RECOMB 2011 Satellite Workshop on Massively Parallel Sequencing (RECOMB-seq)
26-27 March 2011, Vancouver, BC, Canada; Short talk: 2011-03-27 12:10-12:30 (presentation: 15 minutes, questions: 5 minutes)
Slides available online at http://boisvert.info/dropbox/recomb-seq-2011-talk.pdf, version: 2011-03-23-1

Constrained traversal of repeats with paired sequences

Sébastien Boisvert, Élénie Godzaridis, François Laviolette & Jacques Corbeil

Department of Molecular Medicine

Department of Computer Science
and Software Engineering

UNIVERSITÉ LAVAL

Repeats and gene innovation

Biological usefulness:

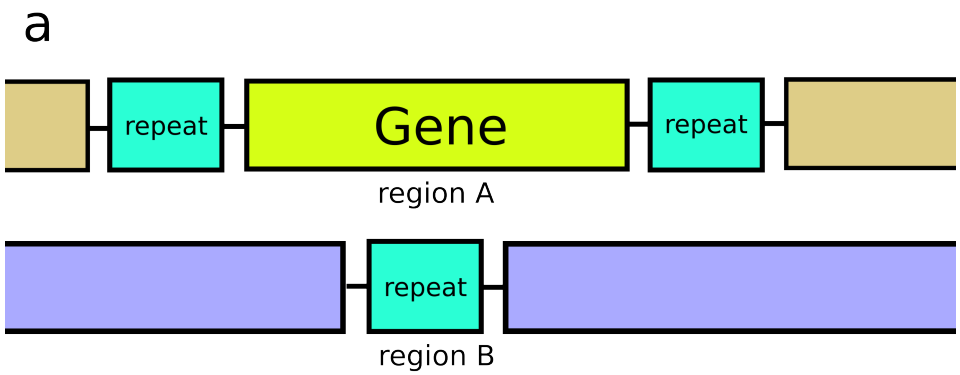
- role in 'gene innovation'

Repeats and gene innovation

Biological usefulness:

- role in 'gene innovation'
- ease copy-and-paste events in genomes

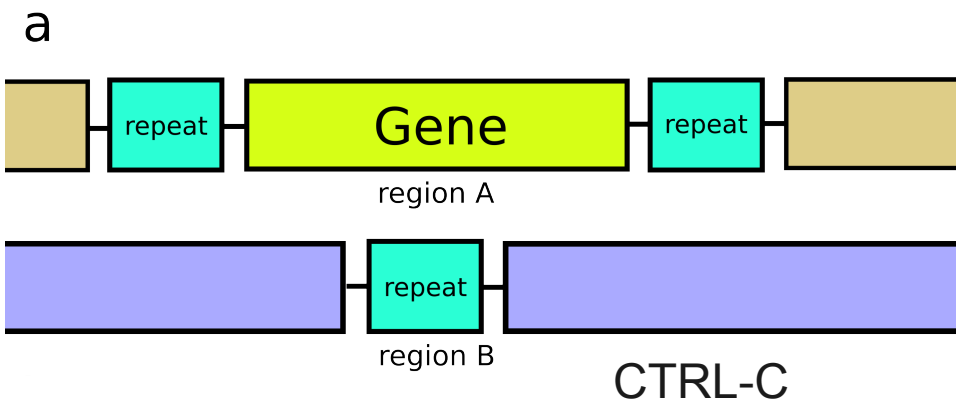
Repeats and gene innovation



Biological usefulness:

- role in 'gene innovation'
- ease copy-and-paste events in genomes

Repeats and gene innovation



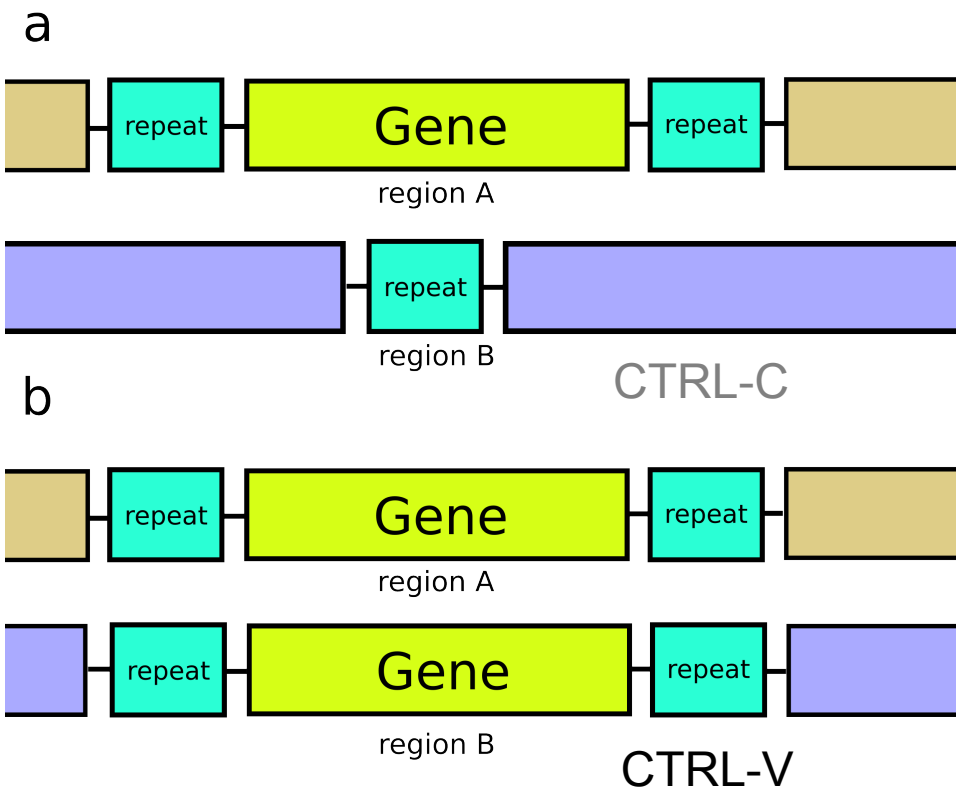
Biological usefulness:

- role in 'gene innovation'
- ease copy-and-paste events in genomes

Repeats and gene innovation

Biological usefulness:

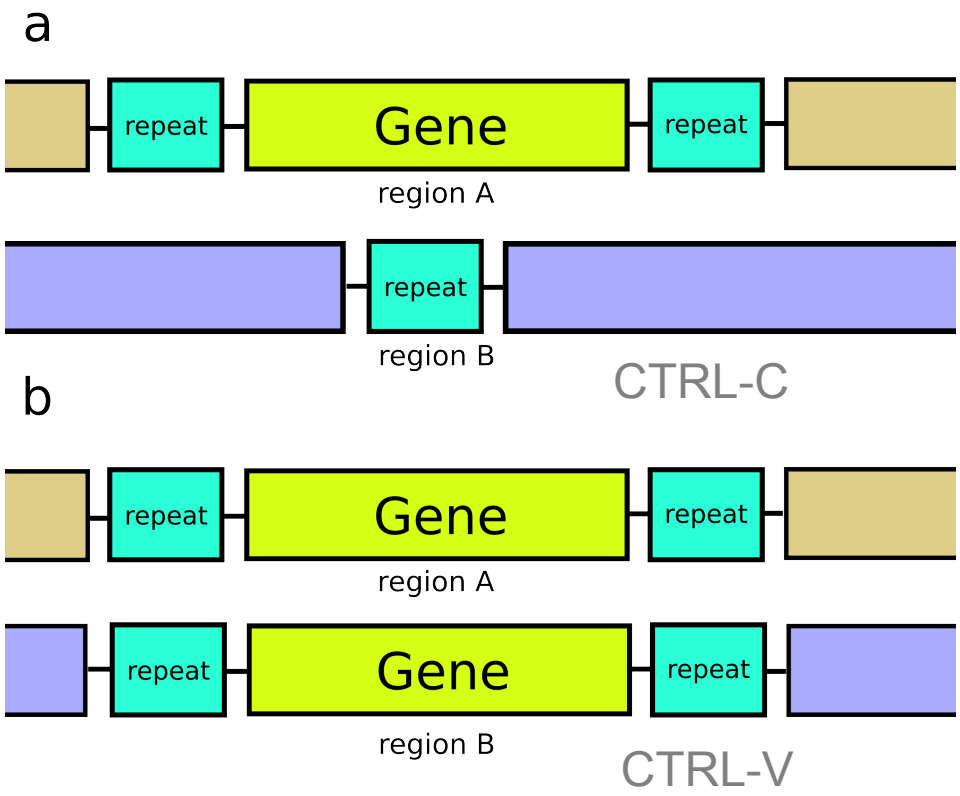
- role in 'gene innovation'
- ease copy-and-paste events in genomes



Repeats and gene innovation

Biological usefulness:

- role in 'gene innovation'
- ease copy-and-paste events in genomes

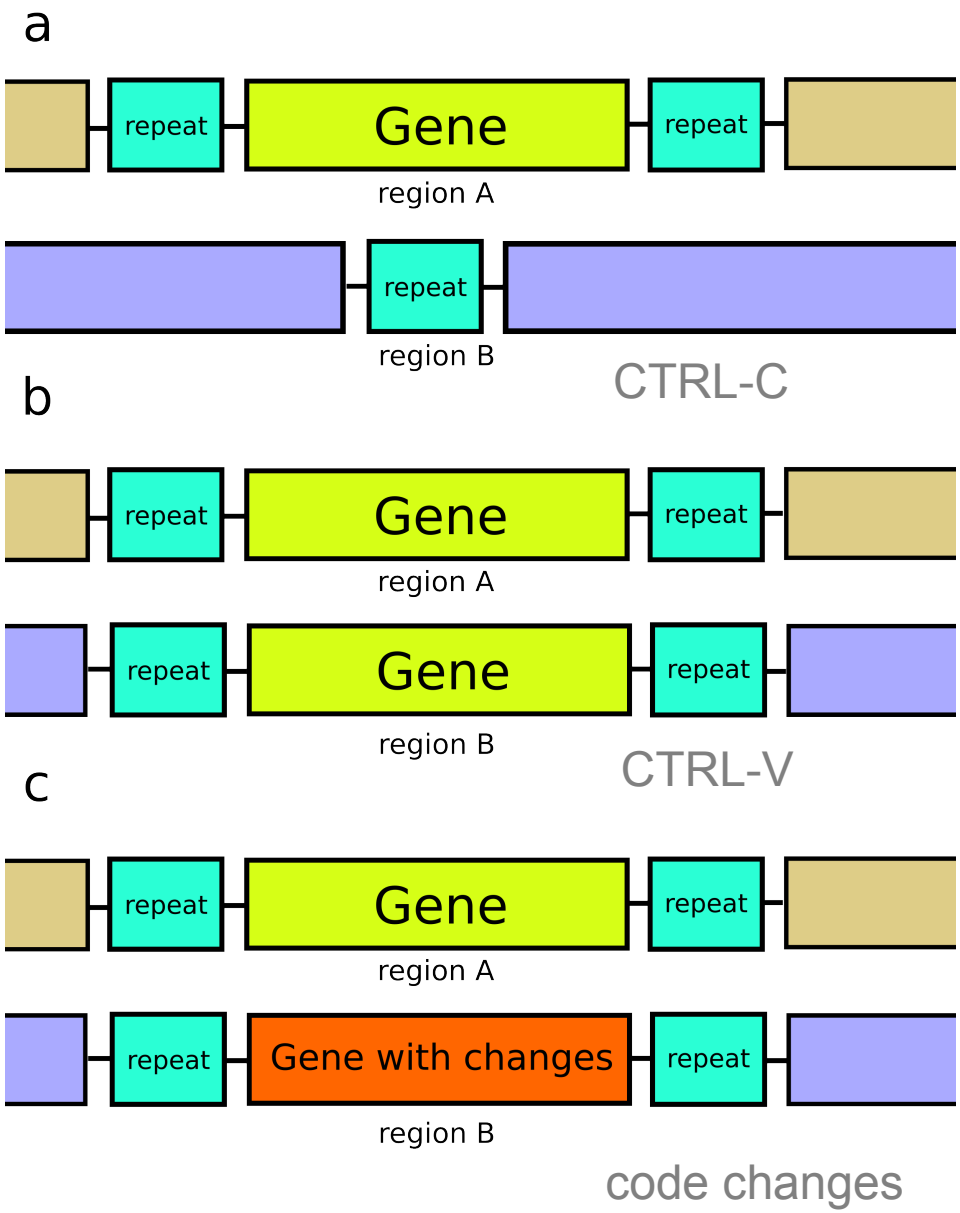


code changes

Repeats and gene innovation

Biological usefulness:

- role in 'gene innovation'
- ease copy-and-paste events in genomes



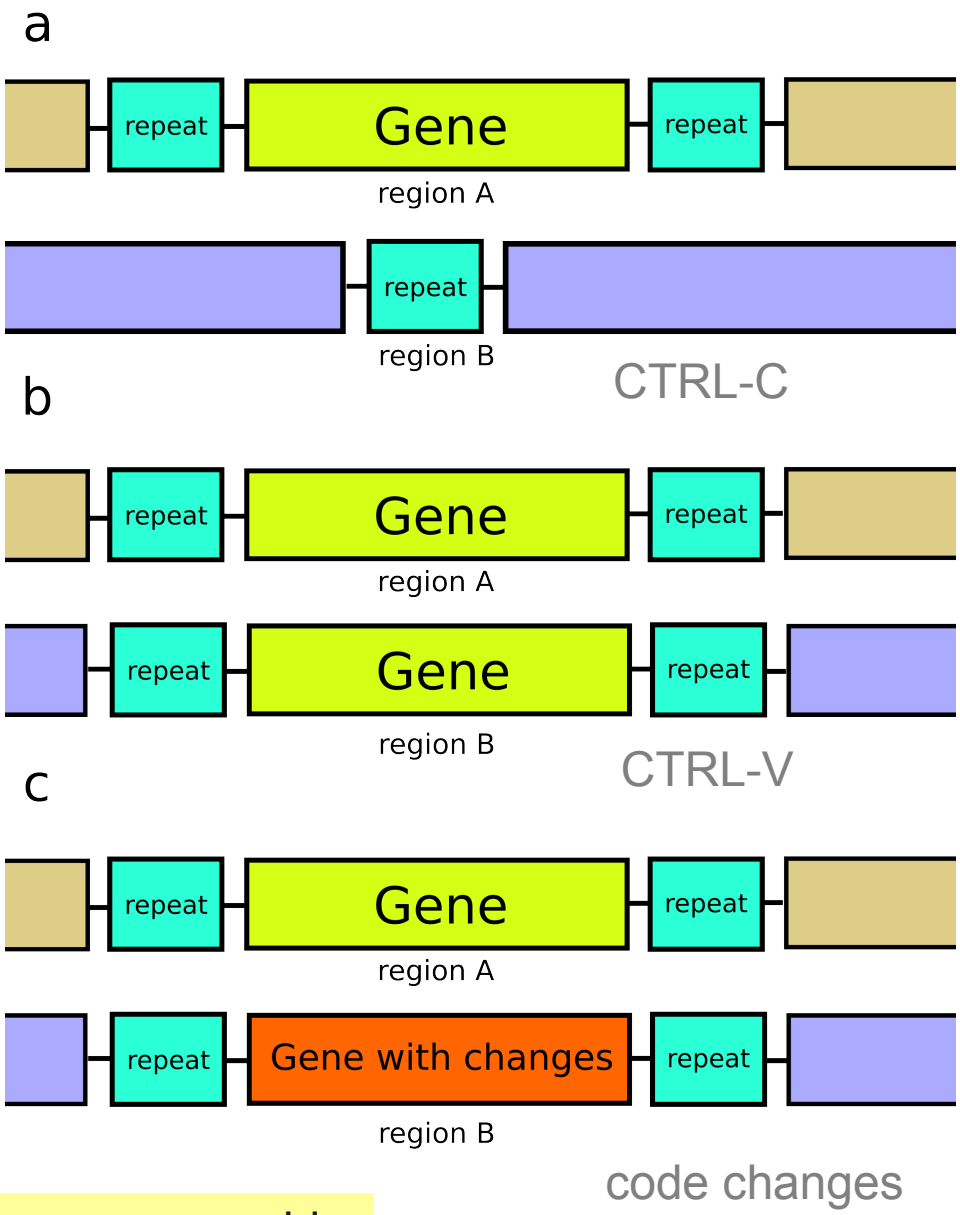
Repeats and gene innovation

Biological usefulness:

- role in 'gene innovation'
- ease copy-and-paste events in genomes

Data analysis point-of-view:

- often collapsed by assembly algorithms



Limitations of next-generation genome sequence assembly
Alkan, Can and Sajjadian, Saba and Eichler, Evan E.
Nature Methods, 2011
http://dx.doi.org/doi:10.1038/nmeth.1527

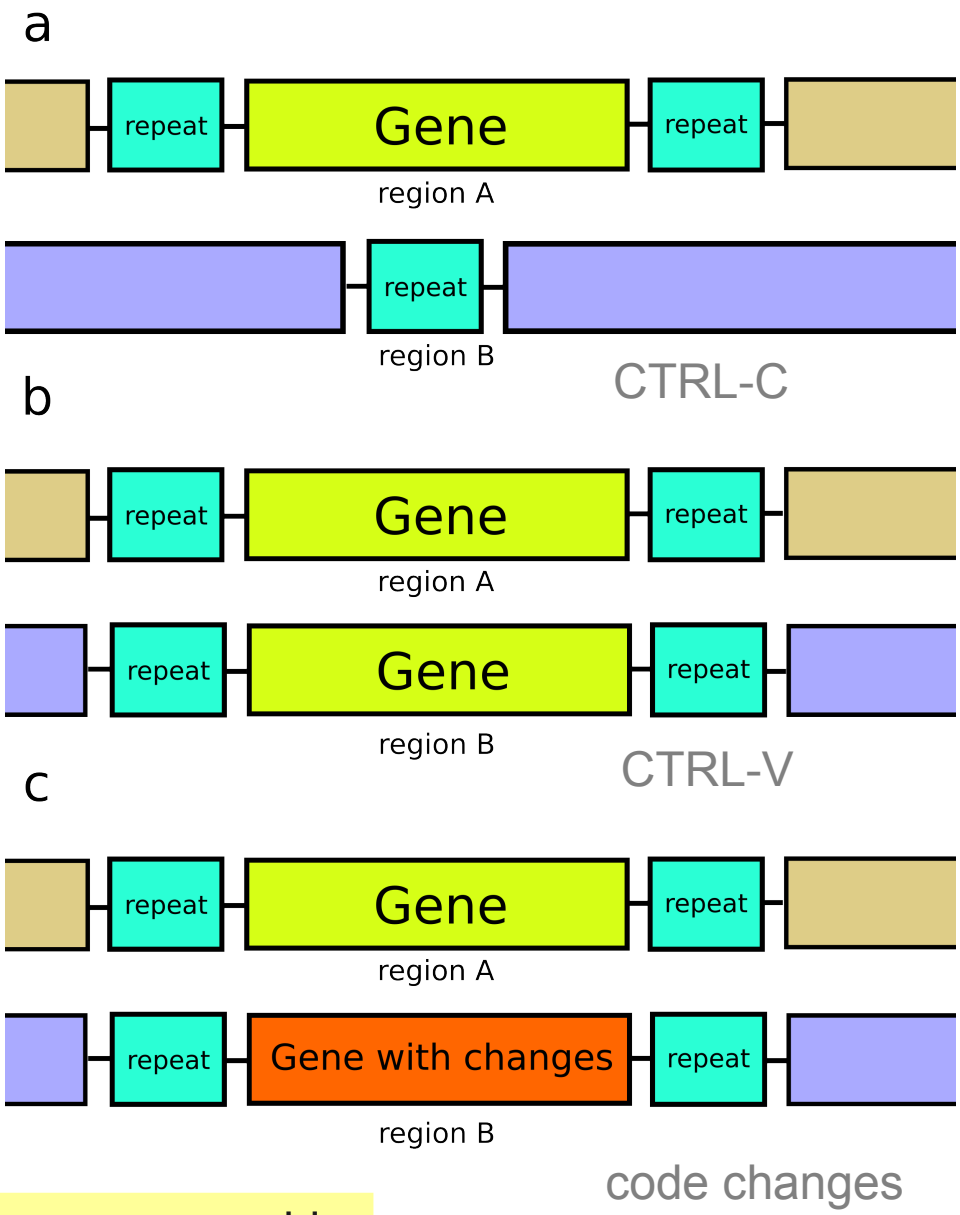
Repeats and gene innovation

Biological usefulness:

- role in 'gene innovation'
- ease copy-and-paste events in genomes

Data analysis point-of-view:

- often collapsed by assembly algorithms
- source of misassemblies

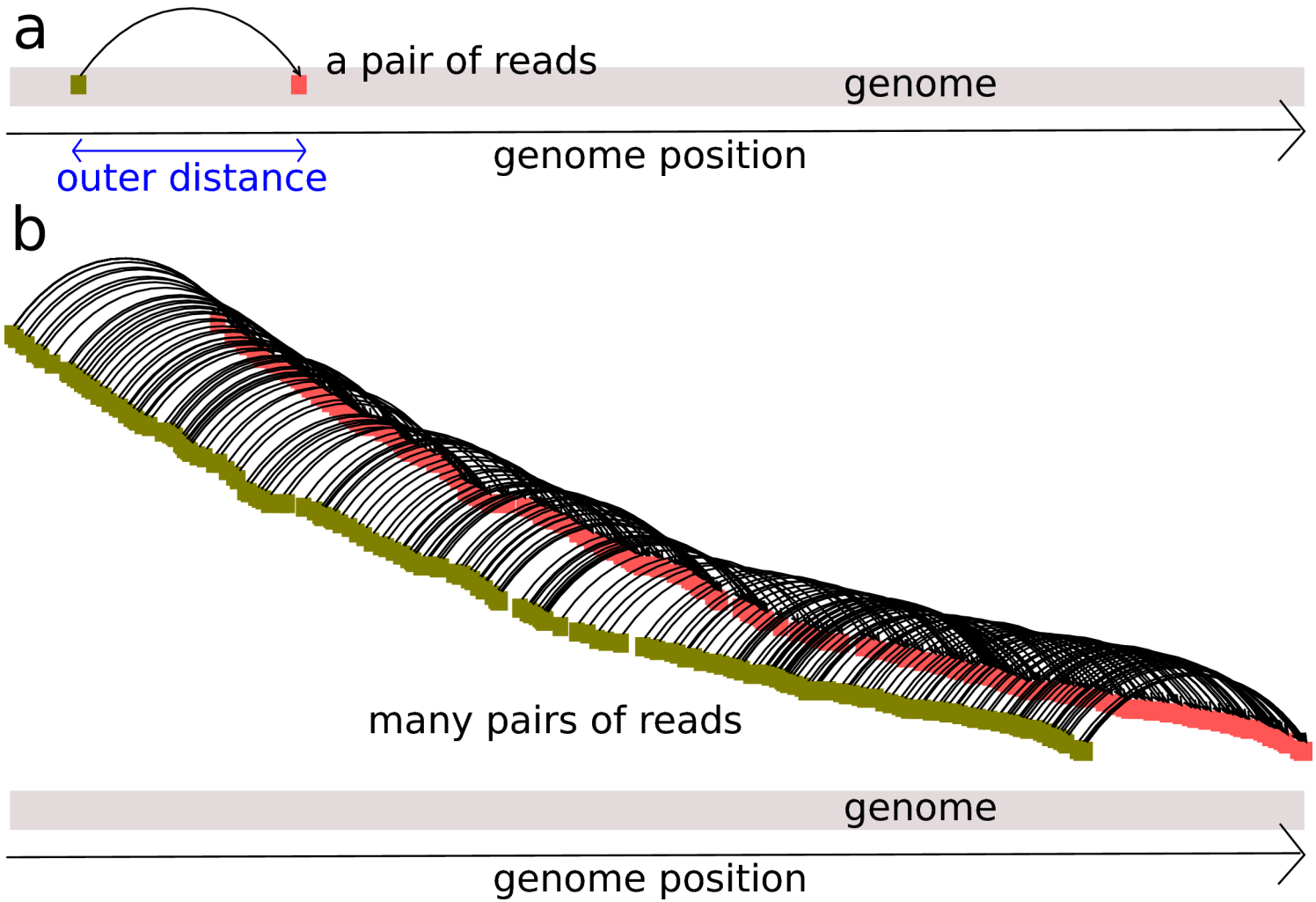


Limitations of next-generation genome sequence assembly
Alkan, Can and Sajjadian, Saba and Eichler, Evan E.
Nature Methods, 2011
http://dx.doi.org/doi:10.1038/nmeth.1527

Background

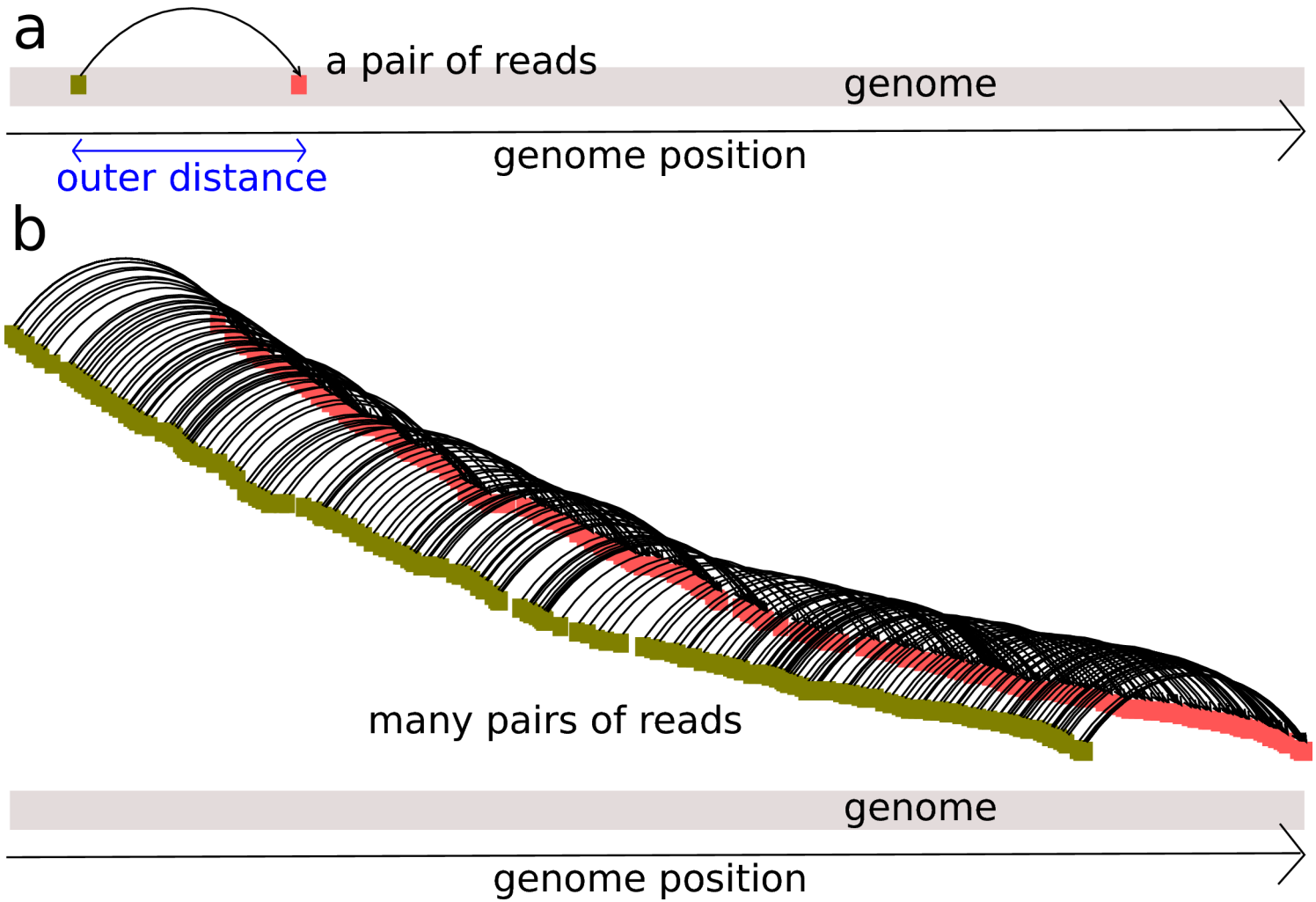
- DNA sequence: contains blueprints of biological systems (genome)

Background



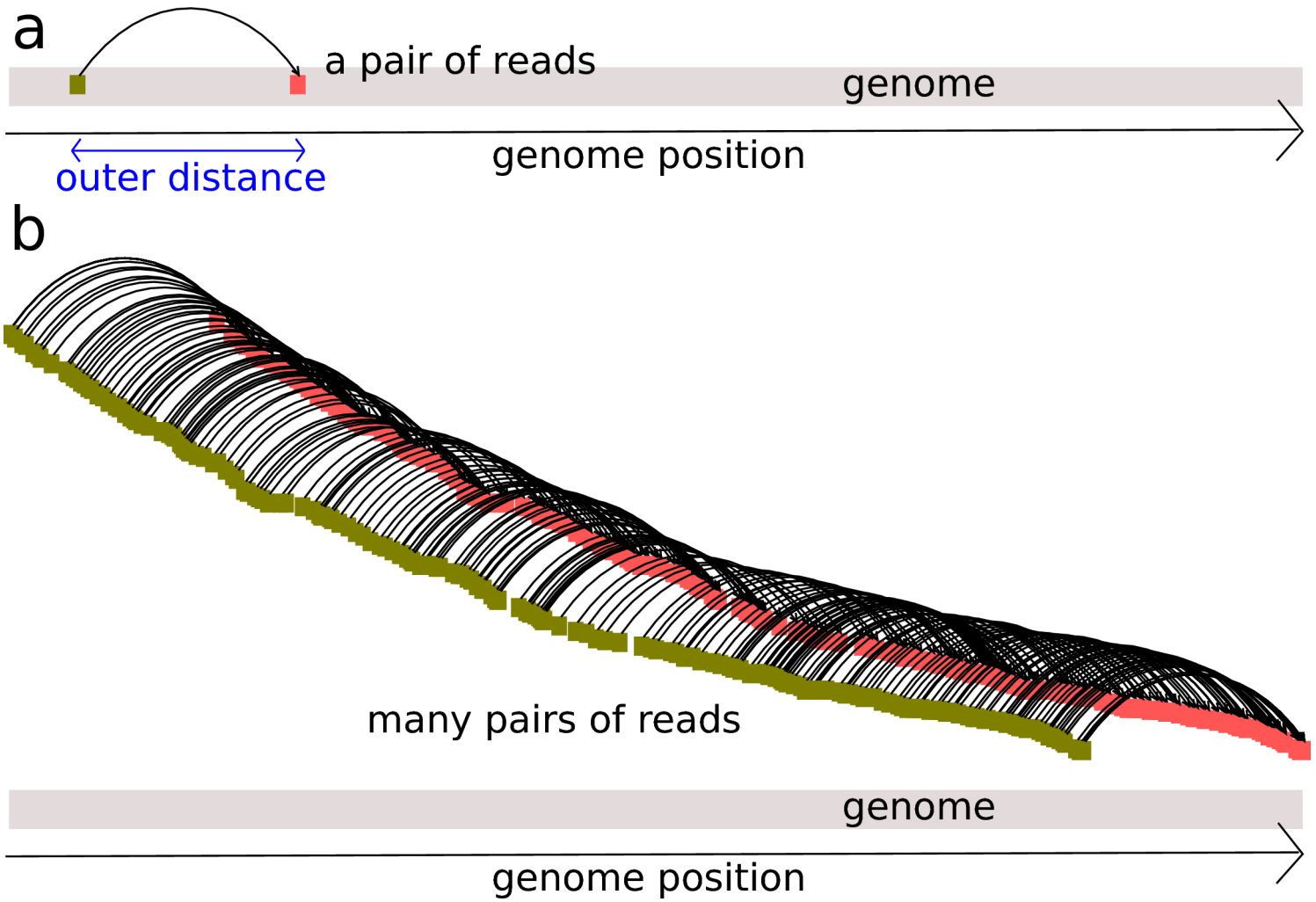
- DNA sequence: contains blueprints of biological systems (genome)
- Goal: get genome sequence from short (50-100 nt) paired reads

Background



- DNA sequence: contains blueprints of biological systems (genome)
- Goal: get genome sequence from short (50-100 nt) paired reads
- Reads contain errors, unknown sampling DNA strand, genome length >>> read length

Background



- DNA sequence: contains blueprints of biological systems (genome)
- Goal: get genome sequence from short (50-100 nt) paired reads
- Reads contain errors, unknown sampling DNA strand, genome length >>> read length
- Outer distances of pairs: not constant, randomly distributed

Seminal work

- Next-generation DNA sequencing deluge started ~2005

Seminal work

- Next-generation DNA sequencing deluge started ~2005
- How to assemble reads into genomes ?

Seminal work

- Next-generation DNA sequencing deluge started ~2005
- How to assemble reads into genomes ?

"we cut the existing pieces of a puzzle into even smaller pieces of regular shape"

- From the landmark paper by Pevzner et al.:

An Eulerian path approach to DNA fragment assembly
Pevzner, Pavel A. and Tang, Haixu and Waterman, Michael S.
Proceedings of the National Academy of Sciences, 2001
http://dx.doi.org/doi:10.1073/pnas.171285098

Discrete systems

- A graph G: vertices V(G) & arcs $E(G) \subseteq V(G) \times V(G)$

Discrete systems

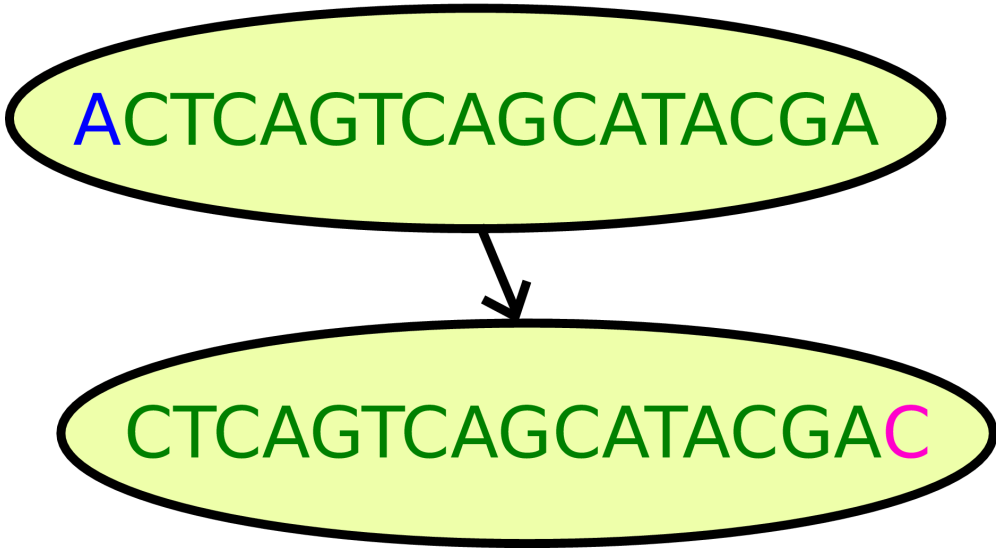
- A graph G: vertices V(G) & arcs $E(G) \subseteq V(G) \times V(G)$
- Alphabet $\Sigma = \{A, T, C, G\}$ & integer k

Discrete systems

- A graph G: vertices V(G) & arcs $E(G) \subseteq V(G) \times V(G)$
- Alphabet $\Sigma = \{A, T, C, G\}$ & integer k
- Vertices: k-mers in reads and reverse-complement reads $V(G) \subseteq \Sigma^k$

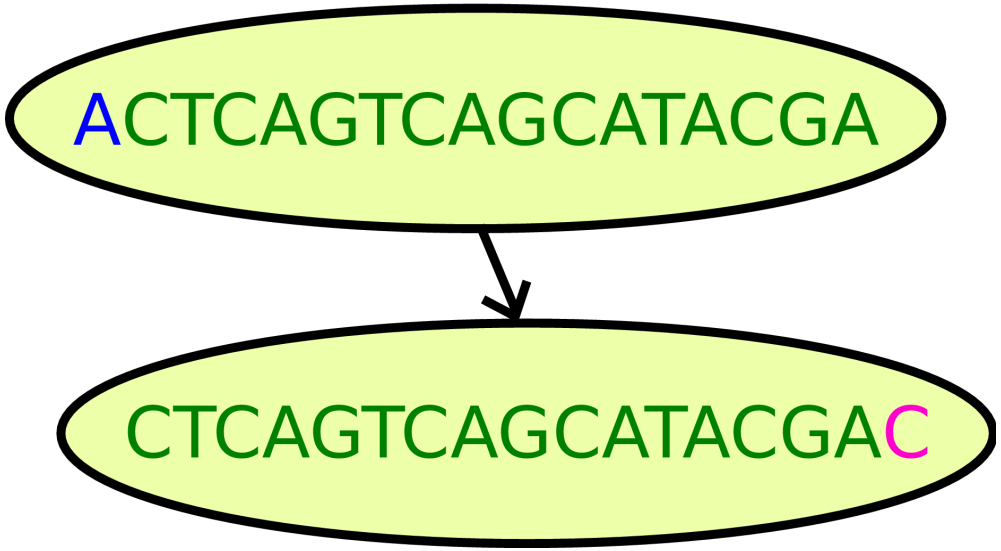
Discrete systems

- A graph G: vertices V(G) & arcs $E(G) \subseteq V(G) \times V(G)$
- Alphabet $\Sigma = \{A, T, C, G\}$ & integer k
- Vertices: k-mers in reads and reverse-complement reads $V(G) \subseteq \Sigma^k$
- Arcs: u and v overlap on k-1 symbols $\Leftrightarrow (u, v) \in E(G)$



Discrete systems

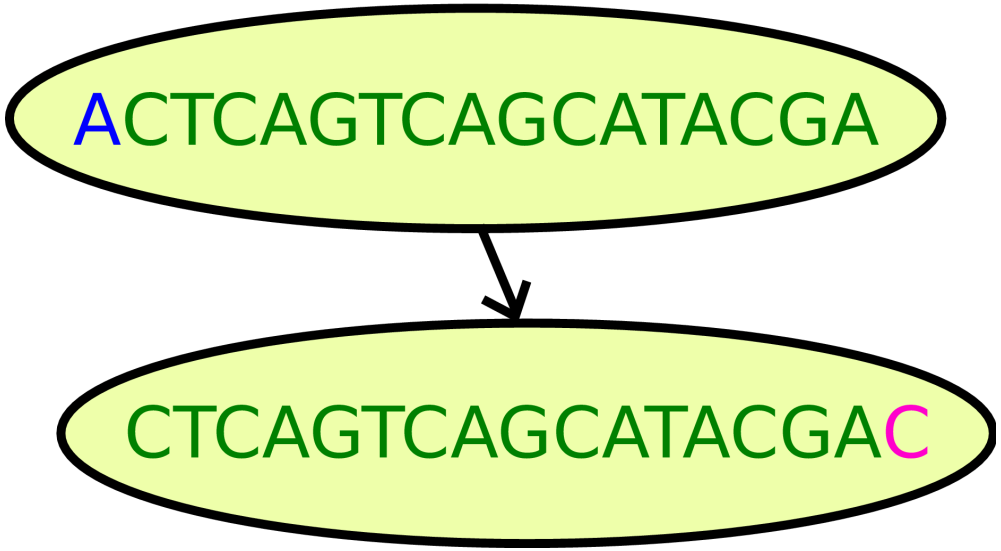
- A graph G: vertices V(G) & arcs $E(G) \subseteq V(G) \times V(G)$
- Alphabet $\Sigma = \{A, T, C, G\}$ & integer k
- Vertices: k-mers in reads and reverse-complement reads $V(G) \subseteq \Sigma^k$
- Arcs: u and v overlap on k-1 symbols $\Leftrightarrow (u, v) \in E(G)$



- Subgraph of the de Bruijn graph

Discrete systems

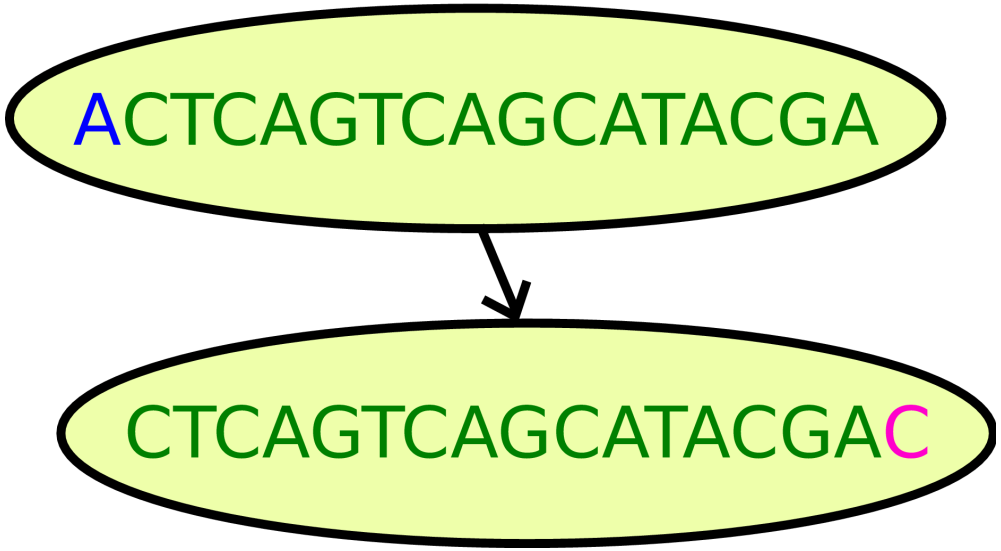
- A graph G: vertices V(G) & arcs $E(G) \subseteq V(G) \times V(G)$
- Alphabet $\Sigma = \{A, T, C, G\}$ & integer k
- Vertices: k-mers in reads and reverse-complement reads $V(G) \subseteq \Sigma^k$
- Arcs: u and v overlap on k-1 symbols $\Leftrightarrow (u, v) \in E(G)$



- Subgraph of the de Bruijn graph
- Genome: path in this graph

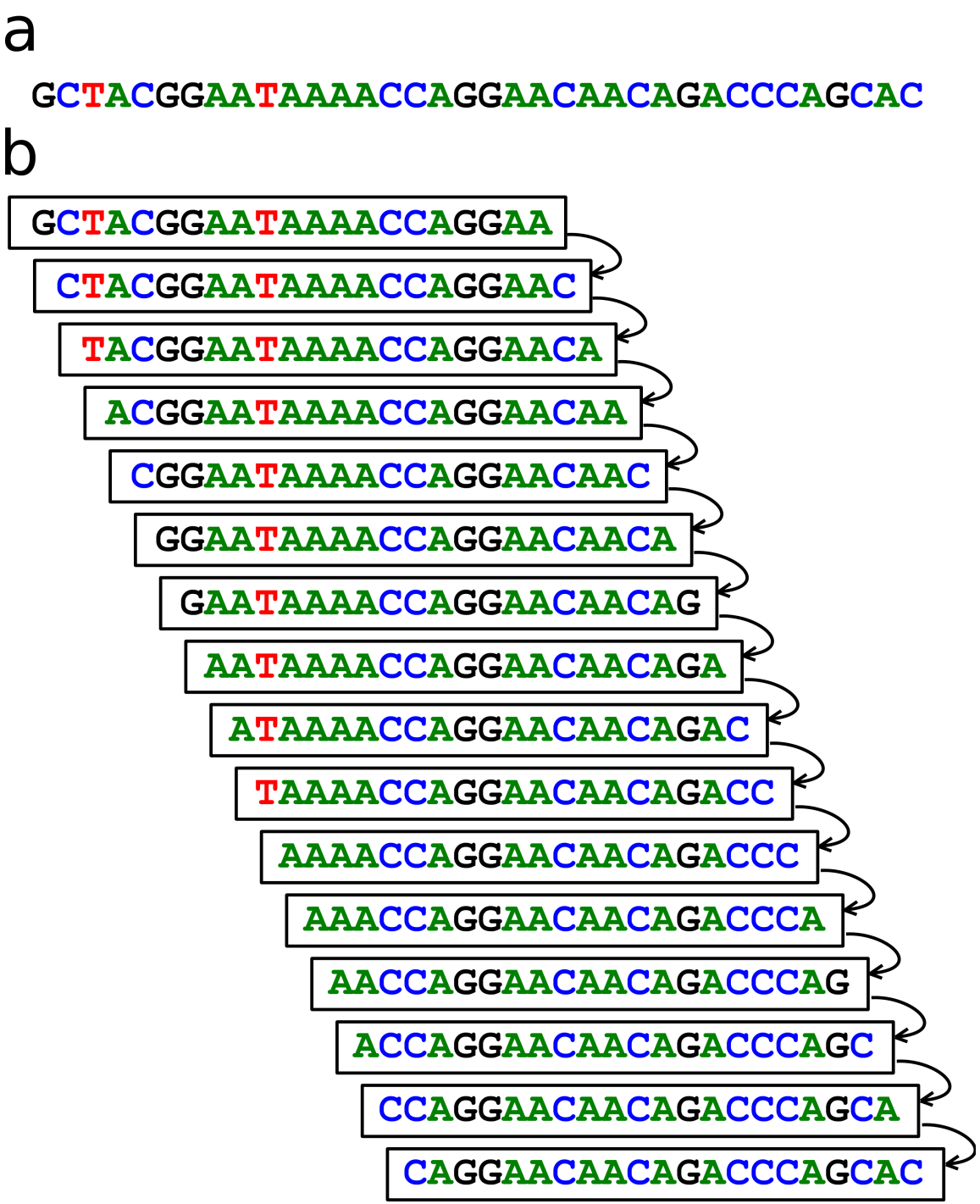
Discrete systems

- A graph G: vertices V(G) & arcs $E(G) \subseteq V(G) \times V(G)$
- Alphabet $\Sigma = \{A, T, C, G\}$ & integer k
- Vertices: k-mers in reads and reverse-complement reads $V(G) \subseteq \Sigma^k$
- Arcs: u and v overlap on k-1 symbols $\Leftrightarrow (u, v) \in E(G)$



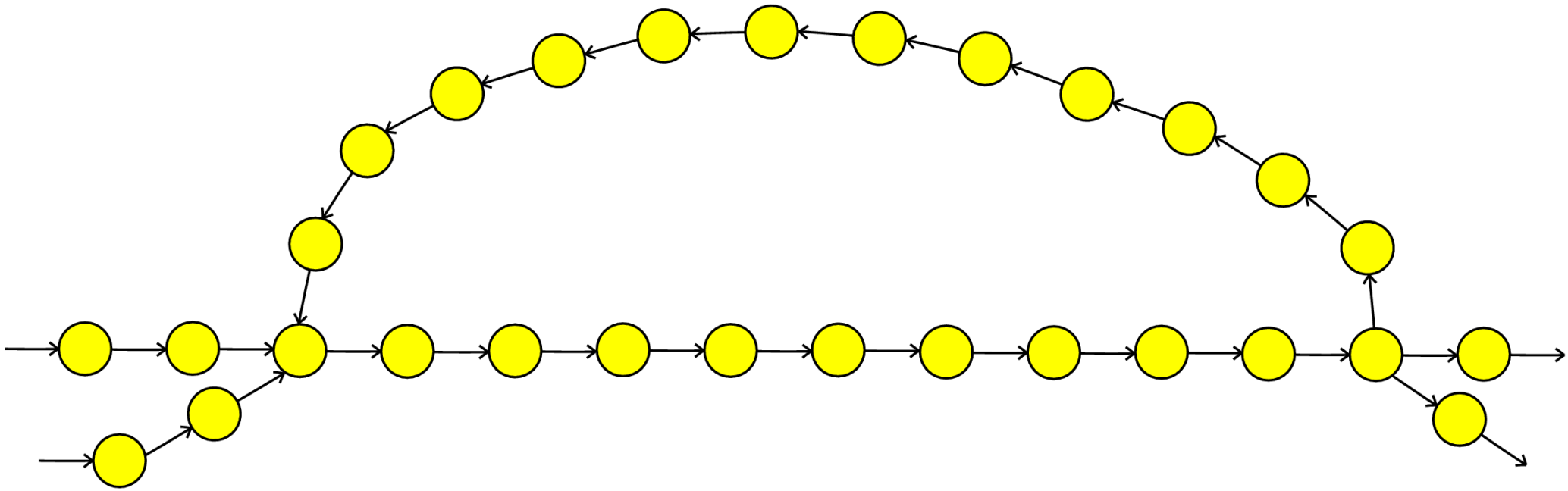
- Subgraph of the de Bruijn graph
- Genome: path in this graph
- Owing to repeats, genome is a set of paths

Read paths



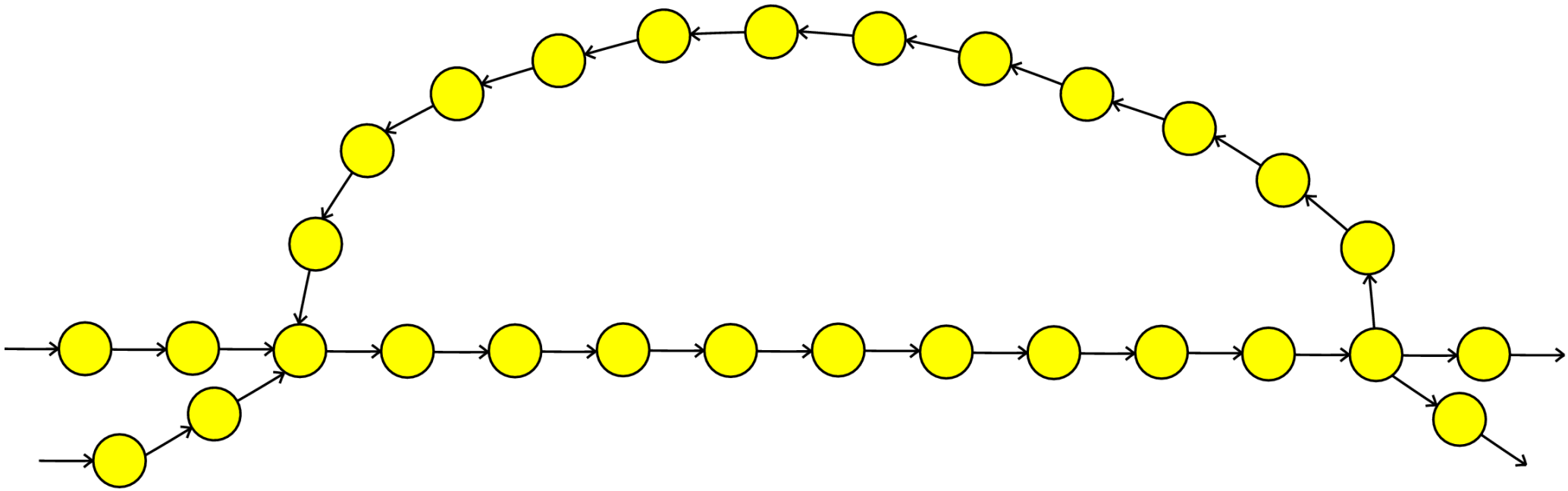
- A read is a path

A repeat in the graph



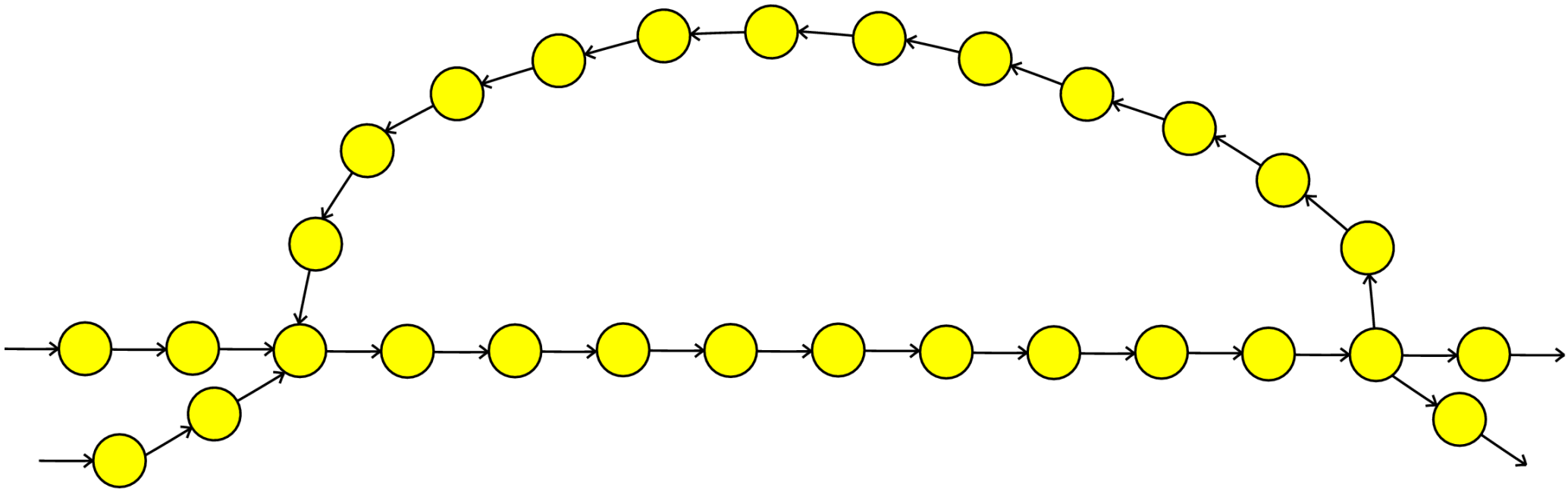
- 2 entry points & 2 exit points

A repeat in the graph



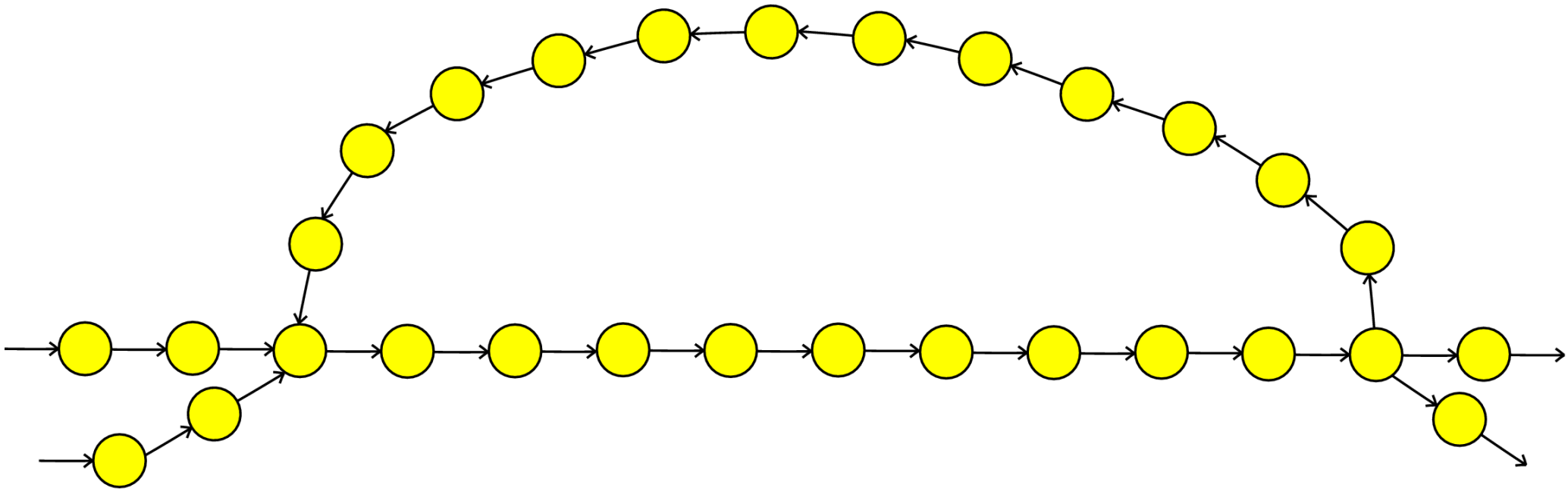
- 2 entry points & 2 exit points
- Matching entry & exit points

A repeat in the graph



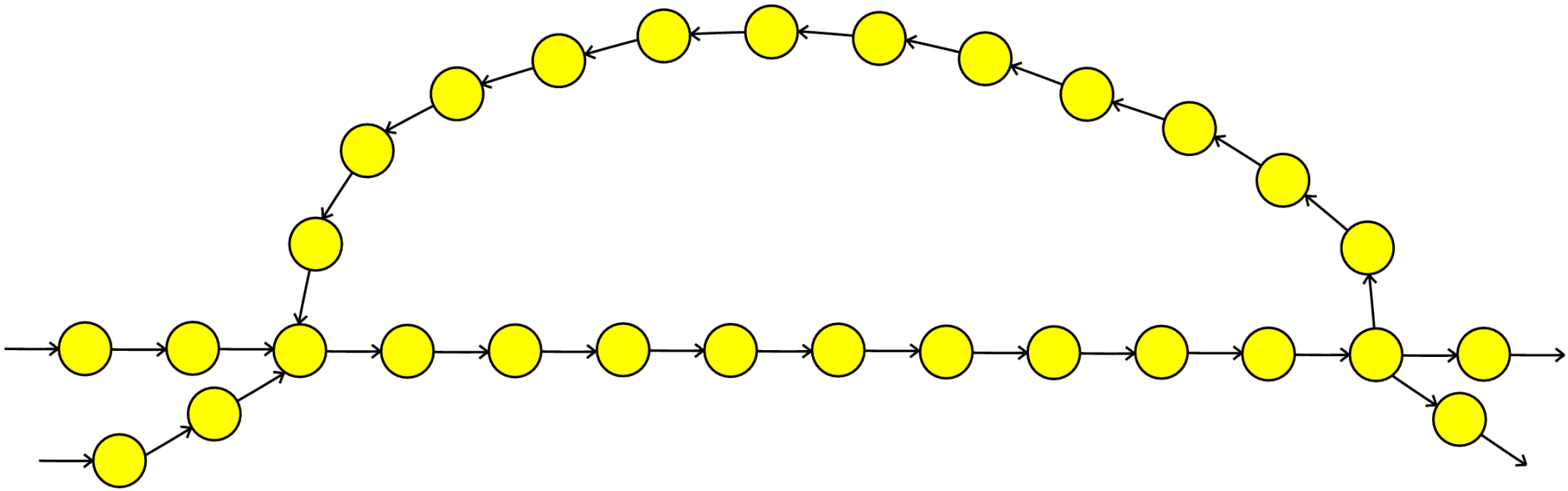
- 2 entry points & 2 exit points
- Matching entry & exit points
- How many copies ?

A repeat in the graph



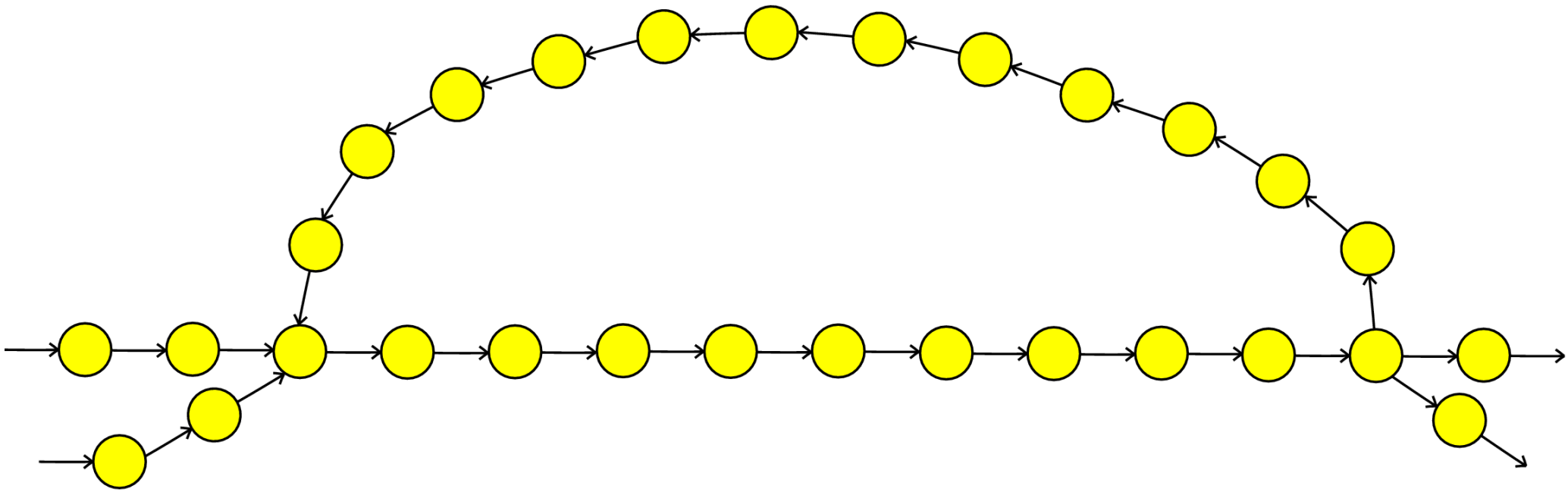
- 2 entry points & 2 exit points
- Matching entry & exit points
- How many copies ?
- Collapsing (example: 1 copy instead of 5)

A repeat in the graph



- 2 entry points & 2 exit points
- Matching entry & exit points
- How many copies ?
- Collapsing (example: 1 copy instead of 5)
- Wrongly matched points are far away in genome

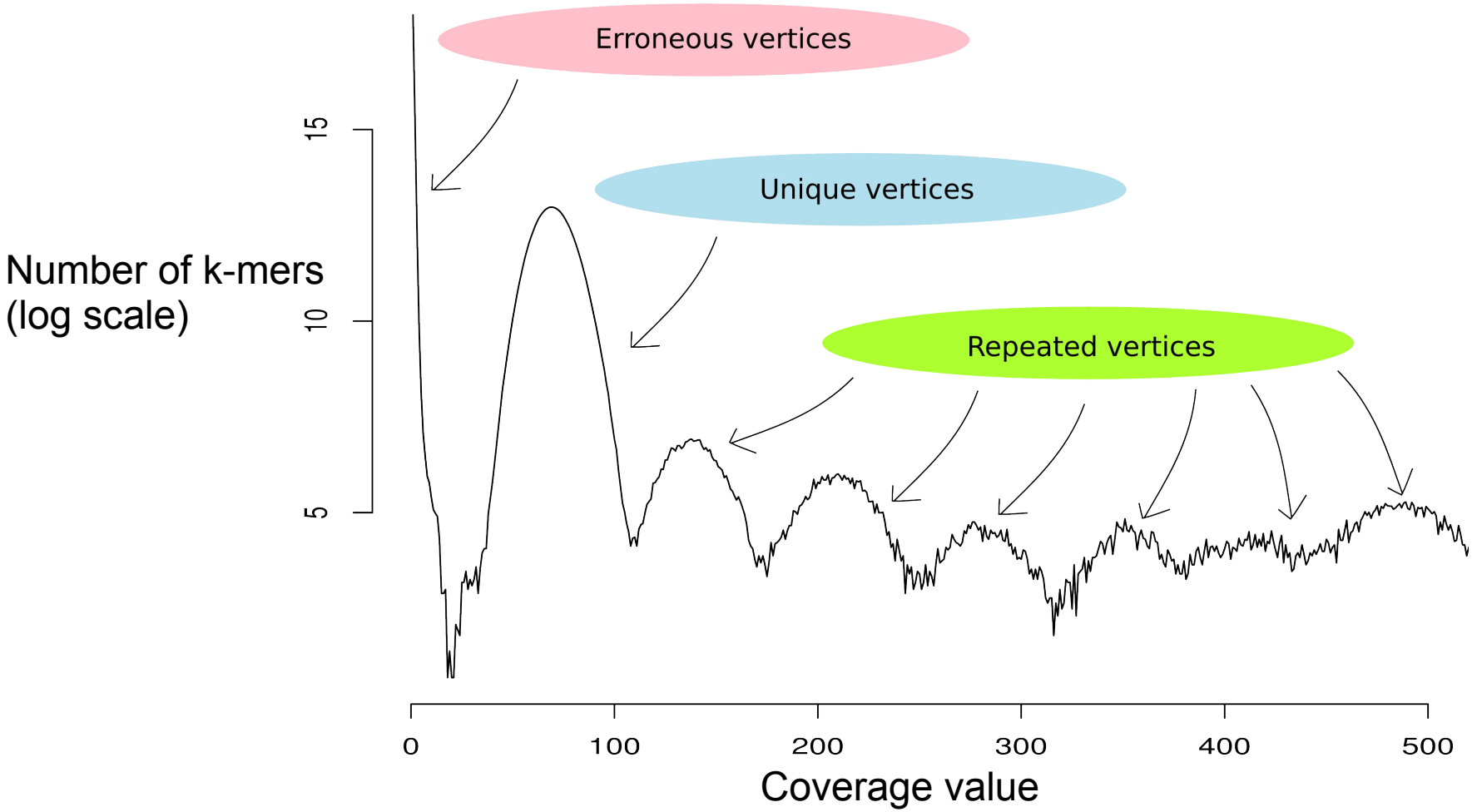
A repeat in the graph



- 2 entry points & 2 exit points
- Matching entry & exit points
- How many copies ?
- Collapsing (example: 1 copy instead of 5)
- Wrongly matched points are far away in genome
- Bad matching = large misassemblies

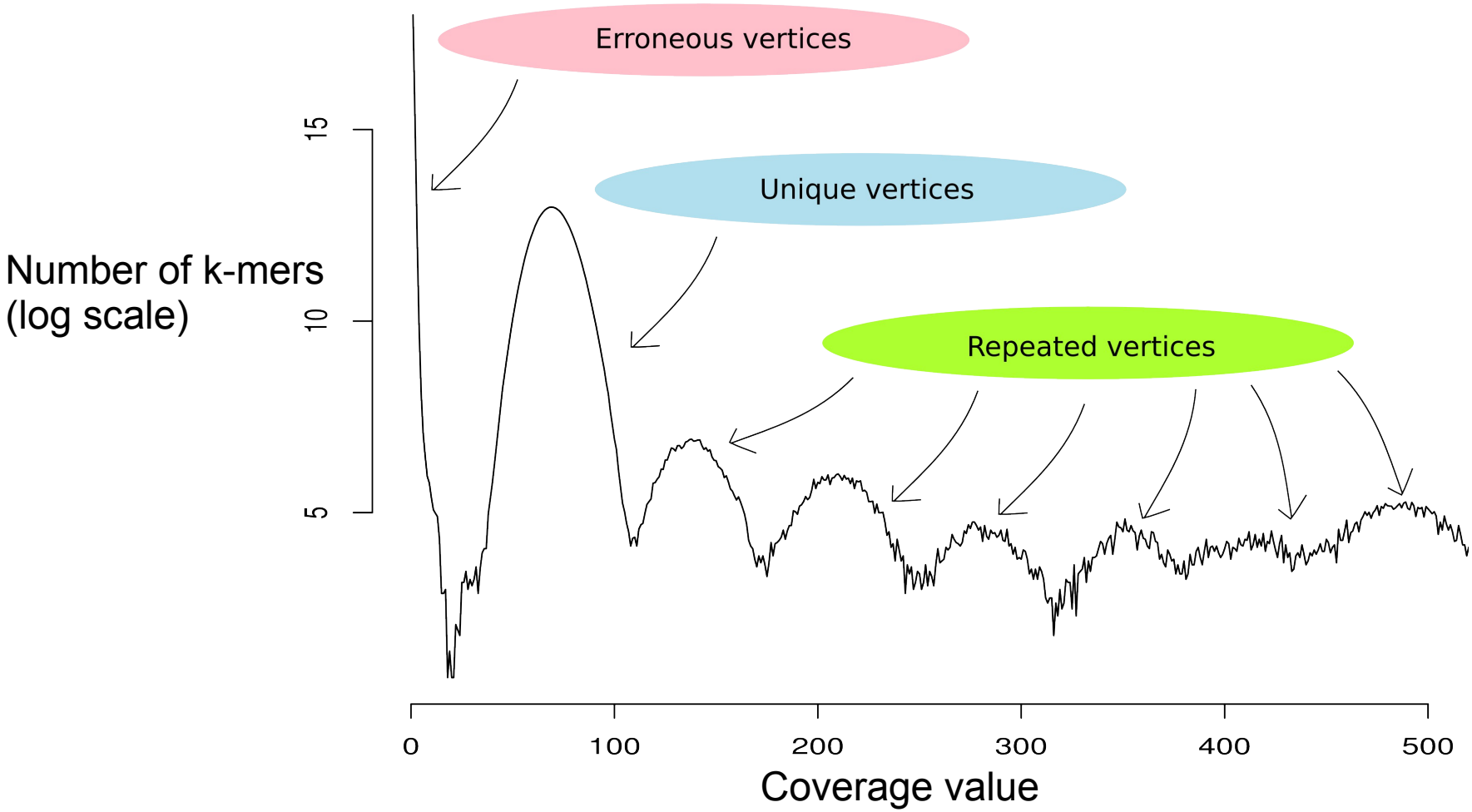
Coverage distribution: assessing k-mer redundancy

a



Coverage distribution: assessing k-mer redundancy

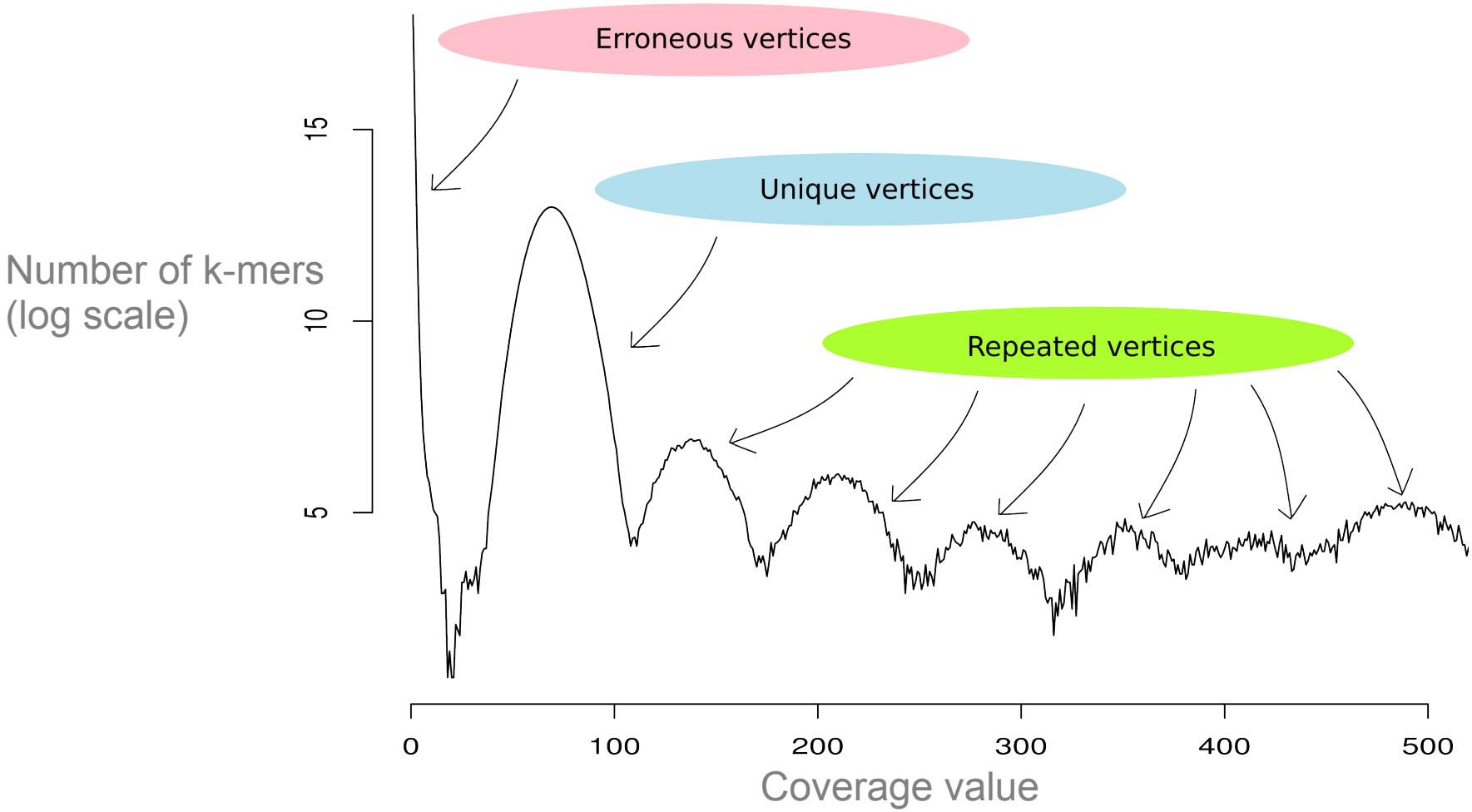
a



➔ Numerous errors, not redundant; many are there only once

Coverage distribution: assessing k-mer redundancy

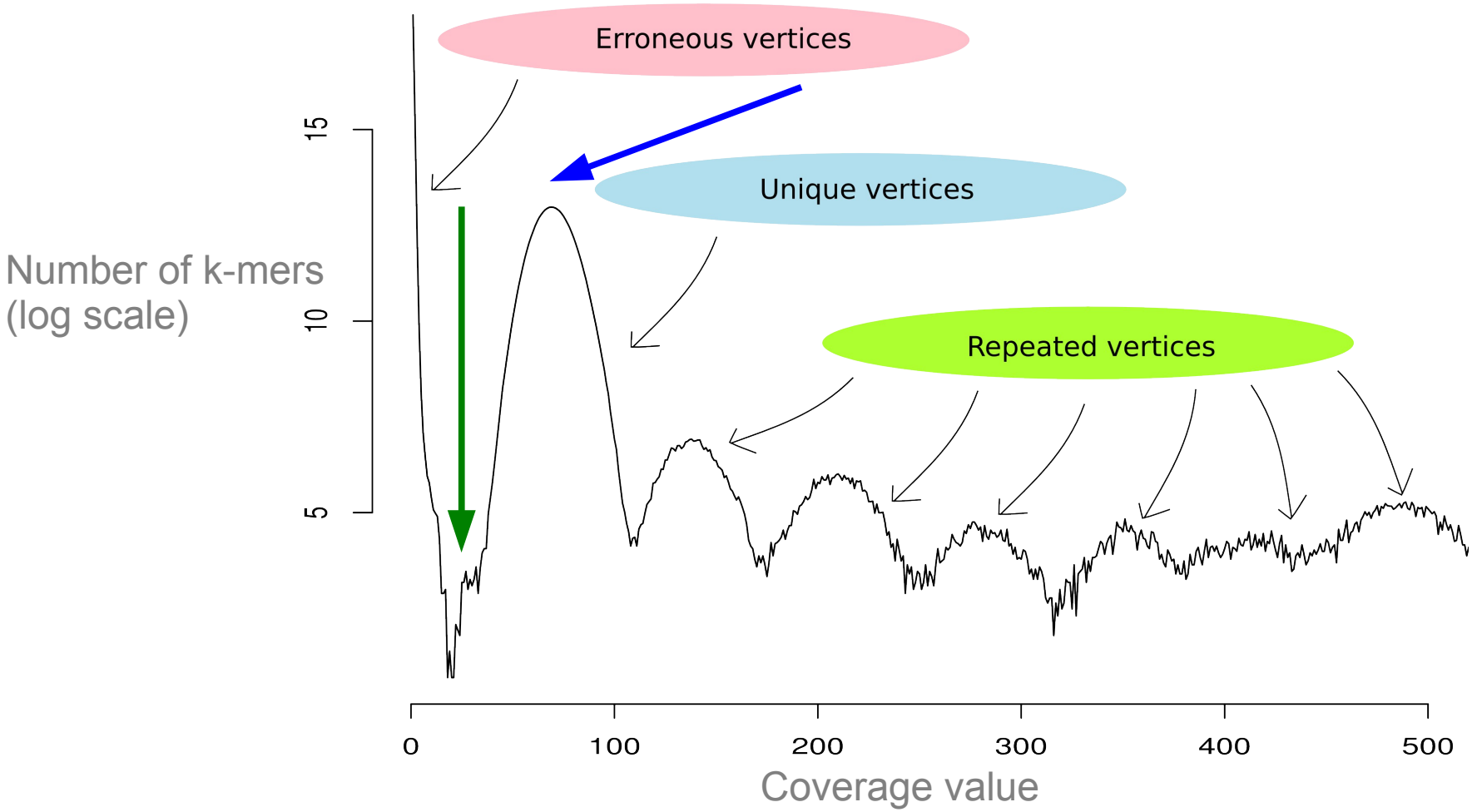
a



- Numerous errors, not redundant; many are there only once
- Redundant genome k-mers

Coverage distribution: assessing k-mer redundancy

a



- Numerous errors, not redundant; many are there only once
- Redundant genome k-mers
- From example: **minimum coverage= 15**, **peak coverage= 69**

Initial aim

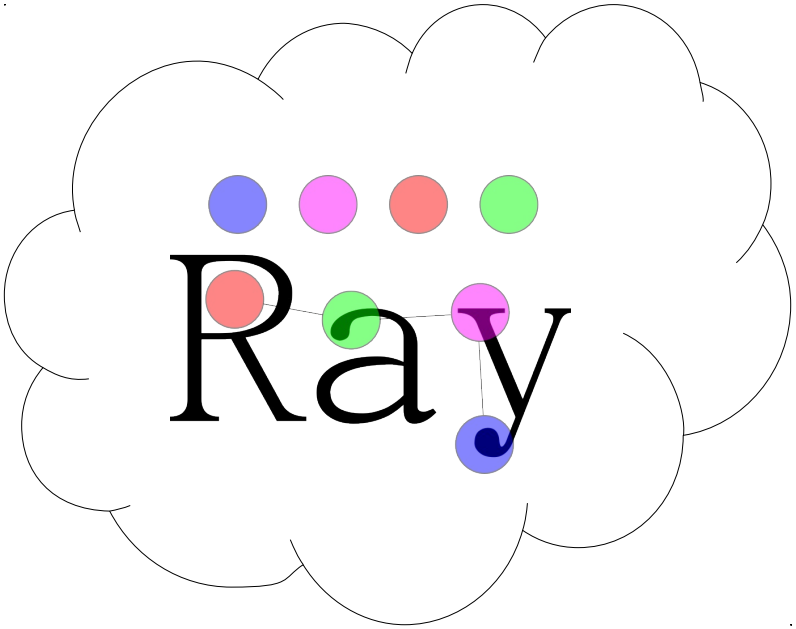
➔ Develop a powerful, parallel, *de novo* assembler able to handle reads from a mix of technologies

Initial aim

➔ Develop a powerful, parallel, *de novo* assembler able to handle reads from a mix of technologies

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 17, Number 11, 2010

Pp. 1519–1533
DOI: 10.1089/cmb.2009.0238



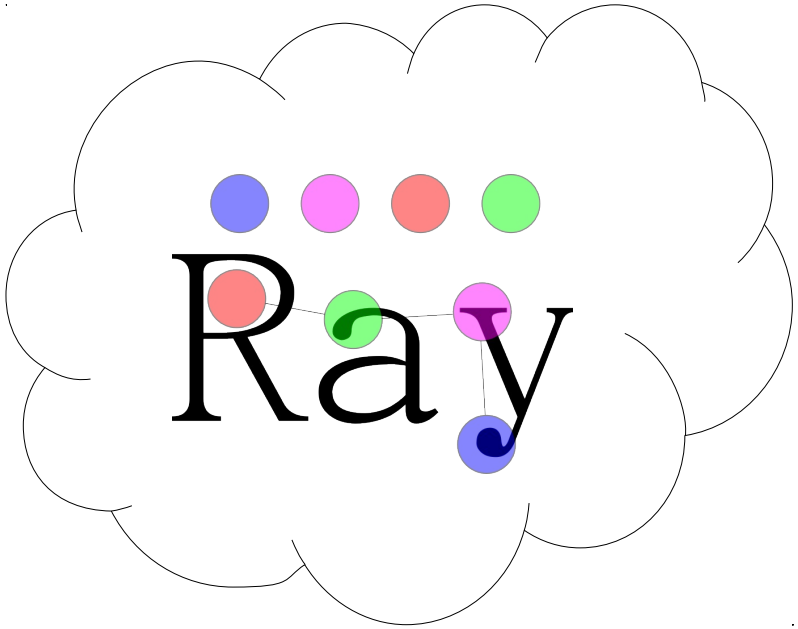
Ray: Simultaneous Assembly of Reads from a Mix of High-Throughput Sequencing Technologies

SÉBASTIEN BOISVERT,[1,2] FRANÇOIS LAVIOLETTE,[3] and JACQUES CORBEIL[1,2]

http://dx.doi.org/doi:10.1089/cmb.2009.0238

Other aims

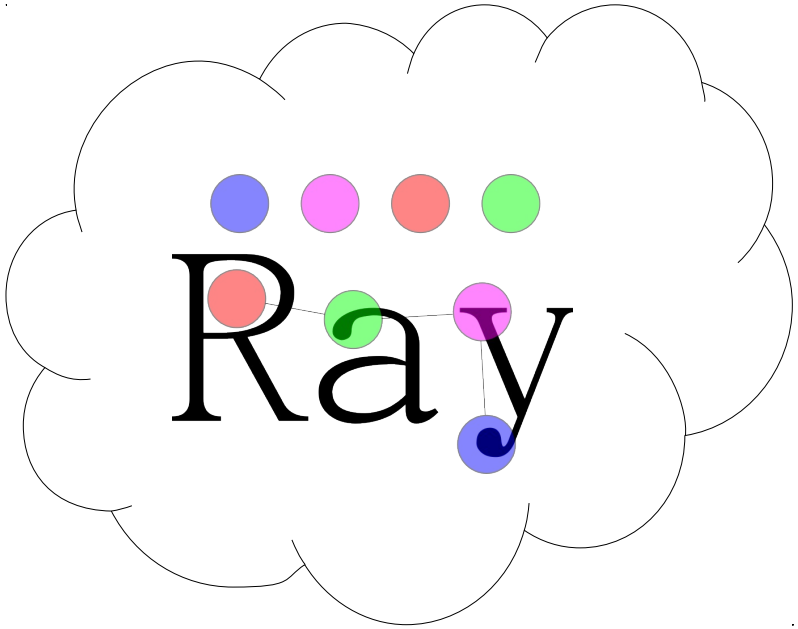
- Improve our assembler to traverse repeated regions more efficiently



Other aims

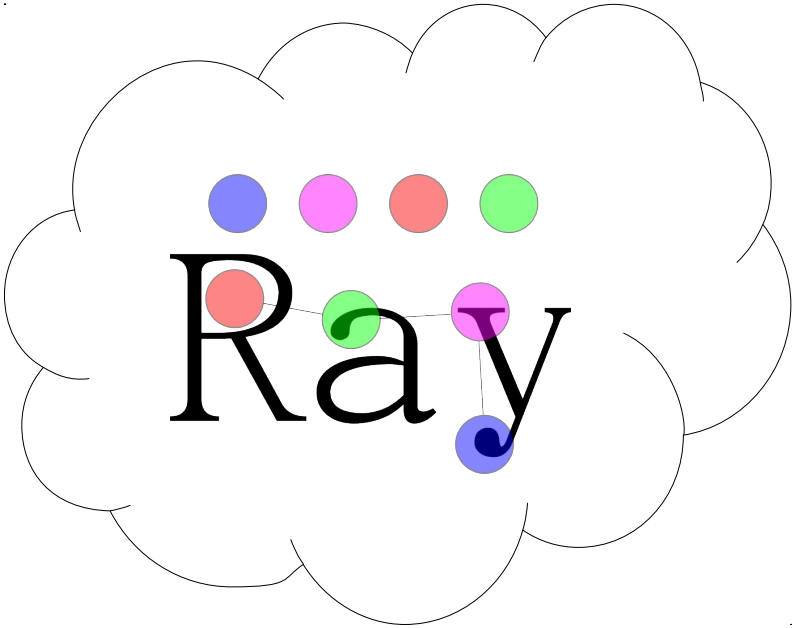
- Improve our assembler to traverse repeated regions more efficiently

- Assemble large genomes: *de novo* assembly of Illumina CEO genome in 11.5 h on 512 processors with Ray (Supplementary slides)



Ray

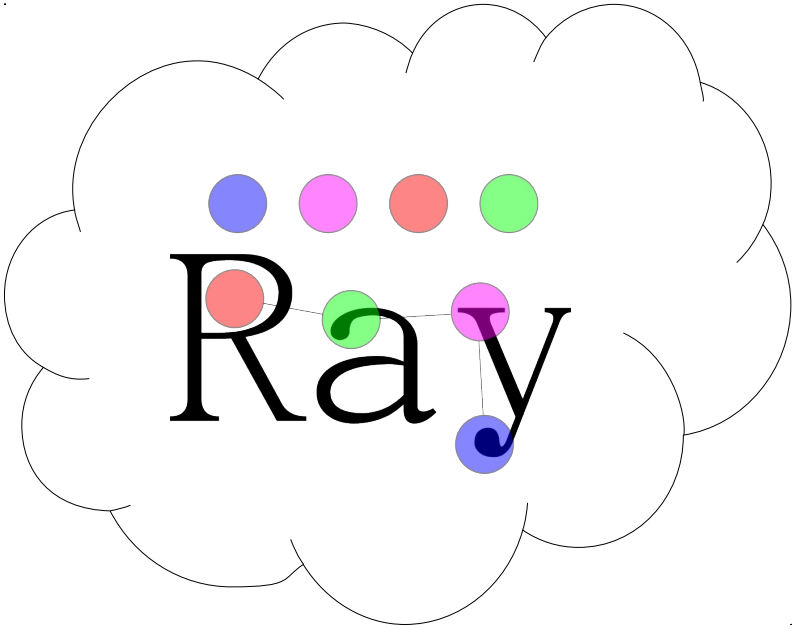
- Computes seeds in the graph



Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies.
Boisvert, Sébastien and Laviolette, François and Corbeil, Jacques
Journal of Computational Biology, 2010, http://dx.doi.org/doi:10.1089/cmb.2009.0238

Ray

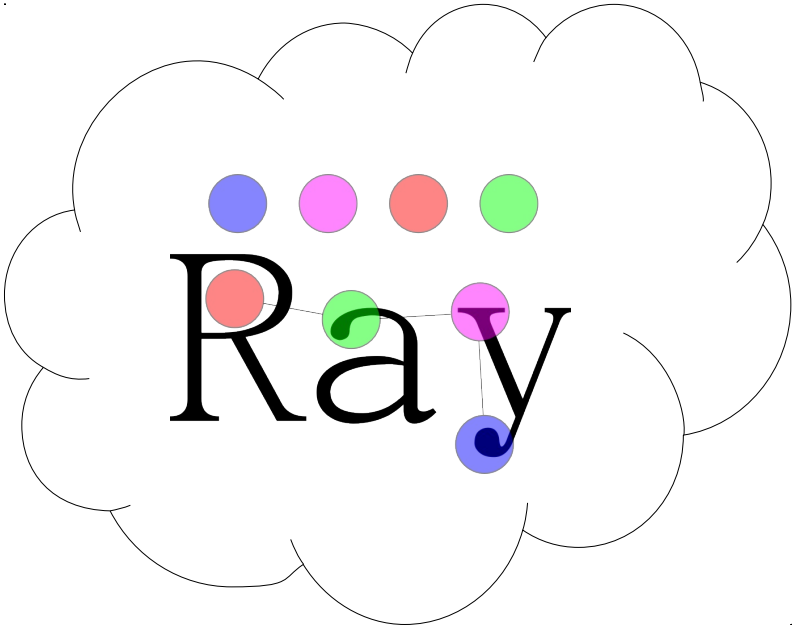
- Computes seeds in the graph
- Extends them using pairs of sequences



Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies.
Boisvert, Sébastien and Laviolette, François and Corbeil, Jacques
Journal of Computational Biology, 2010, http://dx.doi.org/doi:10.1089/cmb.2009.0238

Ray

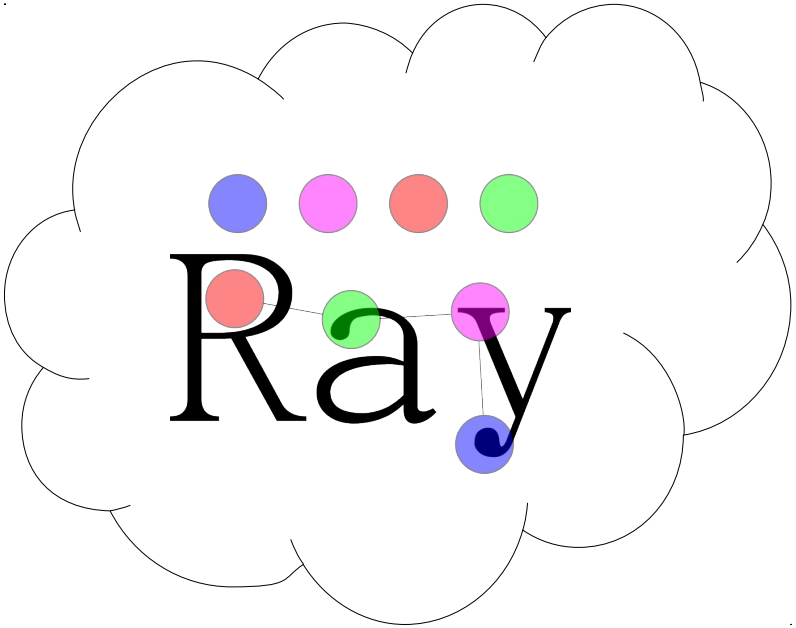
- Computes seeds in the graph
- Extends them using pairs of sequences
- Max. 4 choices for extension: A, T, C, or G
- Selection is heuristic-based



Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies.
Boisvert, Sébastien and Laviolette, François and Corbeil, Jacques
Journal of Computational Biology, 2010, http://dx.doi.org/doi:10.1089/cmb.2009.0238

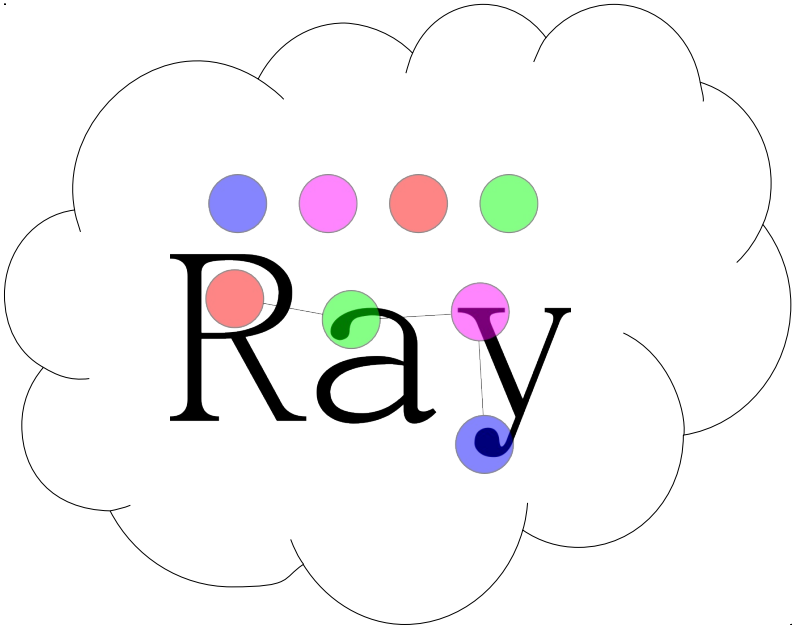
Ray

- Computes seeds in the graph
- Extends them using pairs of sequences
- Max. 4 choices for extension: A, T, C, or G
- Selection is heuristic-based
- Assembly result: no Ns – only As, Ts, Cs and Gs



Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies.
Boisvert, Sébastien and Laviolette, François and Corbeil, Jacques
Journal of Computational Biology, 2010, http://dx.doi.org/doi:10.1089/cmb.2009.0238

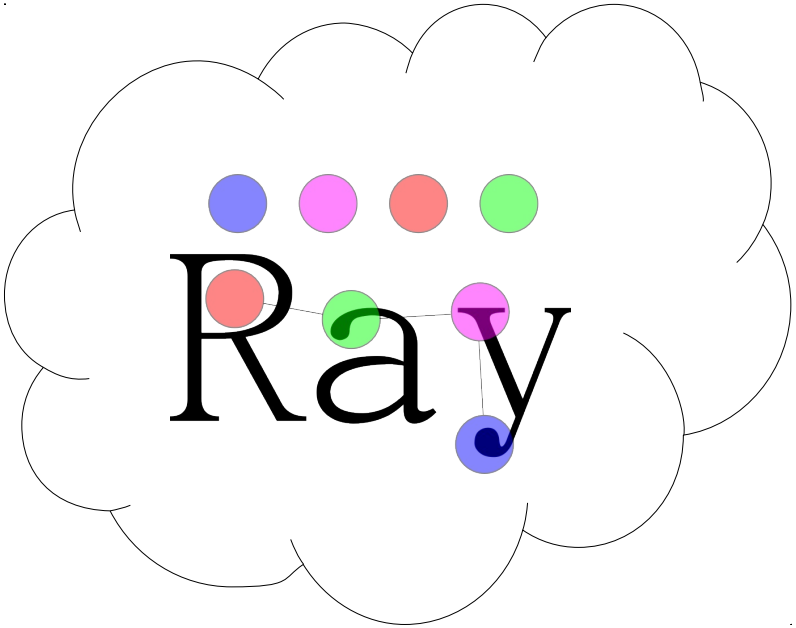
Ray



- Computes seeds in the graph
- Extends them using pairs of sequences
- Max. 4 choices for extension: A, T, C, or G
- Selection is heuristic-based
- Assembly result: no Ns – only As, Ts, Cs and Gs
- Message-Passing Interface (MPI); open system http://tiny.cc/ray-assembler

Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies.
Boisvert, Sébastien and Laviolette, François and Corbeil, Jacques
Journal of Computational Biology, 2010, http://dx.doi.org/doi:10.1089/cmb.2009.0238

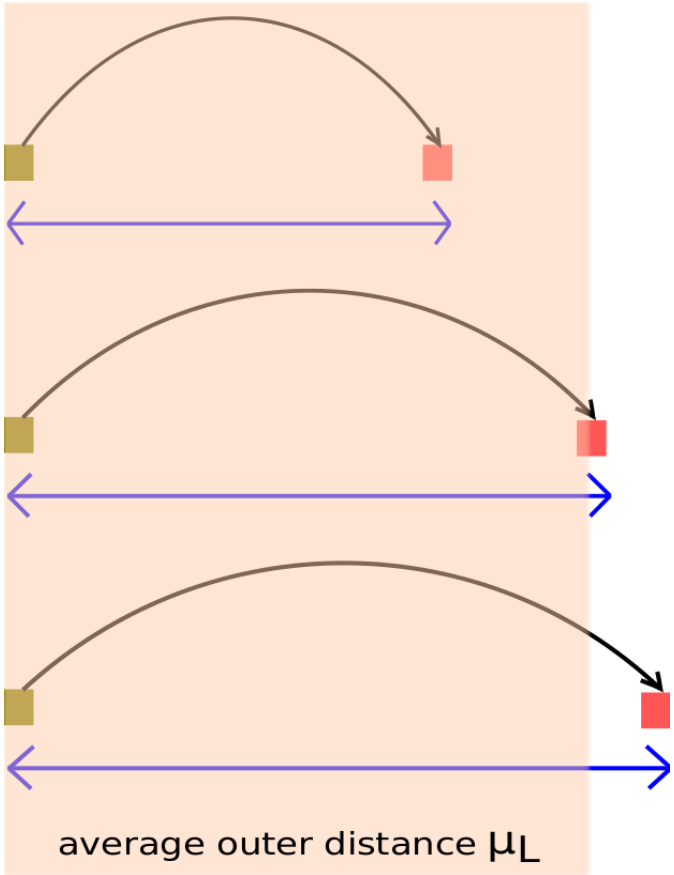
Ray



- Computes seeds in the graph
- Extends them using pairs of sequences
- Max. 4 choices for extension: A, T, C, or G
- Selection is heuristic-based
- Assembly result: no Ns – only As, Ts, Cs and Gs
- Message-Passing Interface (MPI); open system http://tiny.cc/ray-assembler
- Cloud-ready (needs MPI + fast interconnect)

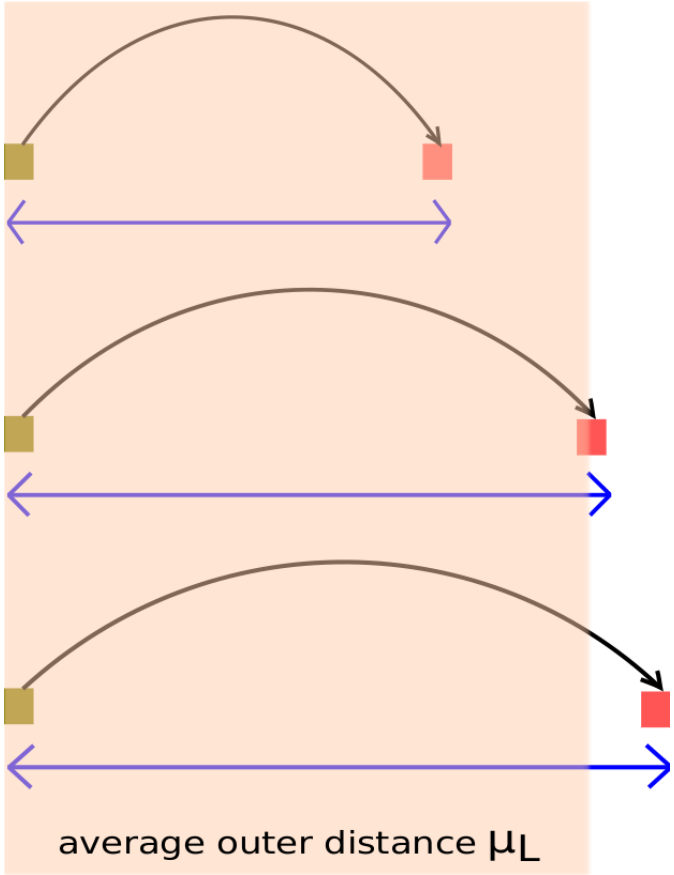
Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies.
Boisvert, Sébastien and Laviolette, François and Corbeil, Jacques
Journal of Computational Biology, 2010, http://dx.doi.org/doi:10.1089/cmb.2009.0238

Paired reads



- For any paired library L:
- average outer distance μ_L
- standard deviation σ_L

Paired reads

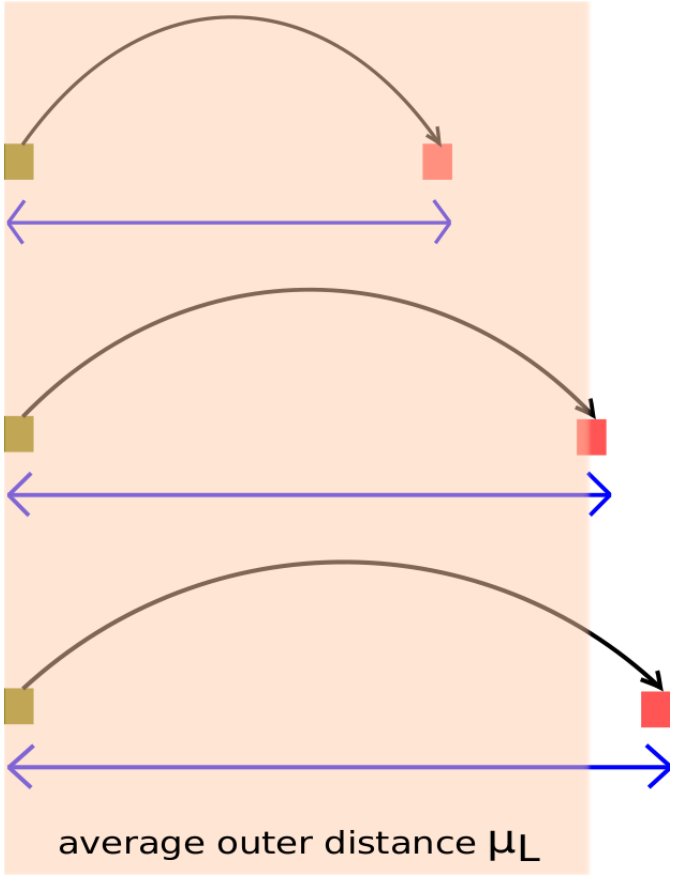


- Chaisson et al. (2009) transform a pair in meta-read

- For any paired library L:
- average outer distance μ_L
- standard deviation σ_L

De novo fragment assembly with short mate-paired reads: Does the read length matter?
Chaisson, Mark J. and Brinza, Dumitru and Pevzner, Pavel A.
Genome Research, 2009
http://dx.doi.org/doi:10.1101/gr.079053.108

Paired reads



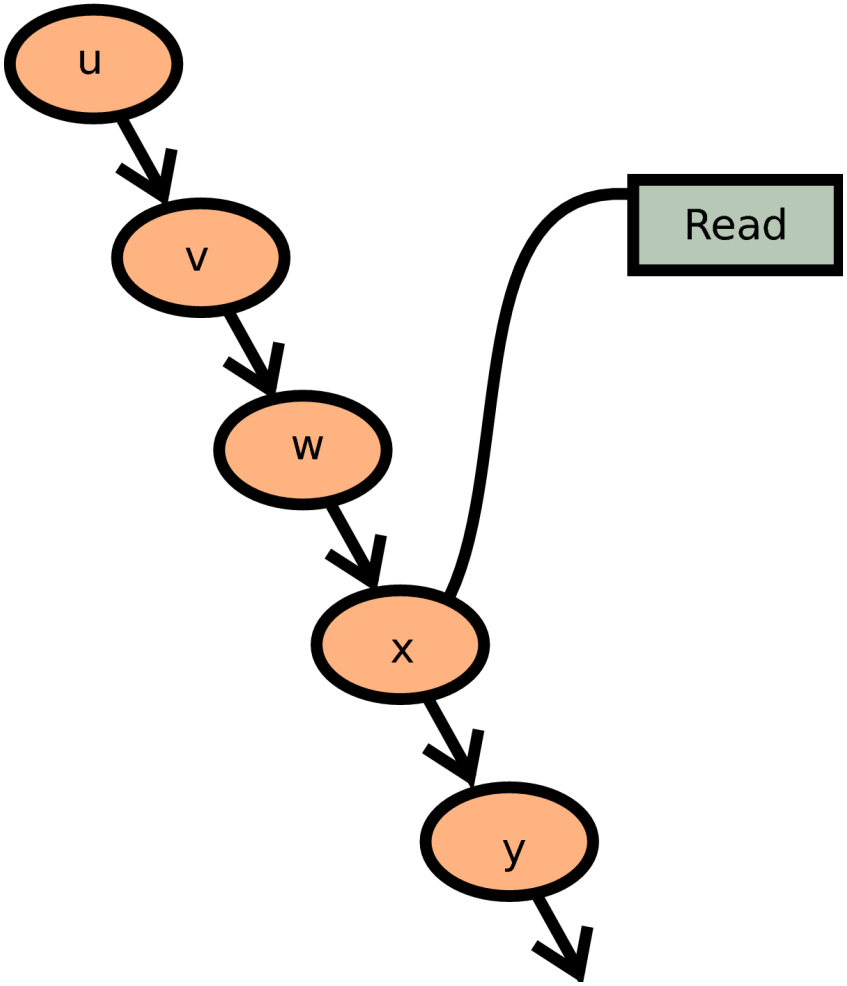
- Chaisson et al. (2009) transform a pair in meta-read
- in Ray: a pair is untransformed, used in synergy with other pairs

- For any paired library L:
- average outer distance μ_L
- standard deviation σ_L

De novo fragment assembly with short mate-paired reads: Does the read length matter?
Chaisson, Mark J. and Brinza, Dumitru and Pevzner, Pavel A.
Genome Research, 2009
http://dx.doi.org/doi:10.1101/gr.079053.108

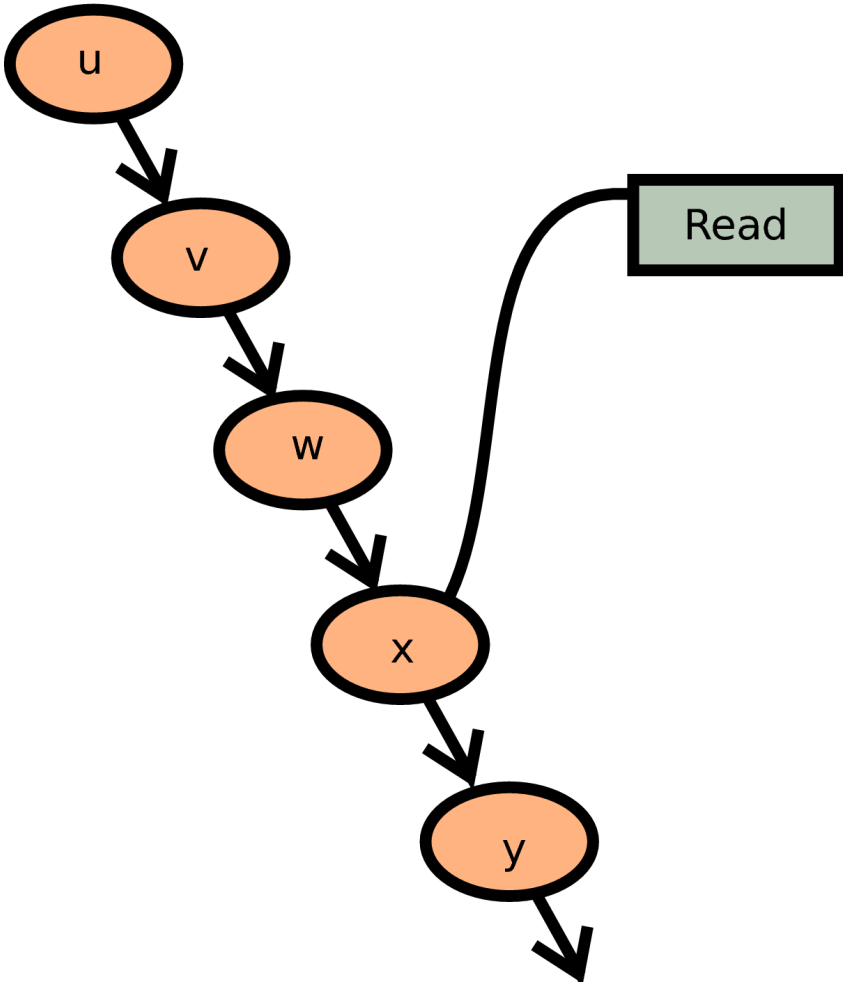
Read markers

- List of reads for a k-mer



Read markers

- List of reads for a k-mer
- Introduced in Daniel Zerbino's PhD thesis

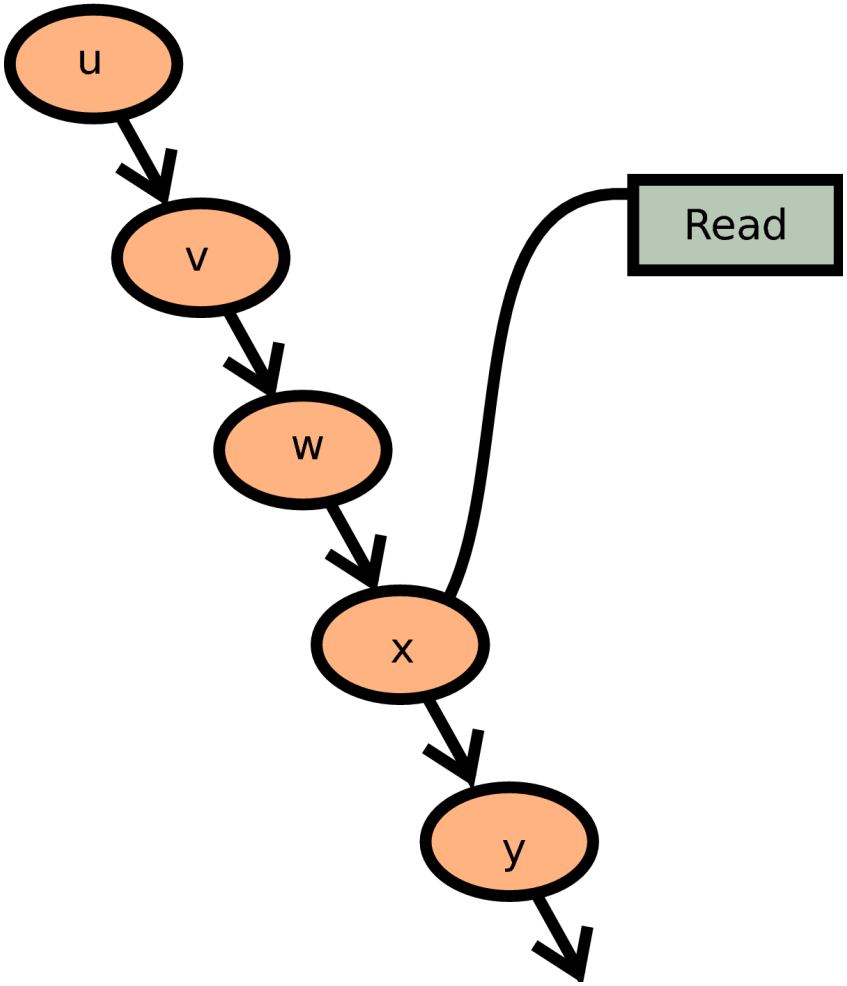


Genome assembly and comparison using de Bruijn graphs
Zerbino, Daniel R.
PhD thesis, University of Cambridge, 2009
http://www.ebi.ac.uk/training/ftp/PhDtheses/Daniel_Zerbino.pdf

Velvet: algorithms for de novo short read assembly using de Bruijn graphs.
Zerbino, Daniel R. and Birney, Ewan
Genome Research, 2008, doi:10.1101/gr.074492.107

Read markers

- List of reads for a k-mer
- Introduced in Daniel Zerbino's PhD thesis
- In Velvet, read offset is 0 (k-mer position in read)

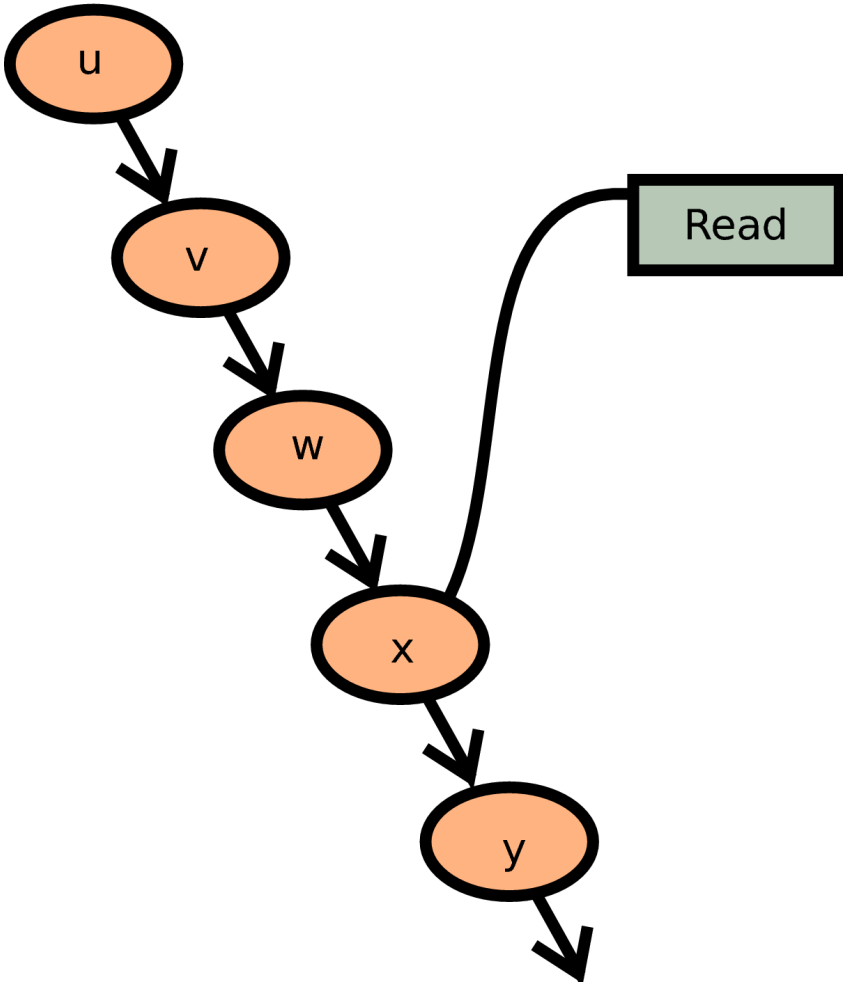


Genome assembly and comparison using de Bruijn graphs
Zerbino, Daniel R.
PhD thesis, University of Cambridge, 2009
http://www.ebi.ac.uk/training/ftp/PhDtheses/Daniel_Zerbino.pdf

Velvet: algorithms for de novo short read assembly using de Bruijn graphs.
Zerbino, Daniel R. and Birney, Ewan
Genome Research, 2008, doi:10.1101/gr.074492.107

Read markers

- List of reads for a k-mer
- Introduced in Daniel Zerbino's PhD thesis
- In Velvet, read offset is 0 (k-mer position in read)
- Not optimal, often a read starts on a repeated vertex
- but may also contains unique vertices



Genome assembly and comparison using de Bruijn graphs
Zerbino, Daniel R.
PhD thesis, University of Cambridge, 2009
http://www.ebi.ac.uk/training/ftp/PhDtheses/Daniel_Zerbino.pdf

Velvet: algorithms for de novo short read assembly using de Bruijn graphs.
Zerbino, Daniel R. and Birney, Ewan
Genome Research, 2008, doi:10.1101/gr.074492.107

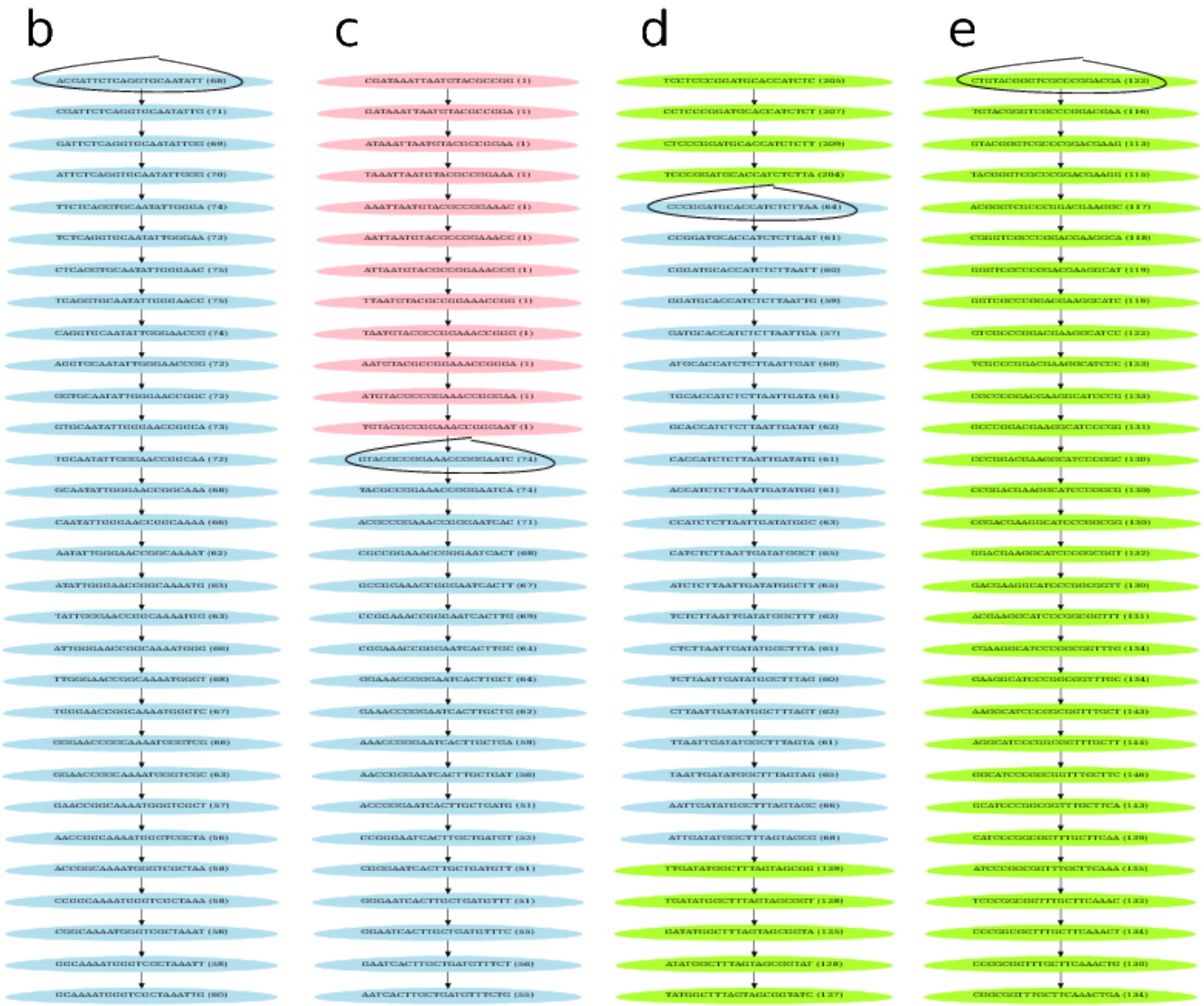
Optimal read markers

- Idea: read markers on non-erroneous non-repeated vertices are more useful

Optimal read markers

- Idea: read markers on non-erroneous non-repeated vertices are more useful

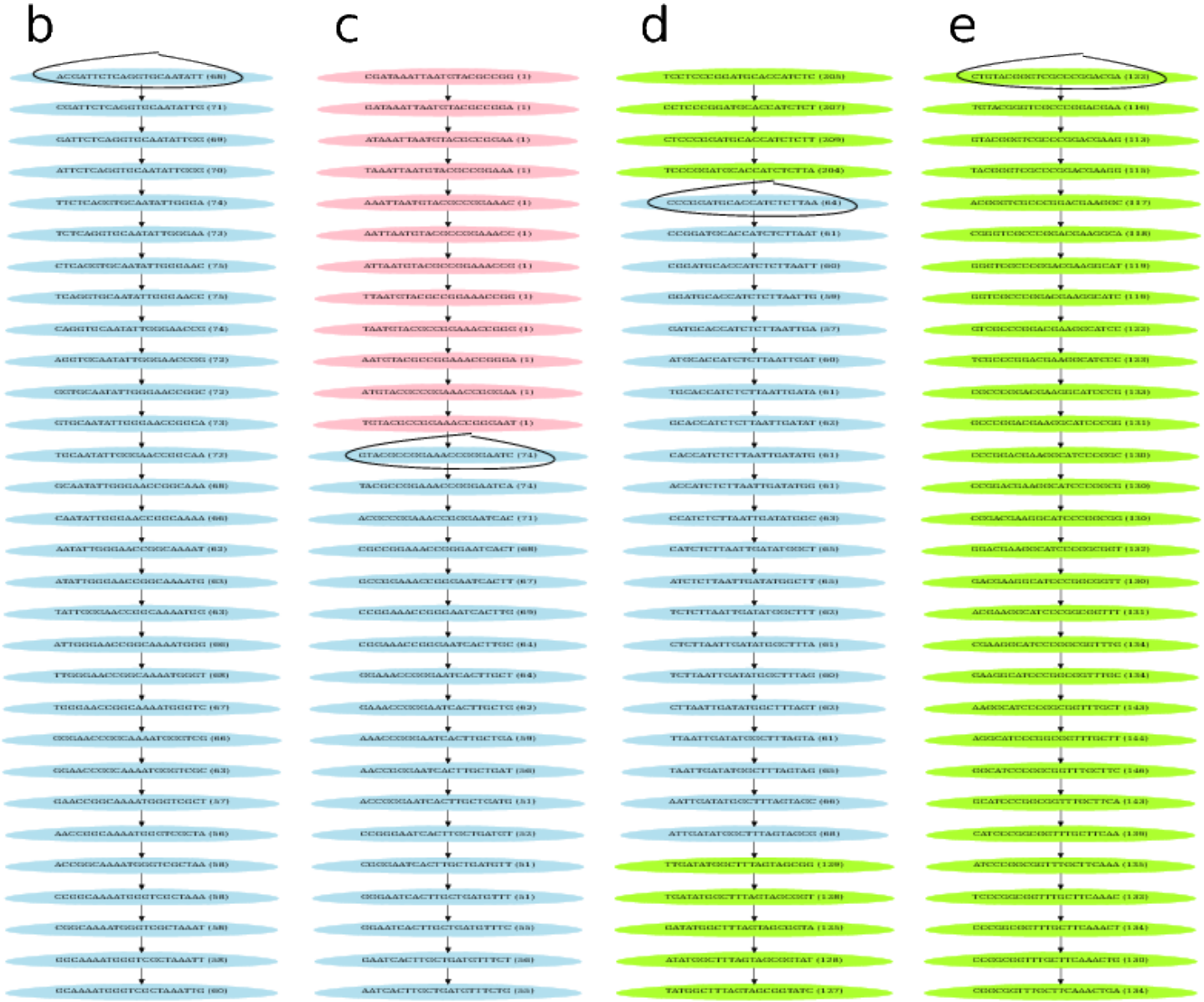
- 4 read path examples



Optimal read markers

- Idea: read markers on non-erroneous non-repeated vertices are more useful

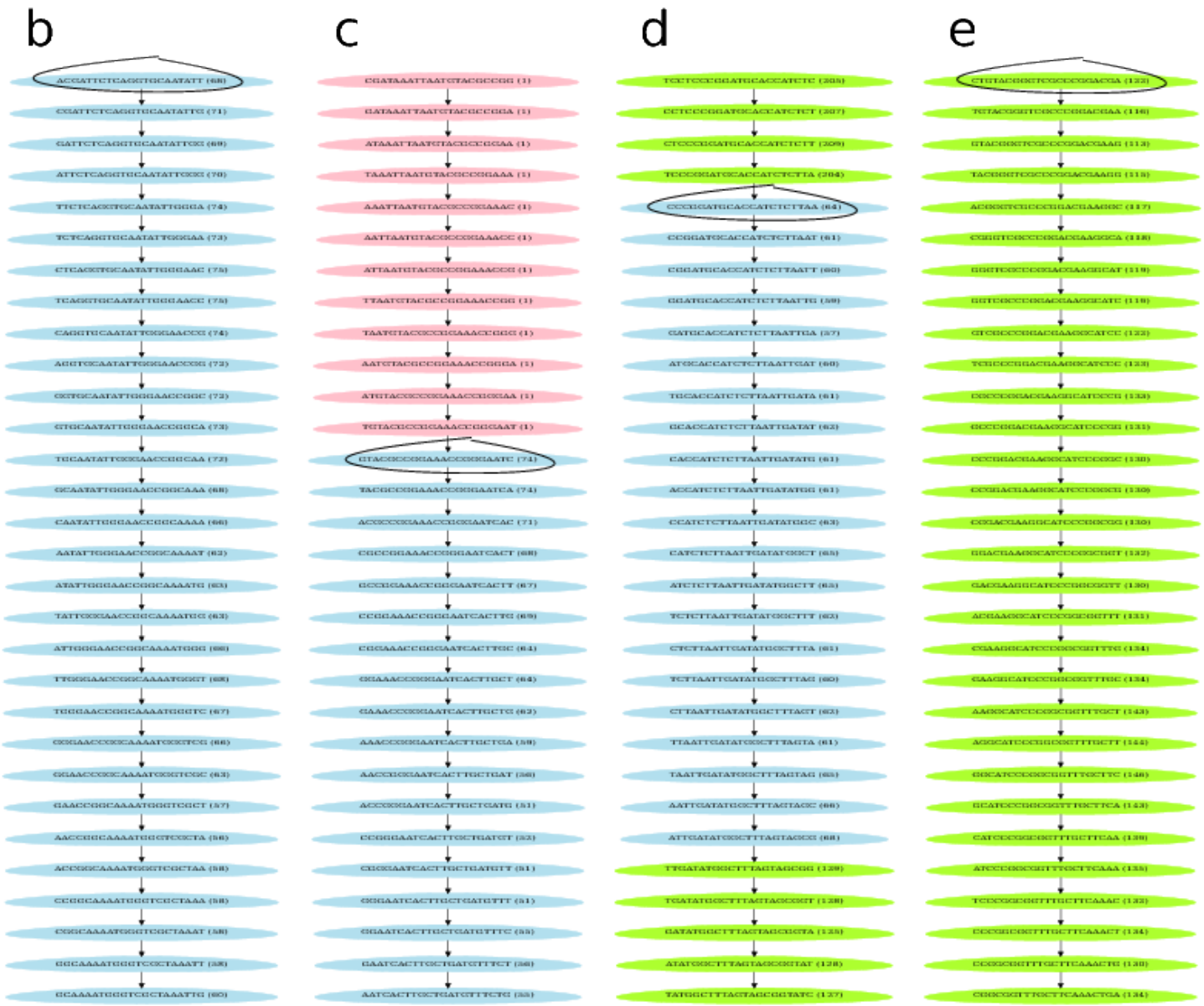
- 4 read path examples
- Blue: unique
- Pink: erroneous
- Green: repeated



Optimal read markers

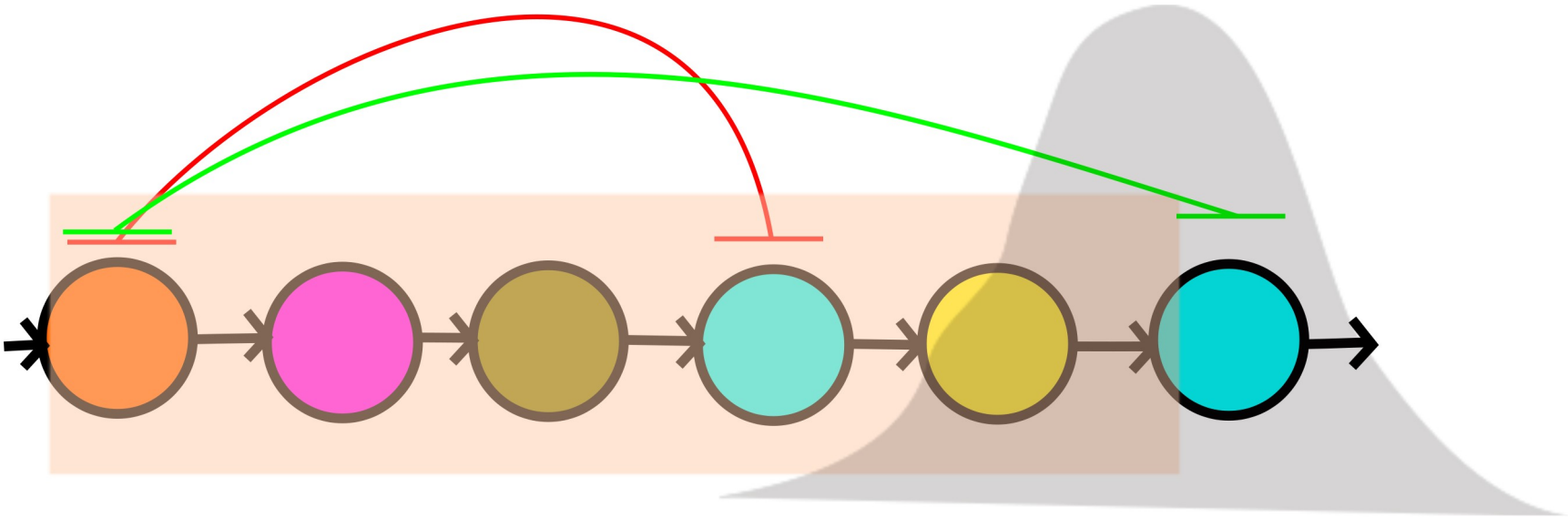
- Idea: read markers on non-erroneous non-repeated vertices are more useful

- 4 read path examples
- Blue: unique
- Pink: erroneous
- Green: repeated
- Circle: read markers



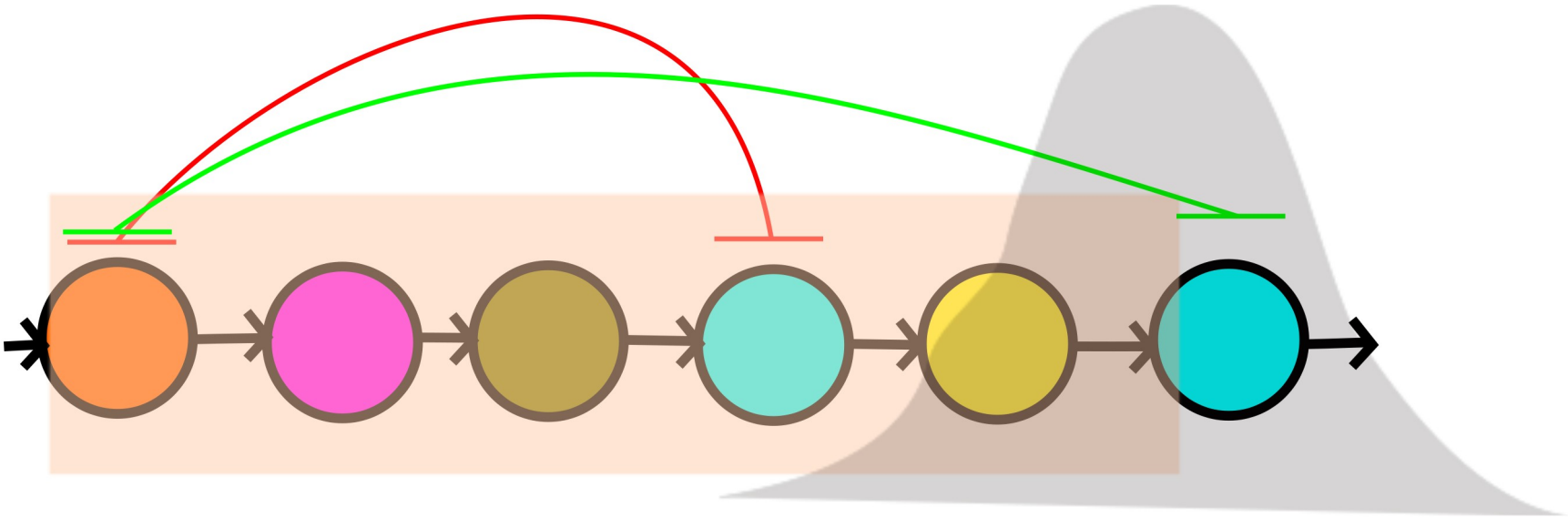
Constraint 1 -- acceptable outer distances

- Outer distance must be within 3 standard deviations from the average



Constraint 1 -- acceptable outer distances

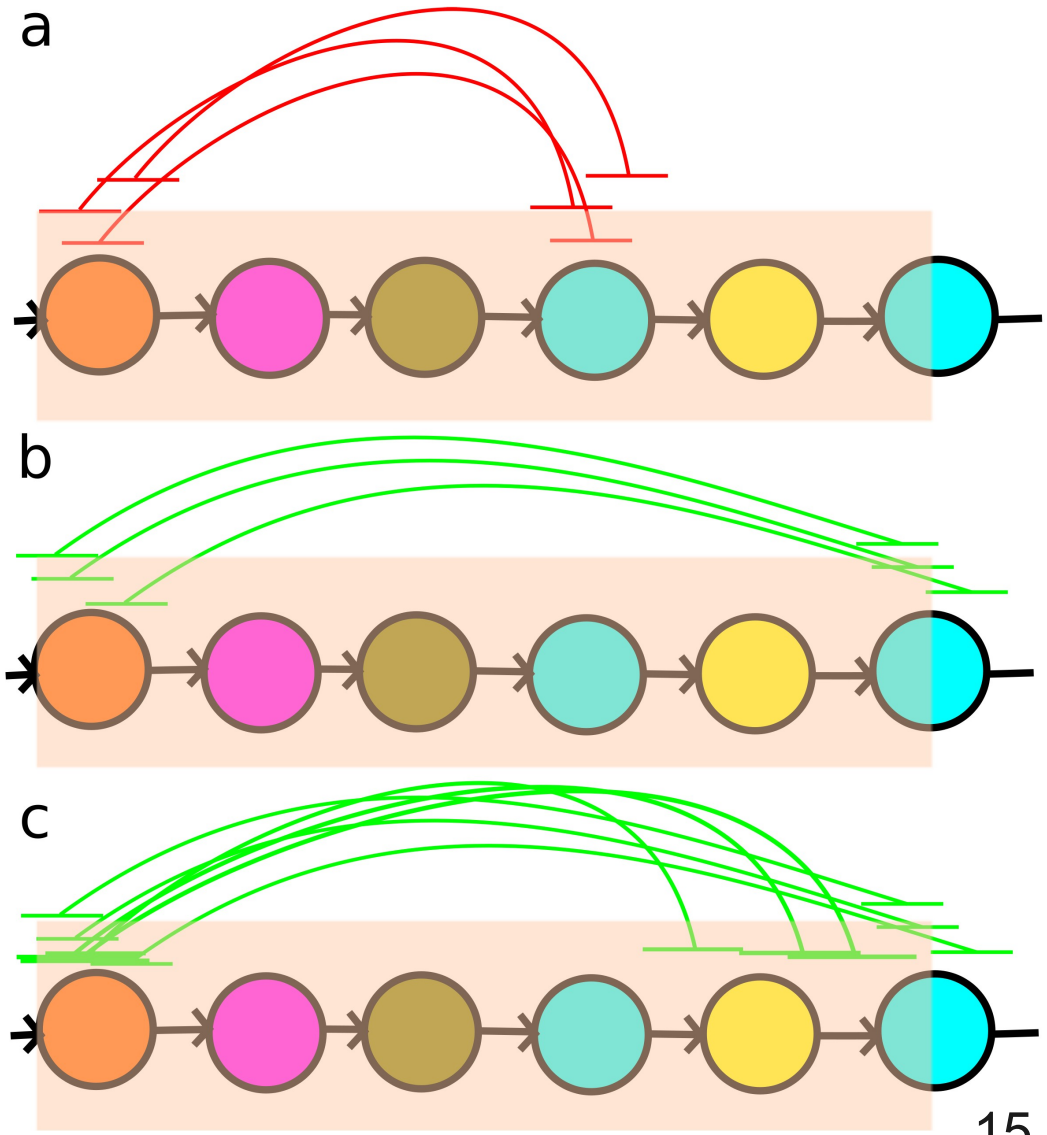
- Outer distance must be within 3 standard deviations from the average



- Indicates course of events

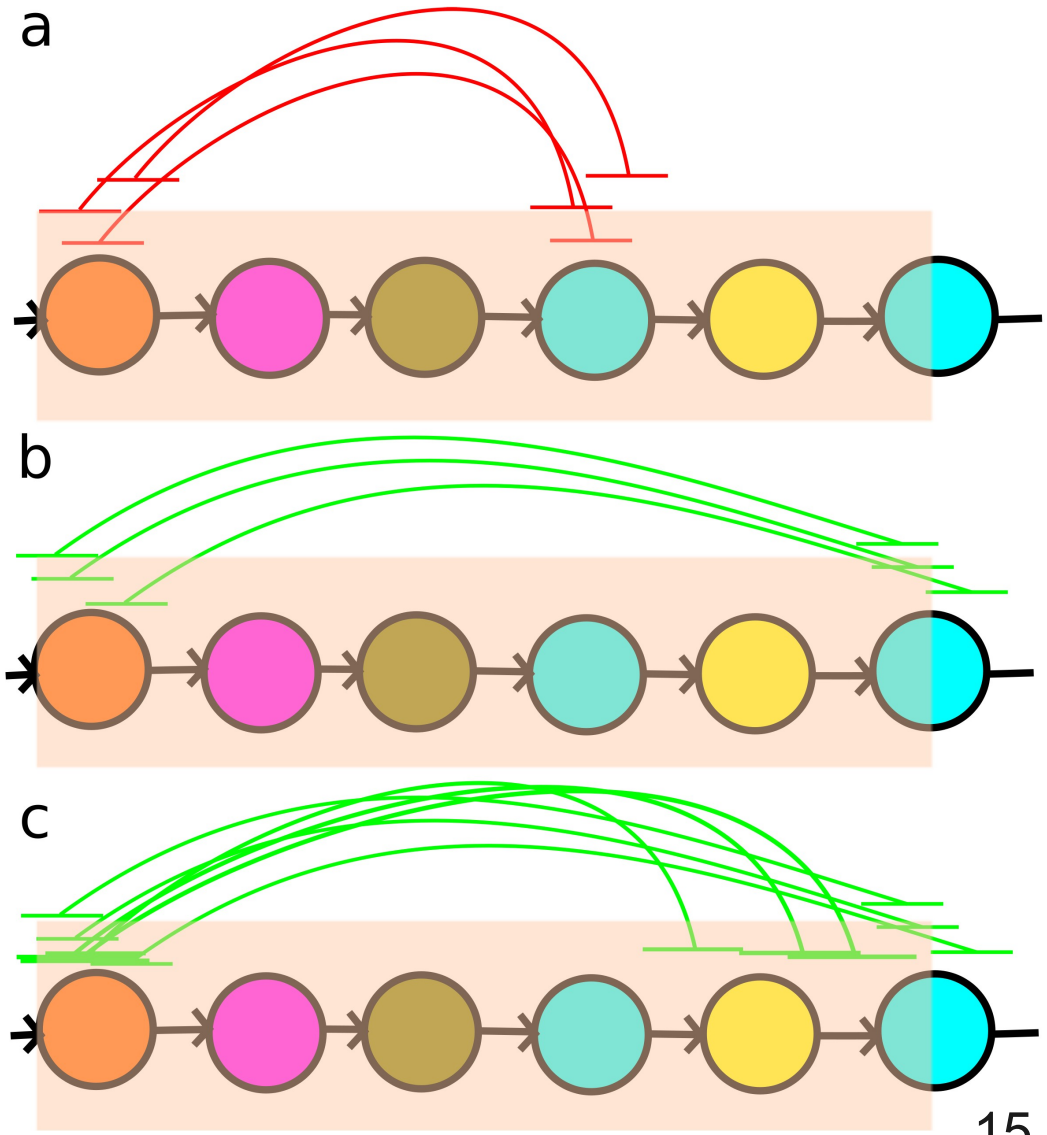
Constraint 2 -- local pair population v. genome-wide pair population

- The average outer distance of local pair population must be within 1 standard deviations from the genome-wide average outer distance



Constraint 2 -- local pair population v. genome-wide pair population

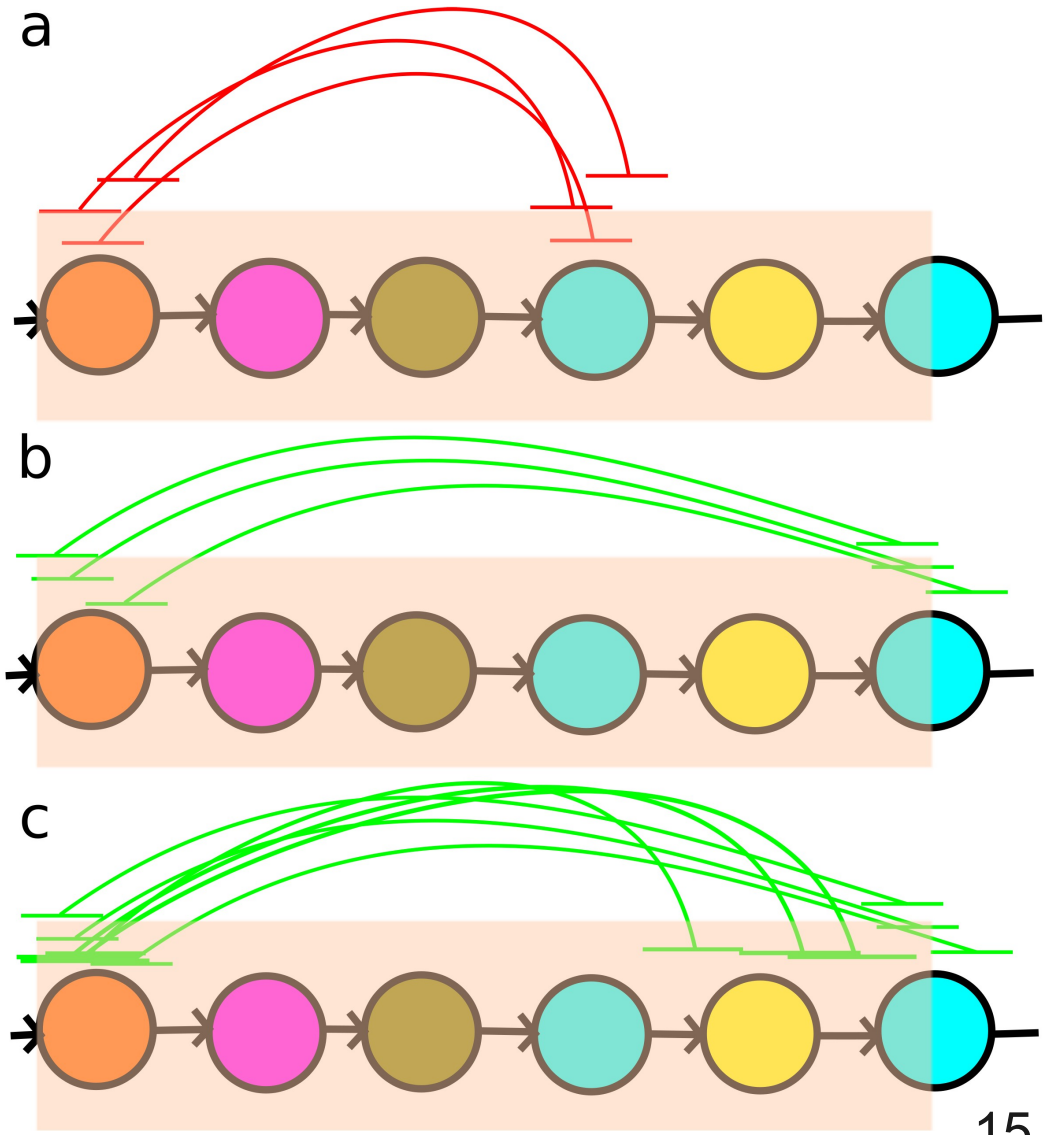
- The average outer distance of local pair population must be within 1 standard deviations from the genome-wide average outer distance



- Constraint on many pairs

Constraint 2 -- local pair population v. genome-wide pair population

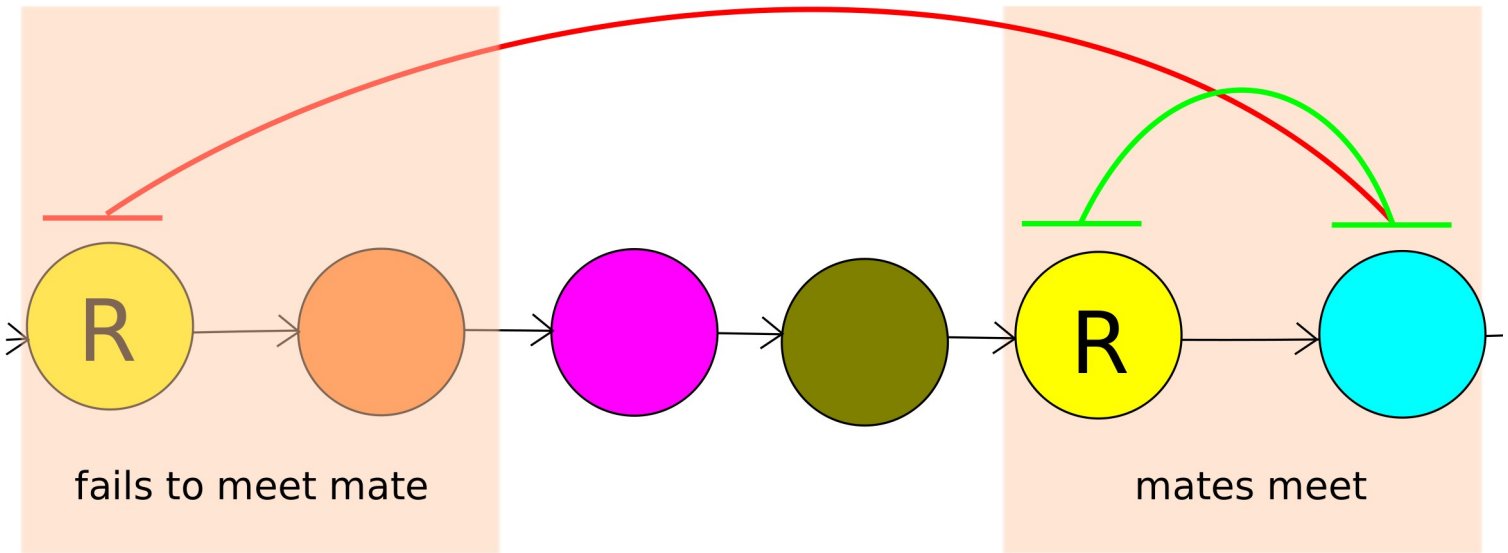
- The average outer distance of local pair population must be within 1 standard deviations from the genome-wide average outer distance



- Constraint on many pairs
- Avoids collapsing of repeats

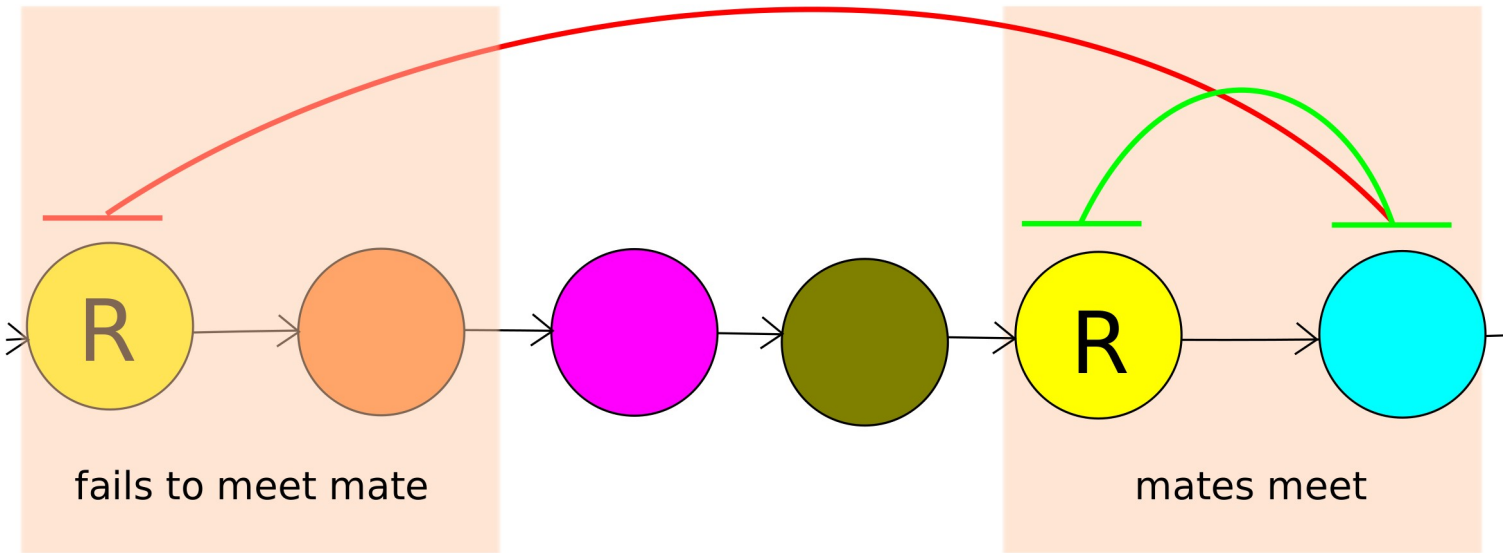
Constraint 3 -- mates meet

- A read that fails to meet its mate within the average + 3 standard deviations is set as unused



Constraint 3 -- mates meet

- A read that fails to meet its mate within the average + 3 standard deviations is set as unused



- A read can meet its mate

Experiment

Table 1 - Description of libraries

Library	Average outer distance	Standard deviation	Read length	Substitution error rate	Number of pairs
L1	200	20	50	0.5%	4000000
L2	1000	100	50	0.5%	4000000
L3	10000	1000	50	0.5%	4000000

→ 3 simulated paired libraries with *E. coli* genome: 200, 1000 & 10000

Experiment

Table 1 - Description of libraries

Library	Average outer distance	Standard deviation	Read length	Substitution error rate	Number of pairs
L1	200	20	50	0.5%	4000000
L2	1000	100	50	0.5%	4000000
L3	10000	1000	50	0.5%	4000000

- 3 simulated paired libraries with *E. coli* genome: 200, 1000 & 10000
- Standard deviation: 10 %; short read length; sequencing errors

Assemblies

Table 2 - Assembly validations

Libraries	Number of contigs	Number of nucleotides	Average contig length	N50 contig length	Maximum contig length	Genome breadth coverage	Large indels	Sub-stitutions	Small indels
L1	108	4559977	42222	87242	269942	98.20%	0	0	1
L1-L2	82	4621035	56354	96588	232618	99.13%	0	5	0
L1-L3	43	4618437	107405	177402	409463	99.59%	0	1	0

Assemblies

Table 2 - Assembly validations

Libraries	Number of contigs	Number of nucleotides	Average contig length	N50 contig length	Maximum contig length	Genome breadth coverage	Large indels	Sub-stitutions	Small indels
L1	108	4559977	42222	87242	269942	98.20%	0	0	1
L1-L2	82	4621035	56354	96588	232618	99.13%	0	5	0
L1-L3	43	4618437	107405	177402	409463	99.59%	0	1	0

- Maximum unique matches with MUMmer (Kurtz et al. 2004)

Versatile and open software for comparing large genomes.
Kurtz, Stefan and Phillippy, Adam and Delcher, Arthur L. and Smoot, Michael and Shumway, Martin and Antonescu, Corina and Salzberg, Steven L.
Genome Biology, 2004 http://dx.doi.org/doi:10.1186/gb-2004-5-2-r12

Assemblies

Table 2 - Assembly validations

Libraries	Number of contigs	Number of nucleotides	Average contig length	N50 contig length	Maximum contig length	Genome breadth coverage	Large indels	Sub-stitutions	Small indels
L1	108	4559977	42222	87242	269942	98.20%	0	0	1
L1-L2	82	4621035	56354	96588	232618	99.13%	0	5	0
L1-L3	43	4618437	107405	177402	409463	99.59%	0	1	0

- Maximum unique matches with MUMmer (Kurtz et al. 2004)
- L1-L3 (L1+L2+L3) yields only 43 contigs; no misassembled contigs

Versatile and open software for comparing large genomes.
Kurtz, Stefan and Phillippy, Adam and Delcher, Arthur L. and Smoot, Michael and Shumway, Martin and Antonescu, Corina and Salzberg, Steven L.
Genome Biology, 2004 http://dx.doi.org/doi:10.1186/gb-2004-5-2-r12

Assemblies

Table 2 - Assembly validations

Libraries	Number of contigs	Number of nucleotides	Average contig length	N50 contig length	Maximum contig length	Genome breadth coverage	Large indels	Sub-stitutions	Small indels
L1	108	4559977	42222	87242	269942	98.20%	0	0	1
L1-L2	82	4621035	56354	96588	232618	99.13%	0	5	0
L1-L3	43	4618437	107405	177402	409463	99.59%	0	1	0

- Maximum unique matches with MUMmer (Kurtz et al. 2004)
- L1-L3 (L1+L2+L3) yields only 43 contigs; no misassembled contigs
- Small increase of breadth of coverage -> sizable impact on number of contigs

Versatile and open software for comparing large genomes.
Kurtz, Stefan and Phillippy, Adam and Delcher, Arthur L. and Smoot, Michael and Shumway, Martin and Antonescu, Corina and Salzberg, Steven L.
Genome Biology, 2004 http://dx.doi.org/doi:10.1186/gb-2004-5-2-r12

Assemblies

Table 2 - Assembly validations

Libraries	Number of contigs	Number of nucleotides	Average contig length	N50 contig length	Maximum contig length	Genome breadth coverage	Large indels	Sub-stitutions	Small indels
L1	108	4559977	42222	87242	269942	98.20%	0	0	1
L1-L2	82	4621035	56354	96588	232618	99.13%	0	5	0
L1-L3	43	4618437	107405	177402	409463	99.59%	0	1	0

- Maximum unique matches with MUMmer (Kurtz et al. 2004)
- L1-L3 (L1+L2+L3) yields only 43 contigs; no misassembled contigs
- Small increase of breadth of coverage -> sizable impact on number of contigs
- Resources necessary: 30 processors, 9 GiB of memory, 11 minutes for L1-L3

Versatile and open software for comparing large genomes.
Kurtz, Stefan and Phillippy, Adam and Delcher, Arthur L. and Smoot, Michael and Shumway, Martin and Antonescu, Corina and Salzberg, Steven L.
Genome Biology, 2004 http://dx.doi.org/doi:10.1186/gb-2004-5-2-r12

Comparison with Velvet

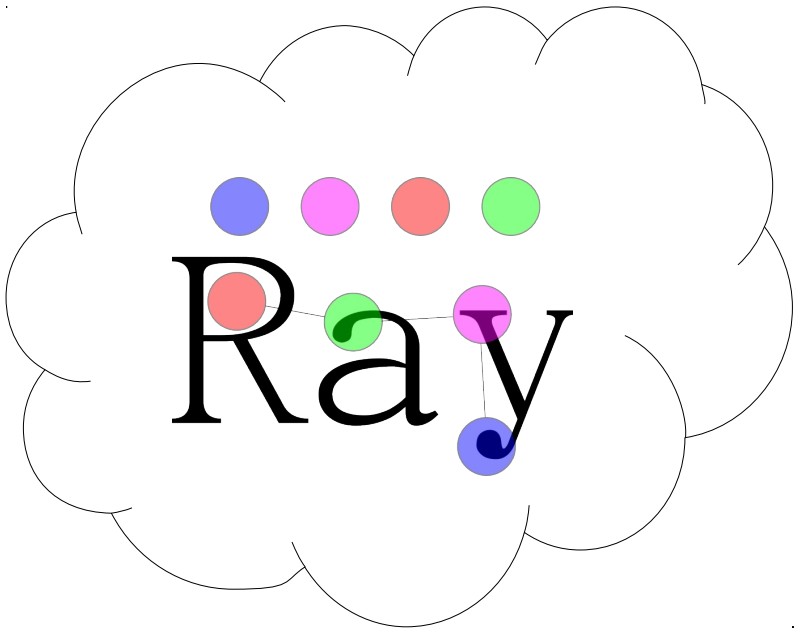
- Comparison of *E. coli* genome assemblies with Velvet and Ray
- with 3 paired libraries (200, 1000 & 10000)

	Velvet	Ray
Contigs	65	43
Misassembled contigs	36	0
Substitution errors	503	1
Small indels	829	0

- Hypothesis: Velvet's bubble merging is unsuitable for repeats

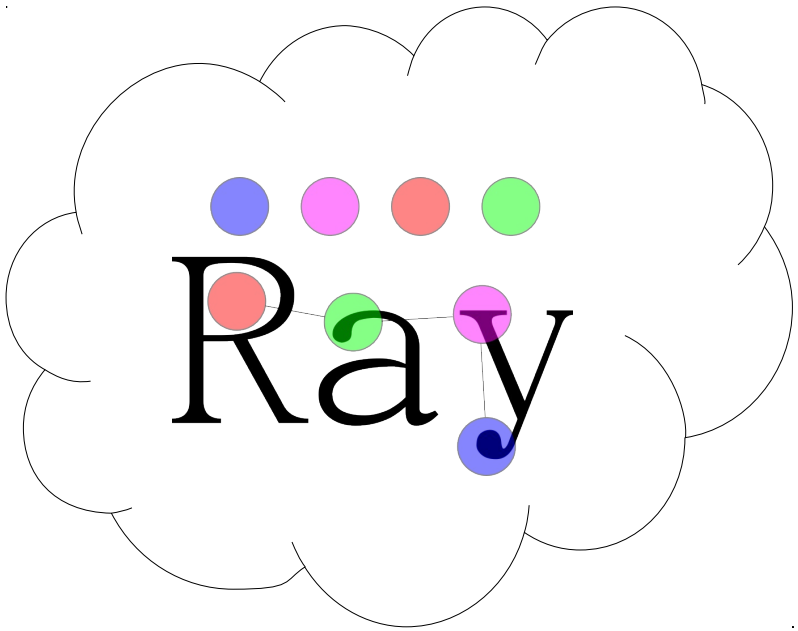
Conclusion

- Optimal read markers place reads on unique vertices if possible



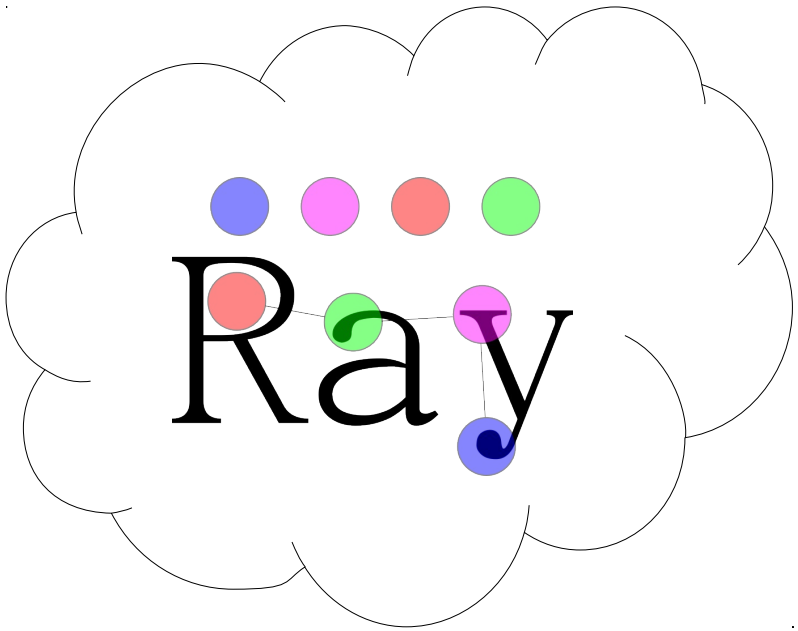
Conclusion

- Optimal read markers place reads on unique vertices if possible
- Statistical constraints allow optimal pair placements



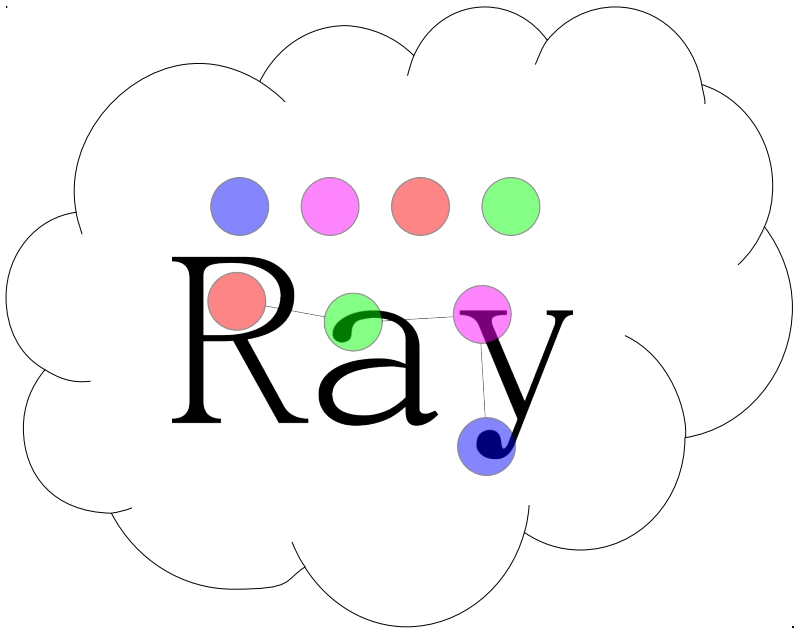
Conclusion

- Optimal read markers place reads on unique vertices if possible
- Statistical constraints allow optimal pair placements
- Treat repeats precisely = large contiguous sequences without misassemblies



Perspective

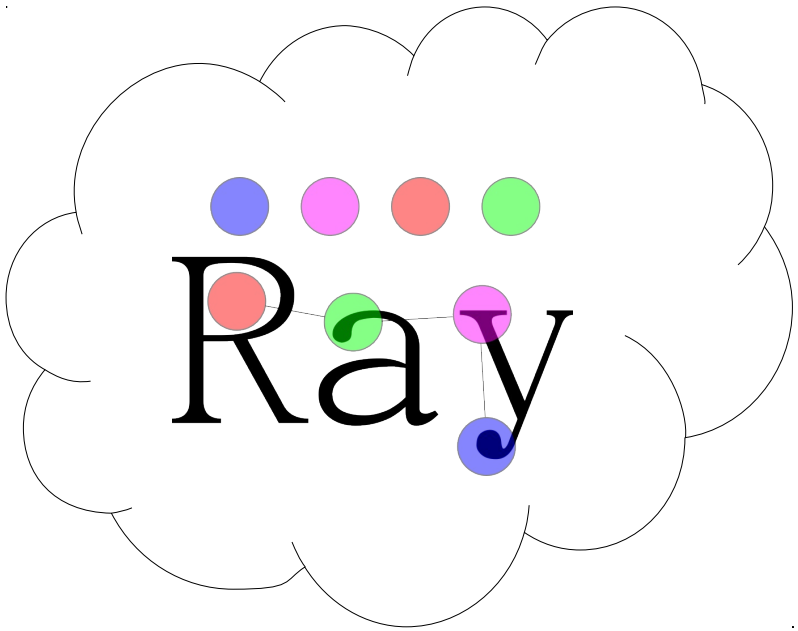
- Cloud-ready (needs MPI + fast interconnect)



Perspective

- Cloud-ready (needs MPI + fast interconnect)

- Assemble large genomes: *de novo* assembly of Illumina CEO genome in 11.5 h with Ray (Supplementary slides)



Acknowledgments

- RECOMB-seq committees for organising this great workshop
- Canadian Institutes of Health Research for doctoral award to S.B.
- J.C. holds the Canada Research Chair in Medical Genomics; funded by the Canadian Institutes of Health Research
- Natural Sciences and Engineering Research Council of Canada for funding to F.L.
- Canadian Foundation for Innovation for infrastructure funding
- Compute Canada (CLUMEQ) for compute resources
- Free/libre software people

Canada Research Chairs
Chaires de recherche du Canada

Canada Foundation for Innovation
Fondation canadienne pour l'innovation

NSERC CRSNG

CLUMEQ

compute calcul CANADA

IRSC CIHR
Instituts de recherche en santé du Canada
Canadian Institutes of Health Research

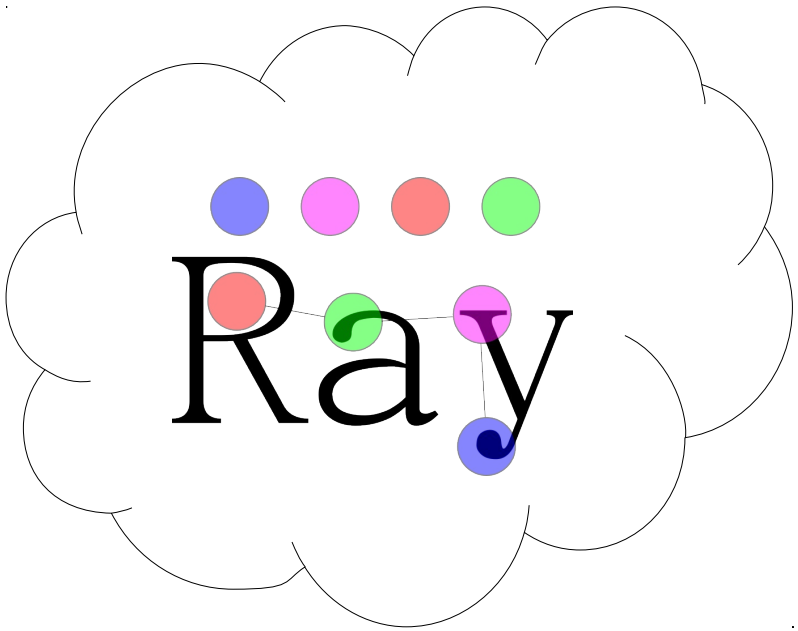
UNIVERSITÉ LAVAL

Questions ?

Merci à vous pour votre attention !

- **Cloud-ready (needs MPI + fast interconnect)**

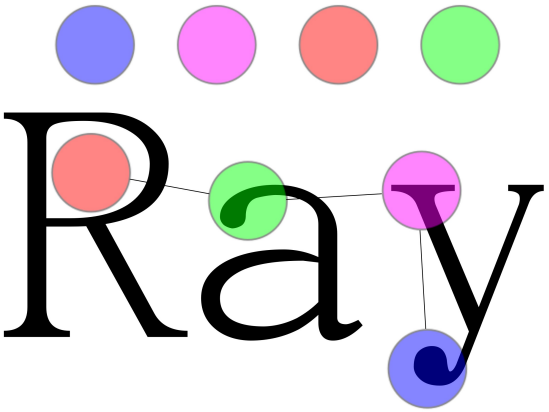
  Example: Amazon EC2 Cluster Compute



- http://tiny.cc/ray-assembler

@sebhtml on Twitter

Supplementary: Ray command




```
$ mpirun -np 30 /software/ray-1.2.4/bin/Ray -k 21 -o L1-L3-assembly \
 -p L1_1.fasta L1_2.fasta -p L2_1.fasta L2_2.fasta -p L3_1.fasta L3_2.fasta
```

Supplementary: Velvet commands

Commands to convert files and run Velvet with automatic calculation of insert lengths and coverage values (3 categories):

```
$ /software/velvet_1.0.19/shuffleSequences_fasta.pl L1_1.fasta L1_2.fasta L1.fasta
$ /software/velvet_1.0.19/shuffleSequences_fasta.pl L2_1.fasta L2_2.fasta L2.fasta
$ /software/velvet_1.0.19/shuffleSequences_fasta.pl L3_1.fasta L3_2.fasta L3.fasta

$ /software/velvet_1.0.19/velveth velvetAssembly 21 -fasta \
  -shortPaired L1.fasta -shortPaired2 L2.fasta -shortPaired3 L3.fasta

$ /software/velvet_1.0.19/velvetg velvetAssembly -exp_cov auto
```

Supplementary: a Velvet incorrect contig

Example of Velvet incorrect contig (# 134; length: 4334 bp)

Reference segment	Contig segment
2116786-2116856	1-71
2296192-2299712	51-3571
2309689-2310402	3582-4295

➔ Velvet seems to jump over repeats, hence skipping important regions

Supplementary: Human genome chromosome 1

```
[1,0]<stdout>: Beginning of computation: 1 seconds
[1,0]<stdout>: Distribution of sequence reads: 1 minutes, 0 seconds
[1,0]<stdout>: Distribution of vertices & edges: 17 minutes, 58 seconds
[1,0]<stdout>: Calculation of coverage distribution: 8 seconds
[1,0]<stdout>: Indexing of sequence reads: 24 minutes, 38 seconds
[1,0]<stdout>: Computation of seeds: 18 minutes, 19 seconds
[1,0]<stdout>: Computation of library sizes: 5 minutes, 9 seconds
[1,0]<stdout>: Extension of seeds: 2 hours, 37 minutes, 12 seconds
[1,0]<stdout>: Computation of fusions: 43 minutes, 52 seconds
[1,0]<stdout>: Collection of fusions: 9 seconds
[1,0]<stdout>: Completion of the assembly: 4 hours, 28 minutes, 26 seconds

Memory usage: ~ 56 GiB

28 CPUs

Peak coverage: 19
```

Supplementary: resource on Compute Canada's colosse

- Ray 1.3.0 on 64 computers
- Linked with InfiniBand QDR (40 Gigabits per second)
- 24 GiB of memory per computer
- 2 Intel Xeon Nehalem-EP (x86_64) processors per computer
- 4 compute cores per processor
- Total: 512 compute cores, 1535 GiB of memory

compute calcul CANADA

CLUMEQ

Supplementary: SRA010766, Illumina CEO genome data

- Illumina CEO
- Jay T. Flatley
- Illumina Genome Analyzer II
- Input: 6 372 129 288 reads (477 909 696 600 nucleotides)

illumina

http://illumina.com/humangenome

Supplementary: *de novo* assembly of Illumina CEO genome (2011-03-21)

- k=21
- Running time: 11.5 h
- Outer distances: 190 +/- 30; read length: 75
- Peak coverage: 22, minimum coverage: 6
- Peak probably too low
- Output: 1 803 534 contiguous sequences
- 1 772 120 417 nucleotides (haploid human genome is 3 Gb)
- N50: 1341, average length: 982, longuest: 14584
- Job identifier: 2814556, job code name: Nitro

Supplementary: next steps

- Try with a higher k-mer length (k=31)

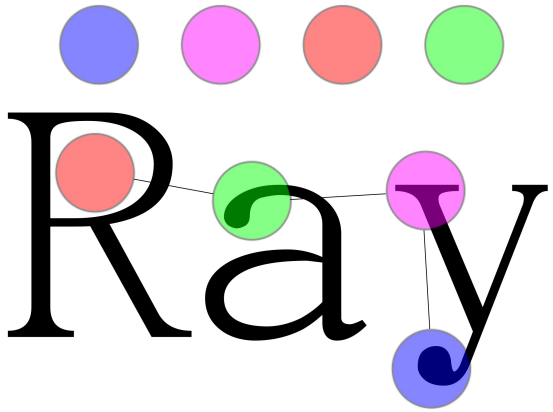
Other dataset:

- Test on the African Genome (SRA000271)
- Yoruban male (NA18507)
- Illumina Genome Analyzer platform

illumina http://illumina.com/humangenome

Supplementary: Ray software systems

- 143 MPI tag types, 28 master modes, 24 slave modes
- Max. 250 paired libraries, max. coverage 65535, max processors: 1000000
- With n processors: n slave rank + 1 master rank (which is also a slave)
- Last revision: 4462 (1.3.0-dev)
- 22792 lines of code (C++; .h, cpp)
- 65 classes, 74 .cpp files, 68 .h files



Supplementary: Ray message transit systems

- Manual message aggregation in some steps
- Virtual communicator on top of MPI_COMM_WORLD (MPI default communicator)
- Allows transparent message aggregation
- Array of workers on each MPI ranks (max. 30000)
- Workers see only 3 methods: pushMessage, isMessageProcessed, getResponseElements
- All the aggregation logic is done by the virtual communicator