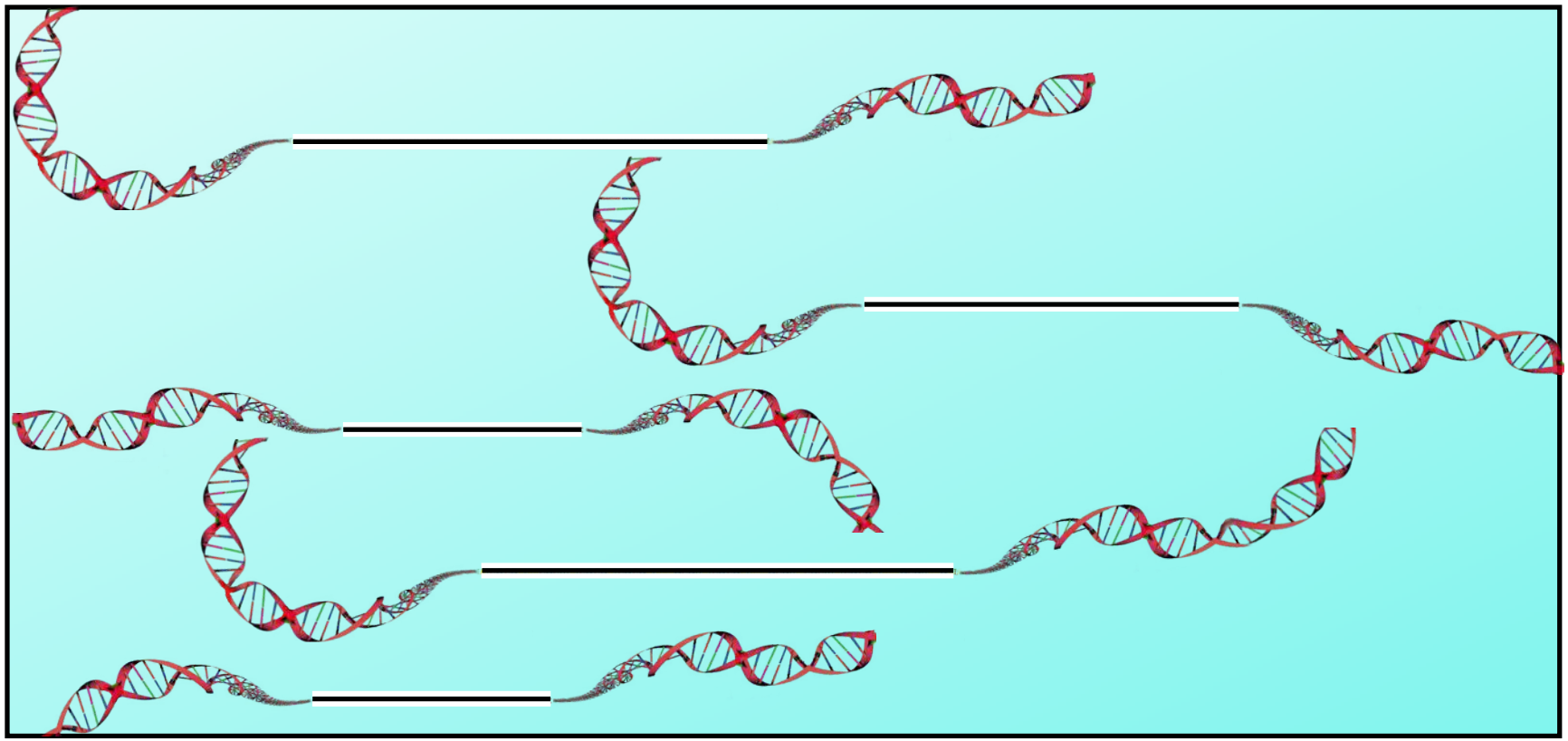
Next Generation Sequencing

- Illumina sequencing throughput at GSC
- Cost of sequencing human genome



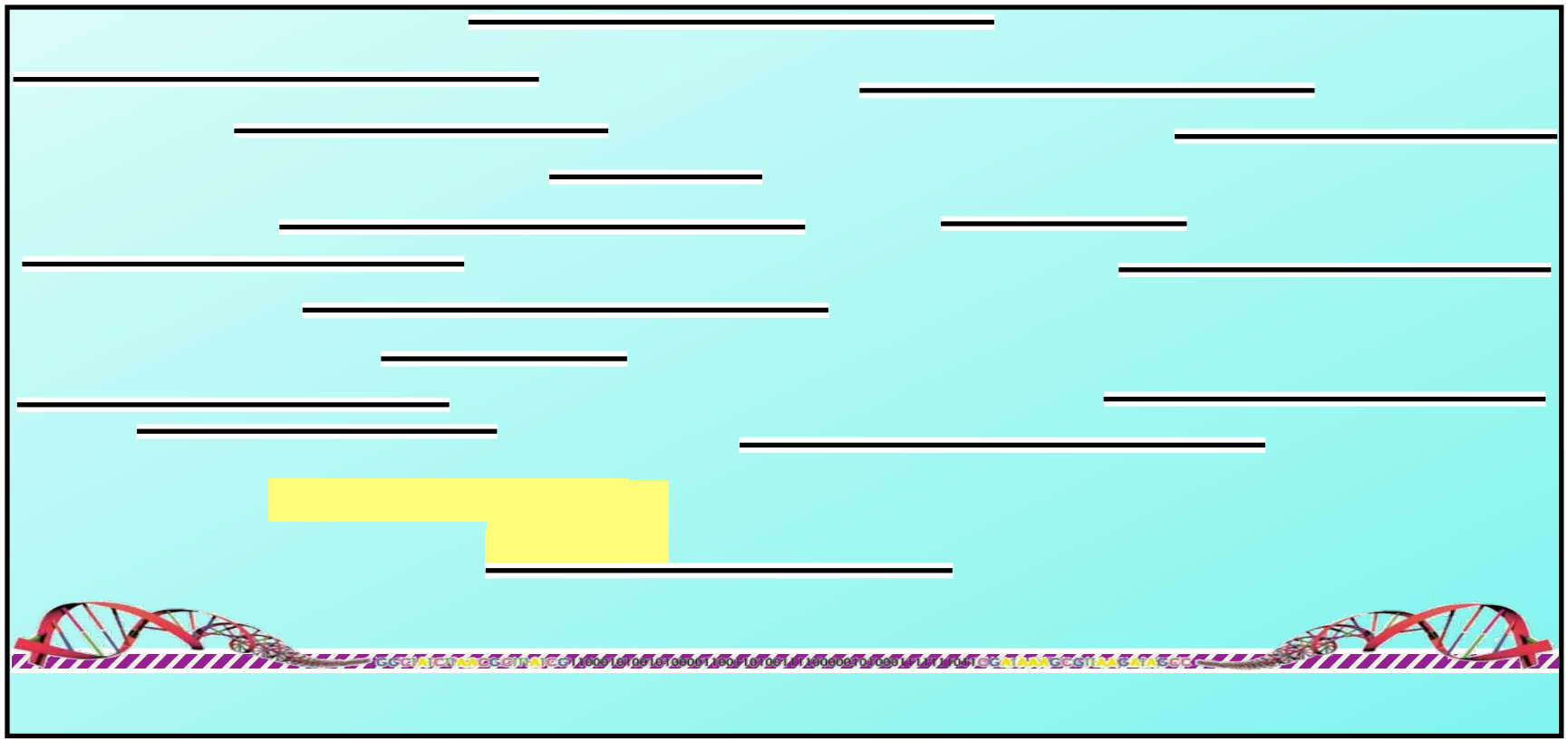


De Novo Assembly Problem



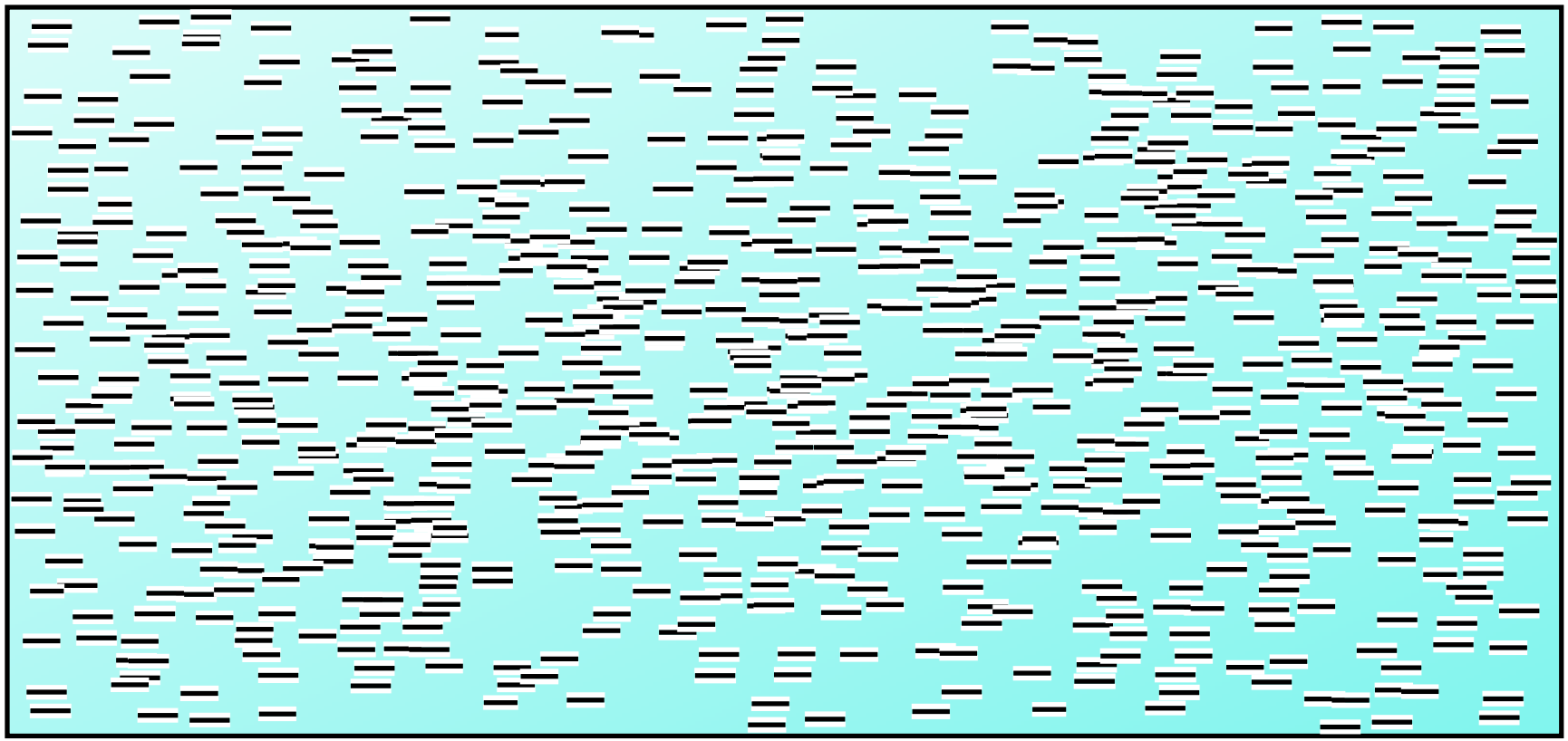
- Old paradigm:
  - long and non-uniform reads (800bp - 1000bp)

De Novo Assembly Problem



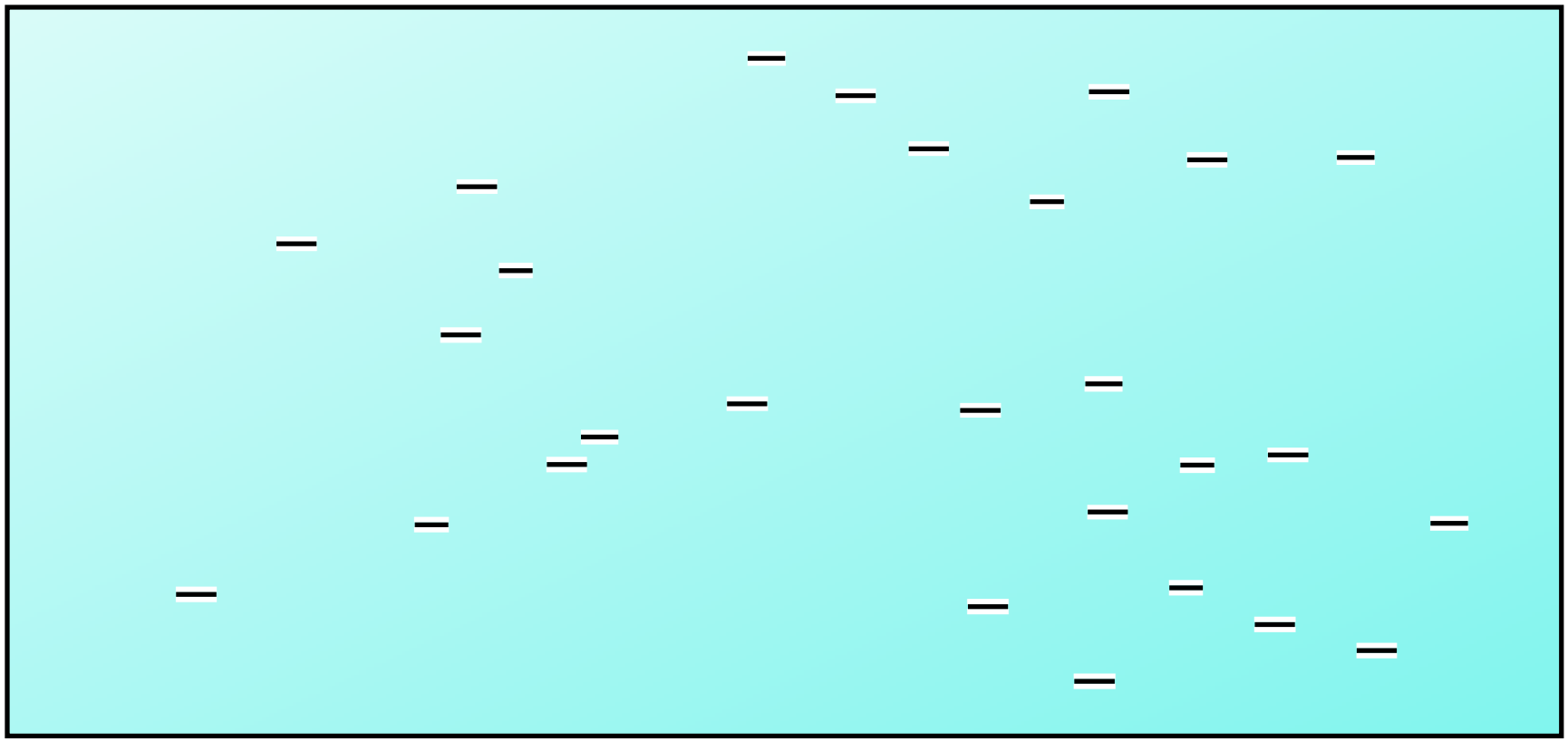
- Old paradigm:
  - long and non-uniform reads (800bp - 1000bp)
  - overlap; overlay; consensus

De Novo Assembly Problem



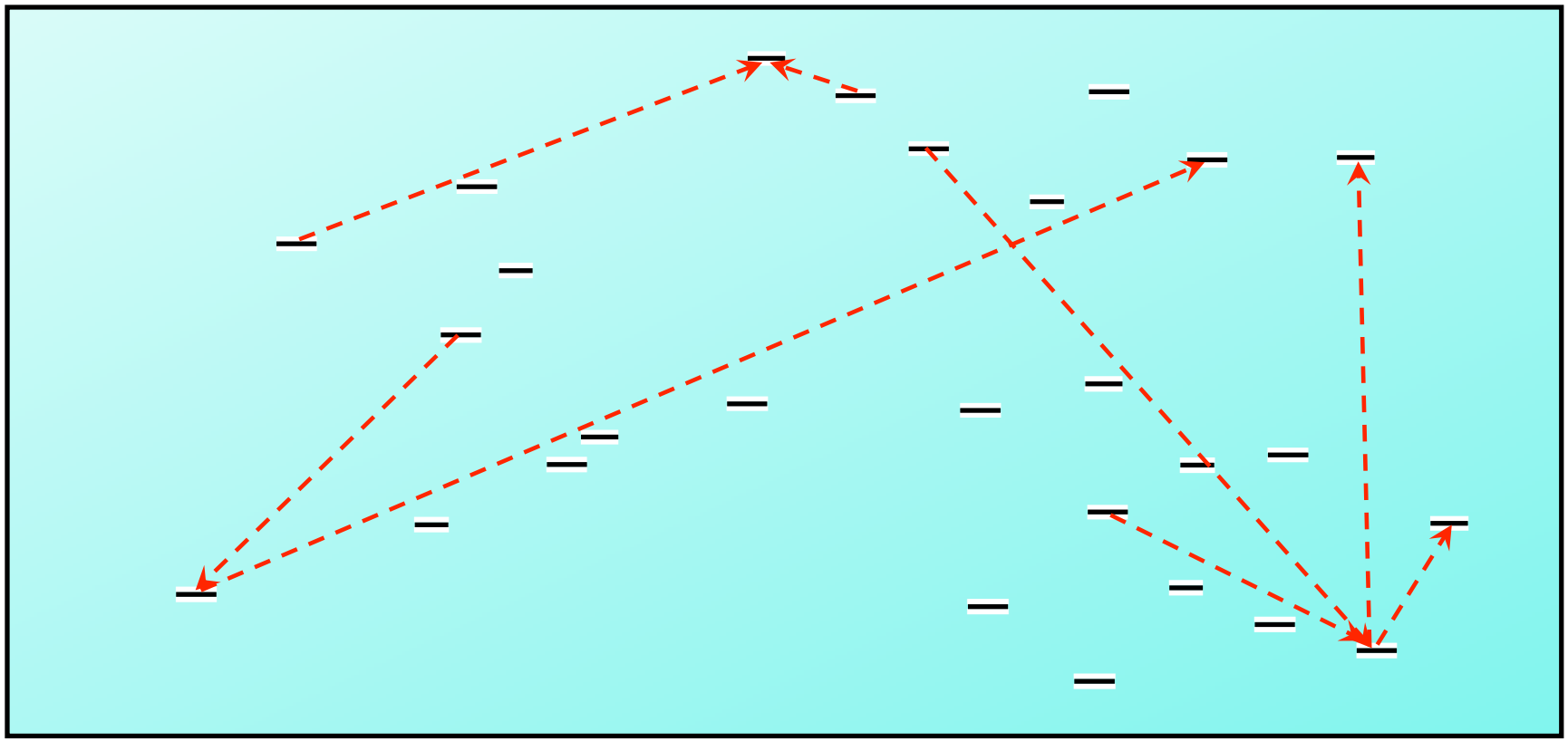
- New paradigm:
  - short and uniform reads (50bp - 150bp)

De Novo Assembly Problem



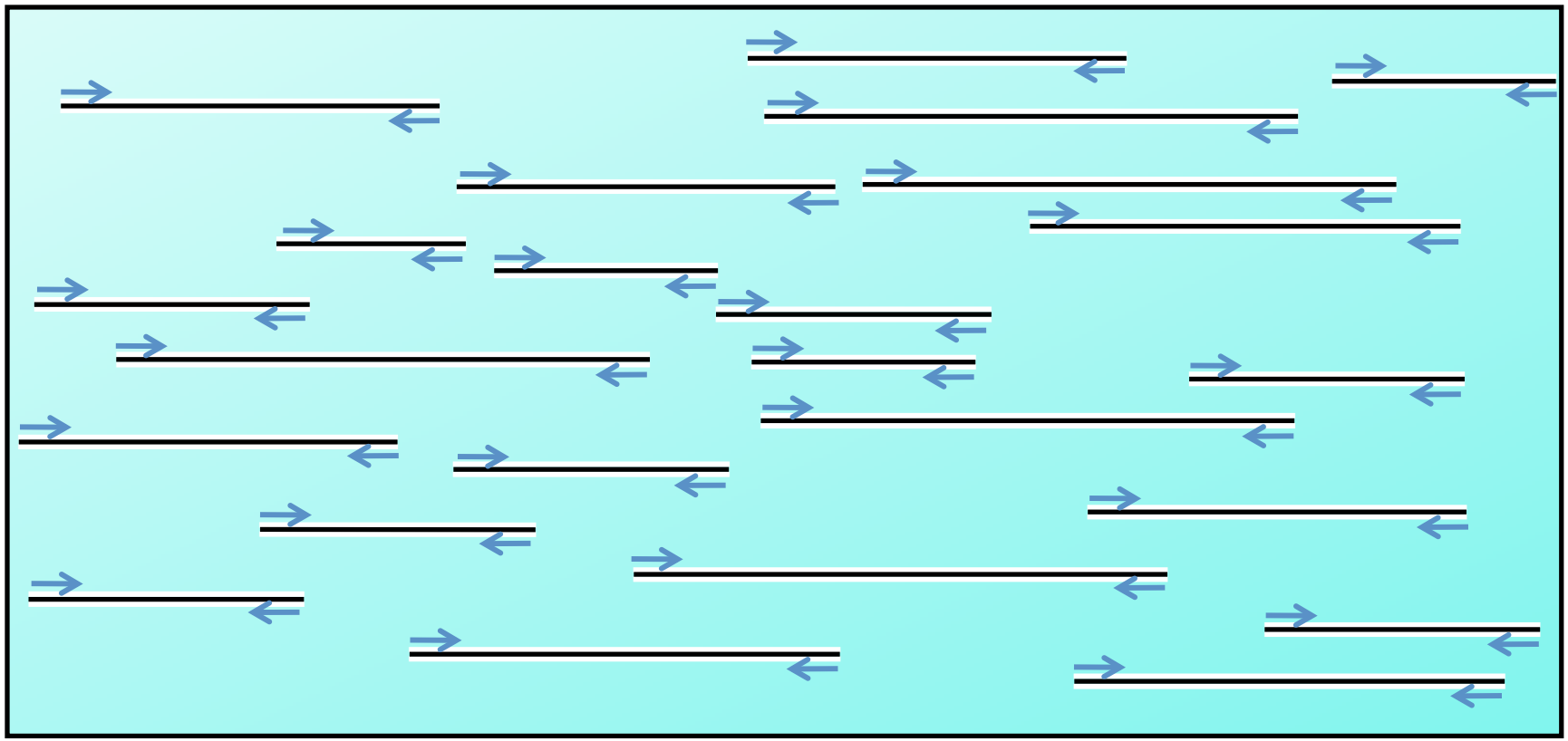
- New paradigm:
  - short and uniform reads (50bp - 150bp)
  - overlap; overlay; consensus

De Novo Assembly Problem



- New paradigm:
  - short and uniform reads (50bp - 150bp)
  - de Bruijn graphs

De Novo Assembly Problem



- New paradigm:
  - long range information through read pairs
  - graph theoretic approaches

Assembly By Short Sequencing

Resource

ABySS: A parallel assembler for short read sequence data

Jared T. Simpson,[1] Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J.M. Jones, and Inanç Birol[2]

Genome Sciences Centre, British Columbia Cancer Agency, Vancouver, British Columbia V5Z 4E6, Canada

Widespread adoption of massively parallel deoxyribonucleic acid (DNA) sequencing instruments has prompted the recent development of de novo short read assembly algorithms. A common shortcoming of the available tools is their inability to

IEEE InfoVis 2009

ABySS-Explorer: Visualizing Genome Sequence Assemblies

Cydney B. Nielsen, Shaun D. Jackman, Inanç Birol, and Steven J.M. Jones

1. ABySS-Explorer employs a novel graph representation enabling biologists to examine the global structure of a genome ence assembly.

tract—One bottleneck in large-scale genome sequencing projects is reconstructing the full genome sequence from the short sub- ences produced by current technologies. The final stages of the genome assembly process inevitably require manual inspection ta inconsistencies and could be greatly aided by visualization. This paper presents our design decisions in translating key data res identified through discussions with analysts into a concise visual encoding. Current visualization tools in this domain focus cal sequence errors making high-level inspection of the assembly difficult if not impossible. We present a novel interactive graph ay, ABySS-Explorer, that emphasizes the global assembly structure while also integrating salient data features such as sequence

Jackman and Birol Genome Biology 2010, 11:202
http://genomebiology.com/2010/11/1/202

Genome Biology

MINIREVIEW

Assembling genomes using short-read sequencing technology

Shaun D Jackman and Inanç Birol*

Bioinformatics Advance Access published June 15, 2009

Original Paper

De novo Transcriptome Assembly with ABySS

Inanç Birol[2,*], Shaun D Jackman[1], Cydney Nielsen[1], Jenny Q Qian[1], Richard Varhol[1], Greg Stazyk[1], Ryan D Morin[1], Yongjun Zhao[1], Martin Hirst[1], Jacqueline E Schein[1], Doug E Horsman[1], Joseph M Connors[2], Randy D Gascoyne[2], Marco A Marra[1] and Steven JM Jones[1]

[1] Genome Sciences Centre, 100-570 W 7th Avenue, Vancouver BC V5Z 4S6, Canada, www.bcgsc.ca

[2] British Columbia Cancer Agency, 600 West 10th Avenue, Vancouver, BC V5Z 4E6, Canada, www.bccancer.ca

Associate Editor: Dr. Alex Bateman

ABSTRACT

Motivation: Whole transcriptome shotgun sequencing data from non-normalized samples offer unique opportunities to study the metabolic states of organisms. One can deduce gene expression levels using sequence coverage as a surrogate, identify coding changes or discover novel isoforms or transcripts. Especially for discovery of novel events, *de novo* assembly of transcriptomes is desirable.

Results: Transcriptome from tumor tissue of a patient with follicular data (Jackson et al., 2009), but it is not yet demonstrated to be applicable to experimental data. Here, we present a *de novo* assembly approach for transcriptome analysis using the ABySS assembler tool (Simpson et al., 2009), which works on experimental data, and we show that transcriptome assembly yields interesting biological insights. ABySS was developed initially for *de novo* assembly of genomes, with a special emphasis on large genomes, and we previously demonstrated its capacity by assembling the human genome using 36bp to 42bp short reads.

BRIEF COMMUNICATIONS

De novo assembly and analysis of RNA-seq data

Gordon Robertson[1], Jacqueline Schein[1], Readman Chiu[1], Richard Corbett[1], Matthew Field[1], Shaun D Jackman[1], Karen Mungall[1], Sam Lee[2], Hisanaga Mark Okada[1], Jenny Q Qian[1], Malachi Griffith[1], Anthony Raymond[1], Nina Thiessen[1], Timothee Cezard[1,4], Yaron S Butterfield[1], Richard Newsome[1], Simon K Chan[1], Rong She[1], Richard Varhol[1], Baljit Kamoh[1], Anna-Liisa Prabhu[1], Angela Tam[1], Yonglun Zhao[1], Richard A Moore[1], Martin Hirst[1], Marco A Marra[1,3], Steven J M Jones[1,3], Pamela A Hoodless[2,3] & Inanc Birol[1]

We describe Trans-ABySS, a *de novo* short-read transcriptome assembly and analysis pipeline that addresses variation in local read densities by assembling read substrings with varying stringencies and then merging the resulting contigs before analysis. Analyzing 7.4 gigabases of 50-base-pair paired-end Illumina reads from an adult mouse liver poly(A) RNA library, we identified known, new and alternative structures in expressed transcripts, and achieved high sensitivity and specificity relative to reference-based assembly methods.

Current methods for sequencing transcriptomes using short-read technologies (RNA-seq) generate millions of short sequence reads. These reads are associated with transcript models after mapping the reads to a reference genome, facilitated by extending the genome sequence to include sequences for junctions between annotated exons[1], as in alternative expression analysis

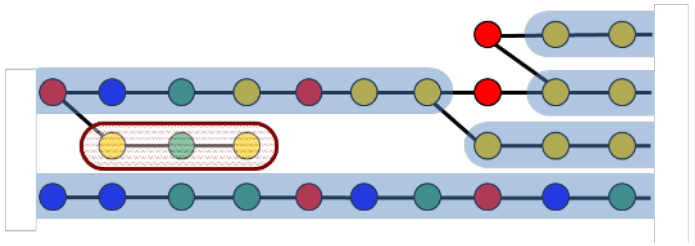
The parameters required for an effective assembly depend on the depth of coverage. For de Bruijn graph assemblers such as ABySS[16,17], which process each read into a set of overlapping substrings (*k*-mers) of length *k* base pairs (bp), the most important parameter is the *k*-mer length. Whole-genome shotgun assembly sequencing libraries attempt to provide a uniform representation of the genome. For these libraries it is reasonable to identify and work with the assembly that corresponds to an optimal *k* value. In non-normalized transcriptome shotgun libraries, however, individual transcripts differ widely in expression and thus present a wide range of sequence representations to an assembler. A single *k* value is therefore unlikely to yield an optimal overall assembly (**Supplementary Note 1**).

Recently, we applied the ABySS short-read assembler to human transcriptome data[18]. Based on an assembly for a single *k* value, in this preliminary analysis we had identified contig structures and alignments that are consistent with alternative isoforms, and thus suggested that ABySS could be effective for transcriptome analysis. However, to make *de novo* assembly practical for characterizing annotated and new transcript structures, we anticipated that it would be necessary to assemble at different *k* values to address variable transcript expression and multiple expressed isoforms. Here we describe the result of work to address these issues, Trans-ABySS, a method and pipeline for assembly and analysis of non-normalized short-read transcriptome data.

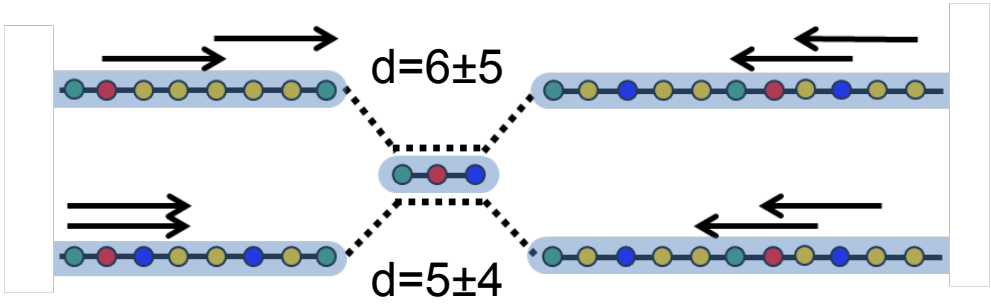
To develop the approach and assess its performance, we generated 147.1 million (7.36 gigabases (Gb)) quality-filtered 50-bp Illumina paired-end reads from a transcriptome library constructed from adult mouse liver poly(A) RNA. We chose this model organism because it has a well-annotated transcriptome and is considered genetically uniform. We assembled the reads using ABySS v1.1.1 (**Supplementary Fig. 1** and **Supplementary Note 1**).

To assess assembly performance as a function of *k* values,

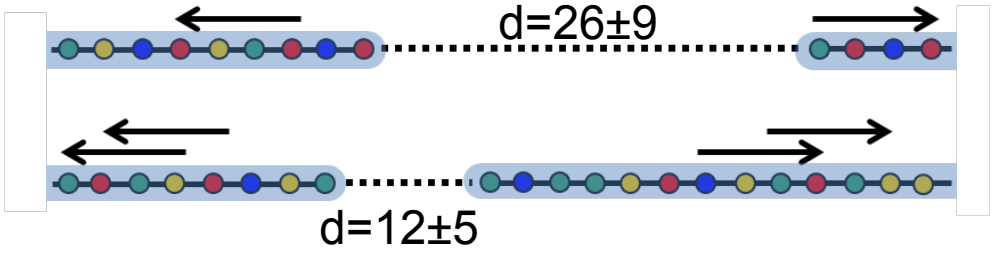
- SE Assembly: k-mer extension on a de Bruijn graph



- PE Assembly: search for unambiguous contig merging along paths



- Scaffolding: search for unambiguous linkage across distant contigs



comments on this story

Stories by subject
- Genetics
- Technology

Stories by keywords
- Genomics
- Genome sequence
- Genome sequencing
- Bioinformatics
- Genome assembly

This article elsewhere
Blogs linking to this article
Add to Connotea
Add to Digg
Add to Facebook

Published online 23 March 2011 | *Nature* **471**, 425 (2011) | doi:10.1038/471425a

News

Genome builders face the competition

Three independent projects seek to contrast approaches in preparation for routine analysis of genetic data.

Erika Check Hayden

Sequencing DNA on an industrial scale is no longer difficult: the challenge is in assembling a full genome from the multitude of short, overlapping snippets that second-generation sequencing machines

At a meeting last week at the University of California, Santa Cruz, three winners emerged: ALLPATHS-LG, developed by the Broad Institute in Cambridge, Massachusetts; ABySS, developed at the British Columbia Cancer Agency's Genome Sciences Centre in Vancouver, Canada; and SOAPdenovo, developed by the Beijing Genomics Institute. But, Korf notes, "it's not just the software, it's how people are running it" that determines the quality of each assembly.

most recent | commented

- Mice with human livers deal with drugs the human way
  11 July 2011
- Kenya set to give green light to GM crops
  11 July 2011
- All eyes on the potato genome
  10 July 2011
- Sudan splits and science community divides
  08 July 2011
- Qatar sets sights on stem cells
  08 July 2011

Related stories

etly embrace whole-genome

2010

es beyond DNA sequence

uencing: the third generation

009

Assembly Problem

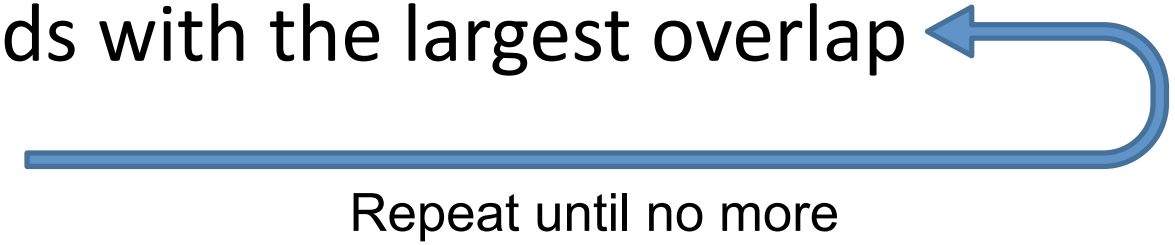
A **partial** and **unambiguous** read-to-read alignment

extends the length of sequence information

- First stage of an assembly algorithm is to find such alignments
- Assembly algorithms differ in the way they find and use these alignments

Greedy Assembly

- Find two reads with the largest overlap
- Merge them



Pro: fast

Con: prone to misassembly

- Assumes largest overlaps are unambiguous

Overlap Overlay Consensus

- Overlap
  Find all pairs of sequences that overlap
- Overlay (a.k.a. Layout)
  Remove redundant and weak overlaps
- Consensus
  Merge pairs of sequences that overlap unambiguously
  Build a consensus sequence from all reads overlaid in a region

Find Overlapping Reads

- Naïve algorithm: make all binary comparisons
  Untenable when too many reads
  - $O(n^2)$
  - RAM
  - CPU

ARACHNE
CAP3
Celera assembler
MIRA
Newbler
Phred/Phrap

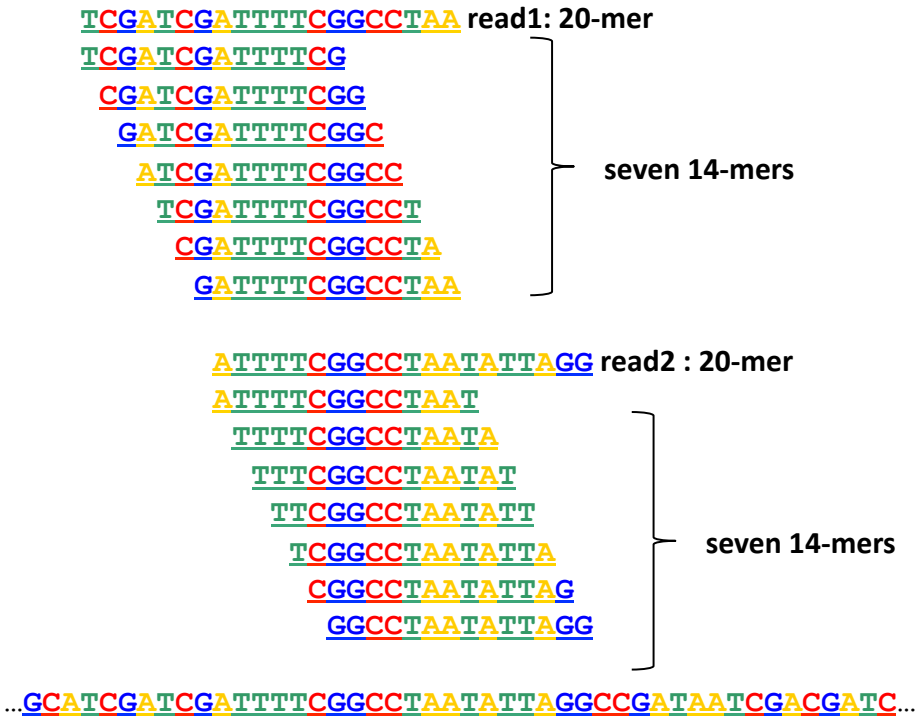
- Ferragina-Manzini index
  - Apply Burrows-Wheeler transform
  - Small memory footprint

SGA

Build an overlap graph

Forget About Overlapping Reads!

- Shred reads to a uniform length *k*
- Build a special overlap graph: de Bruijn Graph

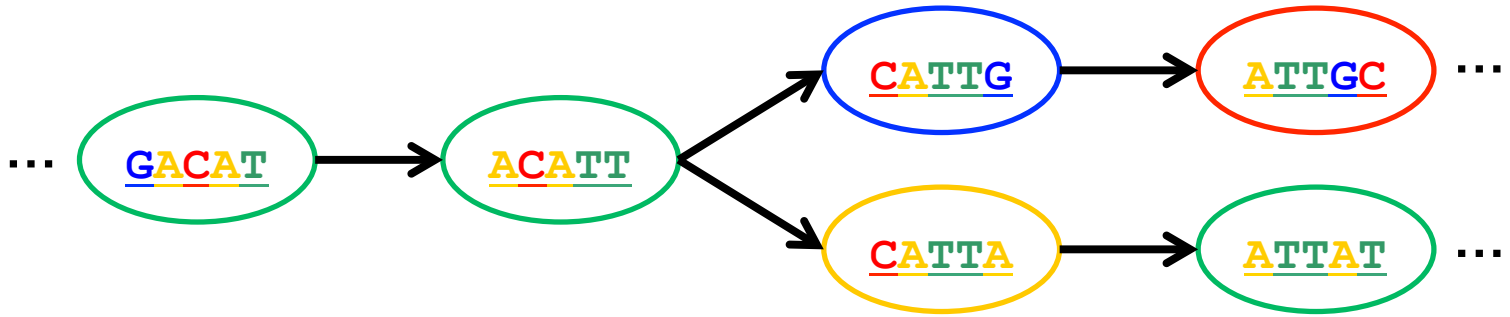


Euler
Velvet
ABySS
SOAPdenovo
ALLPATHS

De Bruijn Graph

- Load *k*-mers in memory
  - 2x4 possible extension of every *k*-mer
- Check if there is a "next" *k*-mer
  - O(*n*) algorithm

…GACATTGC… **seq1**
…GACATTAT… **seq2**



k = 5

Memory Concerns

- Human genome has over 2 billion unique *k*-mers
- If we represent every *k-mer* using, say 50 bytes
  we require over 100 GB RAM
  just to represent *k*-mers

Solution #1: Clustering reads

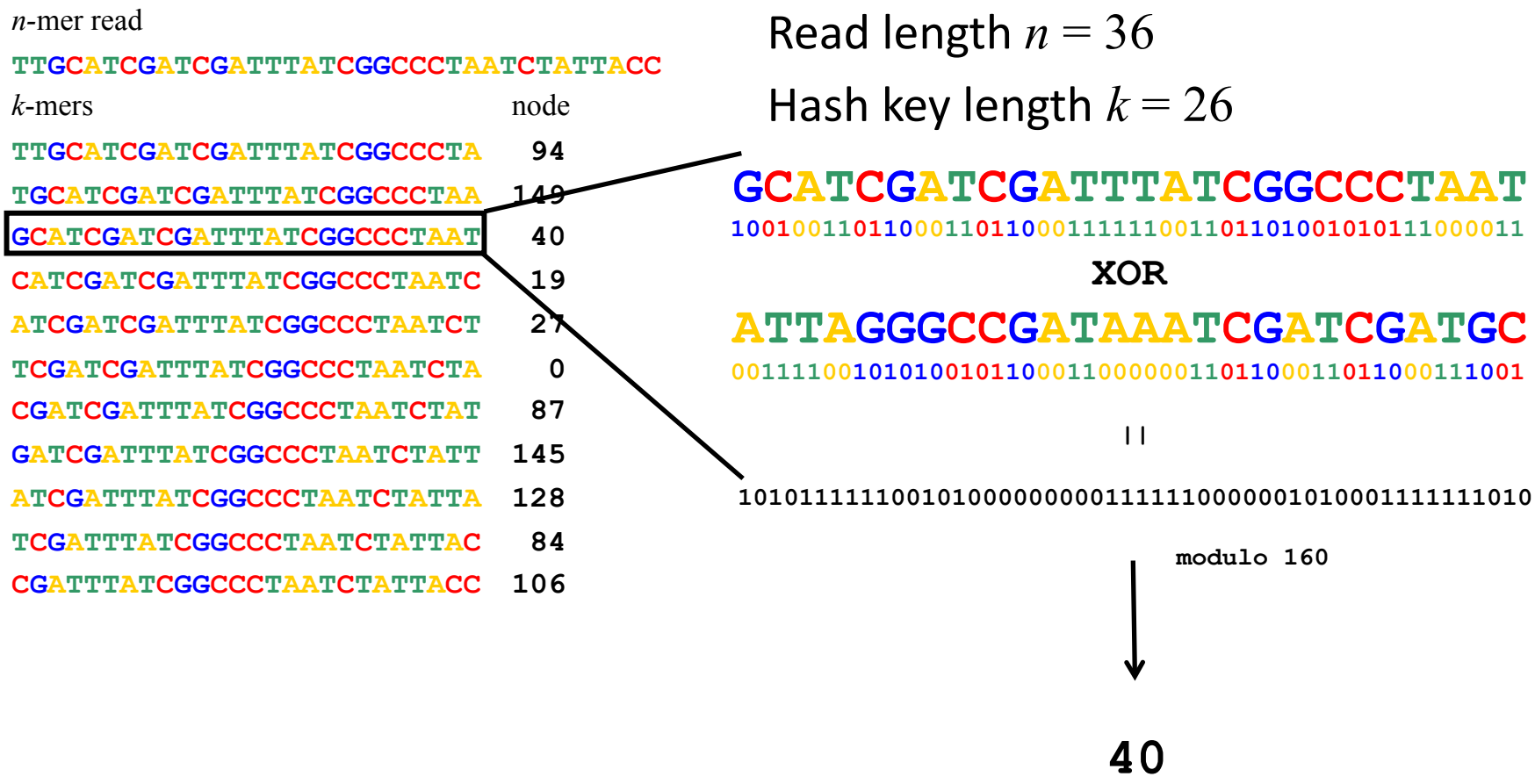
Curtain (w/ Velvet)
Phusion (w/ Phrap)

Solution #2: Distributed computing

ABySS
SOAPdenovo
ALLPATH-LG

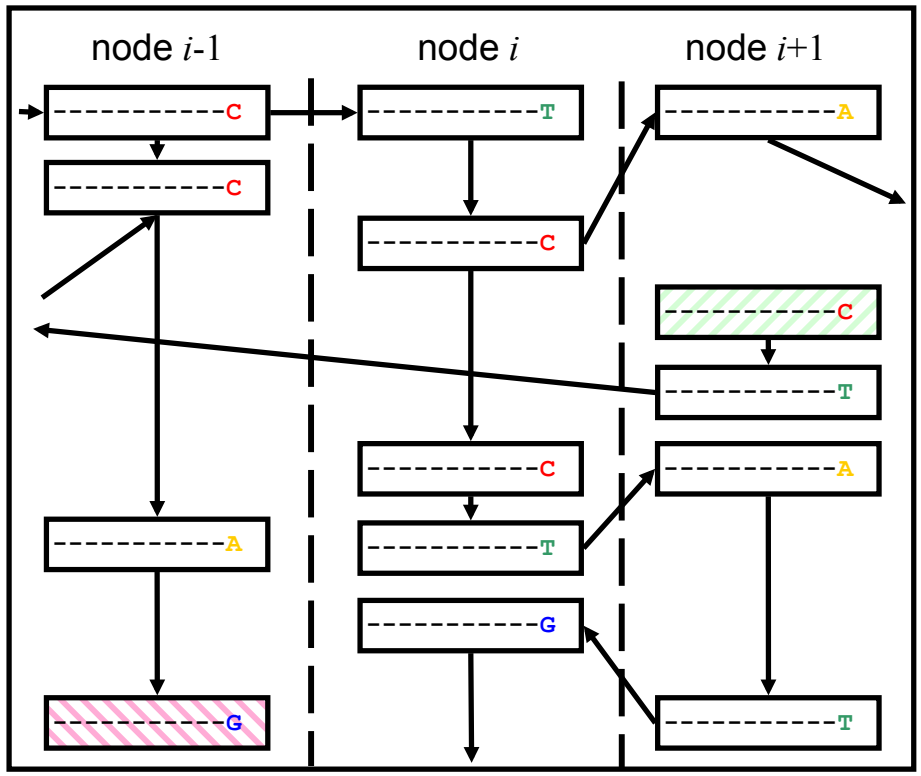
Partitioning Read Space

Distribute sub-reads and reverse-complements over nodes



Graph Generation

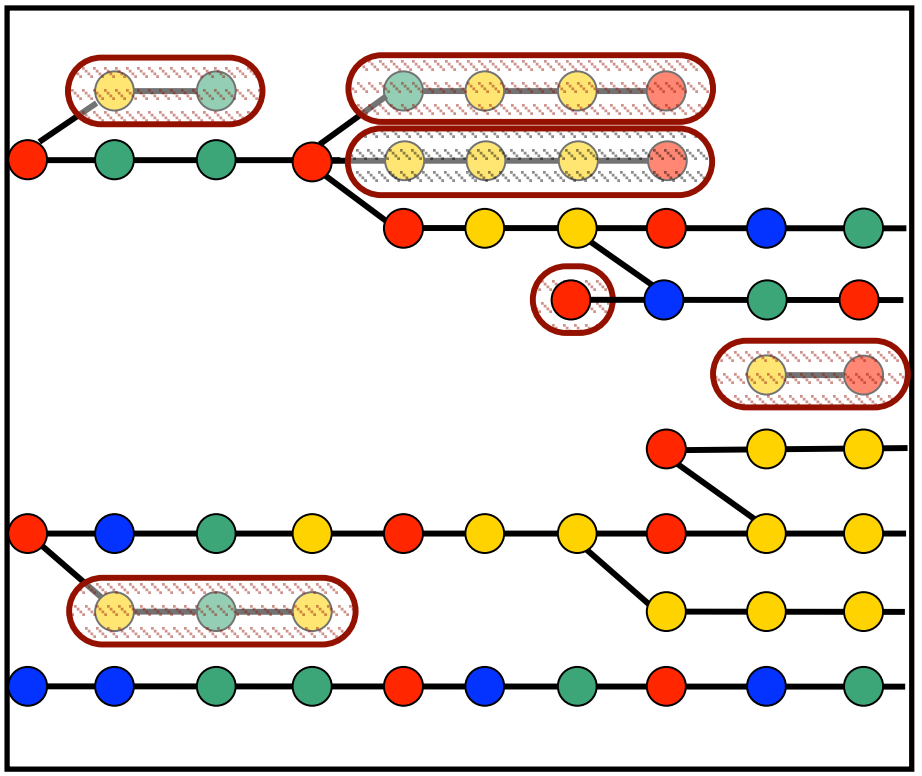
- A given k-mer can have up to 8 extensions
- Each node announces the list of k-mers that it has to the nodes that hold their possible extensions
- Each node records if there are any extensions of the k-mers that it stores



- This forms adjacency information for k-mers over a distributed de Bruijn graph

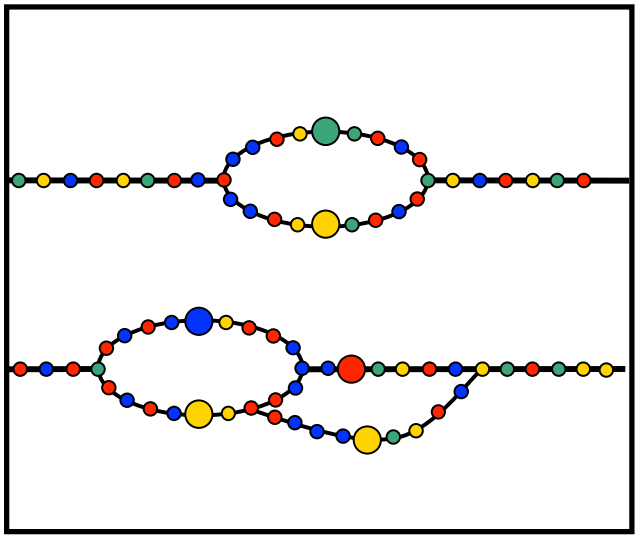
Trimming

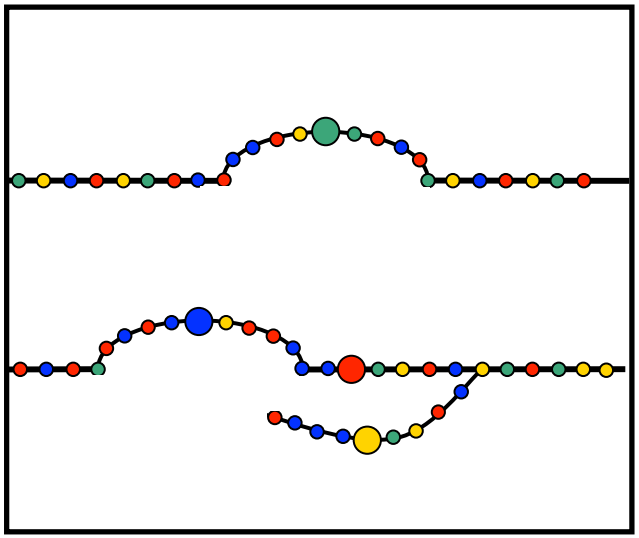
- Data would have experimental noise
- de Bruijn graph would have false branches
- Some read errors are filtered by removing such branches
- Trimming prevents the later assembly step to come to a premature end because of read errors



Bubble Popping

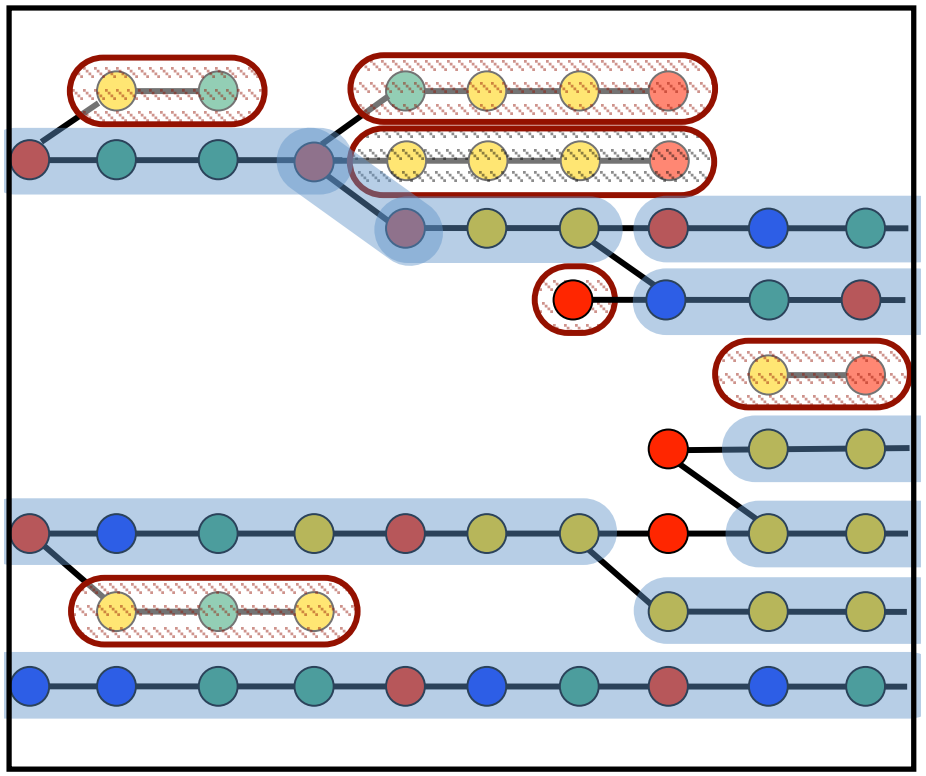
- Repeat read errors and single nucleotide allelic differences would cause “bubbles” of length $2k$-1
- Bubbles are popped by removing either of those branches
- Complex bubbles can form when multiple bubbles intersect
  - Bubble popping step either reduces the bubble orders by one
  - Or creates dead branches
- Popped bubbles are recorded in a log file to study potential allelic differences





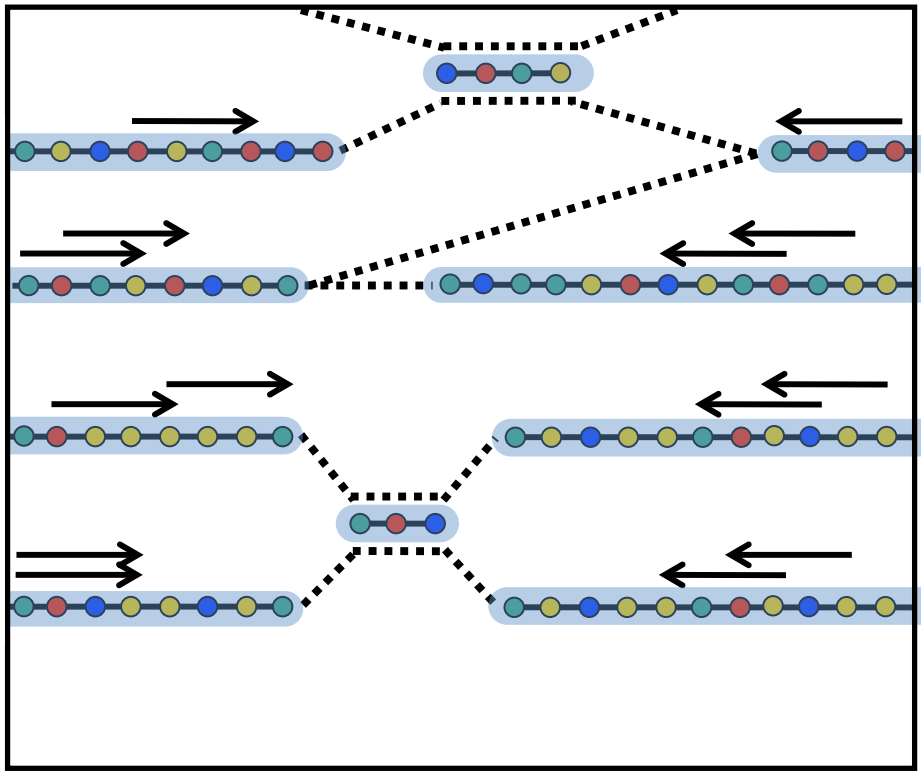
Assembly - SET

- Remaining de Bruijn graph is analyzed for contig extension ambiguities
- If there is a multiplicity in the inbound or outbound contig extensions, then contig growth is terminated



- SET assembly step then concatenates the remaining connected nodes in the di-graph, creating independent contigs that overlap by no more than k-1 bases

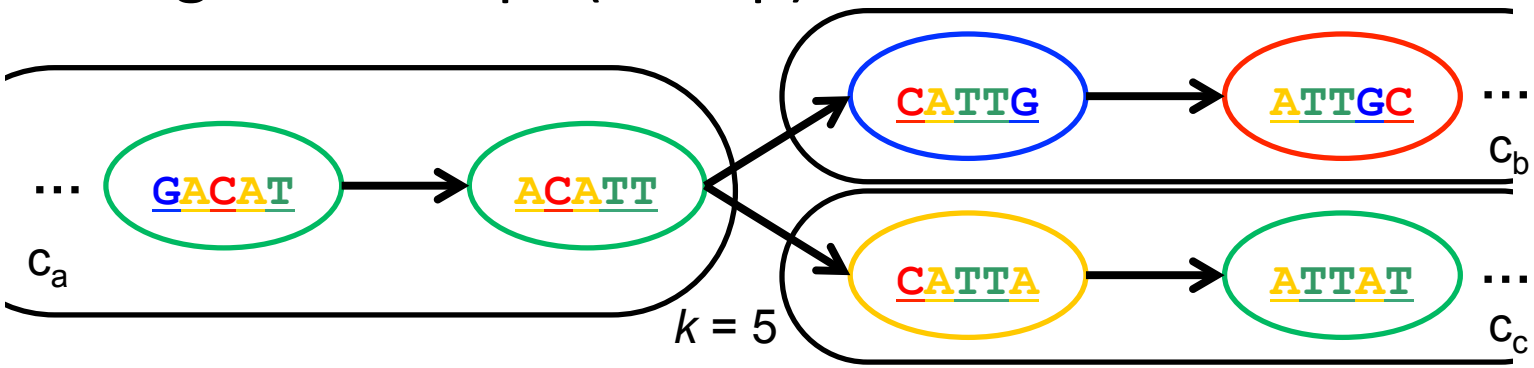
Assembly - PET

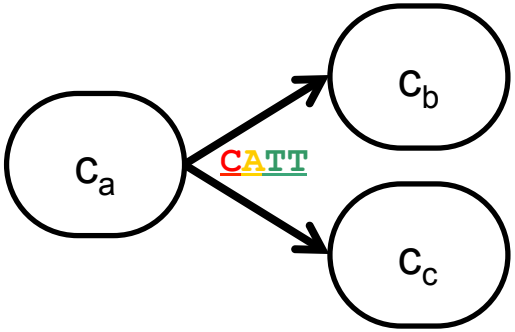


- After SET assembly, reads are aligned to contigs
- Using reads that hit the same contig, empirical fragment size distribution(s) is (are) calculated
- Using reads that hit multiple contigs, inter-contig distances are inferred with a maximum likelihood estimator
- Contigs with coherent and unambiguous distances are joined

Adjacency Graph

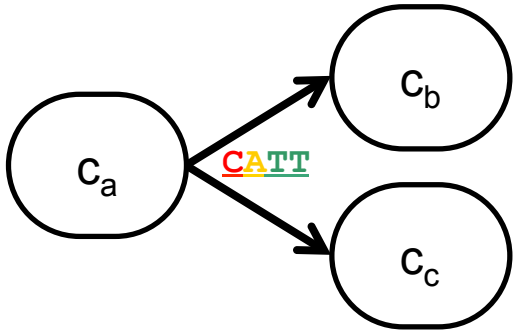
- SET assembly result as a graph
  - Nodes: contigs
  - Edges: overlaps (*k*-1 bp)

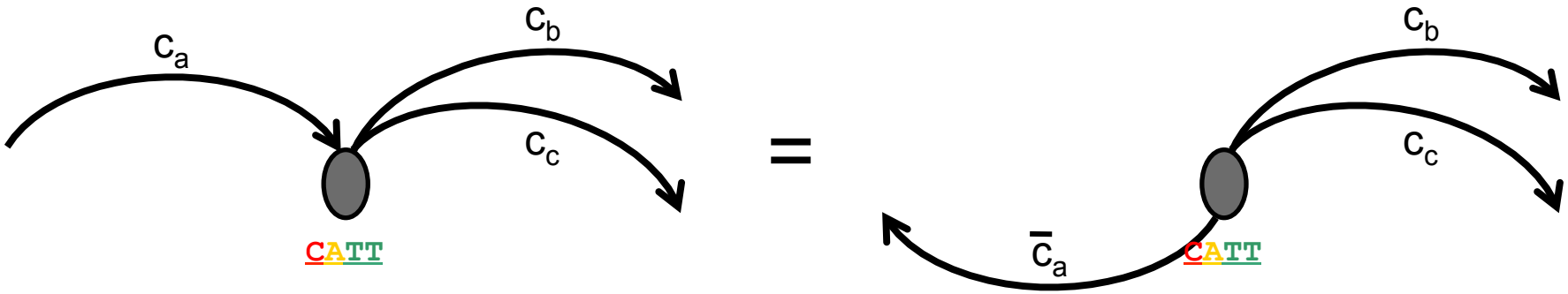




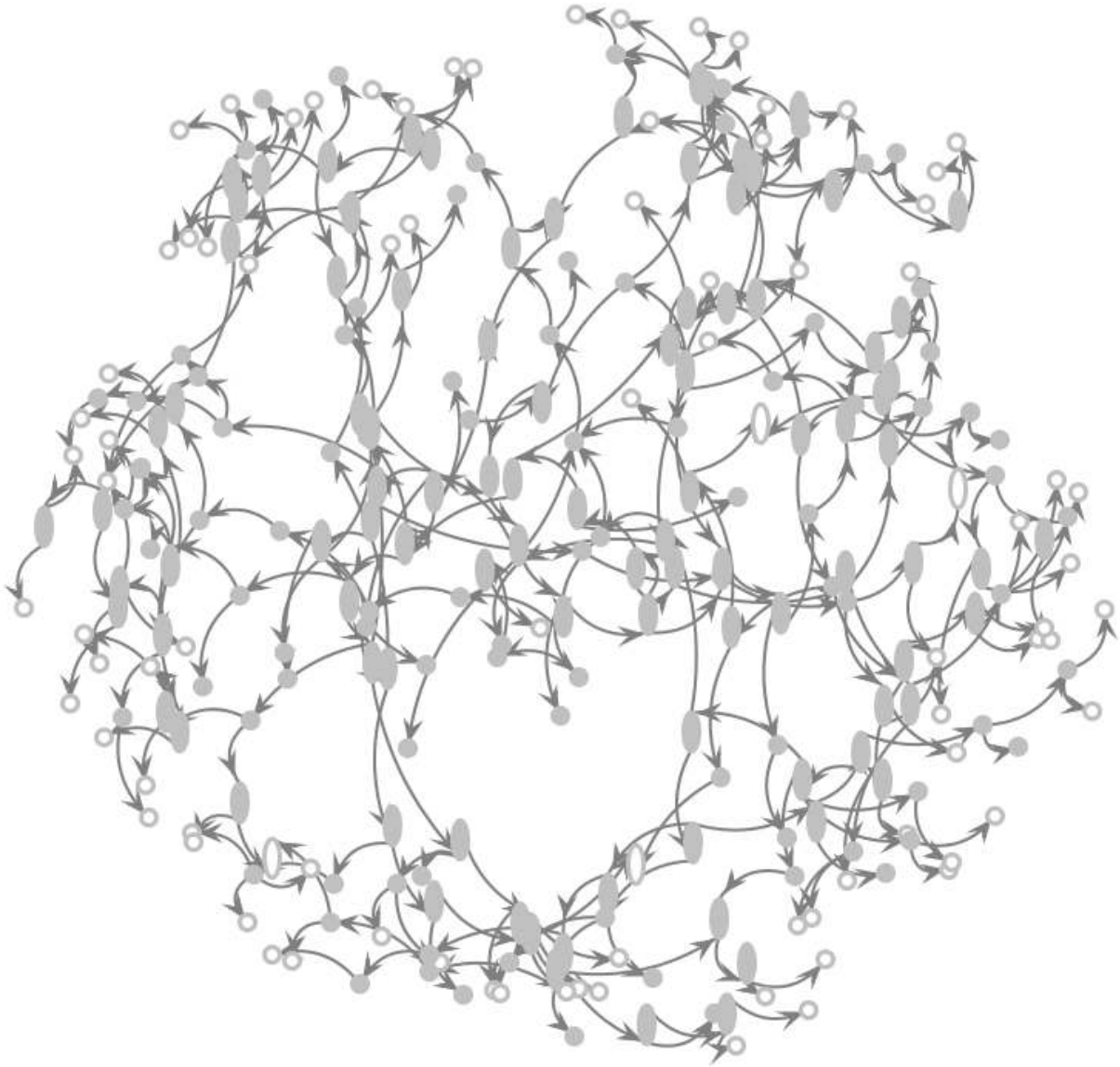
Adjacency Graph

- SET assembly result as a graph
  - Nodes: overlaps (*k*-1 bp)
  - Edges: contigs





Assembly As a Hairball

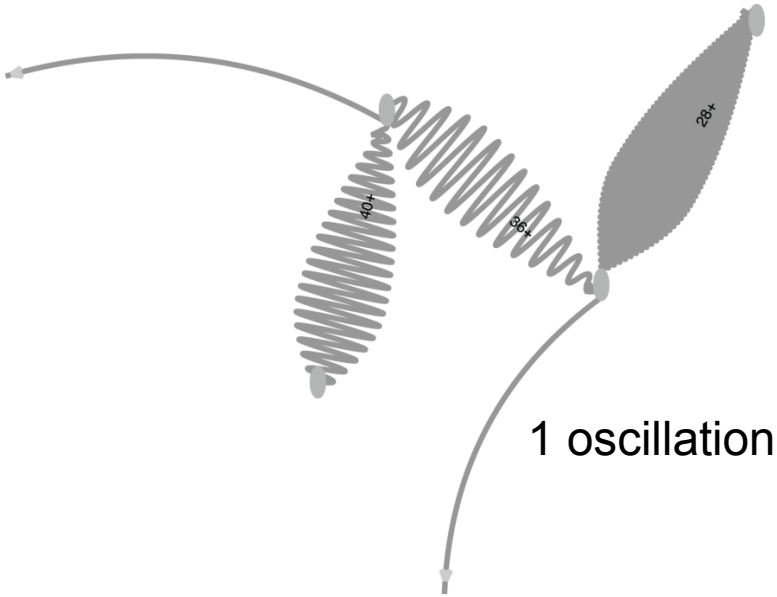


IEEE InfoVis 2009

ABySS-Explorer: Visualizing Genome Sequence Assemblies

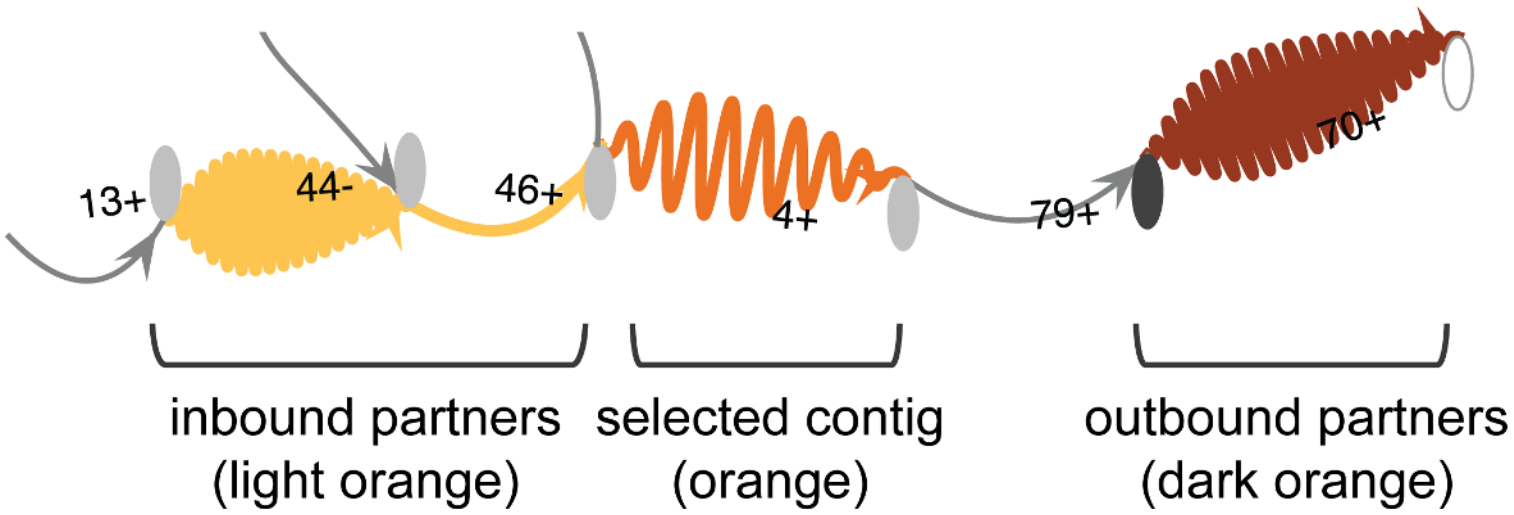
Cydney B. Nielsen, Shaun D. Jackman, Inanç Birol, and Steven J.M. Jones

Contig Length

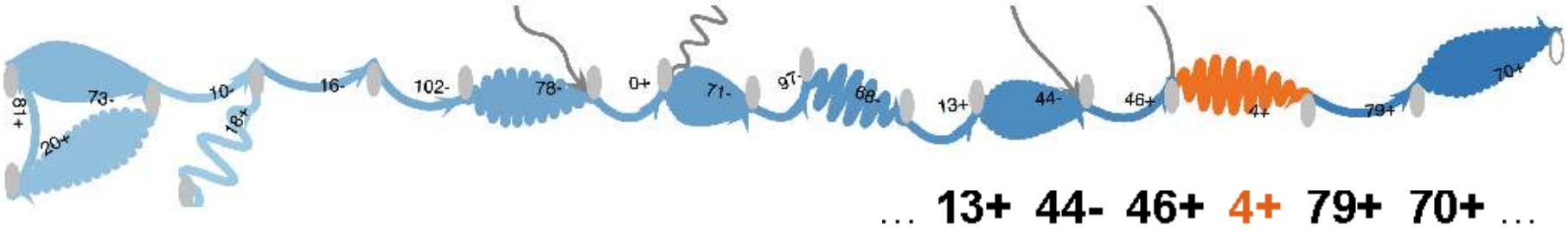


1 oscillation = 100 bp

Paired End Tag Information



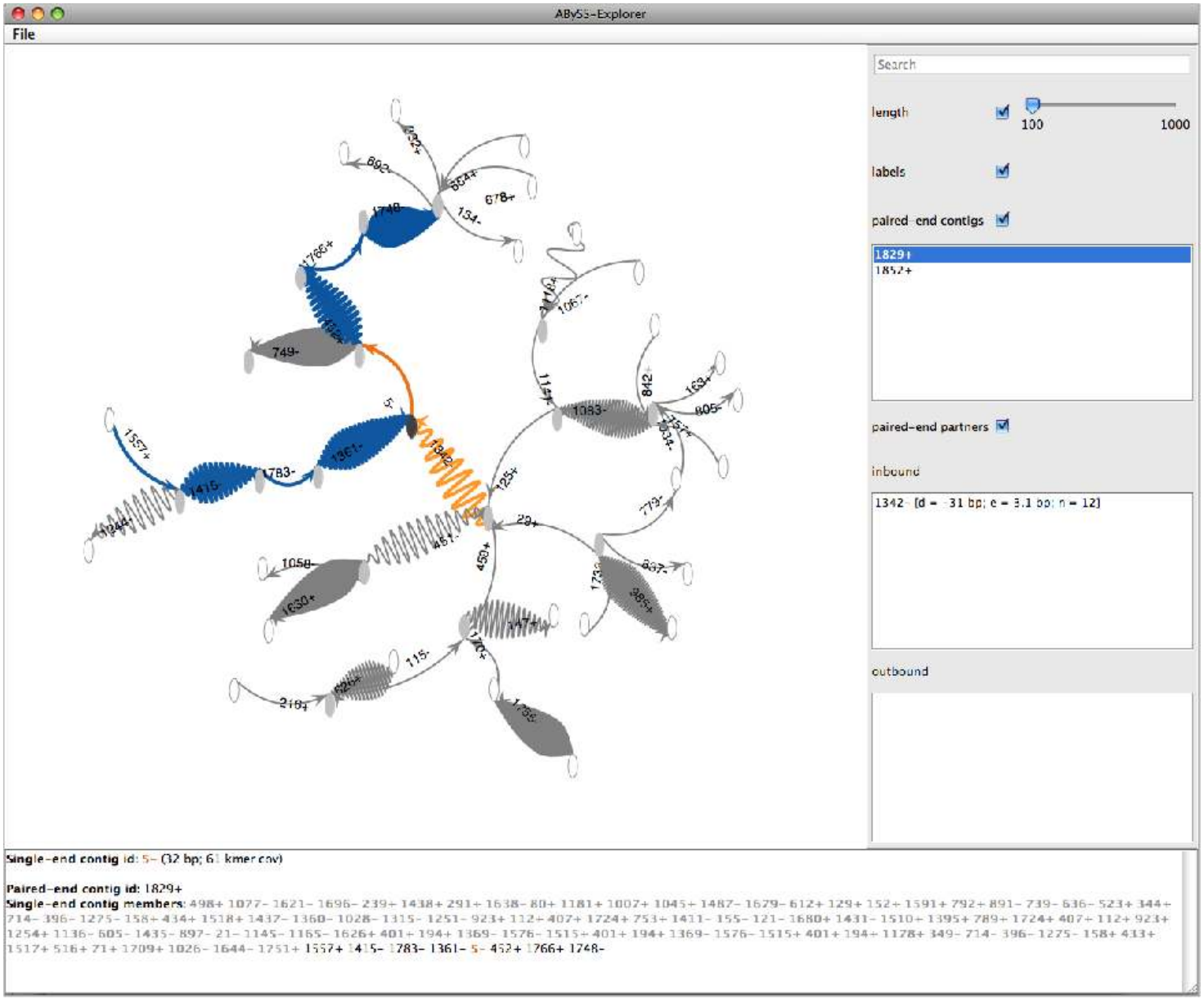
Paired End Contigs



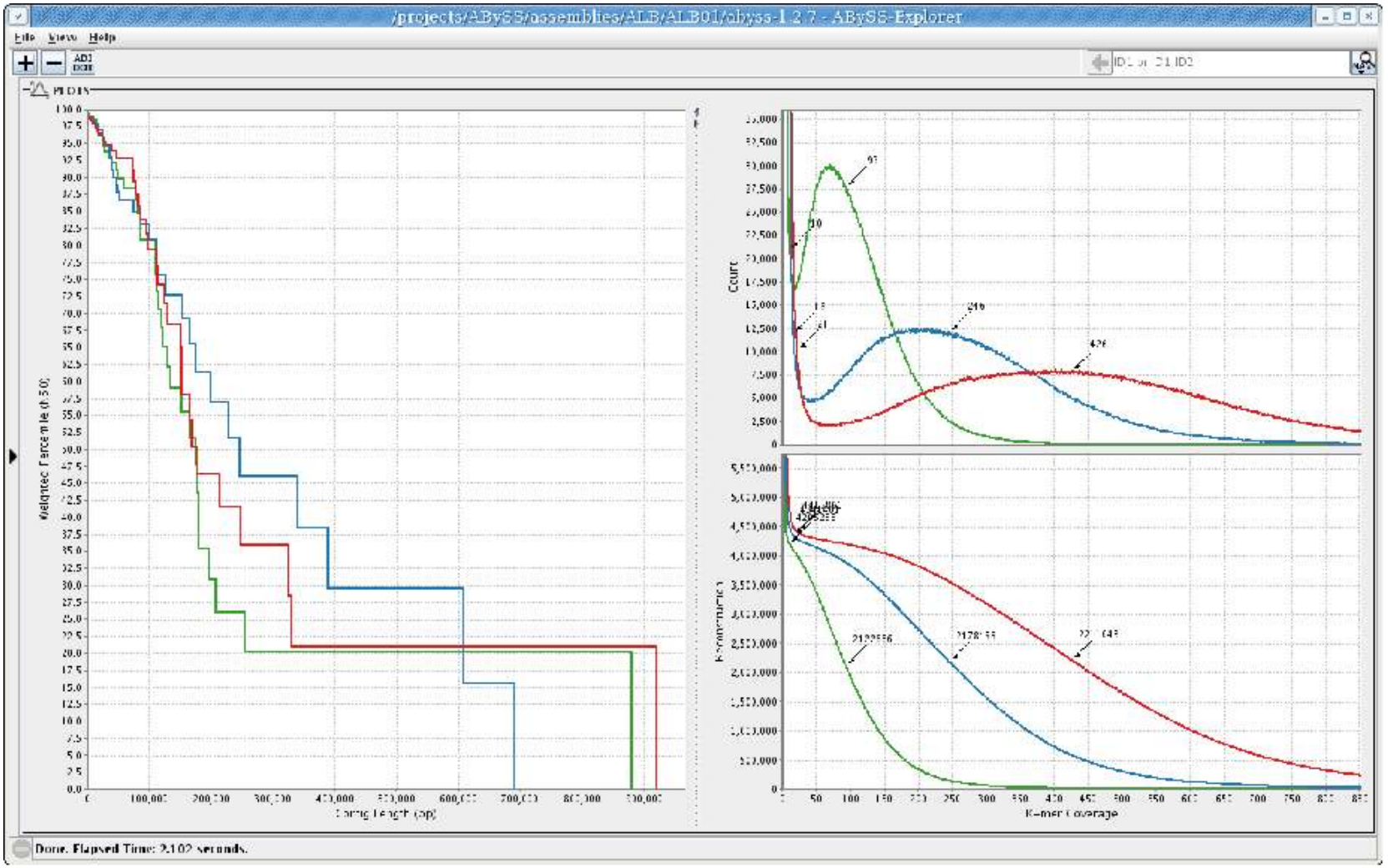
… **13+ 44- 46+ 4+ 79+ 70+** …

Blue gradient: SET contig path in PET assembly
Orange: selected SET contig

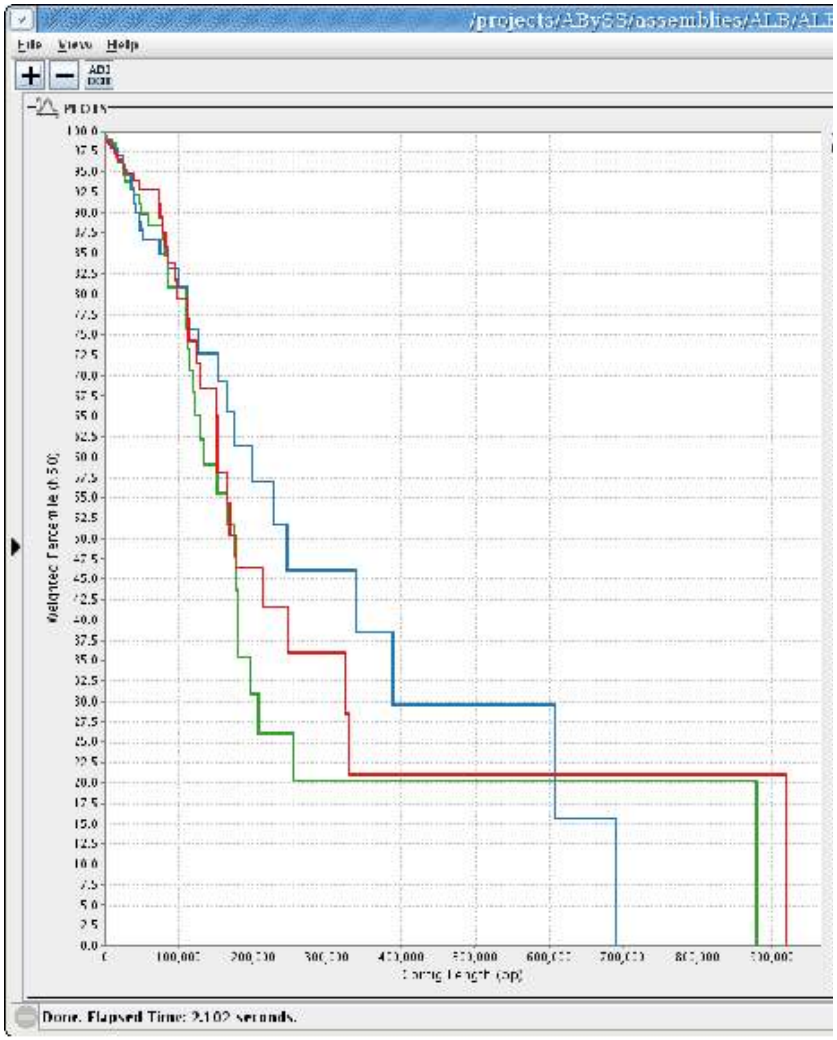
ABySS-Explorer GUI



Statistics Display



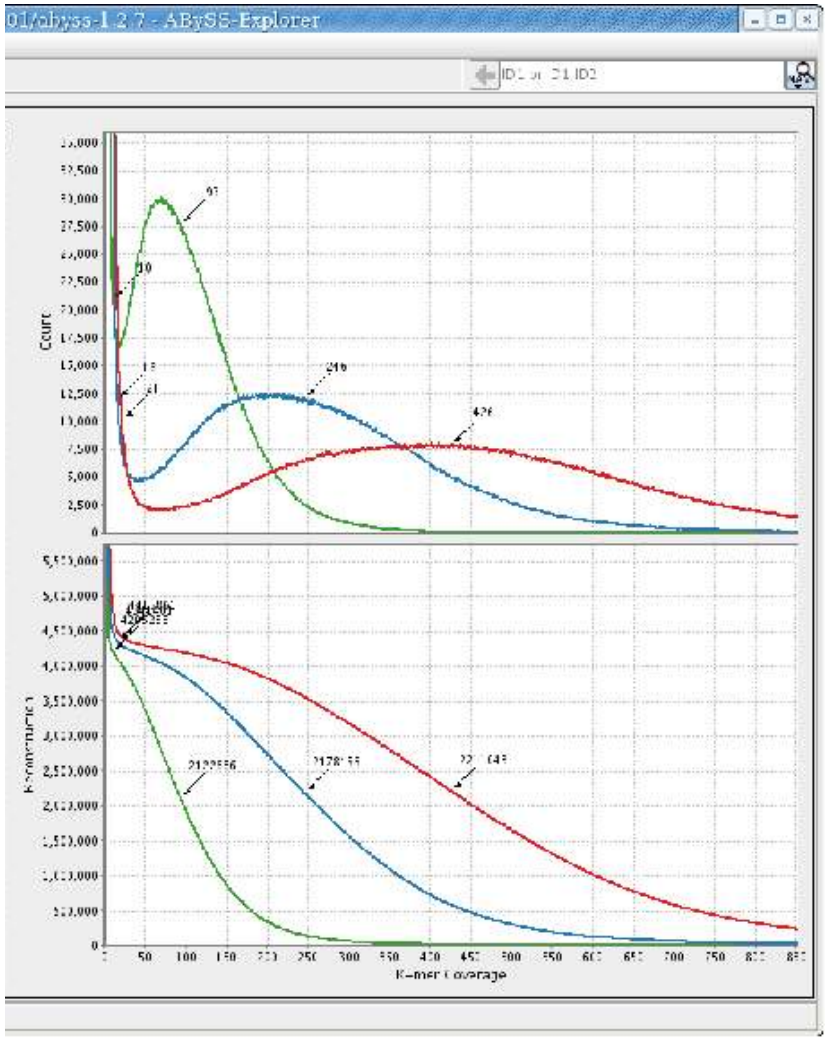
Nxx plot and N50

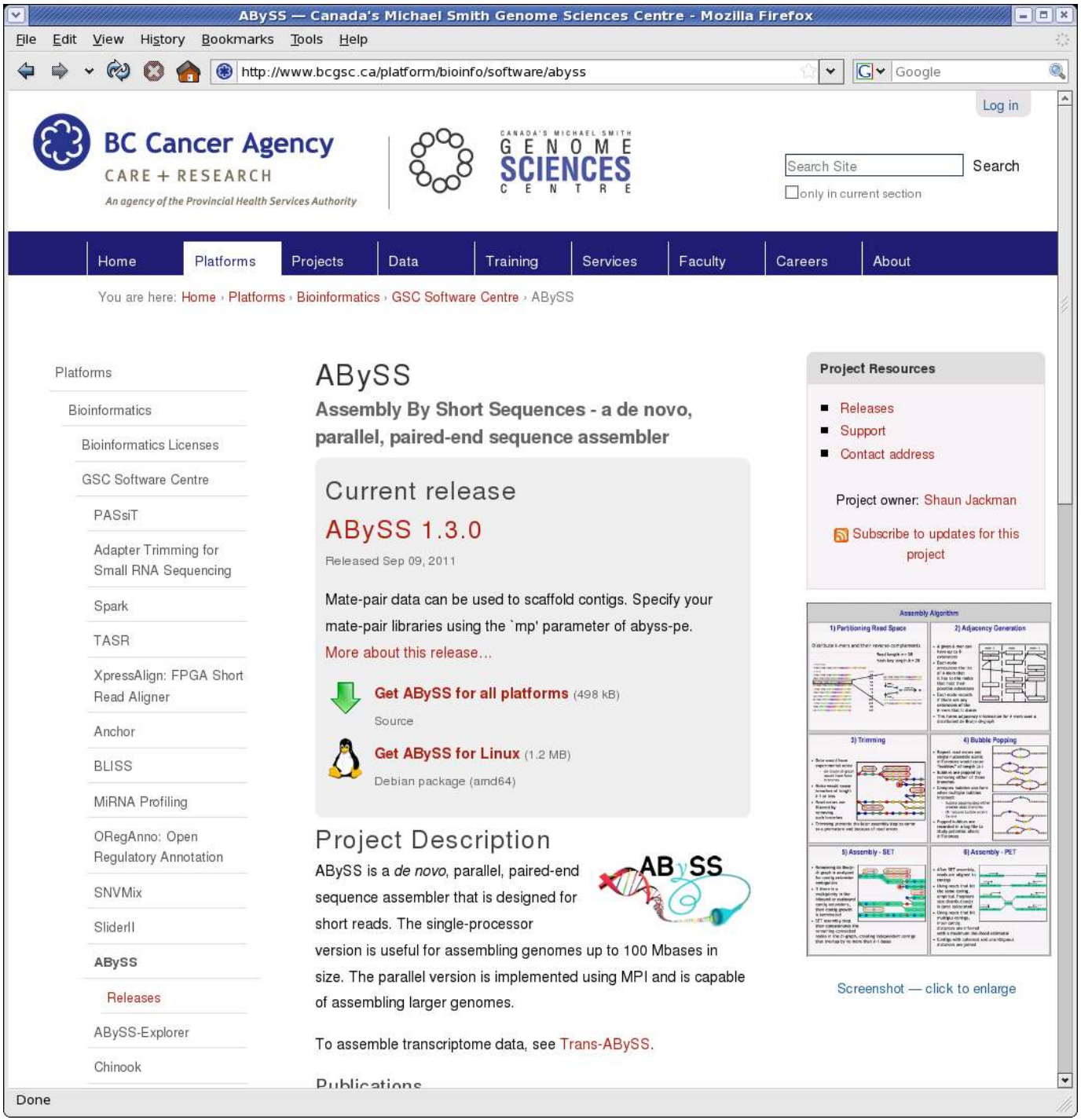


- The N50 is the weighted median of contig sizes
- The N50 summarizes a single point on the Nxx plot
- Better assemblies are further to the right

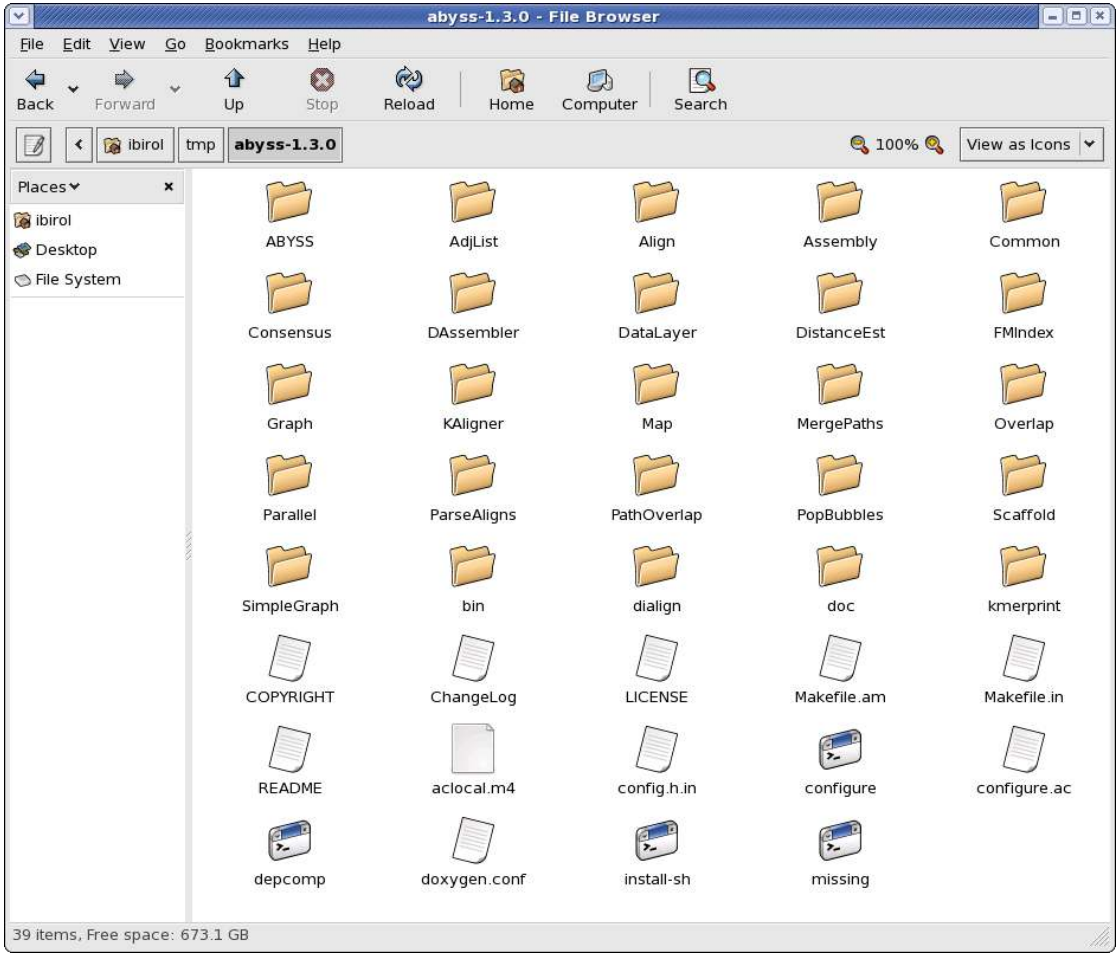
k-mer Coverage Histogram

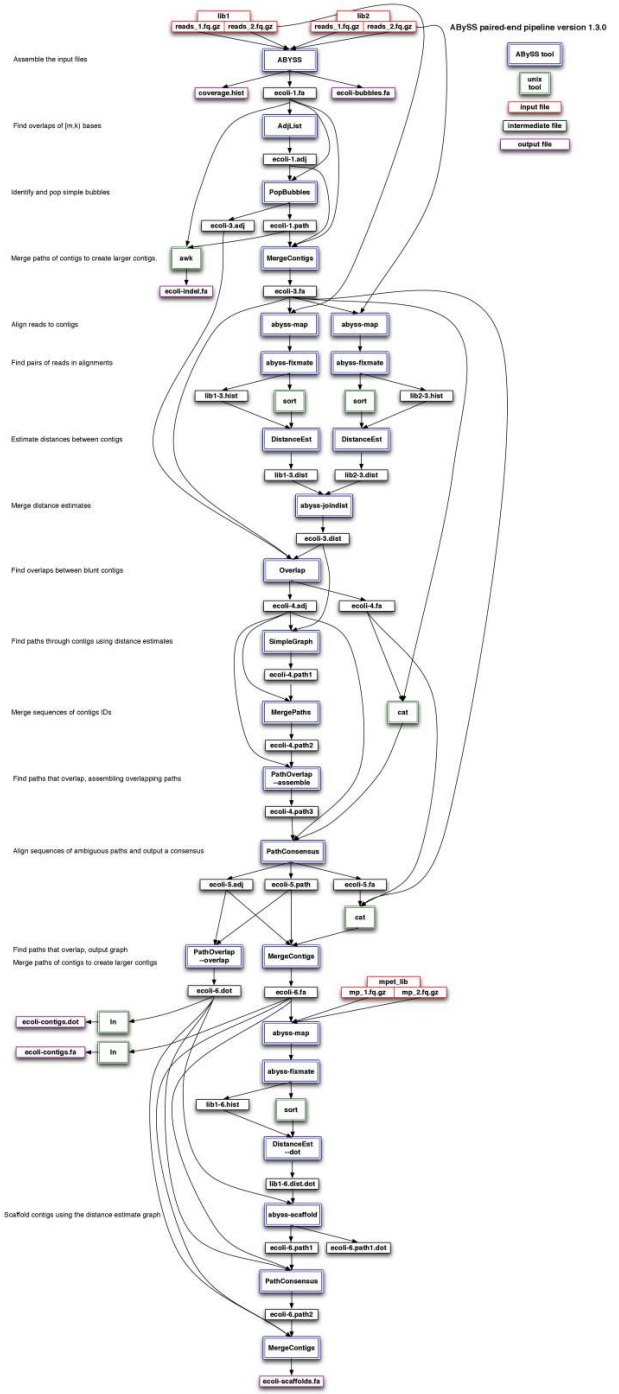
- Counts the number of occurrences of each k-mer
- Useful for
  - estimating the genome size
  - measuring mean coverage
  - library quality control

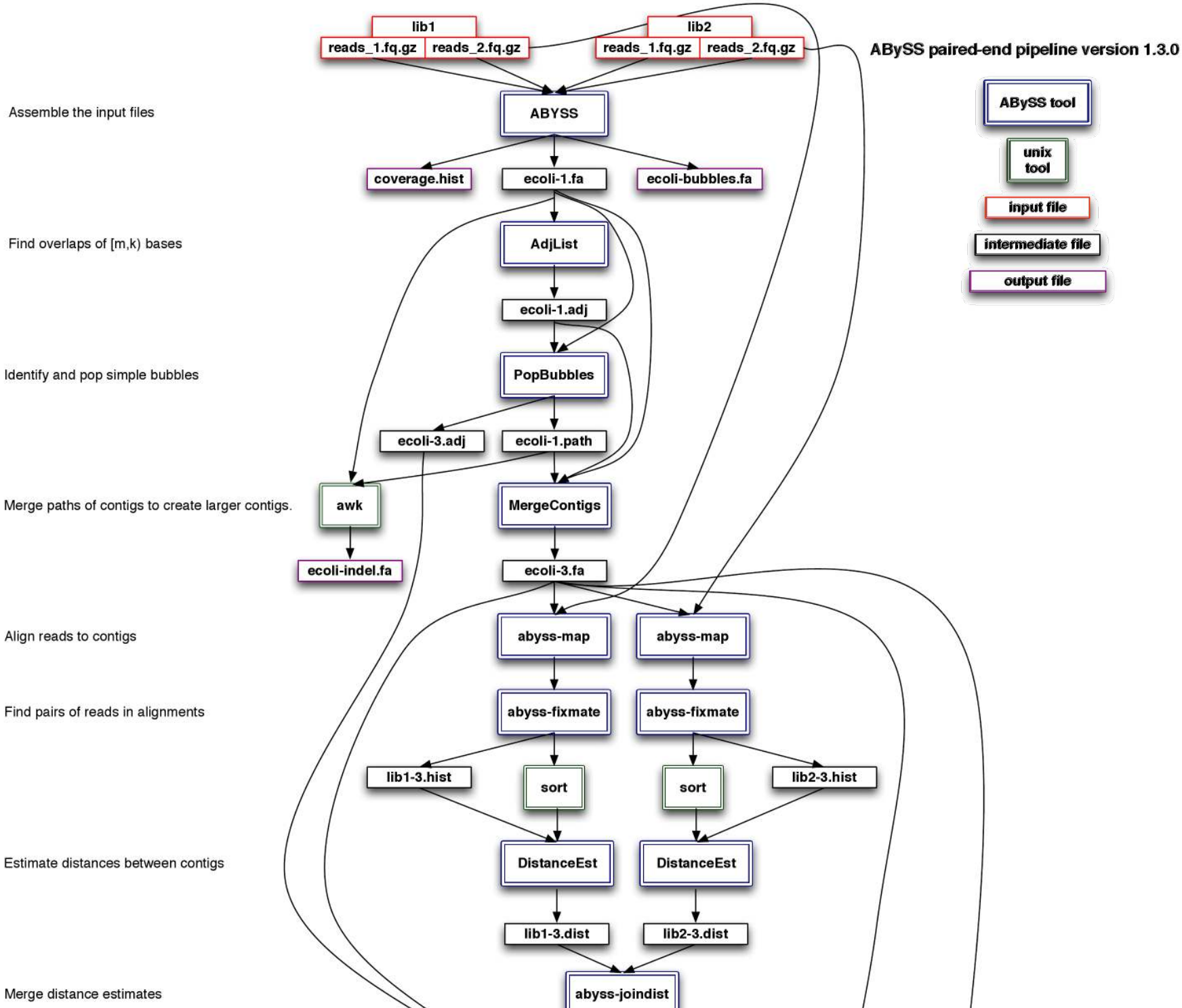



ABySS — Canada's Michael Smith Genome Sciences Centre - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.bcgsc.ca/platform/bioinfo/software/abyss
Google
Log in
BC Cancer Agency
CARE + RESEARCH
An agency of the Provincial Health Services Authority
CANADA'S MICHAEL SMITH GENOME SCIENCES CENTRE
Search Site
Search
only in current section
Home
Platforms
Projects
Data
Training
Services
Faculty
Careers
About
You are here: Home › Platforms › Bioinformatics › GSC Software Centre › ABySS
Platforms
Bioinformatics
Bioinformatics Licenses
GSC Software Centre
PASsiT
Adapter Trimming for Small RNA Sequencing
Spark
TASR
XpressAlign: FPGA Short Read Aligner
Anchor
BLISS
MiRNA Profiling
ORegAnno: Open Regulatory Annotation
SNVMix
SliderII
ABySS
Releases
ABySS-Explorer
Chinook
ABySS
Assembly By Short Sequences - a de novo, parallel, paired-end sequence assembler
Current release
ABySS 1.3.0
Released Sep 09, 2011
Mate-pair data can be used to scaffold contigs. Specify your mate-pair libraries using the `mp' parameter of abyss-pe.
More about this release…
Get ABySS for all platforms (498 kB)
Source
Get ABySS for Linux (1.2 MB)
Debian package (amd64)
Project Description
ABySS is a de novo, parallel, paired-end sequence assembler that is designed for short reads. The single-processor version is useful for assembling genomes up to 100 Mbases in size. The parallel version is implemented using MPI and is capable of assembling larger genomes.
To assemble transcriptome data, see Trans-ABySS.
Publications
Project Resources
Releases
Support
Contact address
Project owner: Shaun Jackman
Subscribe to updates for this project
Assembly Algorithm
1) Partitioning Read Space
2) Adjacency Generation
3) Trimming
4) Bubble Popping
5) Assembly - SET
6) Assembly - PET
Screenshot — click to enlarge
Done

ABySS


abyss-1.3.0 - File Browser
File Edit View Go Bookmarks Help
Back Forward Up Stop Reload Home Computer Search
ibirol tmp abyss-1.3.0
100% View as Icons
Places
ibirol
Desktop
File System
ABYSS
AdjList
Align
Assembly
Common
Consensus
DAssembler
DataLayer
DistanceEst
FMIndex
Graph
KAligner
Map
MergePaths
Overlap
Parallel
ParseAligns
PathOverlap
PopBubbles
Scaffold
SimpleGraph
bin
dialign
doc
kmerprint
COPYRIGHT
ChangeLog
LICENSE
Makefile.am
Makefile.in
README
aclocal.m4
config.h.in
configure
configure.ac
depcomp
doxygen.conf
install-sh
missing
39 items, Free space: 673.1 GB


ABySS paired-end pipeline version 1.3.0
ABySS tool
unix tool
input file
intermediate file
output file
Assemble the input files
ABYSS
Find overlaps of (m,k) bases
AdjList
Identify and pop simple bubbles
PopBubbles
Merge paths of contigs to create larger contigs
MergeContigs
Align reads to contigs
abyss-map
Find pairs of reads in alignments
abyss-fixmate
sort
Estimate distances between contigs
DistanceEst
Merge distance estimates
abyss-joindist
Find overlaps between blunt contigs
Overlap
Find paths through contigs using distance estimates
SimpleGraph
Merge sequences of contigs IDs
MergePaths
cat
Find paths that overlap, assembling overlapping paths
PathOverlap --assemble
Align sequences of ambiguous paths and output a consensus
PathConsensus
Find paths that overlap, output graph
PathOverlap --overlap
Merge paths of contigs to create larger contigs
mpet_lib
ln
abyss-scaffold
Scaffold contigs using the distance estimate graph


ABySS paired-end pipeline version 1.3.0
ABySS tool
unix tool
input file
intermediate file
output file
lib1
reads_1.fq.gz
reads_2.fq.gz
lib2
reads_1.fq.gz
reads_2.fq.gz
Assemble the input files
ABYSS
coverage.hist
ecoli-1.fa
ecoli-bubbles.fa
Find overlaps of [m,k) bases
AdjList
ecoli-1.adj
Identify and pop simple bubbles
PopBubbles
ecoli-3.adj
ecoli-1.path
Merge paths of contigs to create larger contigs.
awk
MergeContigs
ecoli-indel.fa
ecoli-3.fa
Align reads to contigs
abyss-map
abyss-map
Find pairs of reads in alignments
abyss-fixmate
abyss-fixmate
lib1-3.hist
sort
sort
lib2-3.hist
Estimate distances between contigs
DistanceEst
DistanceEst
lib1-3.dist
lib2-3.dist
Merge distance estimates
abyss-joindist

Assembly Operations

- SET contig building: de Bruijn
  - *k*-mer overlap information
- SET error removal: adjacency
- PET contig merging: adjacency & linkage
  - PET alignments
- PET/MPET scaffolding: adjacency & linkage
  - PET/MPET alignments
- Gap closure and contig extensions: read overlap
  - PET alignments

SET Graph Operations

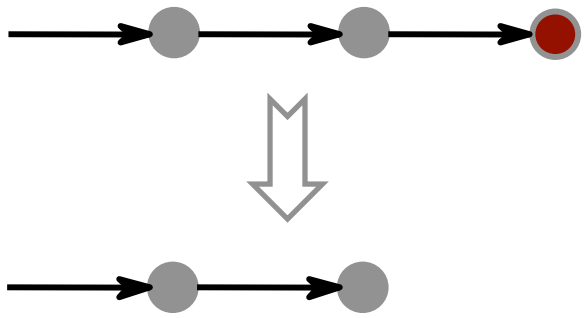
1. Erode low-coverage tips

-e, --erode=COVERAGE

erode bases at the ends of blunt contigs with coverage less than this threshold

-E, --erode-strand=COVERAGE

erode bases at the ends of blunt contigs with coverage less than this threshold on either strand

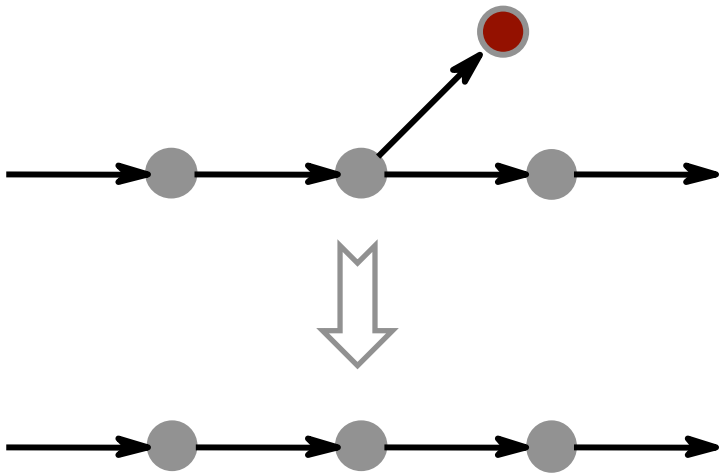


SET Graph Operations

2. Trim tips

-t, --trim-length=TRIM_LENGTH

maximum length of dangling edges to trim

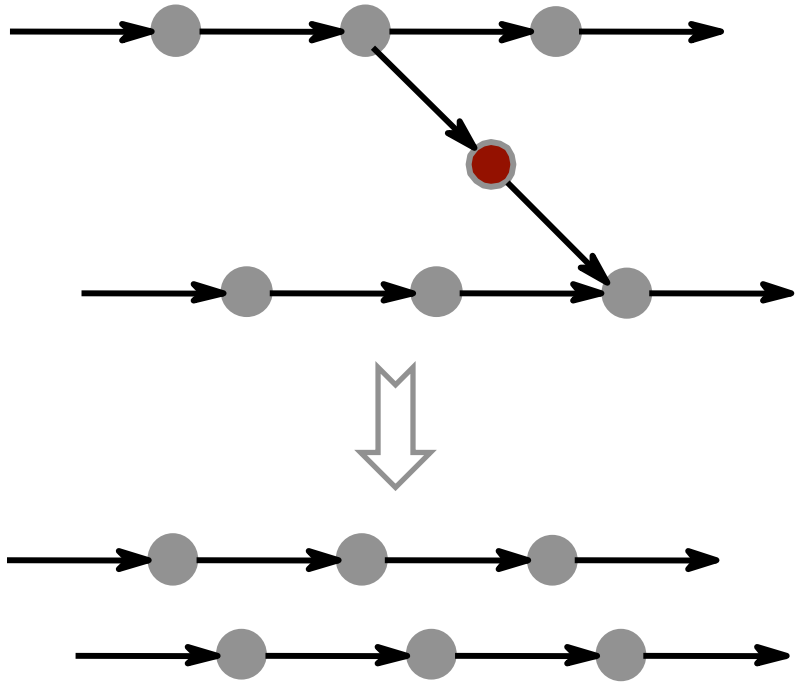


SET Graph Operations

3. Remove low coverage contigs

-c, --coverage=COVERAGE

remove contigs with mean k-mer coverage less than this threshold



SET Graph Operations

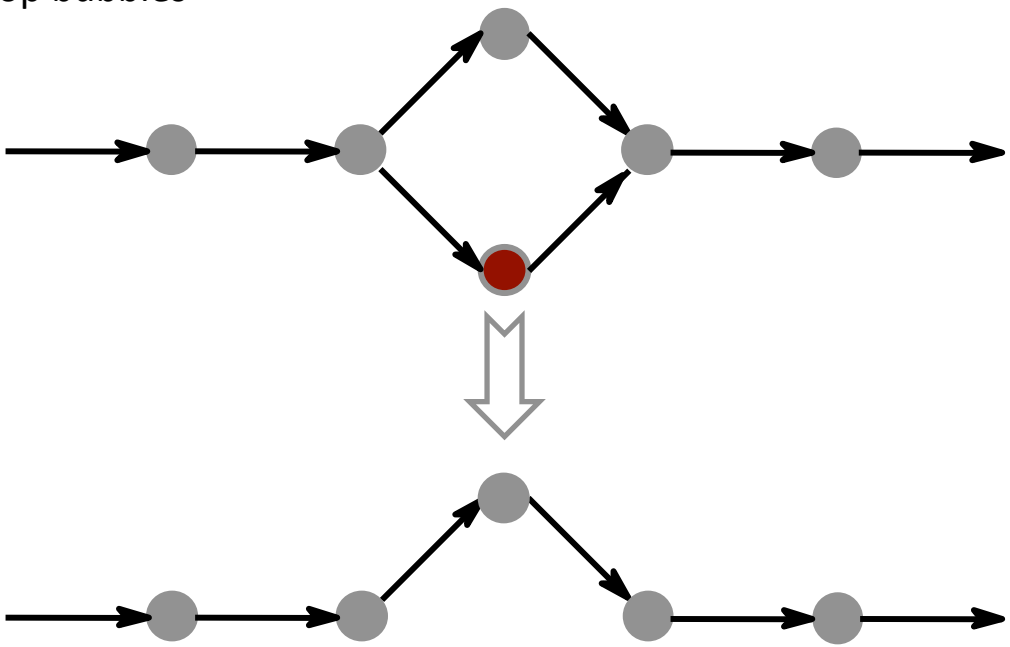
4. Pop bubbles

-b, --bubbles=N

pop bubbles shorter than N bp (default: 3*k)

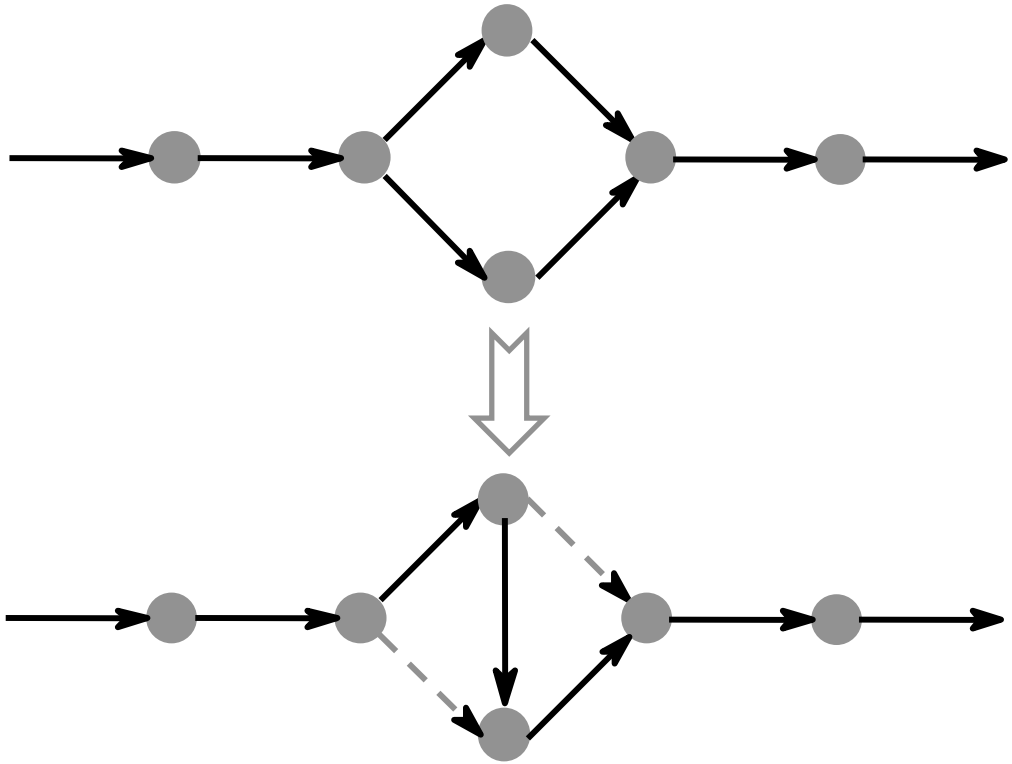
-b0, --no-bubbles

do not pop bubbles



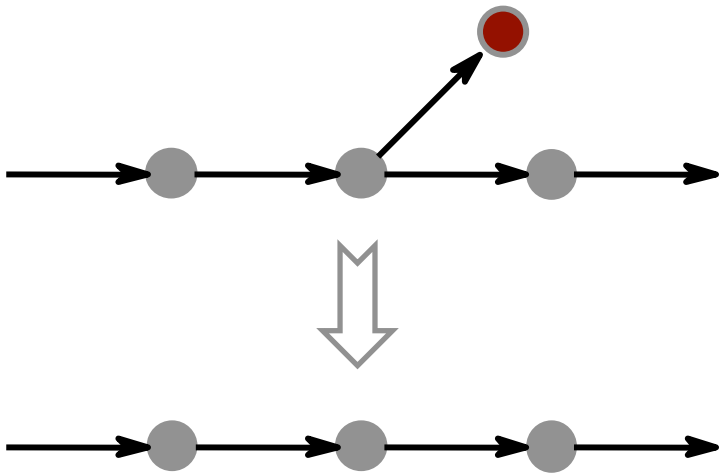
Scaffold Graph Operations

1. Resolve forks



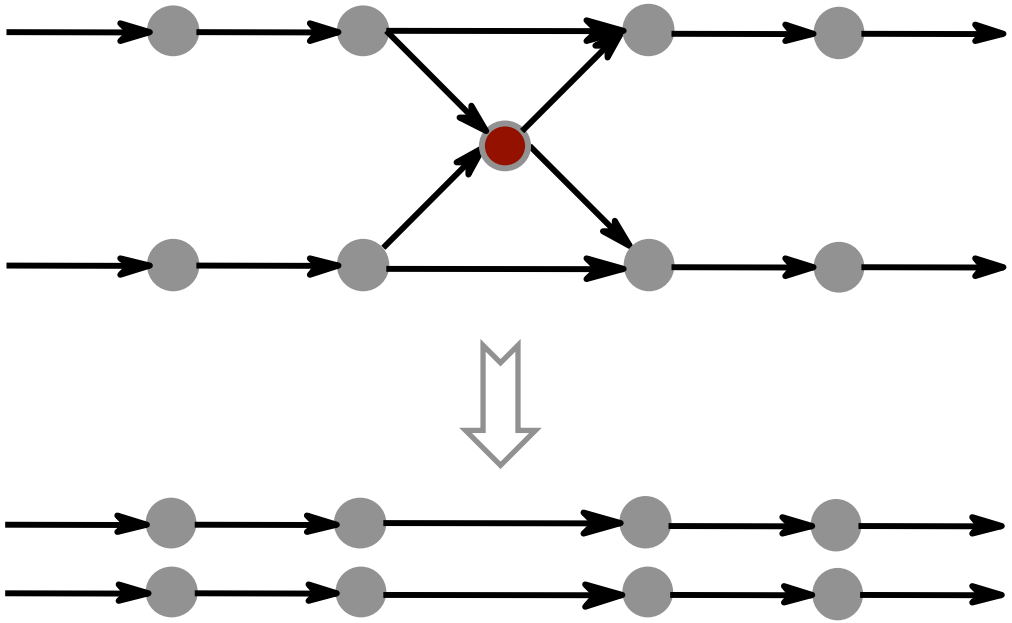
Scaffold Graph Operations

2. Trim tips



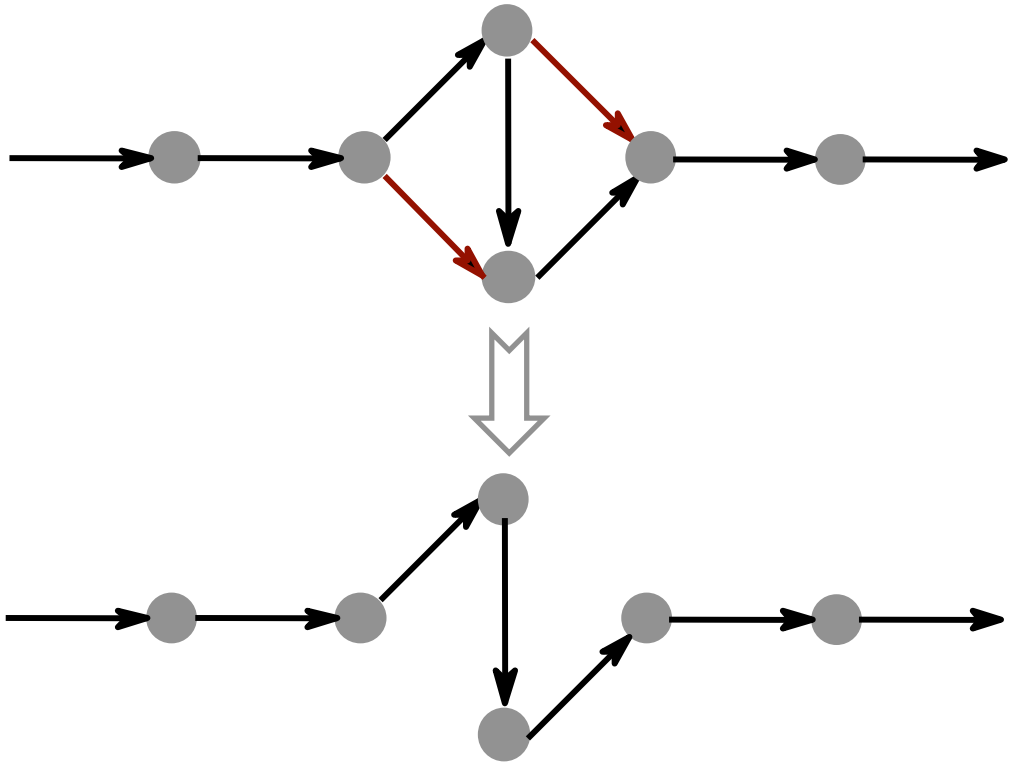
Scaffold Graph Operations

3. Remove repeats



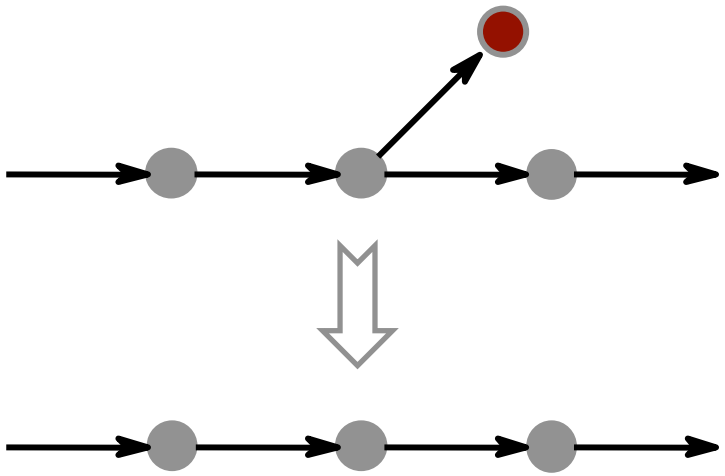
Scaffold Graph Operations

4. Remove transitive edges



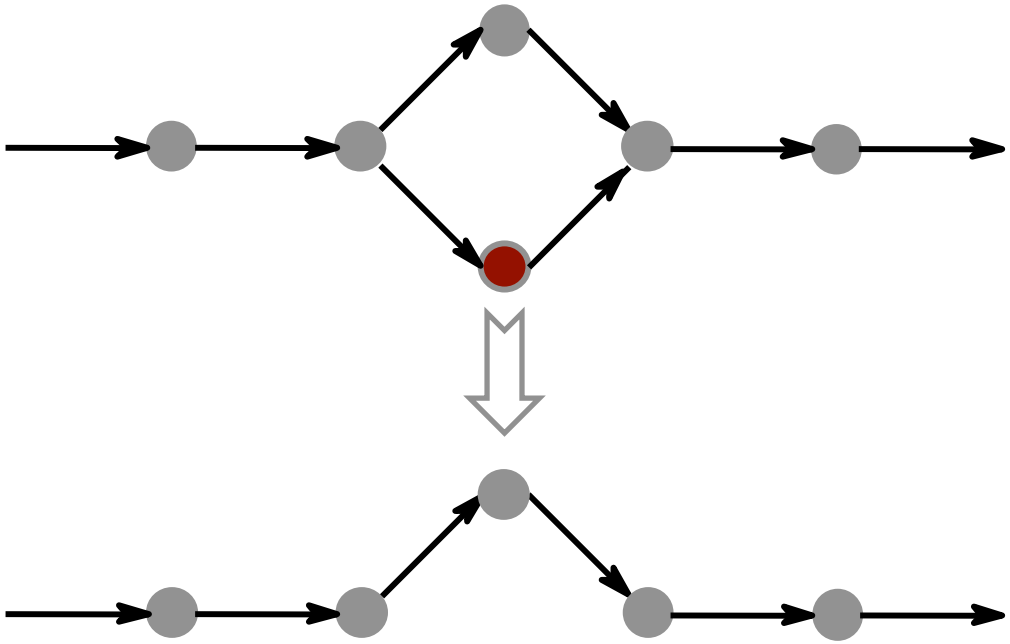
Scaffold Graph Operations

5. Trim tips



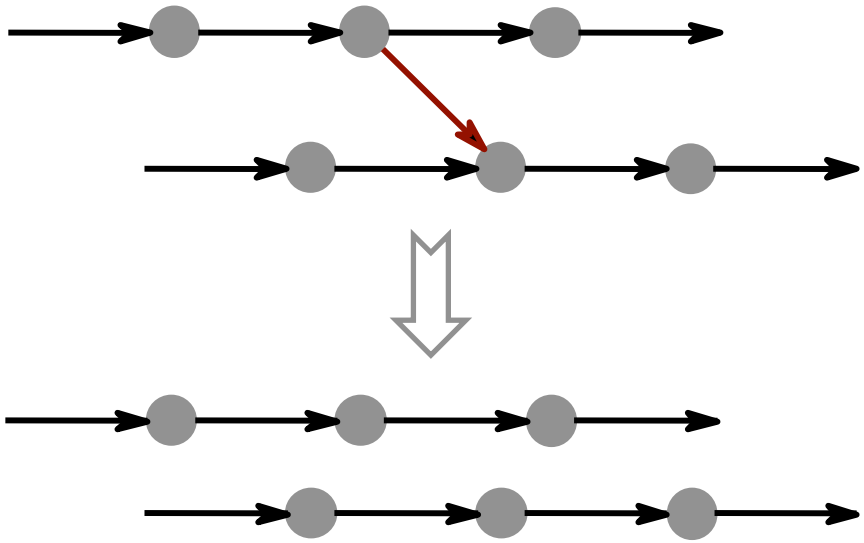
Scaffold Graph Operations

6. Pop bubbles



Scaffold Graph Operations

7. Remove weak edges



Running ABySS

- Assemble the paired-end reads in the file reads.fa

```
> abyss-pe name=ecoli k=32 n=10
  in=reads.fa
```

- Assemble the paired-end reads in the files reads_1.fa and reads_2.fa:

```
> abyss-pe name=ecoli k=32 n=10
  in='reads_1.fa reads_2.fa'
```

Running ABySS in Parallel

- Run ABySS using eight threads

```
> abyss-pe np=8 name=ecoli k=32
  n=10 in='reads_1.fa reads_2.fa'
```

- ABySS uses MPI, the Message Passing Interface. OpenMPI is an open-source implementation of MPI

Running Parallel Jobs on a Cluster

- Run ABySS on a cluster using 8 threads

```
> qsub -pe openmpi 8 -N ecoli
  abyss-pe np=8 name=ecoli k=32 n=10
  in='reads_1.fa reads_2.fa'
```

- abyss-pe uses the environment variables *JOB_NAME* and *NSLOTS* passed to it by SGE as the default values for *name* and *np*

Running for Multiple *k* Values

- Assemble every 8th *k* from 32 to 96

```
> qsub -pe openmpi 8 -N ecoli
  -t 32-96:8 abyss-pe k=32 n=10
  in='reads_1.fa reads_2.fa'
```

- abyss-pe uses the environment variable *SGE_TASK_ID* passed to it by SGE as the default value for *k*

Assembling Multiple Libraries

```
> abyss-pe name=ecoli
  k=32 n=10
  lib='pe200 pe500'
  pe200='pe200_1.fa pe200_2.fa'
  pe500='pe500_1.fa pe500_2.fa'
```

Assembling a Mix of PET and SET

```
> abyss-pe name=ecoli
  k=32 n=10
  lib='pe200 pe500'
  pe200='pe200_1.fa pe200_2.fa'
  pe500='pe500_1.fa pe500_2.fa'
  se='long.fa'
```

Parameters of ABySS

- **name**: name of the assembly
- **lib**: name of the libraries (one or more)
- **se**: paths of the single-end read files
- **${lib}**: paths of the read files for that library
- Example

```
> abyss-pe name=ecoli k=32 n=10
  lib='pe200 pe500'
  pe200='pe200_1.fa pe200_2.fa'
  pe500='pe500_1.fa pe500_2.fa'
  se='long.fa'
```

Parameters of ABySS (SET)

- **k**: the size of a k-mer
- **q**: quality trimming removes low-quality bases from the ends of reads
- **e** and **c**: coverage-threshold parameters
  - e: erosion removes bases from the ends of contigs
  - c: coverage threshold removes entire contigs
- **p**: the minimum identity for bubble popping

Parameters of ABySS (PET)

- **s**: the minimum size of a seed contig
- **n**: the number of pairs required to join two contigs
- Example

```
> abyss-pe name=ecoli
  k=64 q=3 p=0.9 s=100 n=10
  lib='pe200 pe500'
  pe200='pe200_1.fa pe200_2.fa'
  pe500='pe500_1.fa pe500_2.fa'
  se='long.fa'
```

Optimizing *k*

- Assemble every 8th *k* from 32 to 96
  Nine assemblies: 32 40 48 56 64 72 80 88 96
- Find the peak
- Assemble every 2nd *k* around the peak
  For example, if the peak were at *k*=64...
  Eight assemblies: 56 58 60 62 66 68 70 72
- SGE:

```
> qsub -t 32-96:8 qsub-abyss.sh
> qsub -t 56-72:2 qsub-abyss.sh
```

Output Files of ABySS

- ${name}-**contigs.fa**
  The final contigs in FASTA format
- ${name}-**bubbles.fa**
  The equal-length variant sequences (FASTA)
- ${name}-**indel.fa**
  The different-length variant sequences (FASTA)
- ${name}-**contigs.dot**
  The contig overlap graph in Graphviz format

Intermediate Output Files of ABySS

- **.adj**: contig overlap graph in ABySS adj format
- **.dist**: estimates of the distance between contigs in ABySS dist format
- **.path**: lists of contigs to be merged
- **.hist**: fragment-size histogram of a library
- **coverage.hist**: k-mer coverage histogram

Case Study

Mountain Pine Beetle Genome Assembly

Mountain Pine Beetle Genome

Assembly statistics

	contigs	scaffolds
n	1,128,463	1,103,221
n:500bp	33,591	11,657
n:N50	4,324	82
N50 (bp)	11,220	541,443
Max (bp)	276,135	3,583,207
Reconstruction (Gb)	201.9	200.4

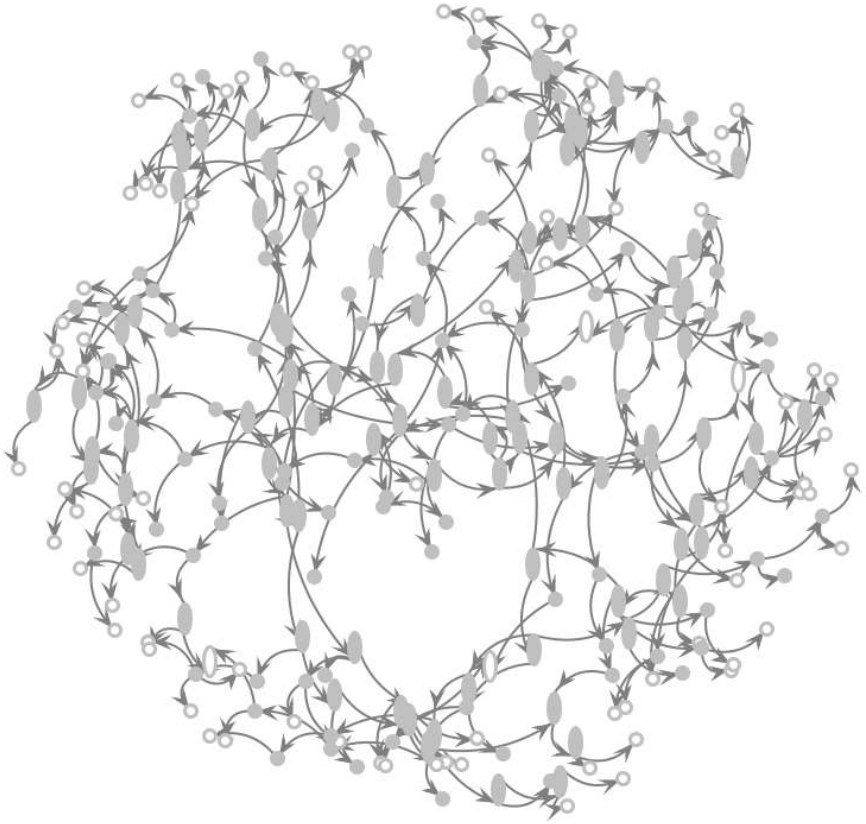
Assembly As a Hairball

- ABySS v1.2.7
  - PET/MPET information disambiguates short contig extensions

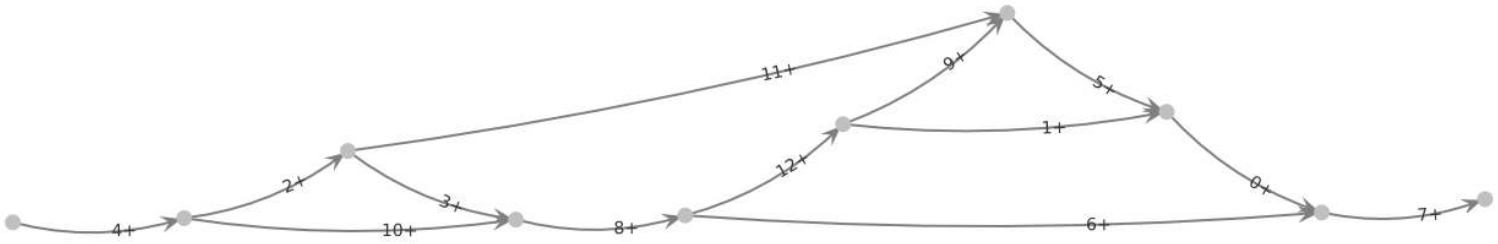
Node connectivity*

in \ out	1	2	3	4	5	6+
1	15822	7354	1882	530	109	1
2	7354	9814	1817	456	72	3
3	1882	1817	1074	238	31	1
4	530	456	238	126	13	1
5	109	72	31	13	10	0
6+	1	3	1	1	0	0

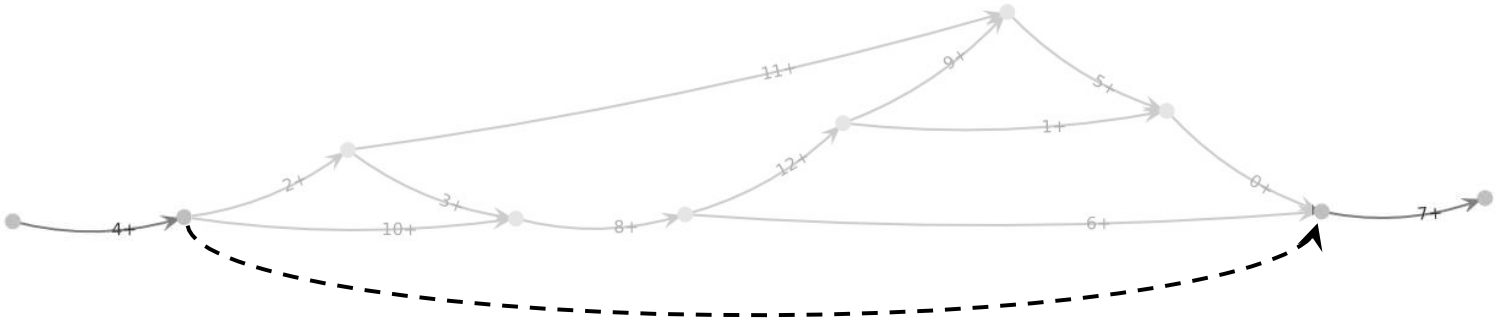
* For contigs ≥ 2 kb

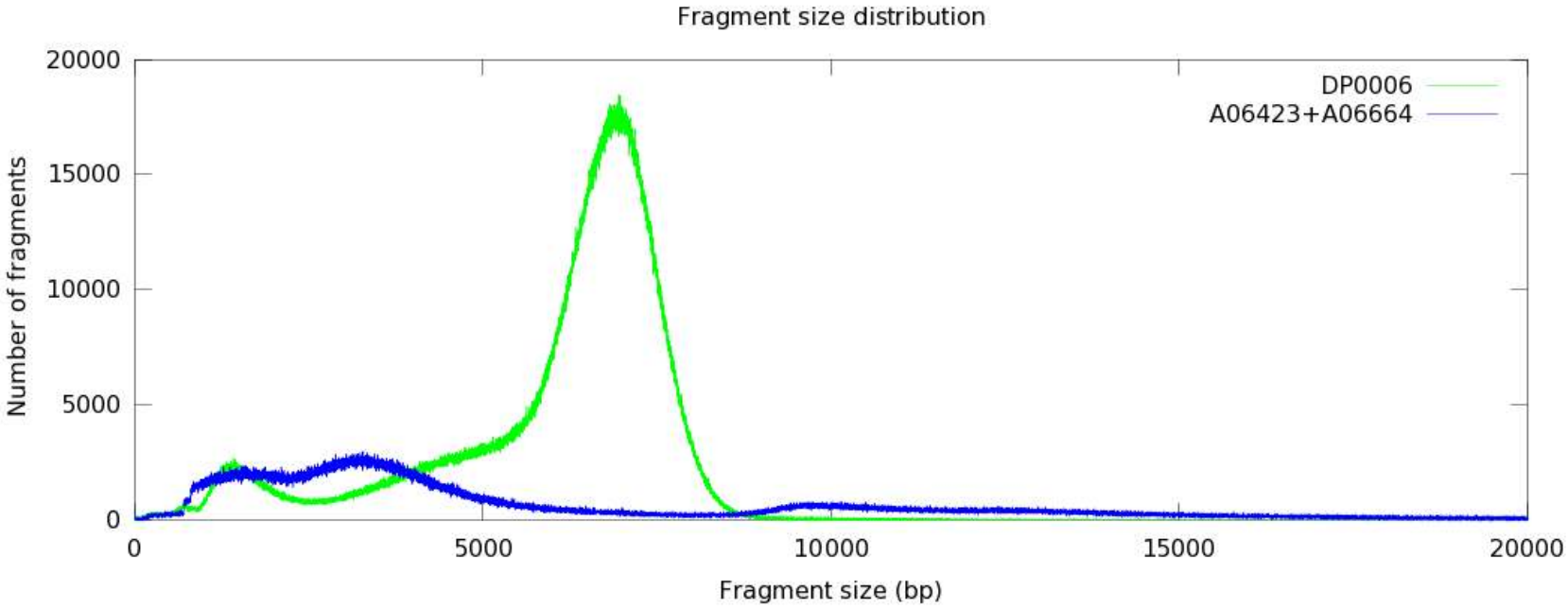


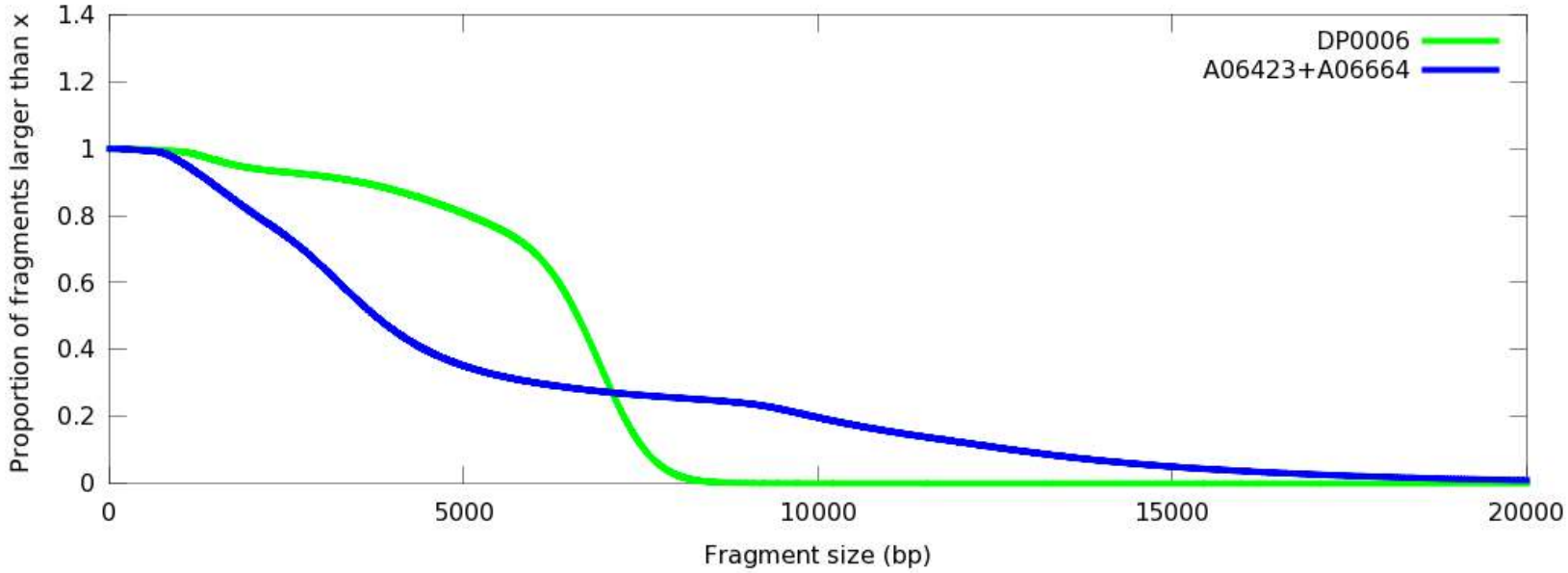
Zoom-in

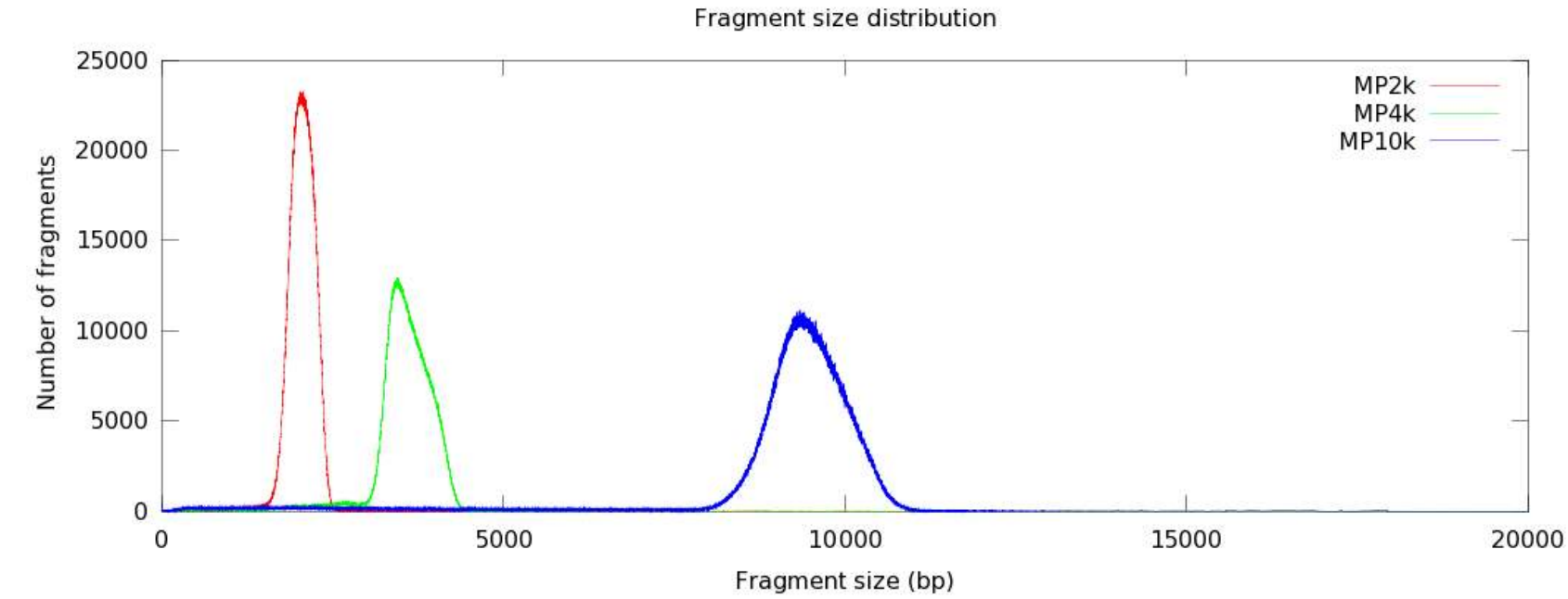


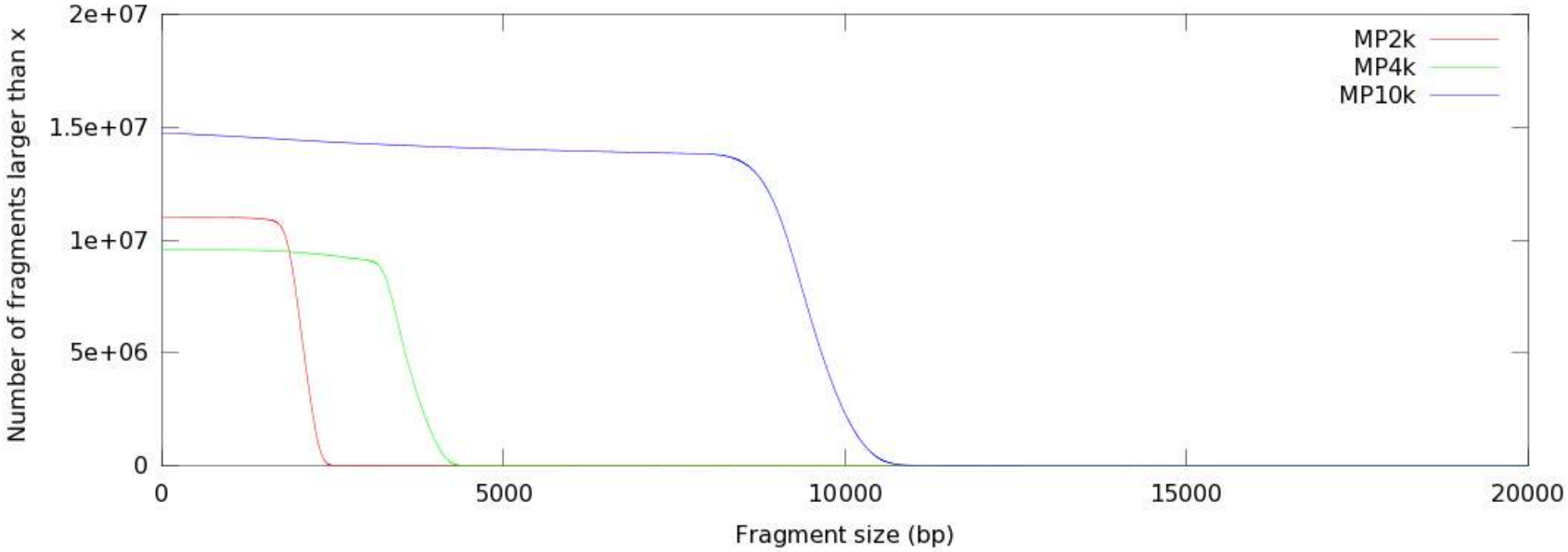
- Contig 4 is (eventually) followed by Contig 7



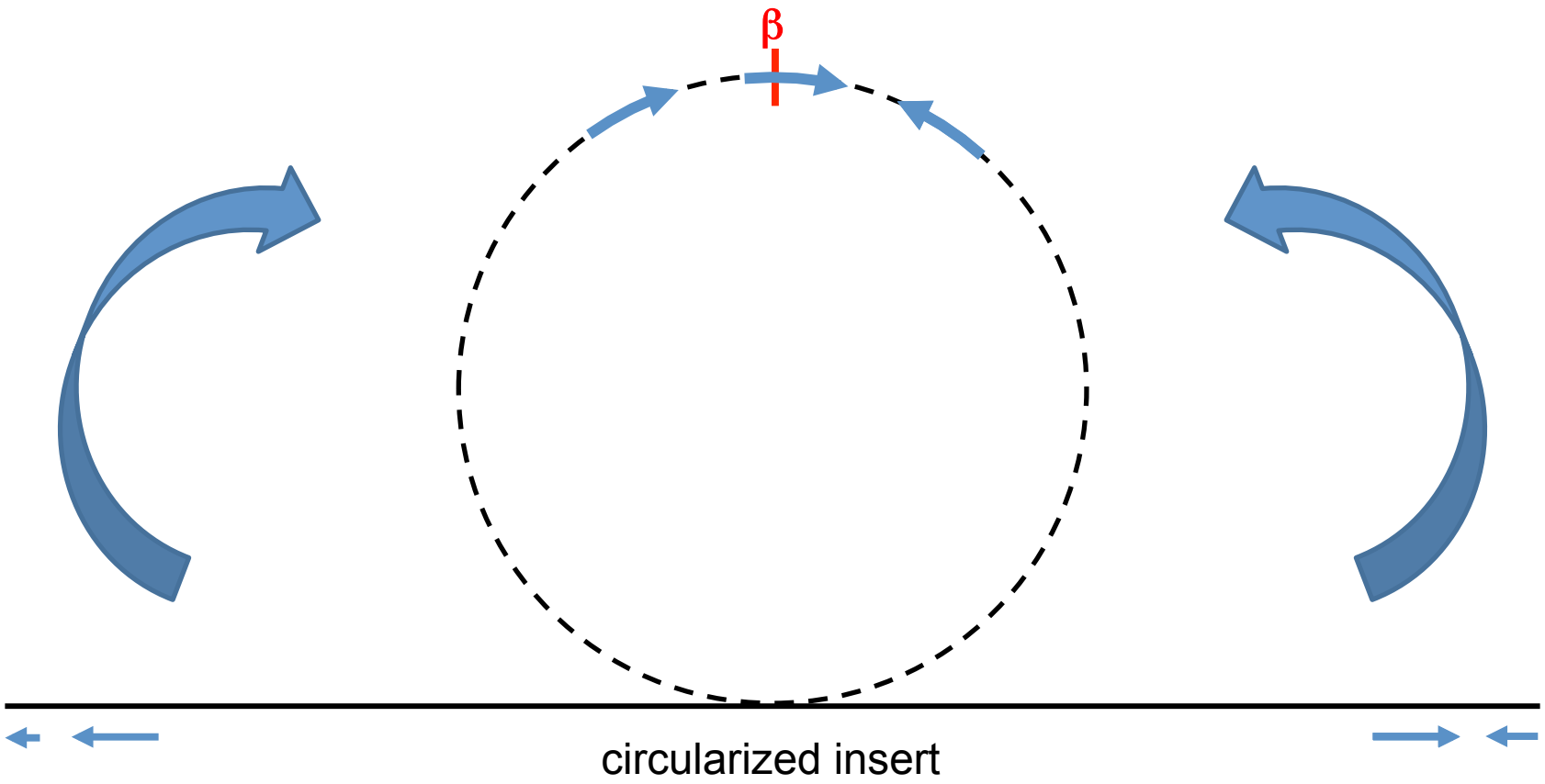

Fragment size distribution
Number of fragments
Fragment size (bp)
DP0006
A06423+A06664
0
5000
10000
15000
20000


Proportion of fragments larger than x
Fragment size (bp)
DP0006
A06423+A06664
0
0.2
0.4
0.6
0.8
1
1.2
1.4
5000
10000
15000
20000


Fragment size distribution
Number of fragments
Fragment size (bp)
MP2k
MP4k
MP10k
0
5000
10000
15000
20000
25000


Number of fragments larger than x
Fragment size (bp)
MP2k
MP4k
MP10k
0
5e+06
1e+07
1.5e+07
2e+07
0
5000
10000
15000
20000

Biotin Read-Through



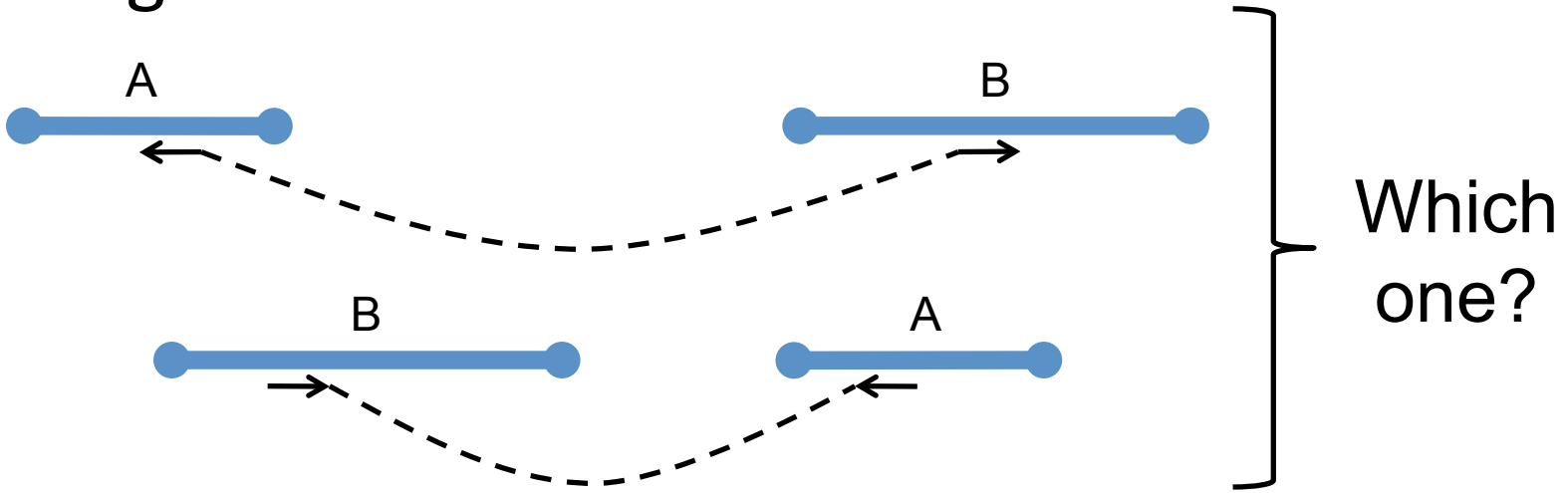
Illumina's

Dr. Strangelove

Or: How I Learned To Stop Worrying And Love The Biotin Read-Through

Triage of MPET Reads

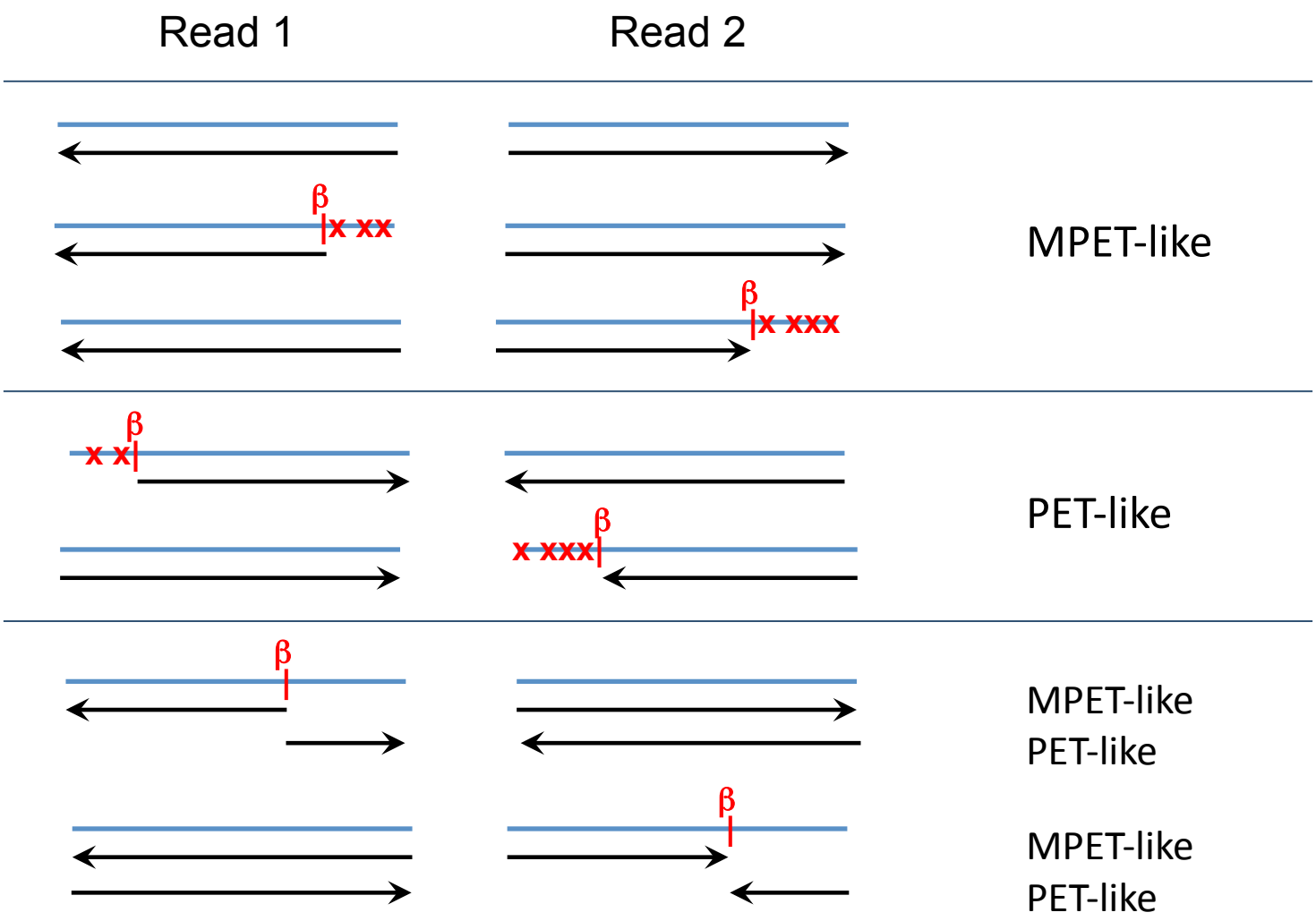
Challenge:



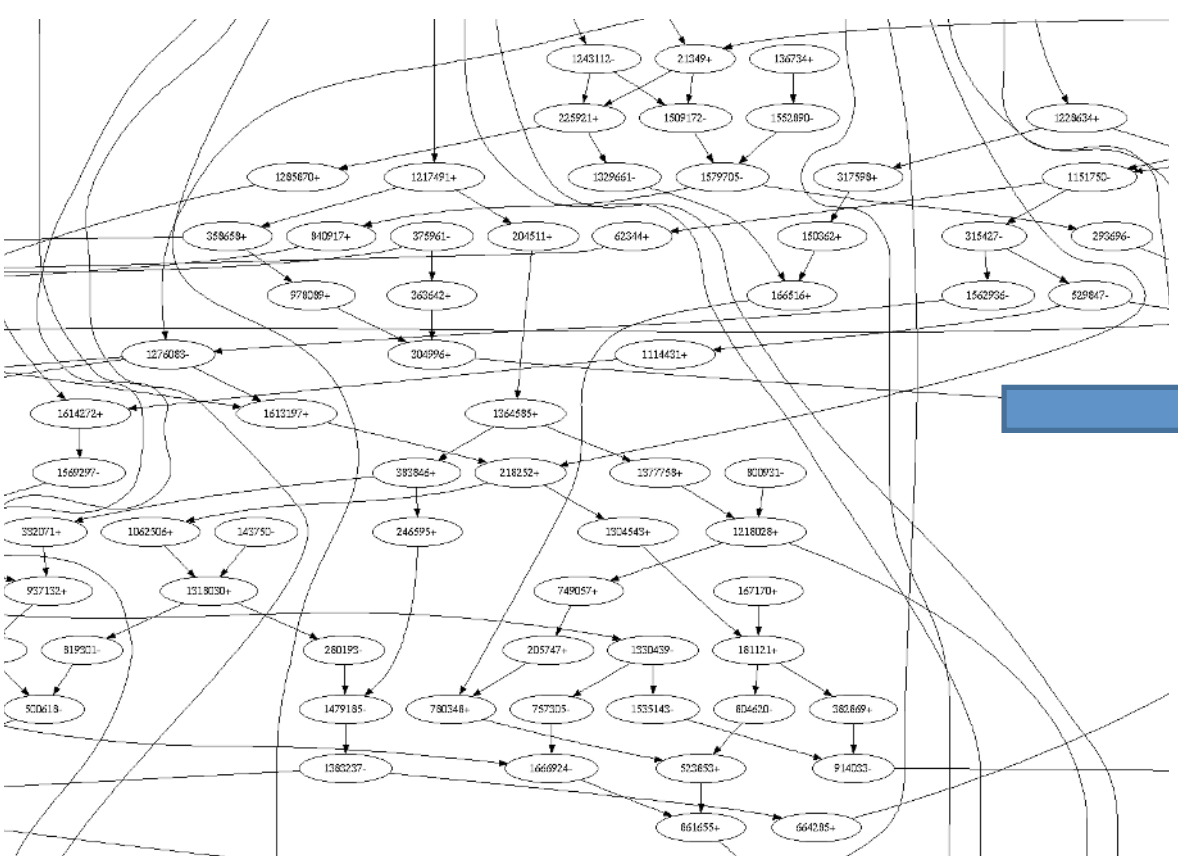
Information:

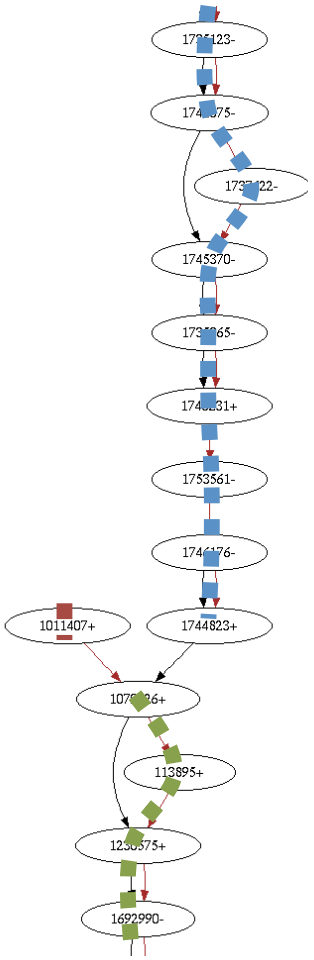
- Distances from contig ends
- Base mismatches on read ends
- Inferred contig orientations

Triage of MPET Reads



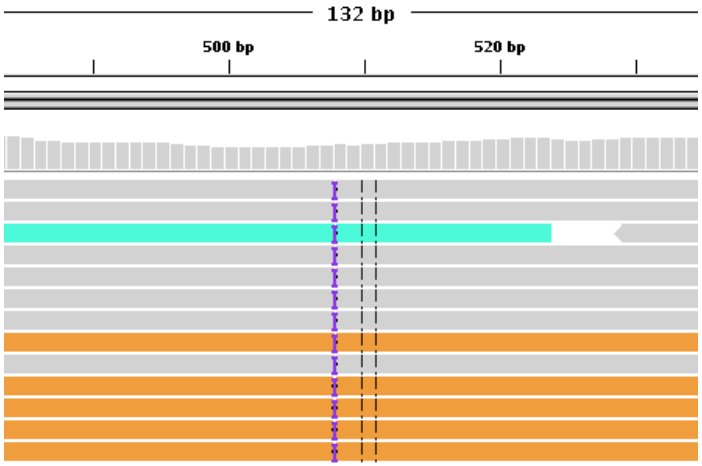
Scaffolding



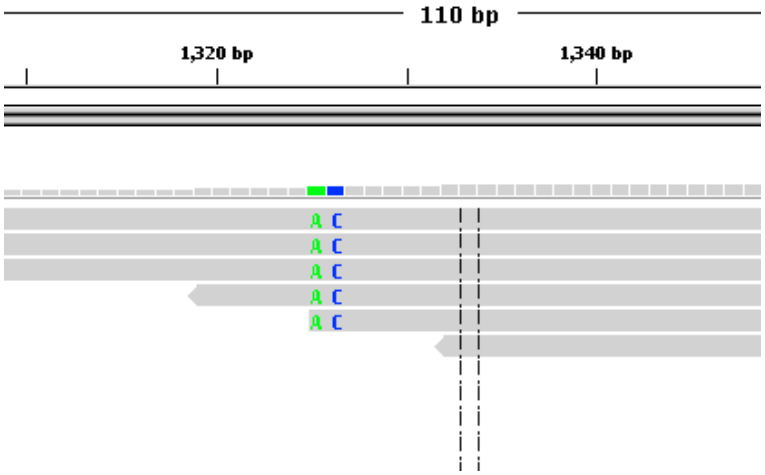


Anchor

- Scrubbing “homozygous” variations



Indel
(2,935)

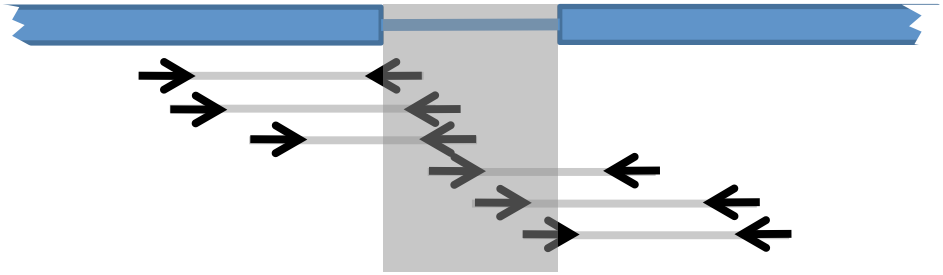


SNPs
(19,715)

Anchor

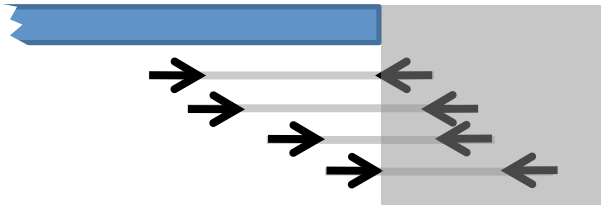
- Local directional assembly
  - scaffold gap filling

(10,499 of 63,986)



  - extension

(20,213 of 53,487)



Quality Assessment

Alignment of 81,047,980 reads

	Before Anchor	After Anchor	Change
Mapped	65,624,456 (80.97%)	66,949,341 (82.60%)	+ 1,324,885
Paired	43,207,118 (53.31%)	44,732,320 (55.19%)	+ 1,525,202
Single-end	9,536,178 (11.77%)	8,846,977 (10.92%)	-689,201

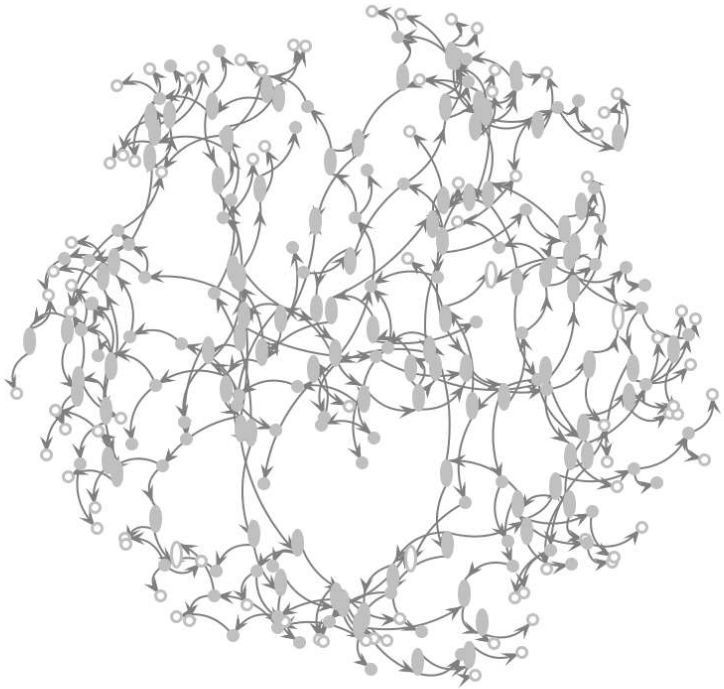
Gene alignments

	2,180 ESTs		248 Conserved Genes	
	Complete	Partial	Complete	Partial
Contigs	968	1169	212	18
Scaffolds	1,481	619	228	5

Final Hairball

- ABySS v1.2.7
  - Read pairs and inferred distances allow for scaffolding

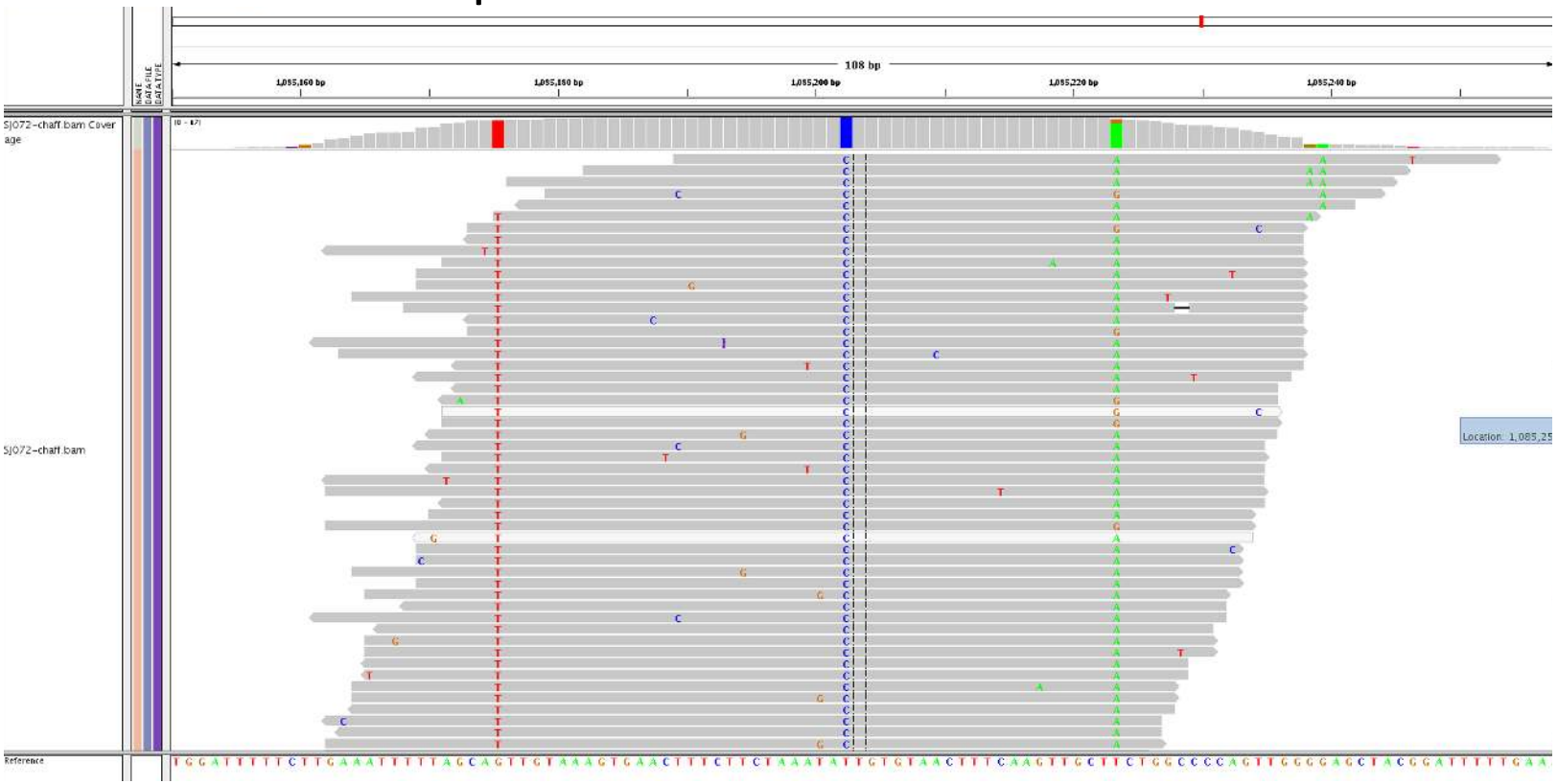
	contigs	scaffolds
n	1,128,463	1,103,221
n:500bp	33,591	11,657
n:N50	4,324	82
N50 (bp)	11,220	541,443
Max (bp)	276,135	3,583,207
Reconstruction (Gb)	201.9	200.4



Date	ABySS Version	Data	n:500	N50	Max	Sum
August 2009	1.0.11	3x GAiix	81,431	1,526	20,755	107.3e6
November 2009	1.0.15	+2x GAiix	104,958	2,333	55,845	195.8e6
February 2010	1.1.1	+4x GAiix	157,081	2,790	136,637	346.3e6
July 2010	1.2.0	+2x GAiix	146,313	3,354	129,008	376.2e6
November 2010	1.2.4	+1x GAiix +1x GAiix (MPET)	100,690	4,474	294,323	268.8e6
May 2011	1.2.7	--	18,660	108,158	1,908,773	201.4e6
July 2011	1.2.7	+ 1x HiSeq +1x HiSeq (MPET)	11,657	541,443	3,583,207	200.4e6
August 2011	1.2.7	--	11,523	561,847	3,746,698	206.5e6

Future Work

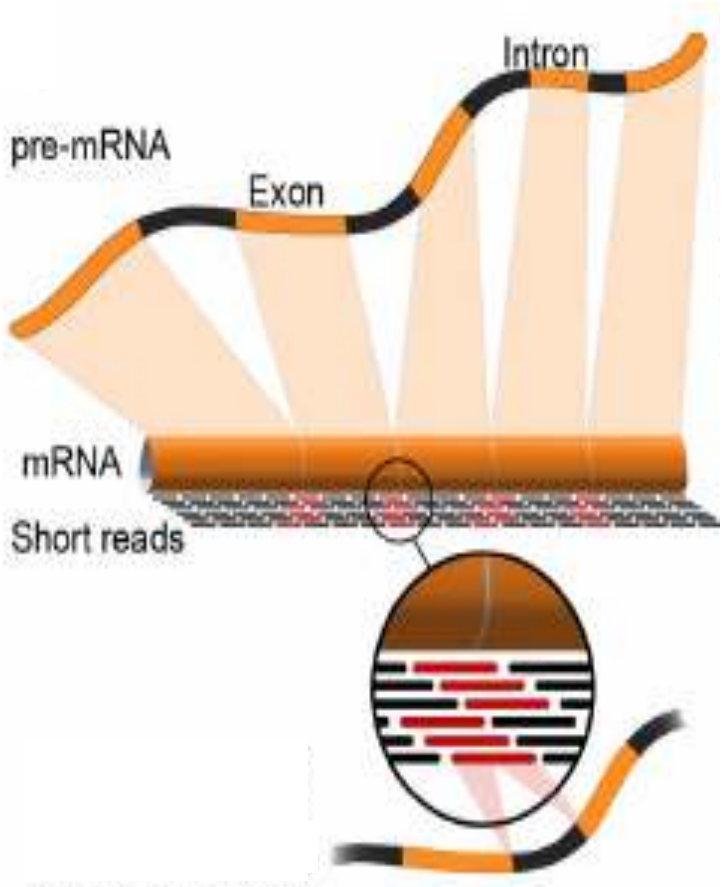
- Clean up the chaff
  - Place short contigs on Anchored scaffolds
  - Annotate repeat elements



Transcriptome Assembly

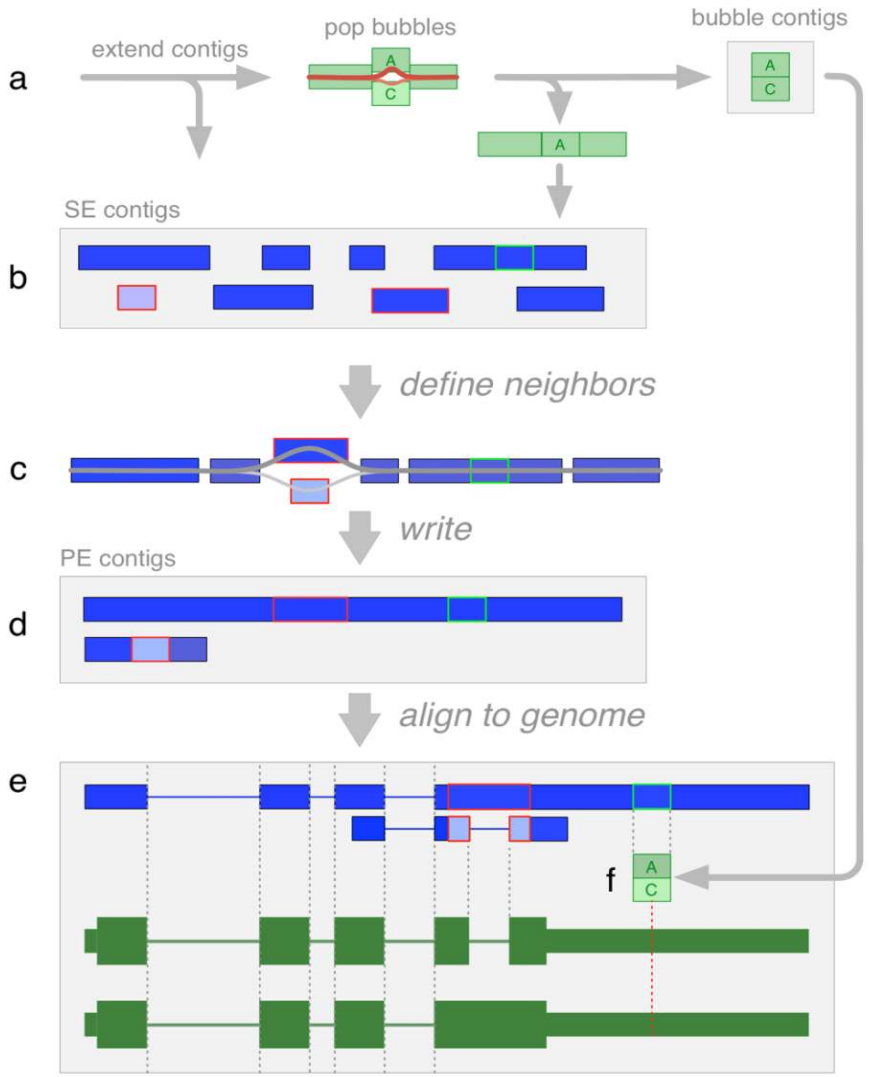
Transcriptome Sequencing

- RNA-seq protocol
- Brings information on how a genome “acts”
  - Expression levels
    - Allelic expression
  - Present isoforms
  - Gene fusions
  - Other transcriptional events
  - Post-transcriptional RNA editing



Rodrigo Goya

Transcriptome Assembly

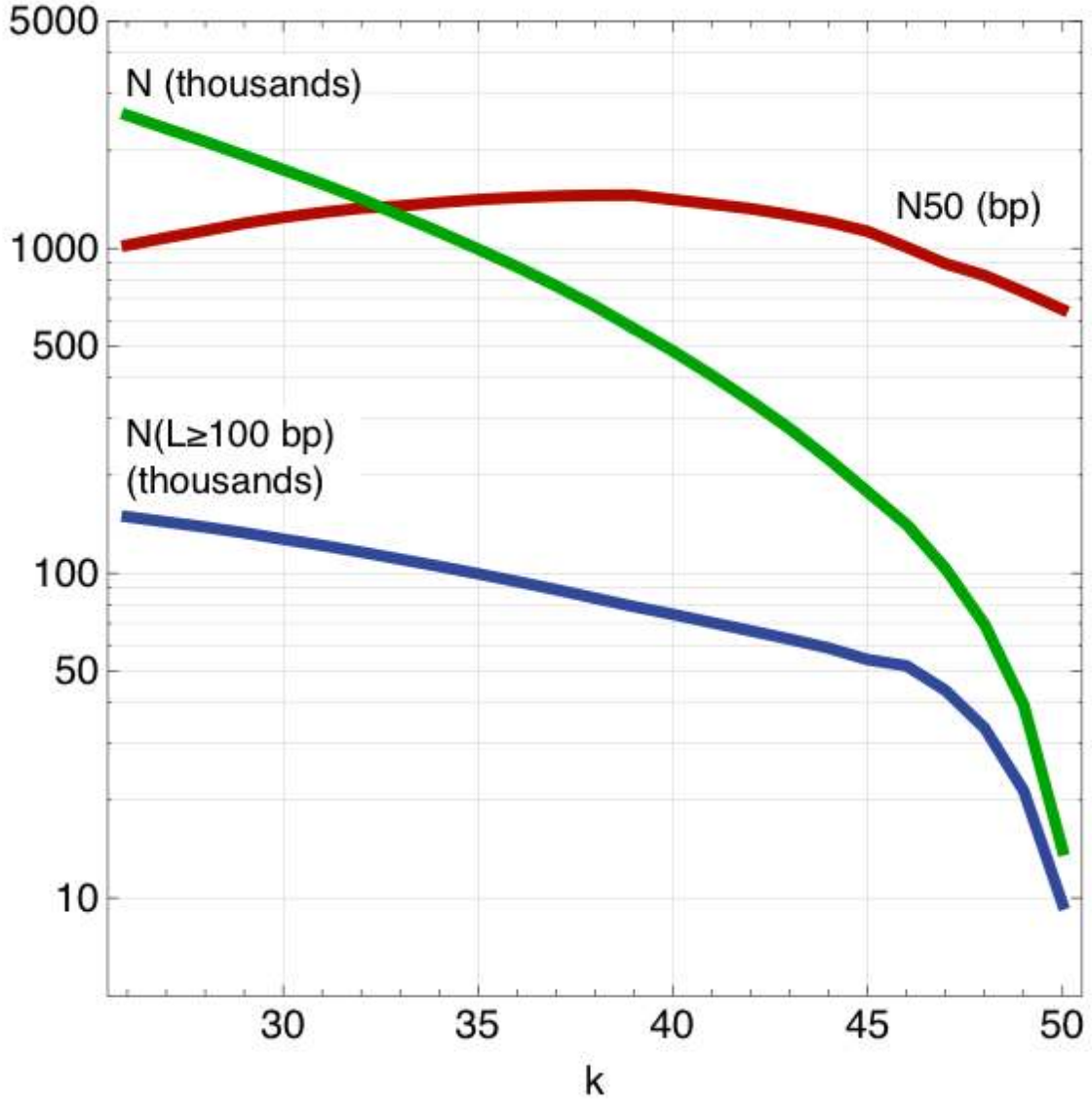


Transcript models

Transcriptome assembly is different from genome assembly

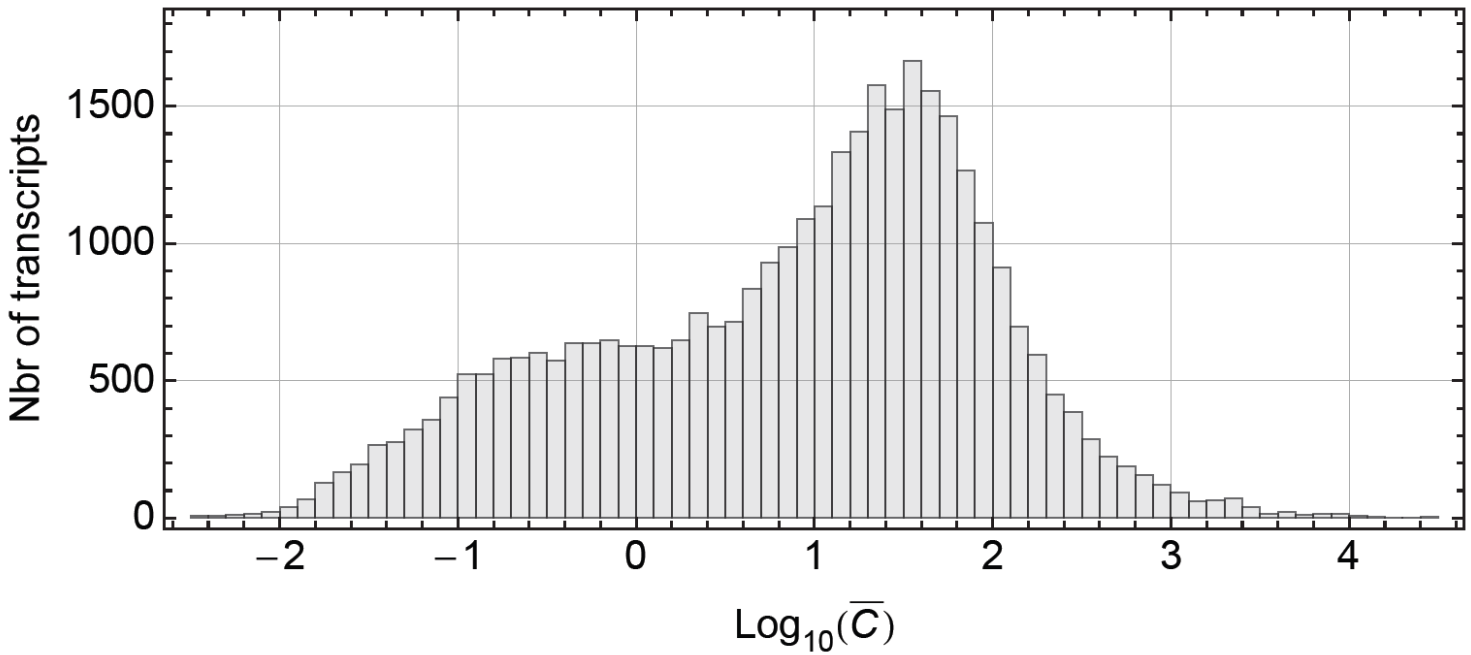
- varying coverage levels
  ⇒ varying expression levels
- split assembly paths
  ⇒ isoforms/splice variants
- small contig sizes
  ⇒ small product sizes

What Overlap to Choose?

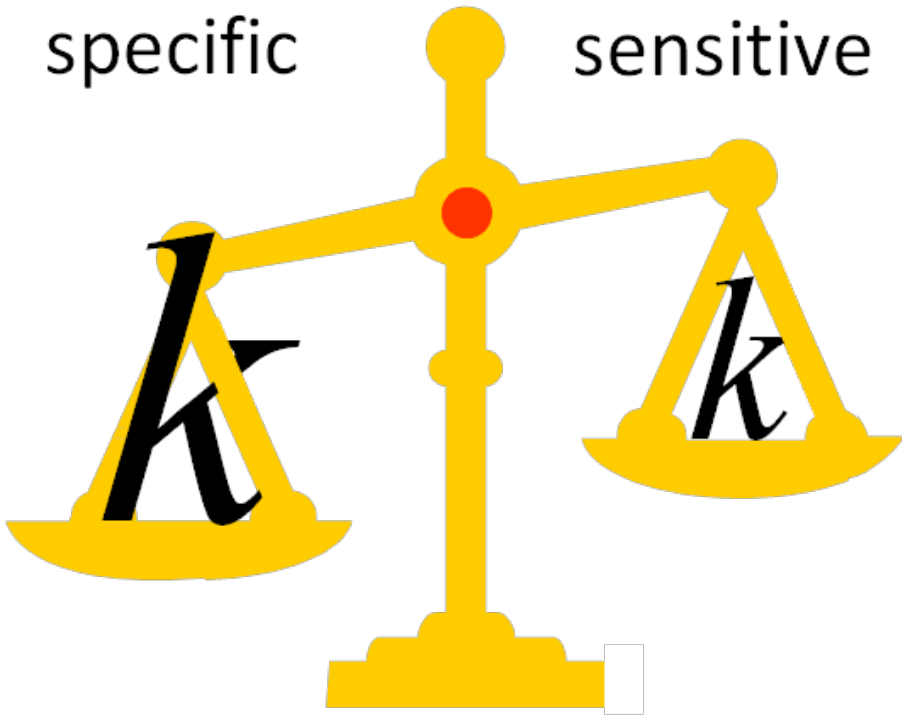


What Overlap to Choose?

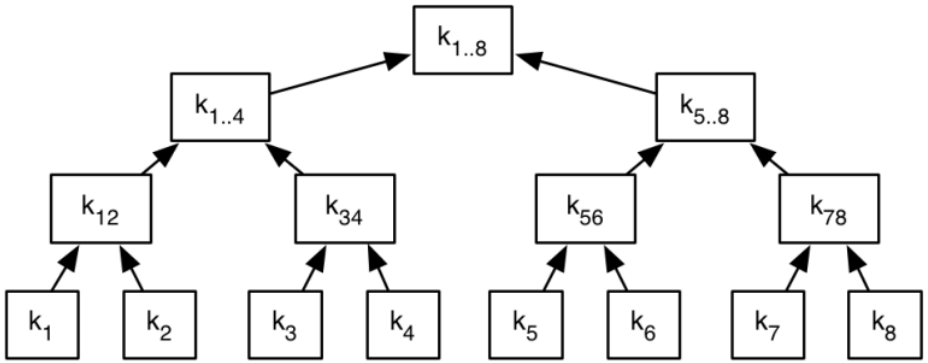
- Selection of parameter *k* depends on read coverage depth
- Expression levels vary over 5 orders of magnitude

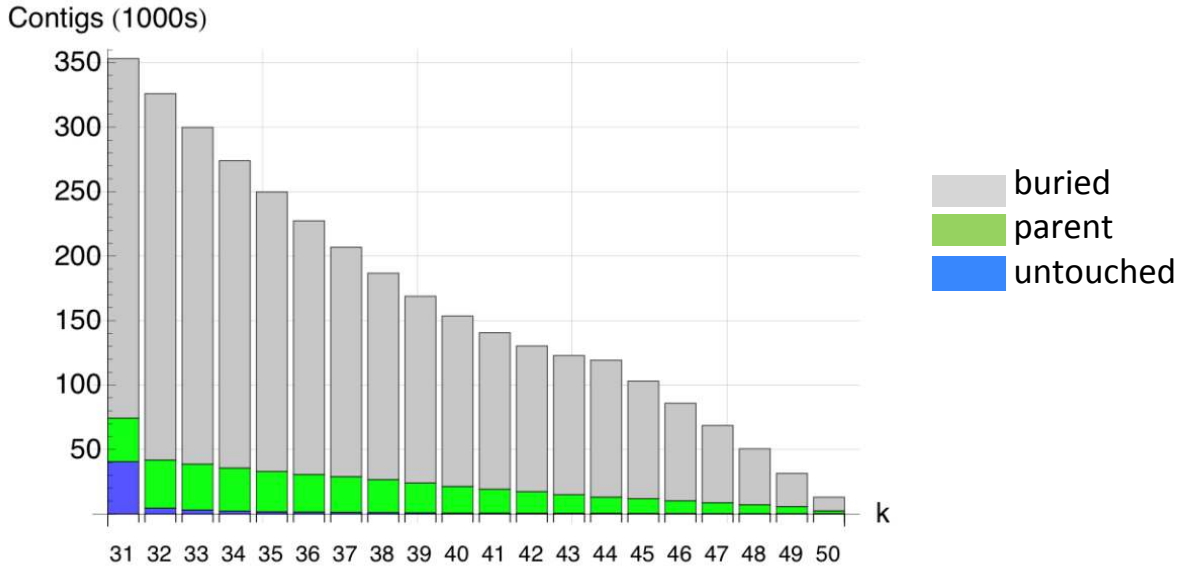


Selection of *k*



Assembly Merging

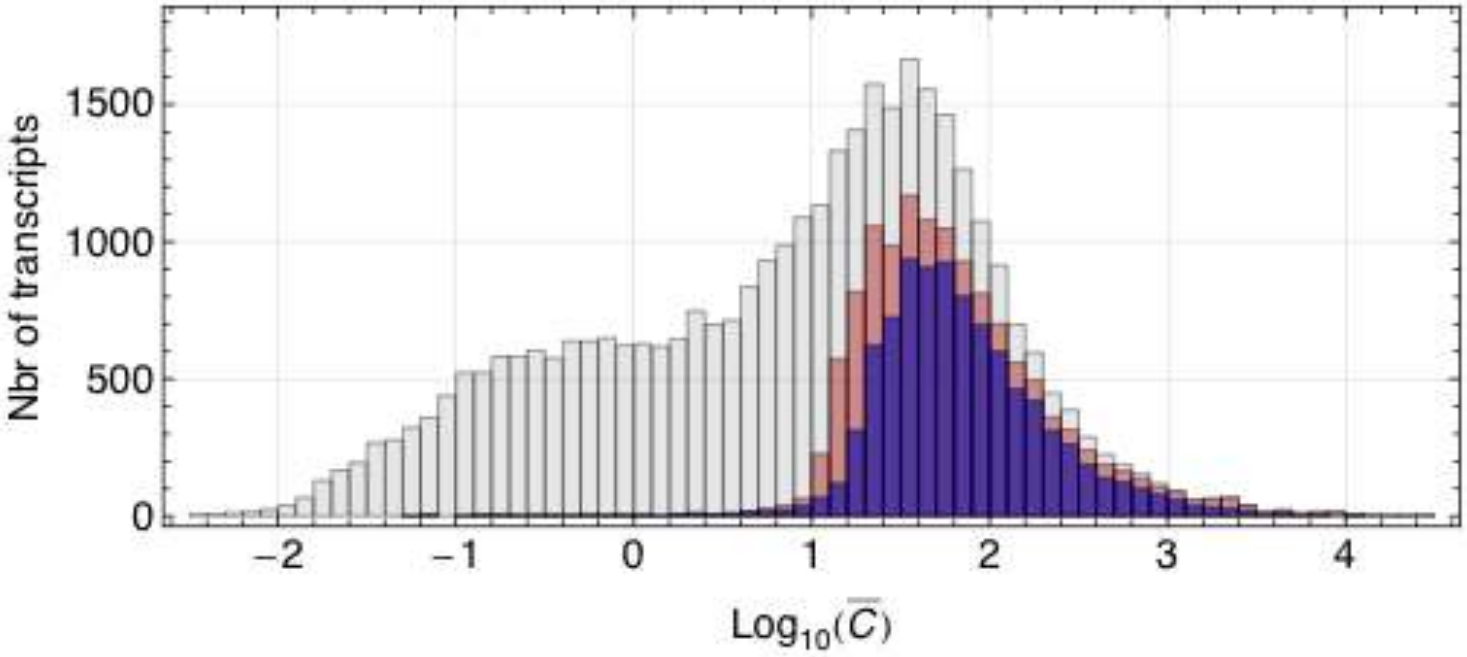




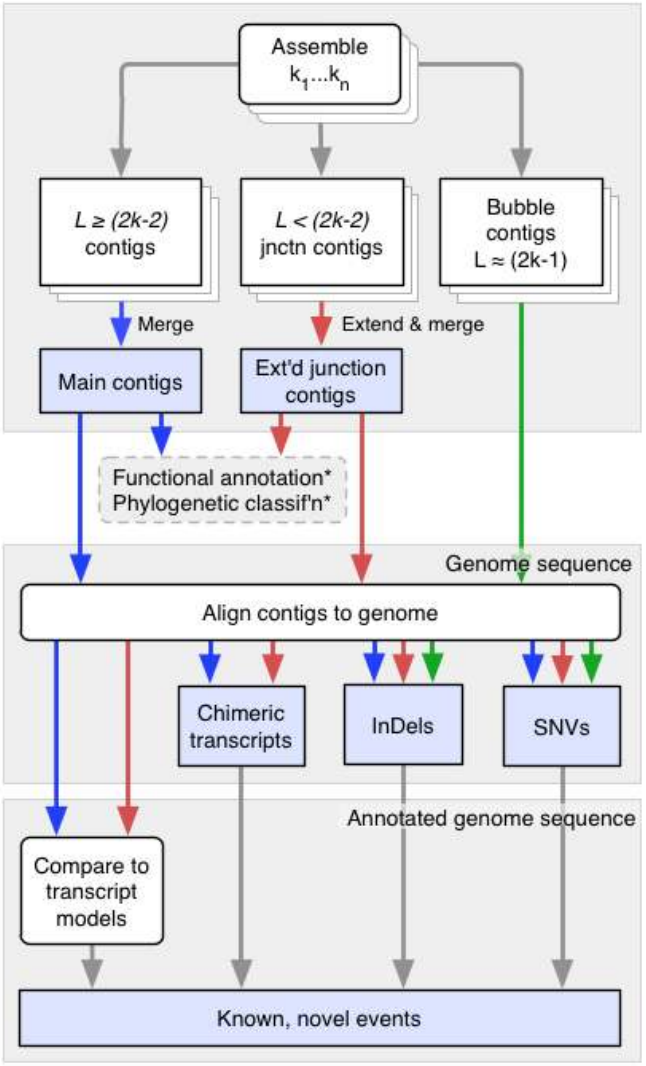
Multi-*k* Assembly

We capture a wide range of expression levels

- Gray: all transcripts with a read alignment
- Blue: at least 80% of a transcript in a single contig
- Red: at least 80% of a transcript is reconstructed



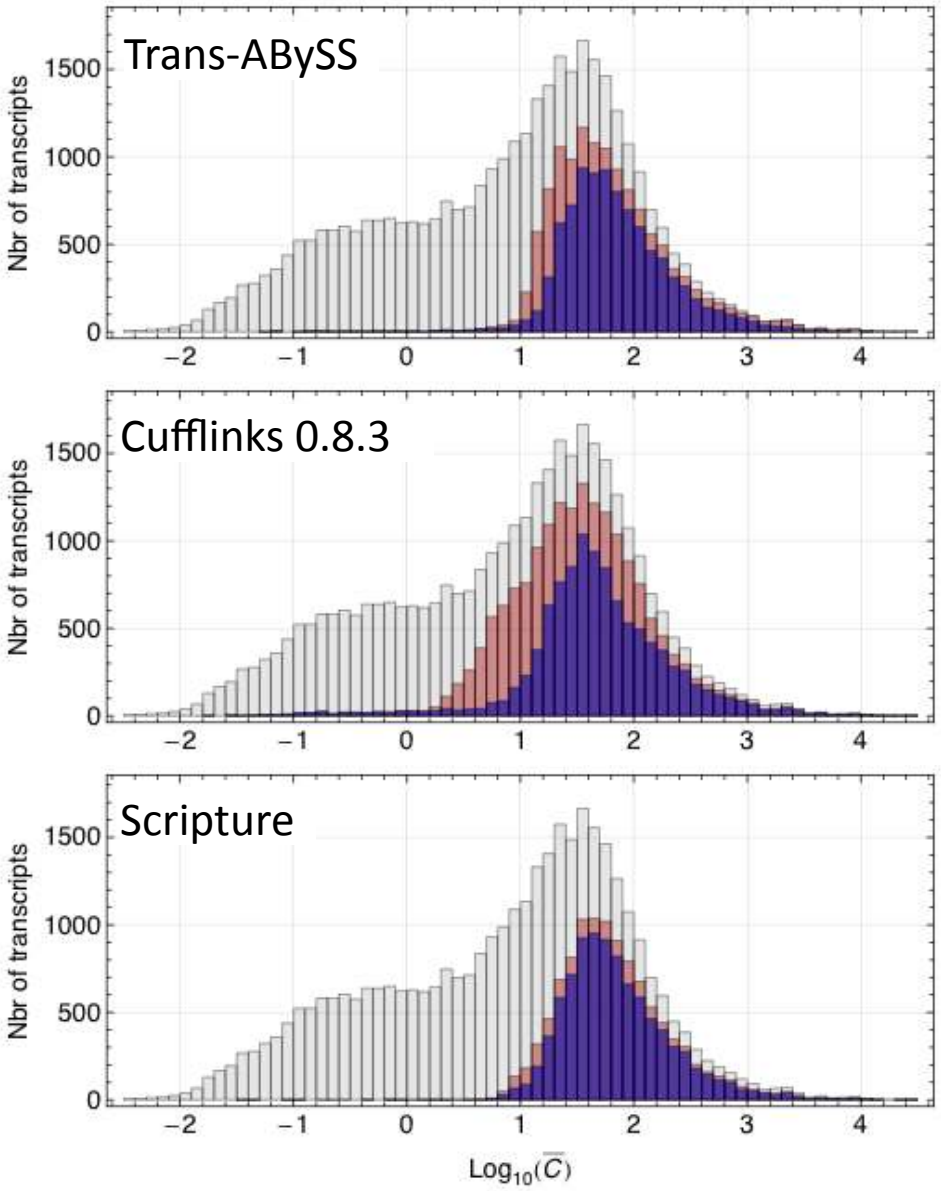
Trans-ABySS



A versatile tool for

- Transcript reconstruction
- Gene identification
- InDel and SNV discovery
- Chimeric transcript discovery
  - Gene fusions
  - Trans-splicing
- Expression analysis

Transcriptome Assembly

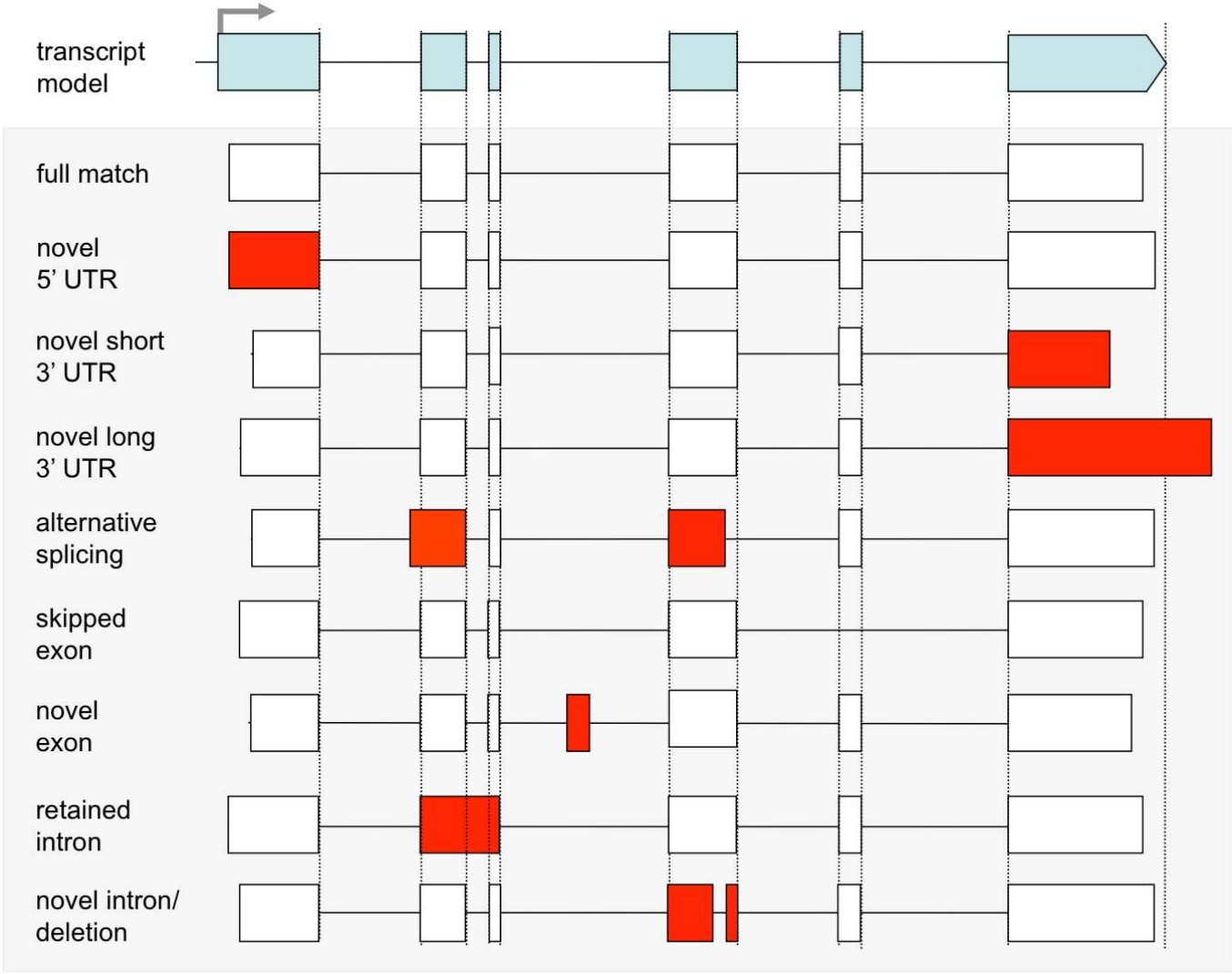


De novo assembly based on ABySS

Reference-based assembly based on TopHat alignments

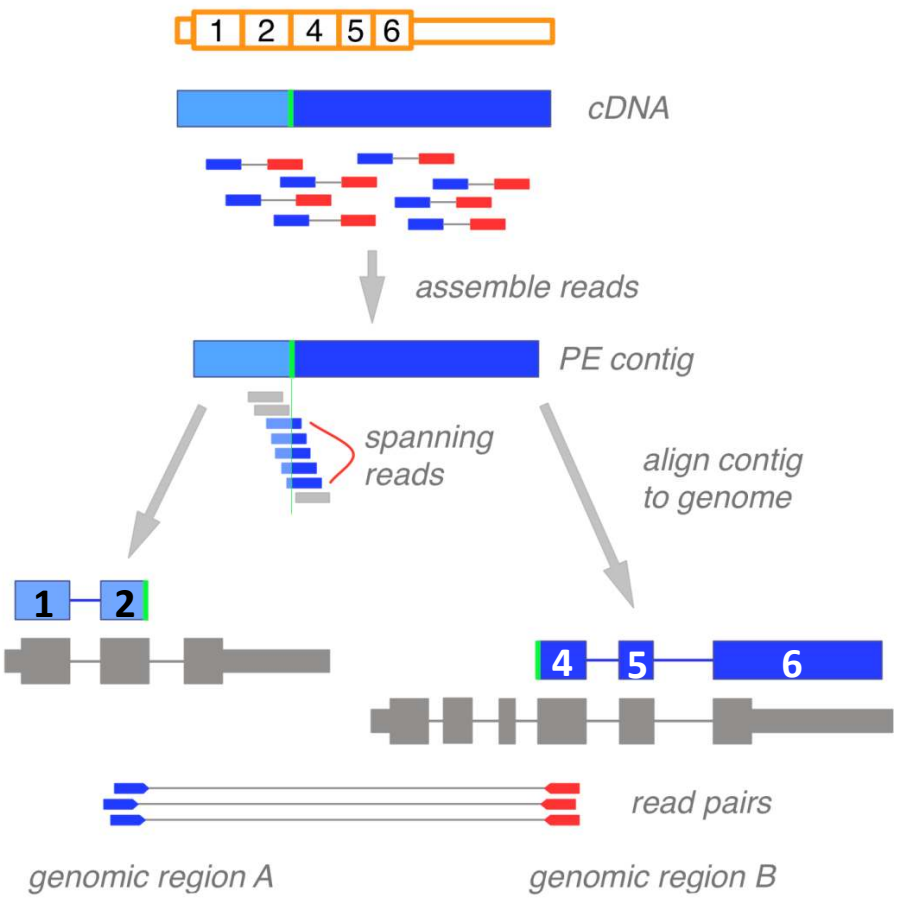
[Trapnell et al., 2010; Guttman et al., 2010; Trapnell et al., 2009]

Events



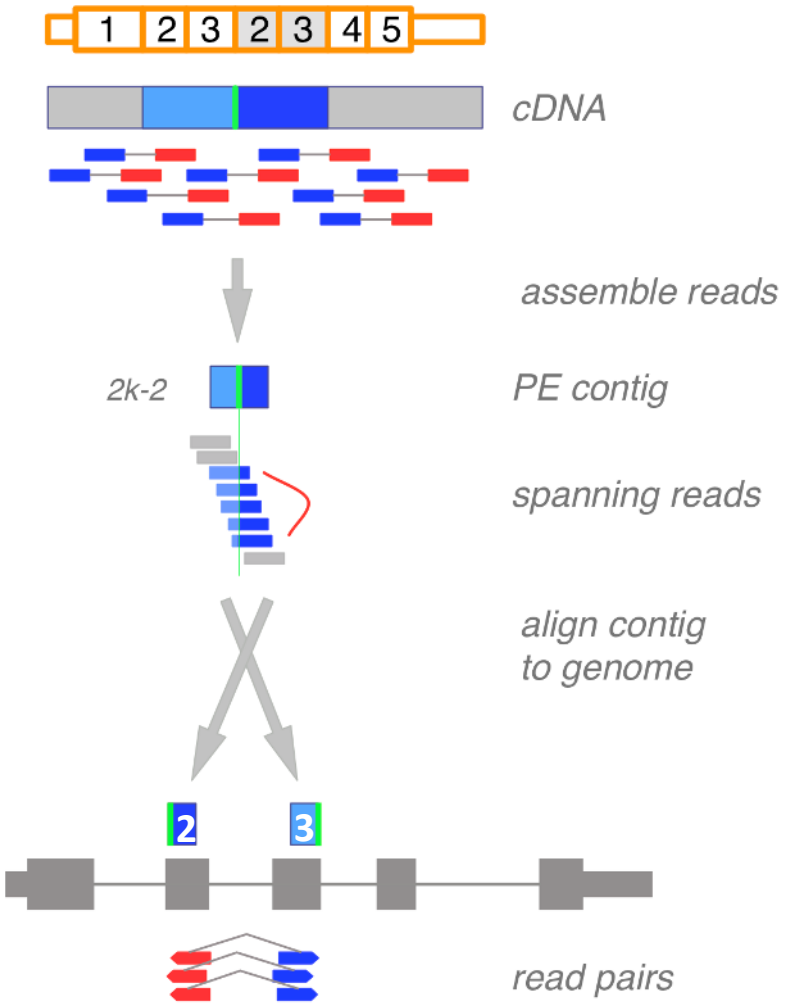
+ chimeric transcripts

Detecting Fusions



- Conventionally detected through identifying translocations in genomes
- Assembled transcriptome contigs span multiple genes
- Break points (usually) correspond to exon boundaries
- Break points are supported by
  - Spanning reads
  - Read pairs linking regions

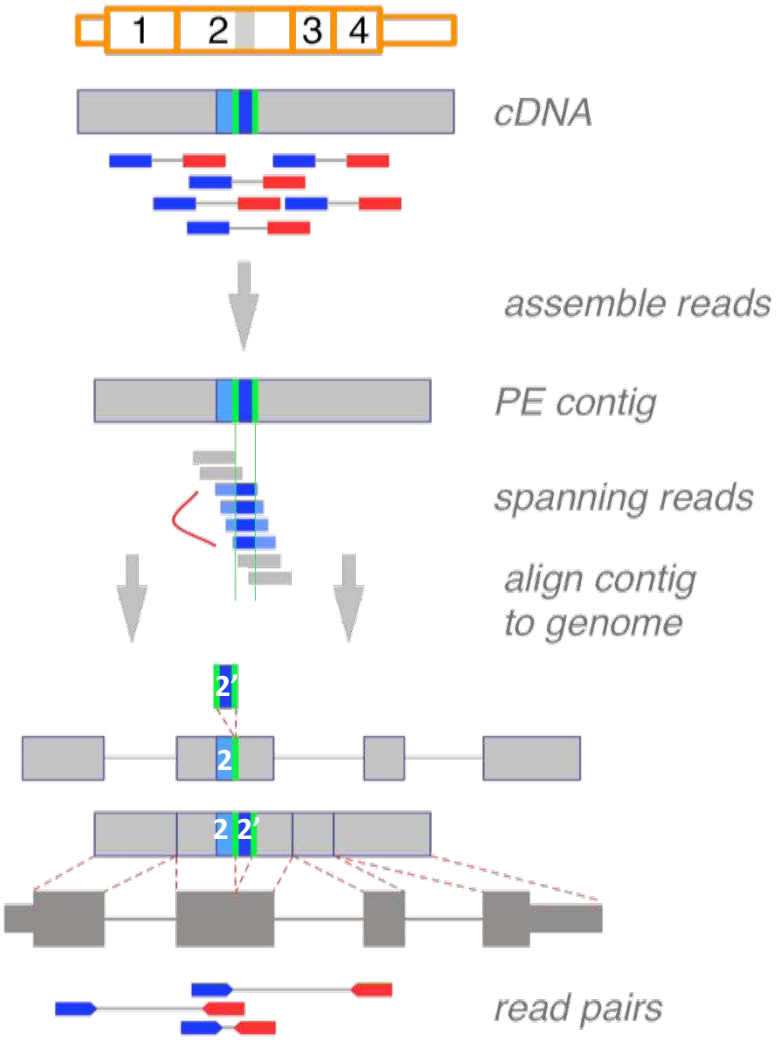
Detecting Partial Tandem Duplications



- One or more exons get repeated in their entirety
- Usually coexist with the wild-type
- PTD events are manifested in a particular contig type
  - A short contig with 50/50 split alignment
- Break points are supported by
  - Spanning reads
  - Read pairs in opposite orientation

Lucas Swanson, Readman Chiu and Gordon Robertson

Detecting Internal Tandem Duplications

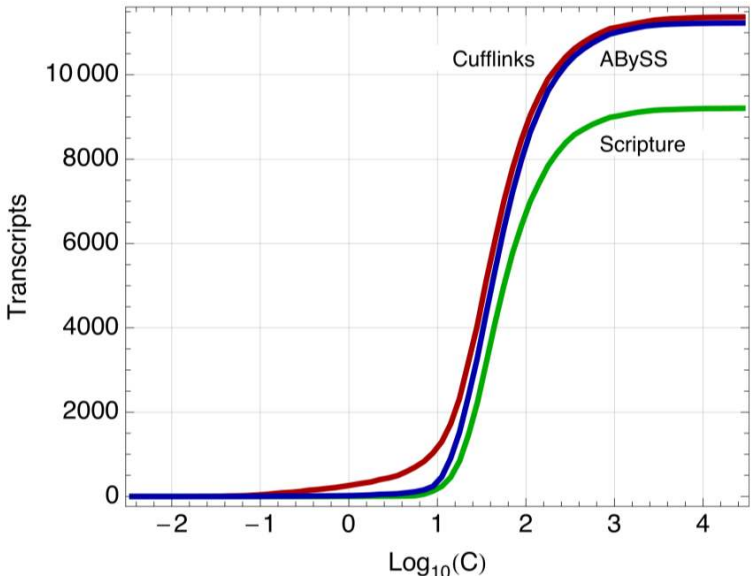


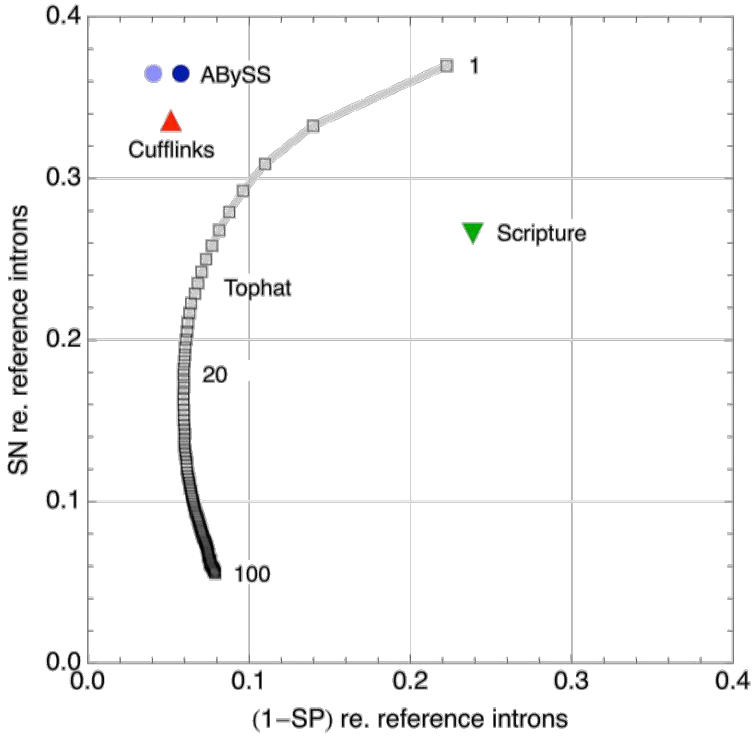
- Tandem duplications internal to exons
- Contig alignments result in
  - Query gaps
  - Contiguous target blocks
- Read support on break point(s)
- Aberrant read pair distances

Lucas Swanson, Readman Chiu and Gordon Robertson

Performance

- Compared to mapping-based analysis tools Trans-ABySS constructs
  - as many transcripts
  - with better sensitivity and specificity





[Trapnell et al., 2010; Guttman et al., 2010; Trapnell et al., 2009]

Sequence Alignment

http://en.wikipedia.org/wiki/Sequence_alignment_software

Short-Read Sequence Alignment

Name	Description
BFAST	Explicit time and accuracy tradeoff with a prior accuracy estimation, supported by indexing the reference sequences. Optimally compresses indexes. Can handle billions of short reads. Can handle insertions, deletions, SNPs, and color errors (can map ABI SOLiD color space reads). Performs a full Smith Waterman alignment.
BLASTN	BLAST's nucleotide alignment program, slow and not accurate for short reads, and uses a sequence database (EST, sanger sequence) rather than a reference genome.
BLAT	Made by Jim Kent. Can handle one mismatch in initial alignment step.
Bowtie	Uses a Burrows-Wheeler transform to create a permanent, reusable index of the genome; 1.3 GB memory footprint for human genome. Aligns more than 25 million Illumina reads in 1 CPU hour. Supports Maq-like and SOAP-like alignment policies (can be run from inside Geneious Server).
BWA	Uses a Burrows-Wheeler transform to create an index of the genome. It's a bit slower than bowtie but allows indels in alignment (can be run from inside Geneious Server).
CASHX	Quantify and manage large quantities of short-read sequence data. CASHX pipeline contains a set of tools that can be used together or as independent modules on their own. This algorithm is very accurate for perfect hits to a reference genome.
CUDA-EC	Short-read alignment error correction using GPUs.
drFAST	Read mapping alignment software that implements cache obliviousness to minimize main/cache memory transfers like mrFAST and mrsFAST, however designed for the SOLiD sequencing platform (color space reads). It also returns all possible map locations for improved structural variation discovery.
ELAND	Implemented by Illumina. Includes ungapped alignment with a finite read length.
GNUMAP	Accurately performs gapped alignment of sequence data obtained from next-generation sequencing machines (specifically that of Solexa/Illumina) back to a genome of any size. Includes adaptor trimming, SNP calling and Bisulfite sequence analysis.
GEM	High-quality alignment engine (exhaustive mapping, that is 100% of sensitivity, for any number of substitutions; 1 non-exhaustive indel). Several standalone applications (mapper, split mapper, mappability, and other) provided.
GMAP and GSNAP	Robust, fast, short-read alignment. GMAP: longer reads, with multiple indels and splices (see entry above under Genomics analysis); GSNAP: shorter reads, with a single indel or up to two splices per read. Useful for digital gene expression, SNP and indel genotyping. Developed by Thomas Wu at Genentech. Used by the National Center for Genome Resources (NCGR) in Alpheus.
Geneious Assembler	Fast, accurate overlap assembler with the ability to handle any combination of sequencing technology, read length, any pairing orientations, with any spacer size for the pairing, with or without a reference genome.
LAST	
MAQ	Ungapped alignment that takes into account quality scores for each base (can be run from inside Geneious Server).
mrFAST and mrsFAST	Gapped (mrFAST) and ungapped (mrsFAST) alignment software that implements cache obliviousness to minimize main/cache memory transfers. They are designed for the Illumina sequencing platform and they can return all possible map locations for improved structural variation discovery.

Sequence alignment

- Global
- Local
- Glocal

Global alignment

- Base-by-base alignment of one sequence to another allowing for both mismatches and gaps
- Example:

  AGAGTGCTGCCGCC

  AGATGTACTGCGCC
- Alignment:

  AGA-GTGCTGCCGCC

  ||| || |||| |||

  AGATGTACTGC-GCC
- 12 matches of 15 bp = 80% identity

Local Alignment

- Given two sequences, find a matching substring from each of those two sequences
- Example:

```
AGATGTGCTGCCGCC
TTTGTACTGAAA
AGATGTGCTGCCGCC
    ||| ||||
  TTTGTACTGAAA
```

- 6 matches of 7 bp = 86% identity

Glocal Alignment

- Given a query sequence and a reference sequence, identify a substring of the reference sequence that matches the entirety of the query sequence
- Example:

  Reference: AGATGTGCTGCCGCCACGT

  Query: TTTGTACTGAAA

  ```
  ACGTAGATGTGCTGCCGCCACGT
        |||  |||
     TTTGTACTGAAA
  ```

- 6 matches of 12 bp = 50% identity

Criteria for Choosing an Aligner

- Global, local or glocal alignment
- Aligning short sequences to long sequences
- Aligning long sequences to long sequences
- Handling small gaps (insertions and deletions)
- Handling large gaps (introns)
- Handling split alignments (chimera)
- Speed and ease of use

Popular Alignment Software

Short reads

- BWA
- GSNAP
- Bowtie
  - TopHat
- SOAP

Long sequence

- BWA-SW
- GMAP
- BLAT
- BLAST
- Exonerate
- MUMmer

Seed and Extend

- For large sequences, an exhaustive alignment is very slow
- Many aligners start by finding perfect or near perfect matches to seeds
- The seeding strategy has a large effect on the sensitivity of the aligner
  - e.g. BLAT requires two perfect nearby 11-mer matches

Memory Use

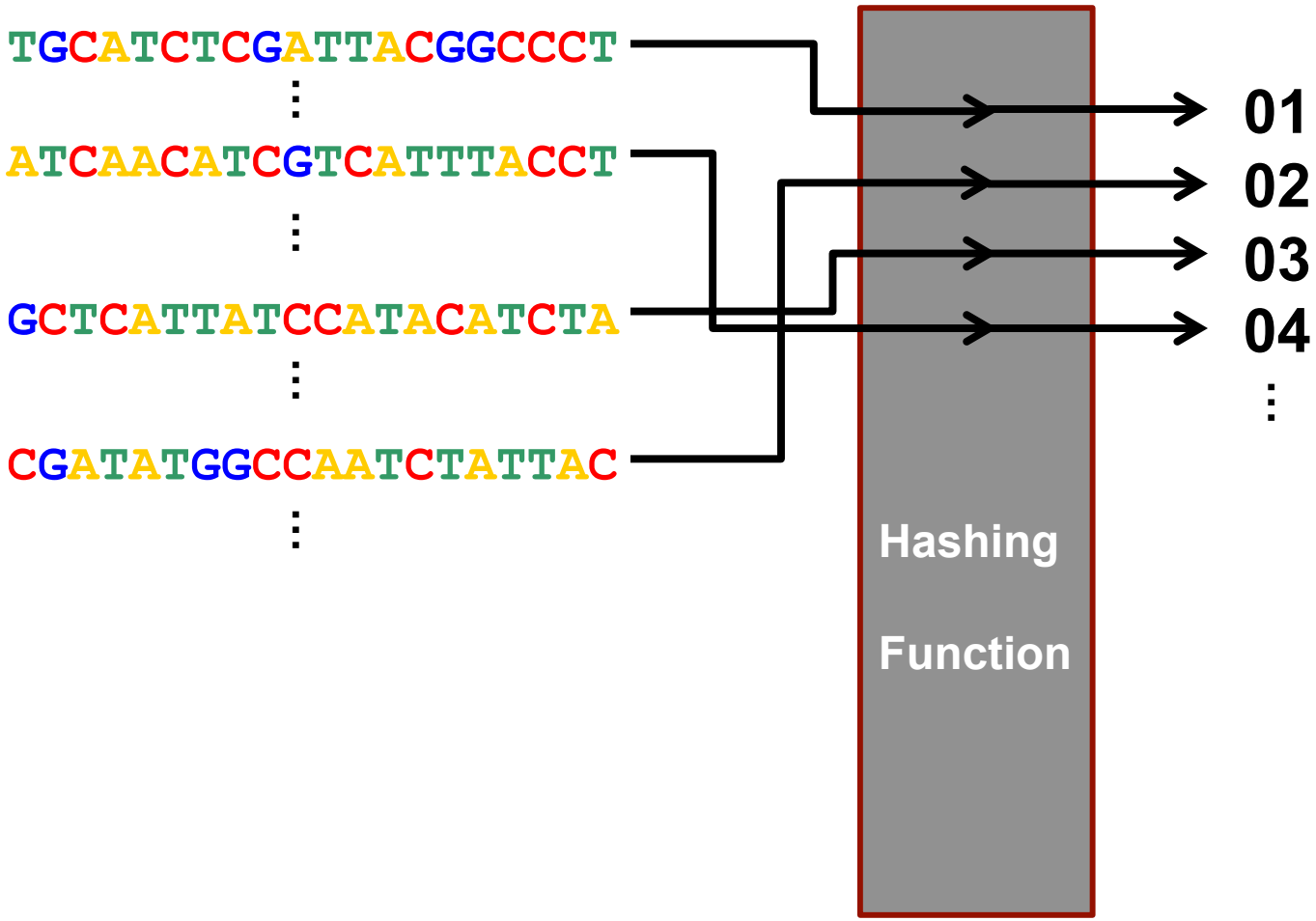
Hashing

- Load a representation of all the reads and/or the reference into memory
  - GSNAP
  - SOAP
  - mr/mrsFAST
  - KAligner

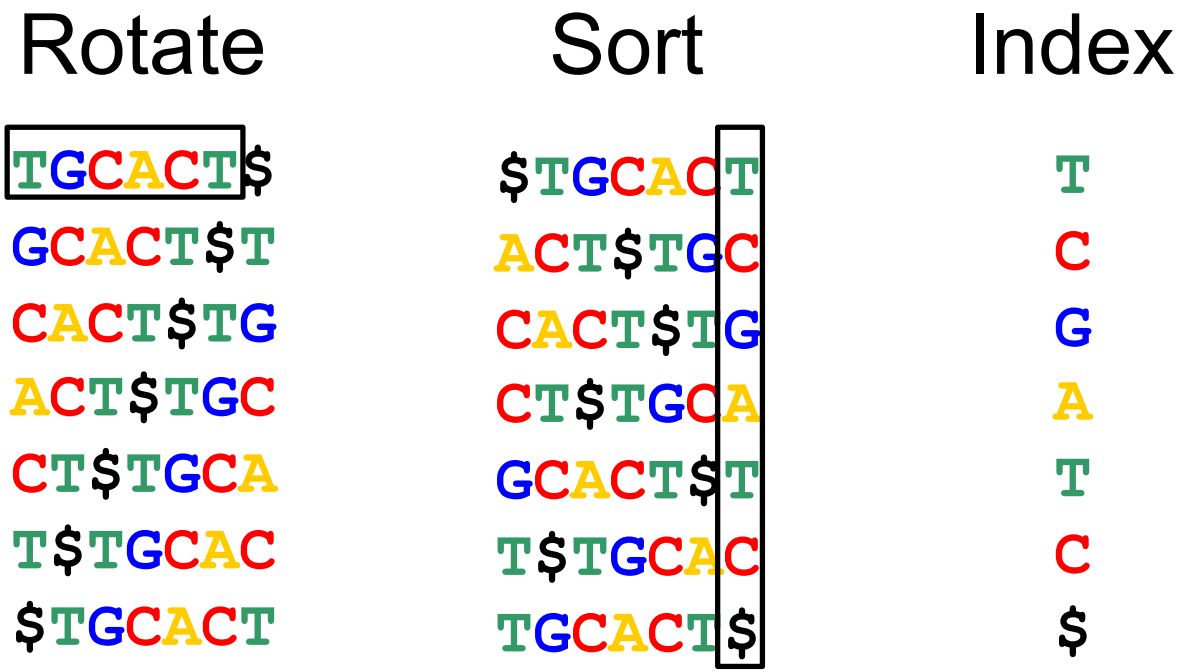
Burrows-Wheeler Transformation (Ferragina-Manzini, indexing)

- Compress reads and/or the reference before loading
  - BWA
  - Bowtie
  - Abyss-map

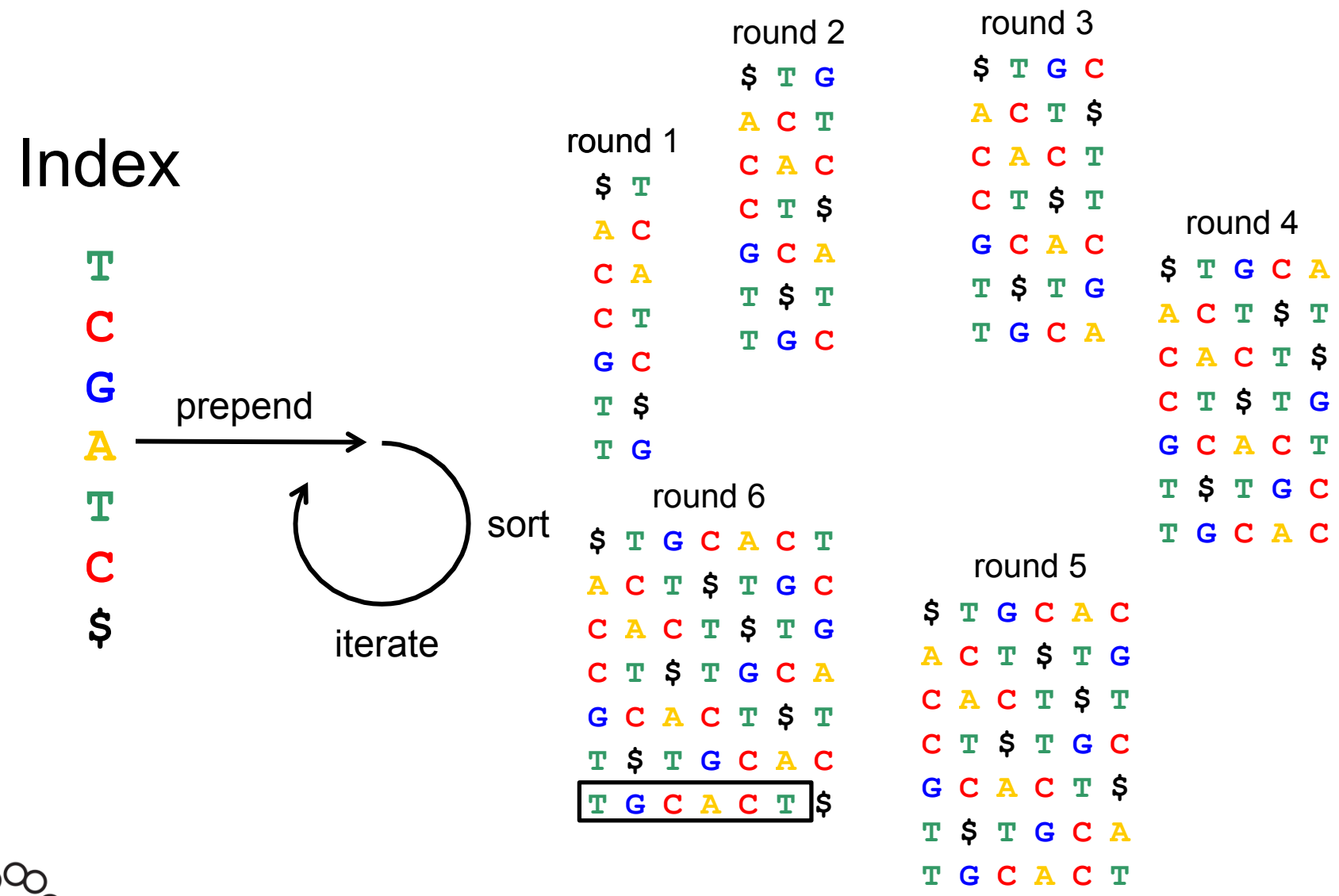
Hashing


TGCATCTCGATTACGGCCCT
ATCAACATCGTCATTTACCT
GCTCATTATCCATACATCTA
CGATATGGCCAATCTATTAC
Hashing
Function
01
02
03
04

BW Transform

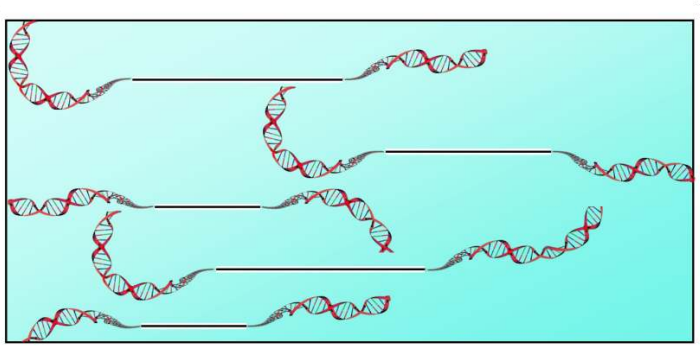


Inverse BW Transform



Summary

De Novo Assembly Problem

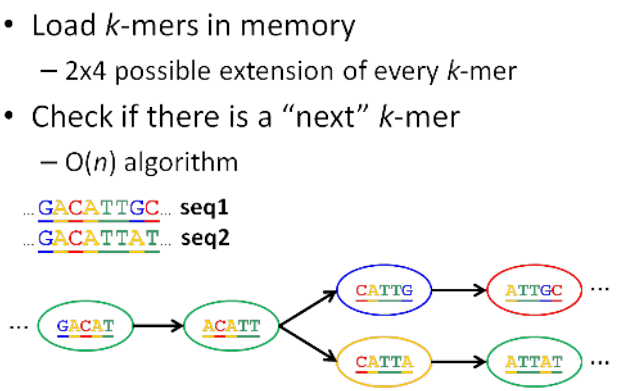


- Old paradigm:
  - long and non-uniform reads (800bp - 1000bp)

5

De Bruijn Graph

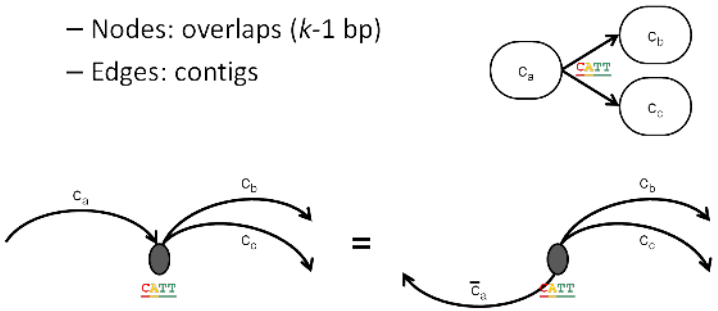
- Load *k*-mers in memory
  - 2x4 possible extension of every *k*-mer
- Check if there is a "next" *k*-mer
  - O(*n*) algorithm



19

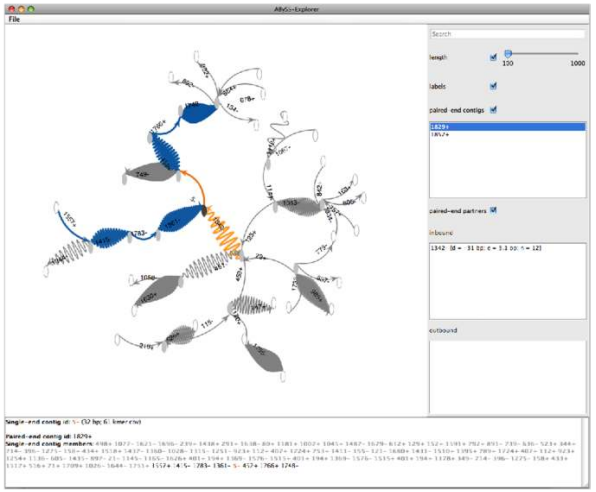
Adjacency Graph

- SET assembly result as a graph
  - Nodes: overlaps (*k*-1 bp)
  - Edges: contigs



28

ABySS-Explorer GUI



33

Summary

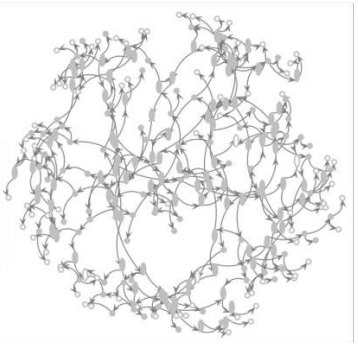
Assembly As a Hairball

- ABySS v1.2.7
  - PET/MPET information disambiguates short contig extensions

Node connectivity*

in \ out	1	2	3	4	5	6+
1	15822	7354	1882	530	109	1
2	7354	9814	1817	456	72	3
3	1882	1817	1074	238	31	1
4	530	456	238	126	13	1
5	109	72	31	13	10	0
6+	1	3	1	1	0	0

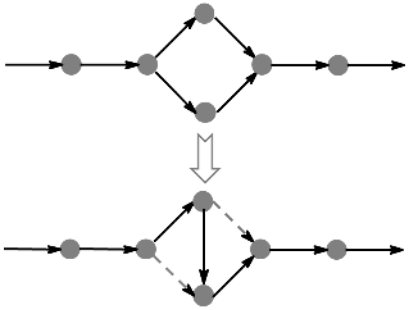
* For contigs ≥ 2 kb



66

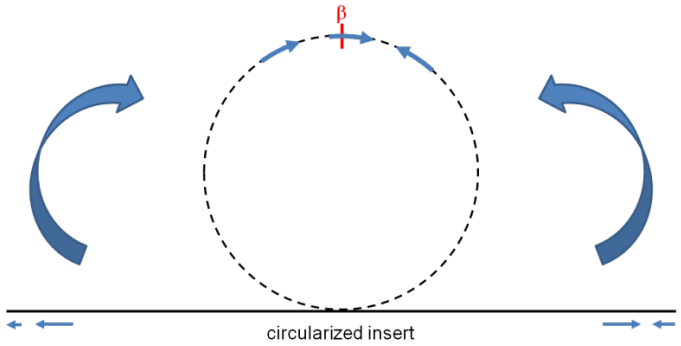
Scaffold Graph Operations

1. Resolve forks



45

Biotin Read-Through

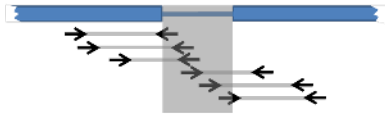


70

Anchor

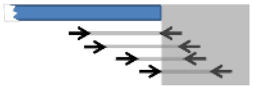
- Local directional assembly
  - scaffold gap filling

(10,499 of 63,986)



  - extension

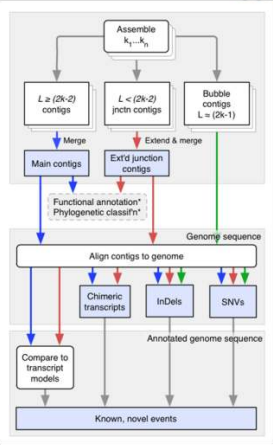
(20,213 of 53,487)



76

Summary

Trans-ABySS

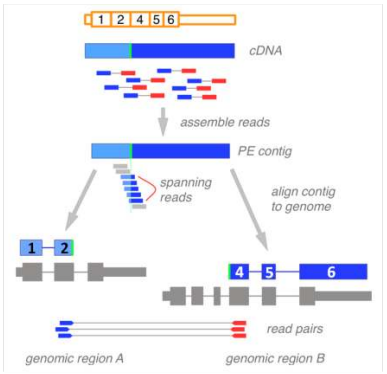


A versatile tool for

- Transcript reconstruction
- Gene identification
- InDel and SNV discovery
- Chimeric transcript discovery
  - Gene fusions
  - Trans-splicing
- Expression analysis

89

Detecting Fusions



- Conventionally detected through identifying translocations in genomes
- Assembled transcriptome contigs span multiple genes
- Break points (usually) correspond to exon boundaries
- Break points are supported by
  - Spanning reads
  - Read pairs linking regions

92 Lucas Swanson, Readman Chiu and Gordon Robertson

BW Transform

Rotate	Sort	Index
TGCACT$	$TGCACT	T
GCACT$T	ACT$TGC	C
CACT$TG	CACT$TG	G
ACT$TGC	CT$TGCA	A
CT$TGCA	GCACT$T	T
T$TGCAC	T$TGCAC	C
$TGCACT	TGCACT$	$

107

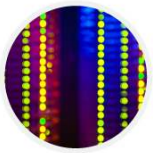
www.bcgsc.ca -> software

ABySS,
Trans-ABySS,
ABySS-Explorer,
Anchor

ABySS and Trans-ABySS Google Groups

ibirol@bcgsc.ca

sjackman@bcgsc.ca





CANADA'S MICHAEL SMITH GENOME SCIENCES CENTRE
WWW.BCGSC.CA

ACKNOWLEDGEMENTS

ABySS Team:
Shaun Jackman
Tony Raymond
Rod Docking

Cydney Nielsen

Beetle Project:
Joerg Bohlmann
Chris Keeling

Trans-ABySS Team:
Readman Chiu
Karen Mungall
Gordon Robertson
Ka Ming Nip
Jenny Qian
Rong She
Lucas Swanson

Nancy Liao
Greg Taylor
Simon Chan
Diana Palmquist

GSC:
Sequencing Team
Library Core

Steven Jones
Marco Marra


GenomeCanada


GenomeBritishColumbia