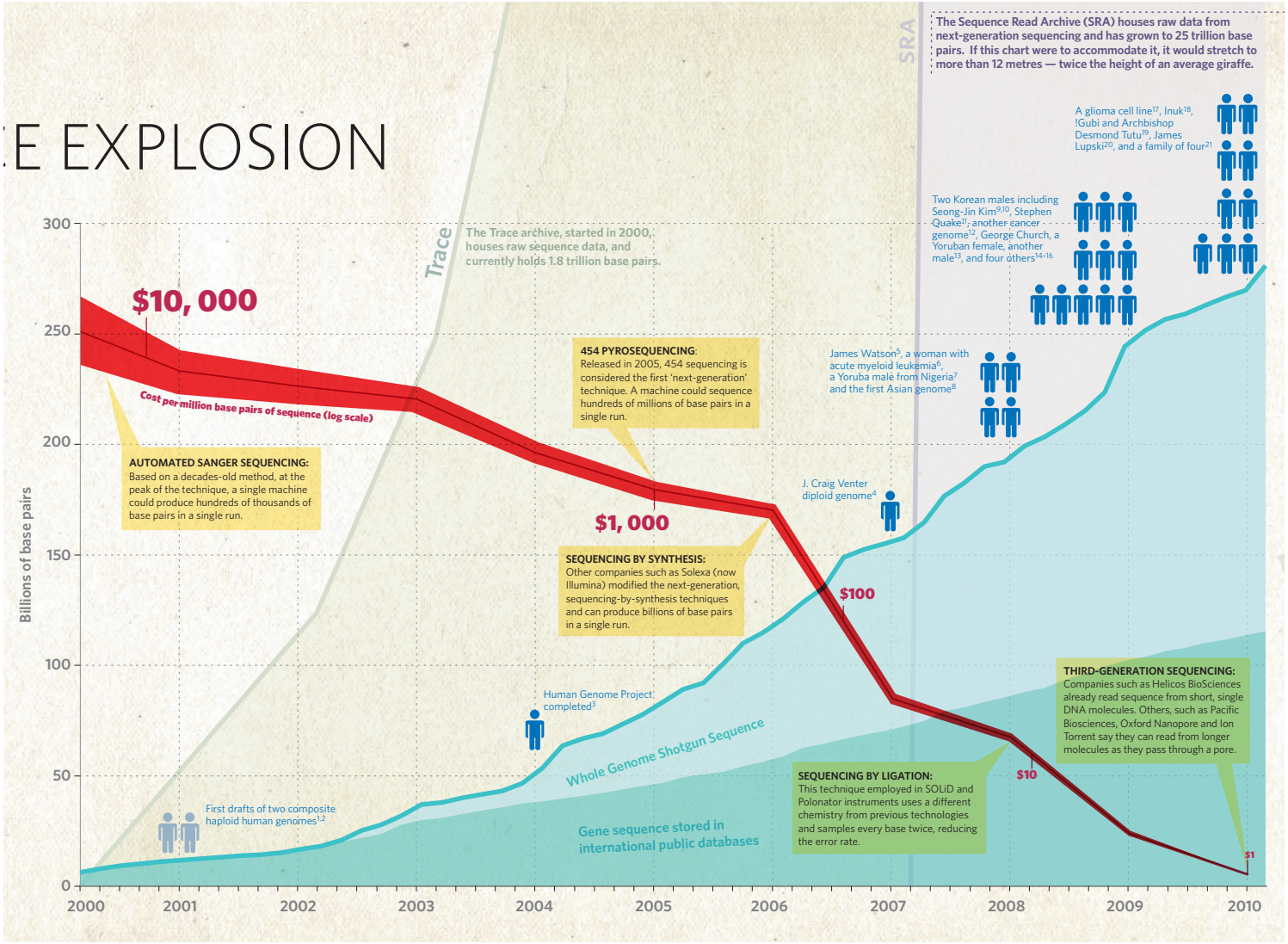
Next generation sequencing: assembly by mapping reads

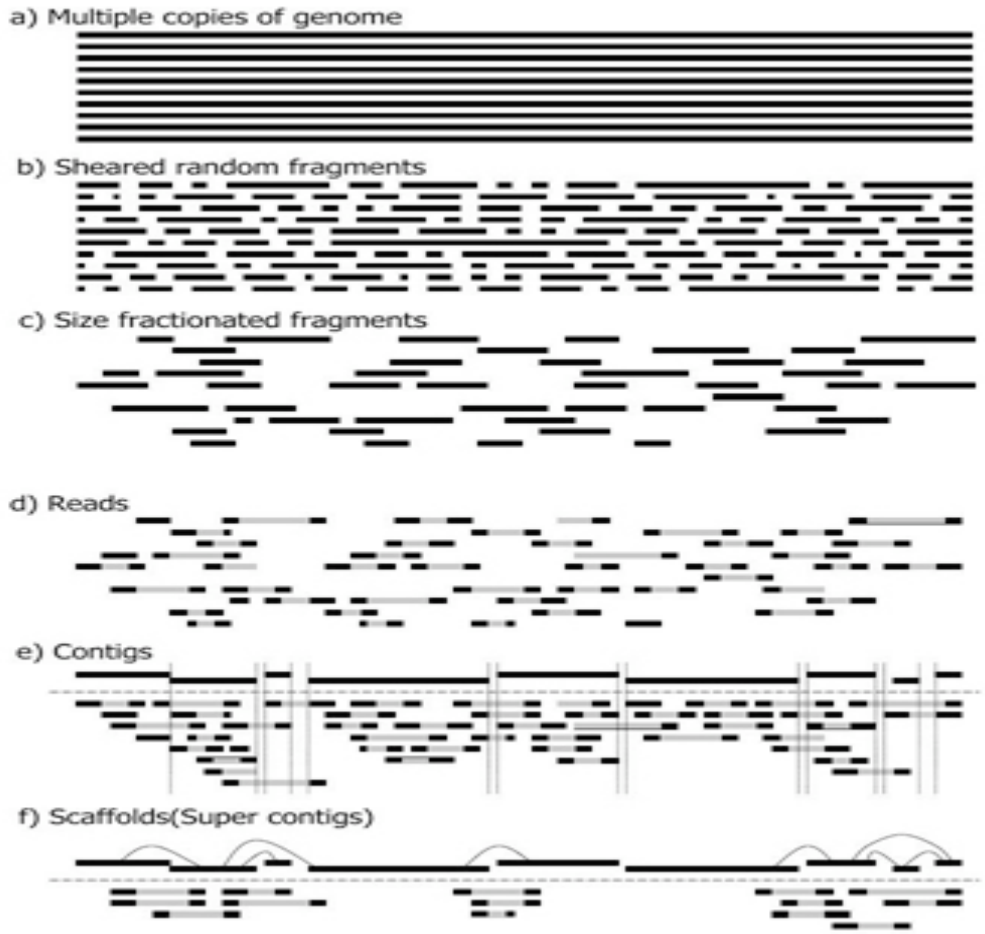
Laurent Falquet, Vital-IT
Bogota, March 21, 2011

Swiss Institute of Bioinformatics

Evolution of DNA sequencing

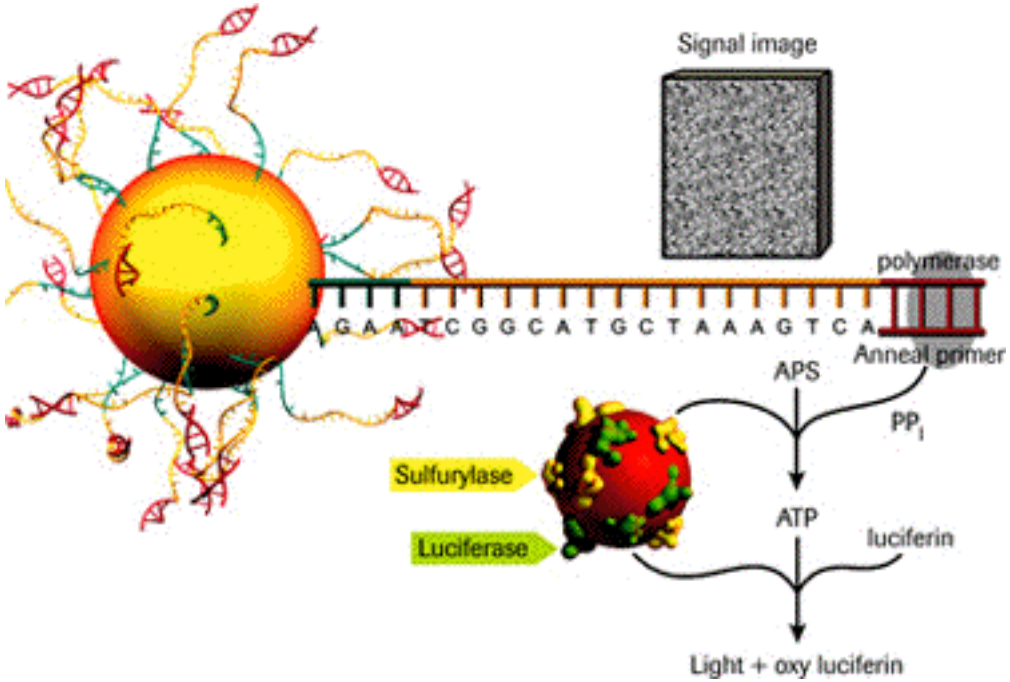


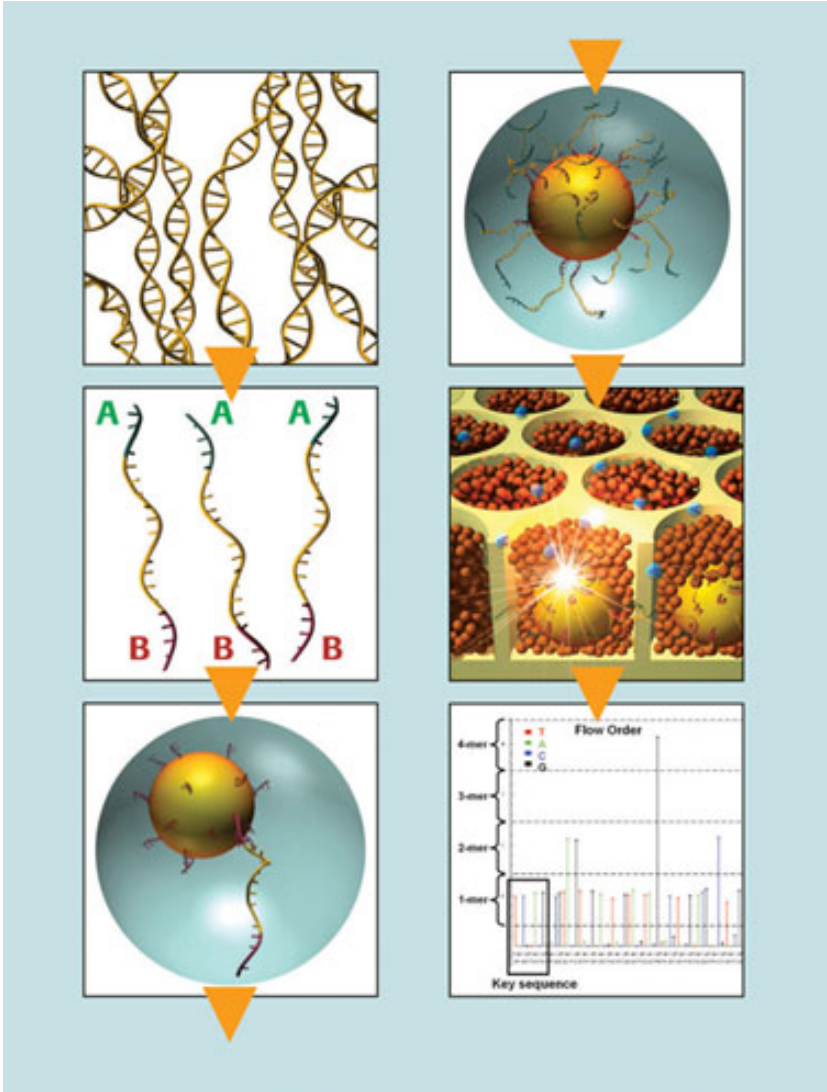
Ultra High Throughput Sequencing (WGS)



- http://www.k.u-tokyo.ac.jp/pros-e/person/shinichi_morishita/shinichi_morishita.htm

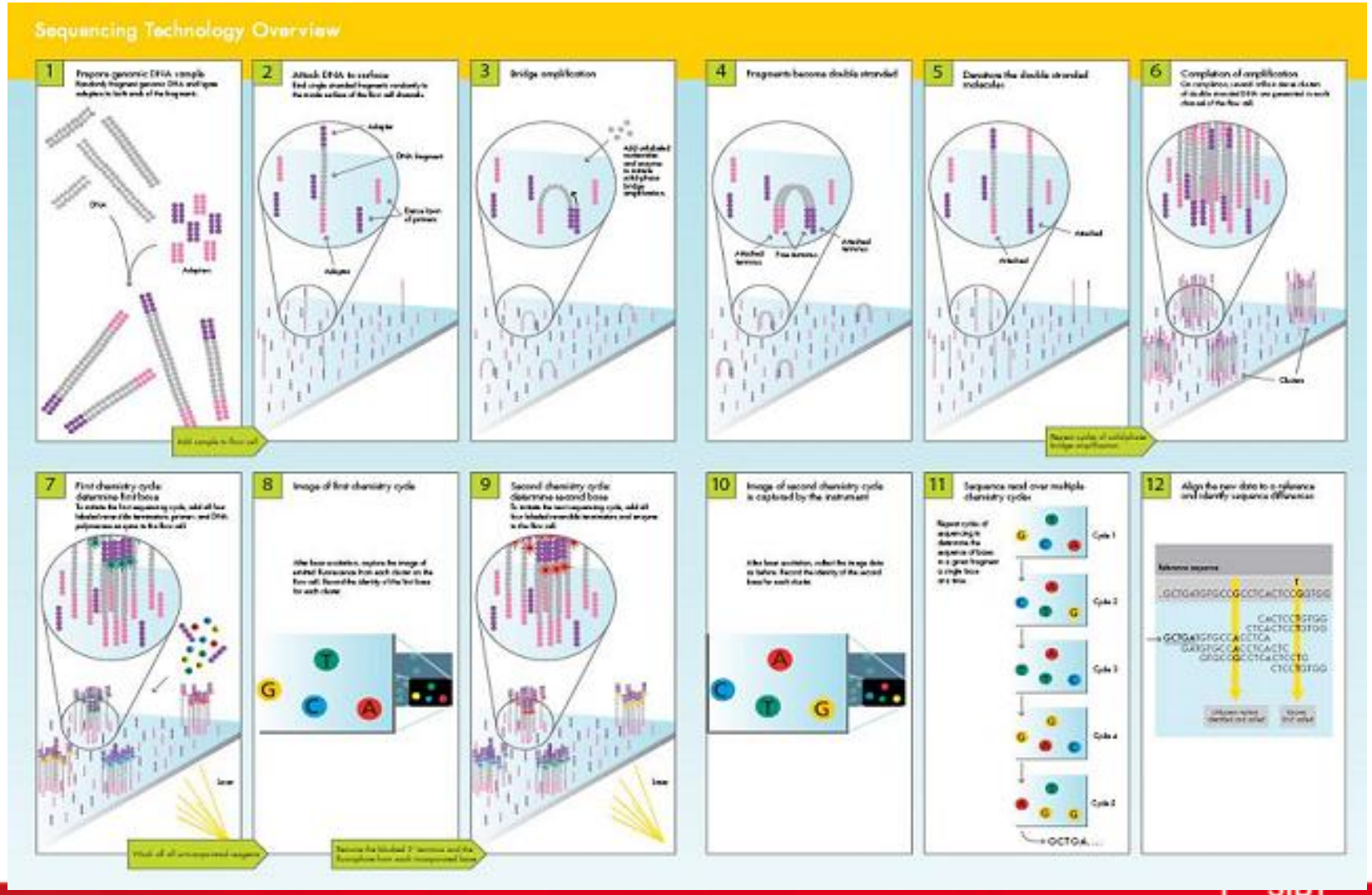
454: Pyrosequencing with pre-amplification step (emulsion PCR)



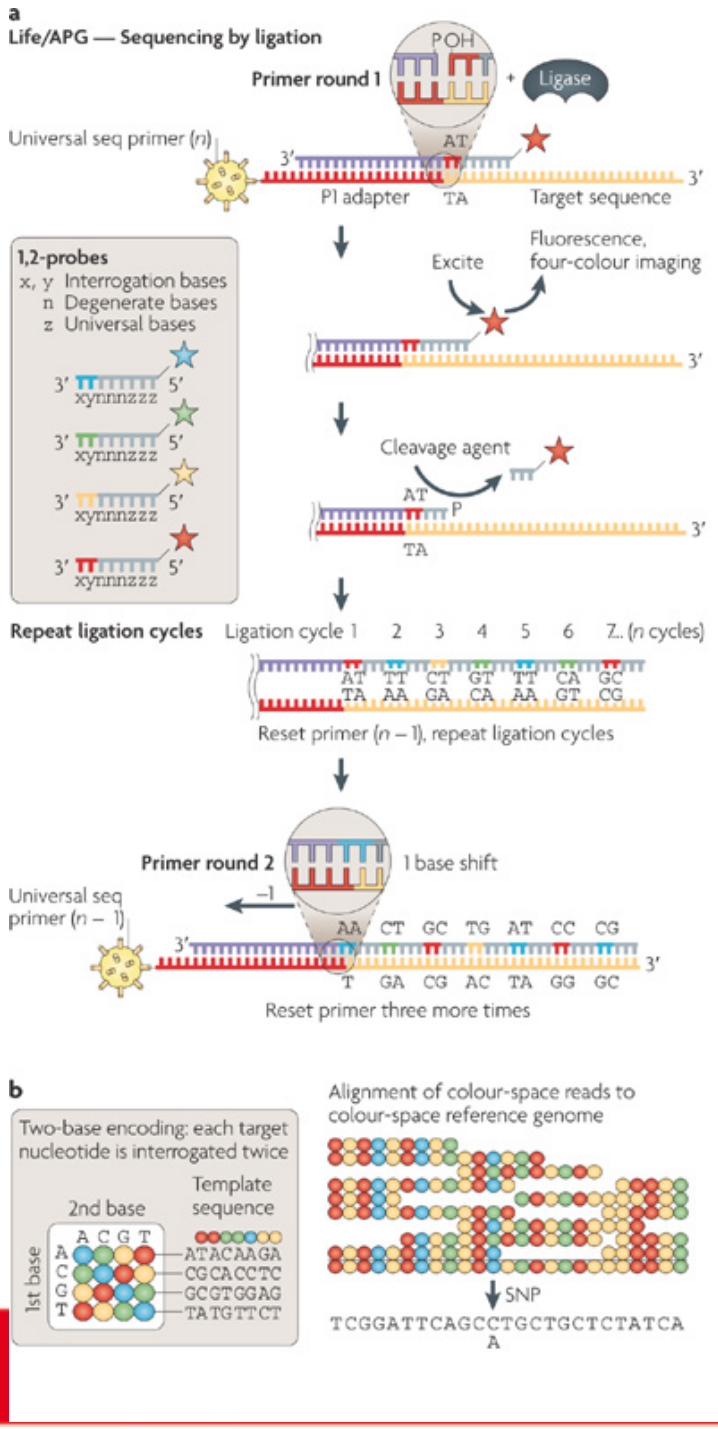


Illumina: sequencing by synthesis on a surface with pre-amplification step

illumina



SOLiD: Sequencing by ligation with pre-amplification step (emulsion PCR) and double read of each base



What are NextGenerationSequencing short reads data?

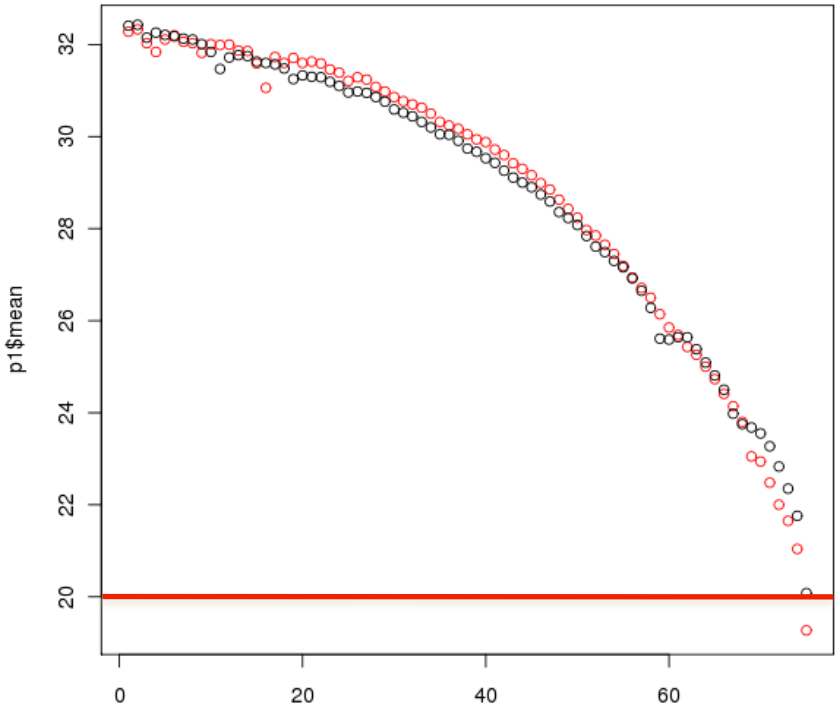
Sequencing platform	ABI3730xl Genome Analyzer	Roche (454) FLX	Illumina Genome Analyzer and HiSeq	ABI SOLiD (Life technologies)	IonTorrent (Life technologies)
Sequencing chemistry	Automated Sanger sequencing	Pyrosequencing on solid support	Sequencing-by-synthesis with reversible terminators	Sequencing by ligation	Pyrosequencing converted to current on chip
Template amplification method	In vivo amplification via cloning	Emulsion PCR	Bridge PCR	Emulsion PCR	None (single molecule)
Read length	700–900 bp	200–500 bp	36-150 bp	35-75 bp	50–100 bp
Sequencing throughput (old numbers)	0.03–0.07 Mb/h	13 Mb/h	25 Mb/h	21–28 Mb/h	? Mb/h
Advantage by price	700 bp / $	16'000 bp / $	500'000 bp / $	1'000'000 bp / $	? bp / $
Nr of installed machines (estimation)	??	243	926	268	5

Limitations of the techniques

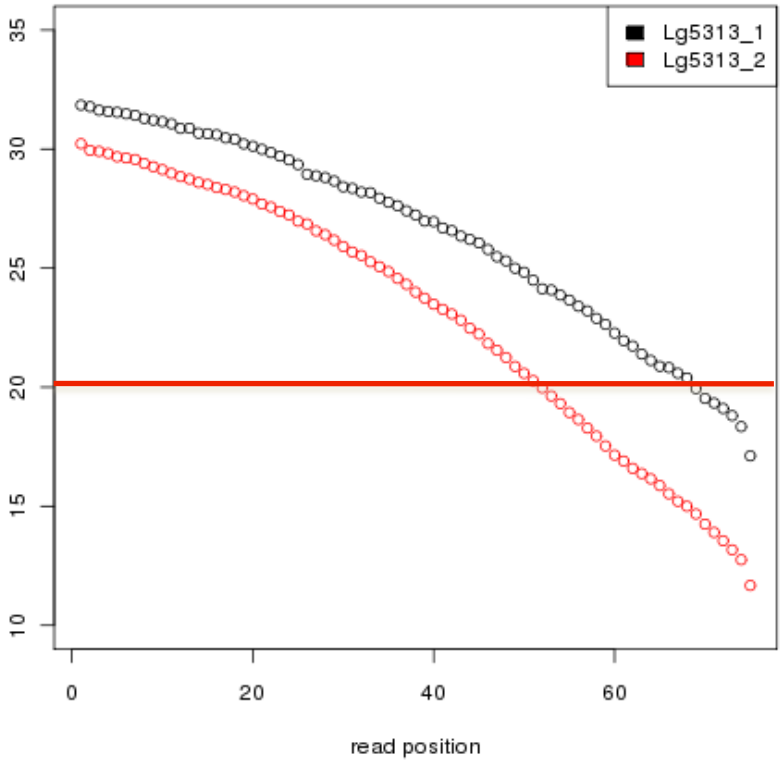
- All methods
  - Sequencing errors
  - Missing data (sampling/coverage bias)
- Roche 454
  - long (>12) mononucleotide repeats
- Illumina
  - short reads (36-150bp)
- SOLiD
  - very short reads (25-50bp)
  - biased paired-ends (50/25)

Variability in the quality (mean value per position)

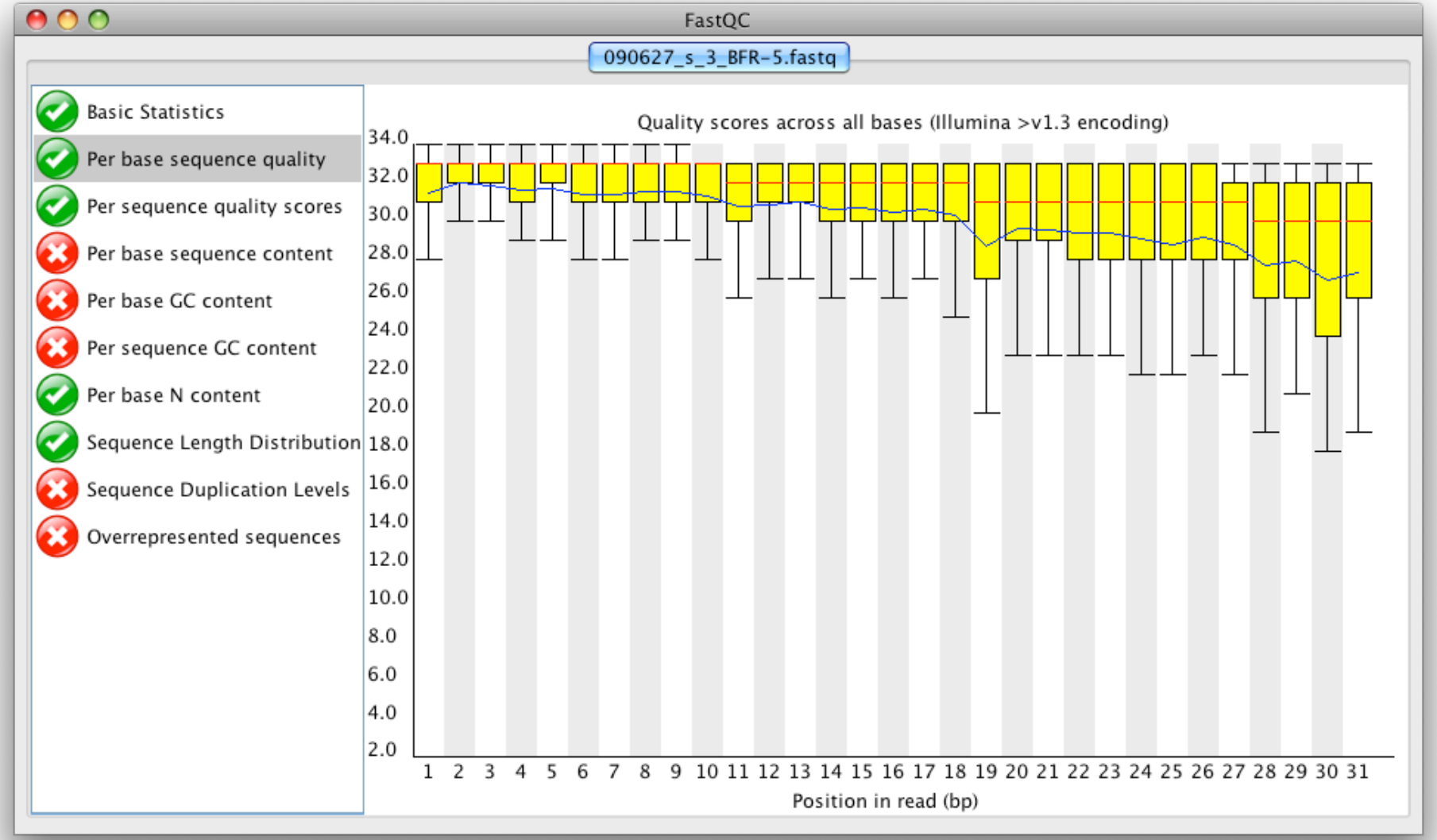
- Good example



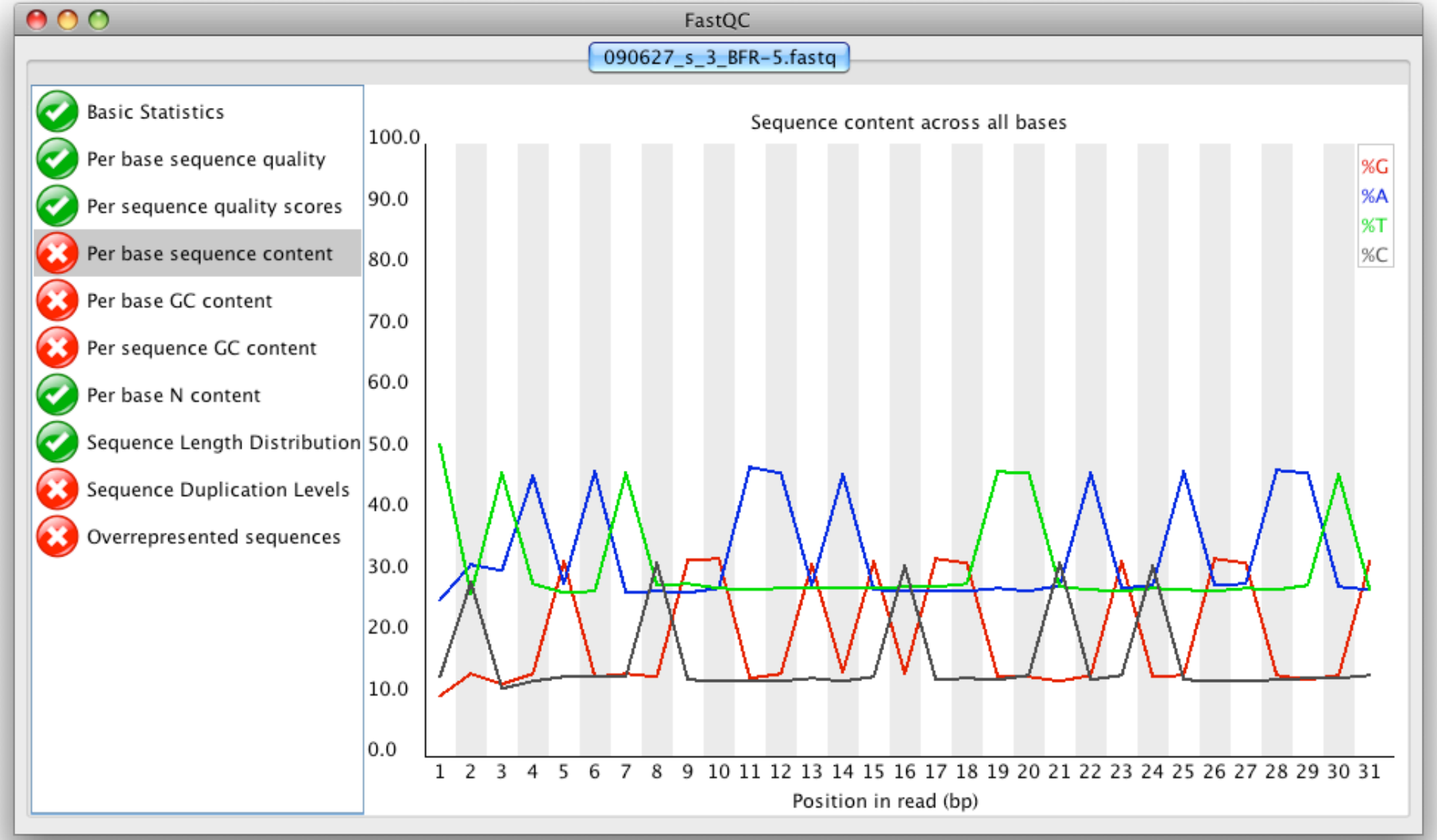
- Less good example…



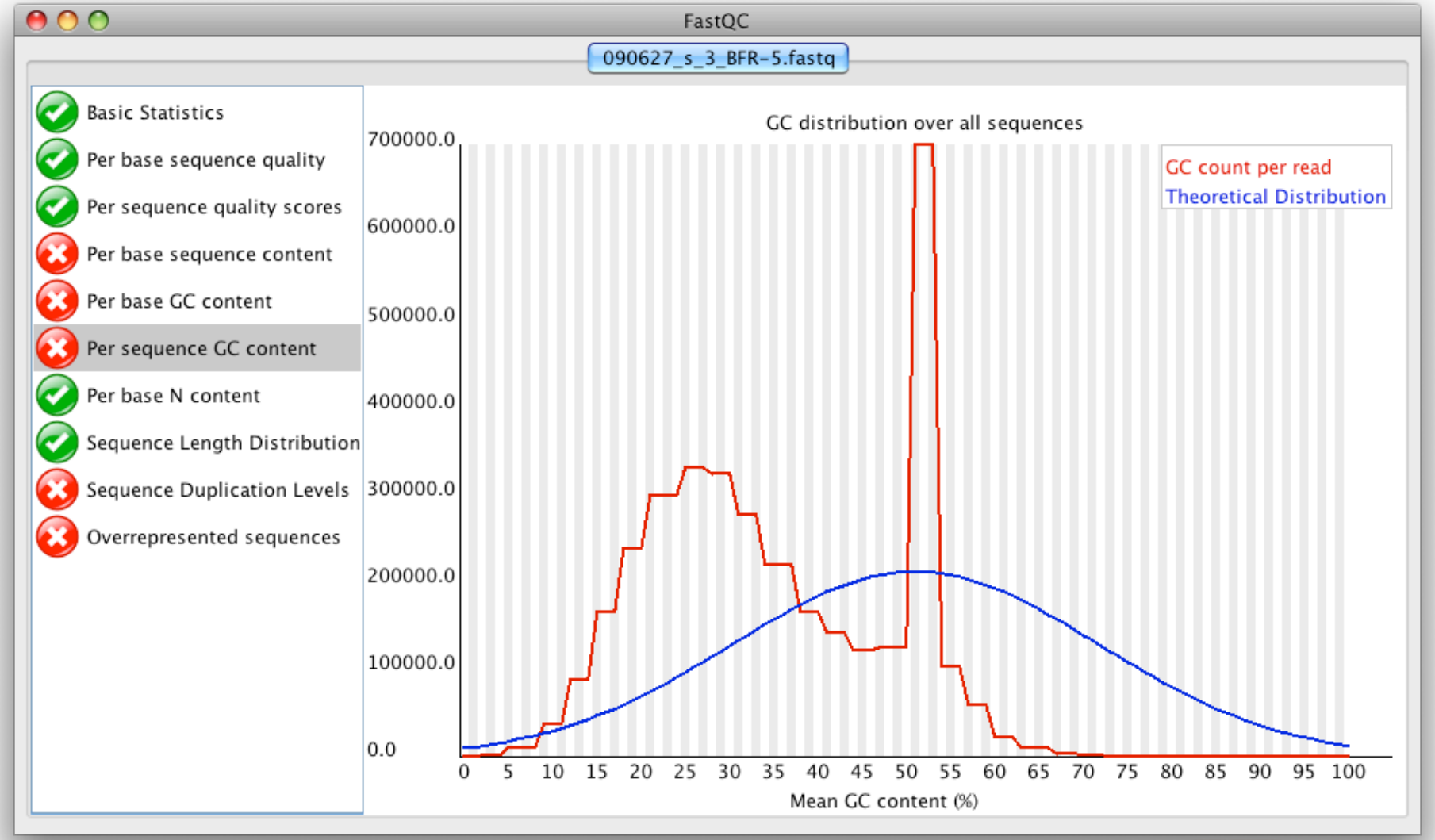
Quality Control example (fastqc)



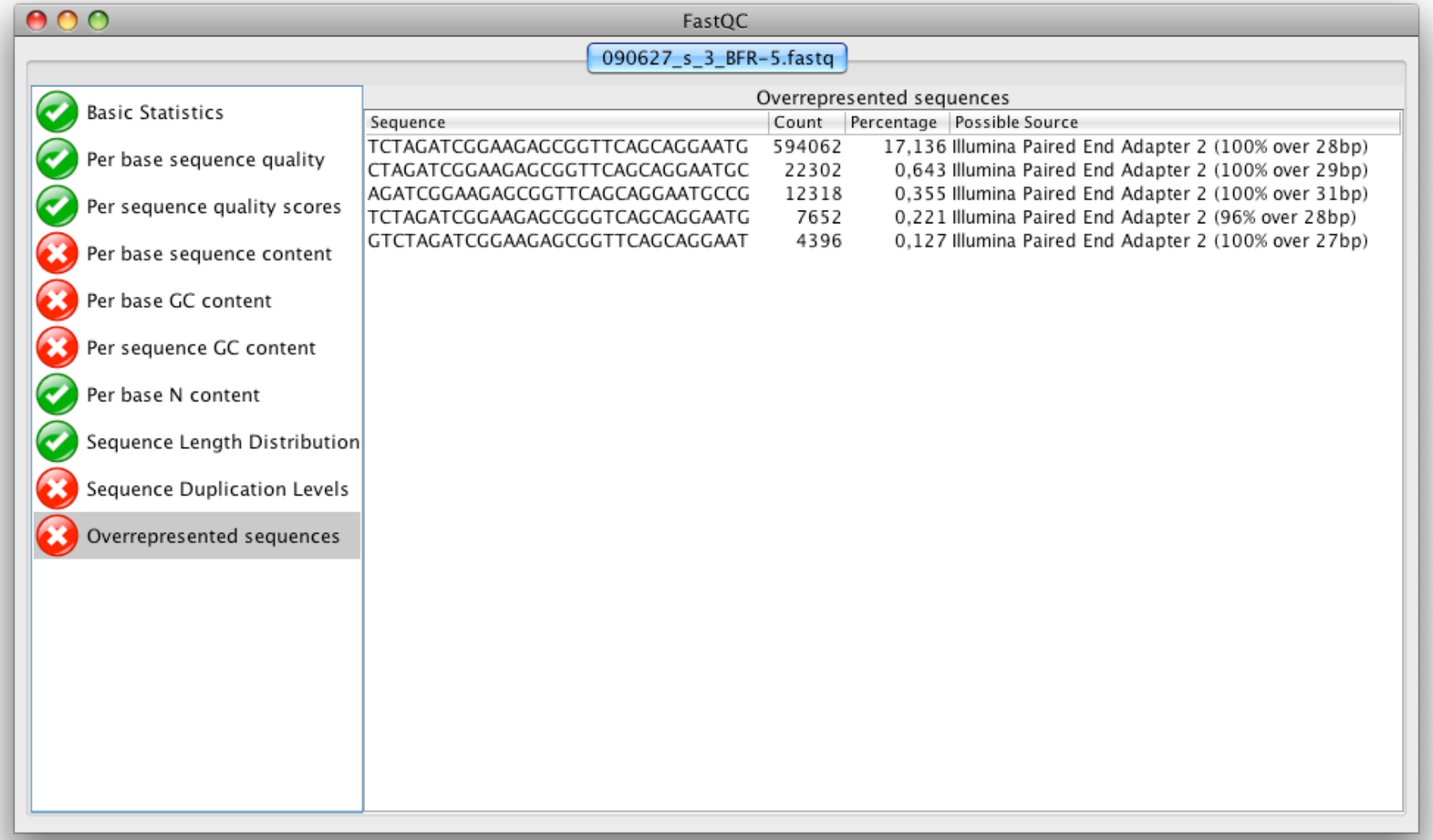
Quality Control example



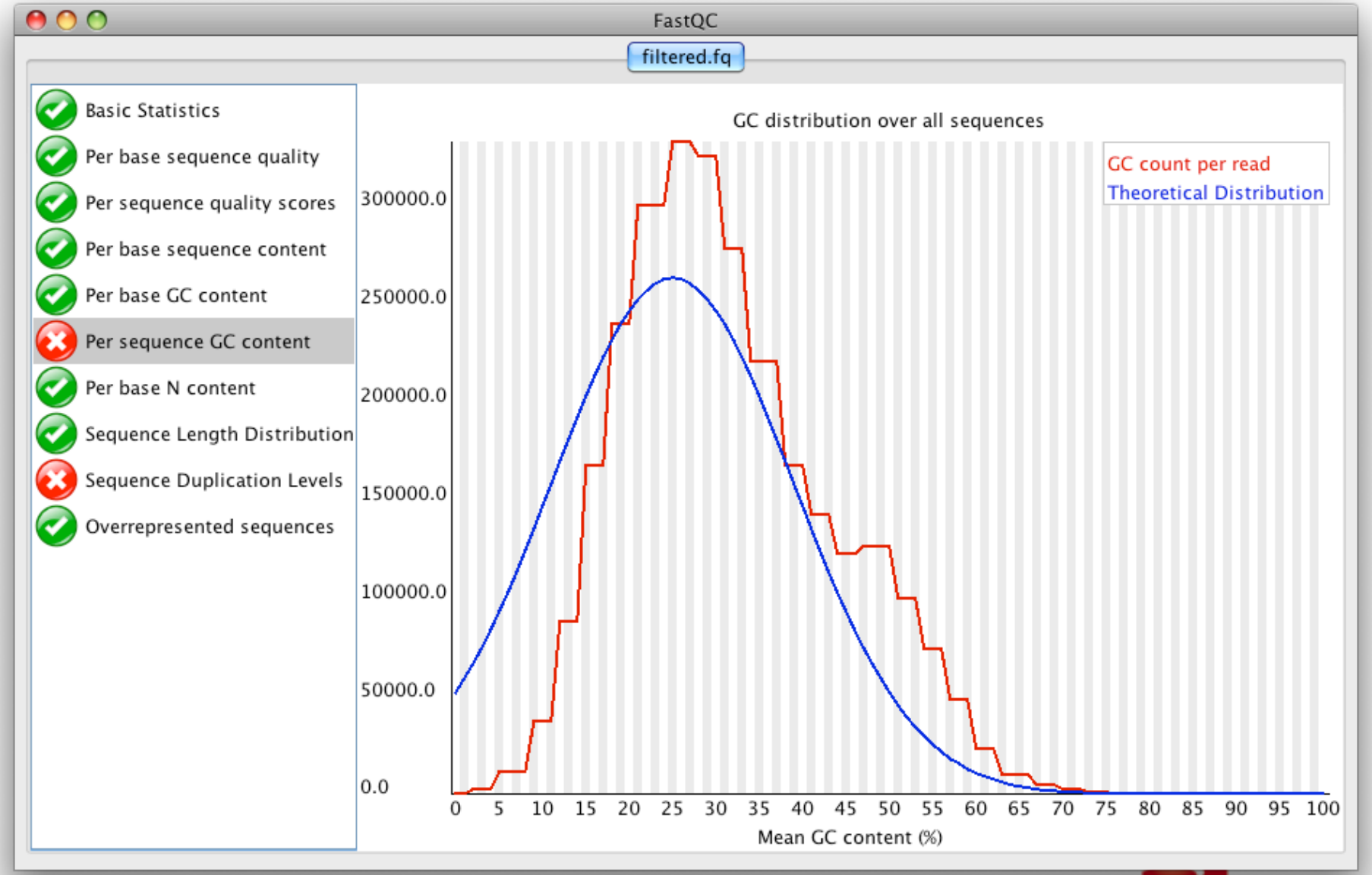
Quality Control example



Quality Control example



Quality Control example



Filtering data can help

http://pathogenomics.bham.ac.uk/blog/2009/09/tips-for-de-novo-bacterial-genome-assembly/

- Illumina reads quality decrease with length
  - Trim 3' ends of reads according to quality
  - Remove reads with average low quality
  - If coverage is high, remove orphan reads

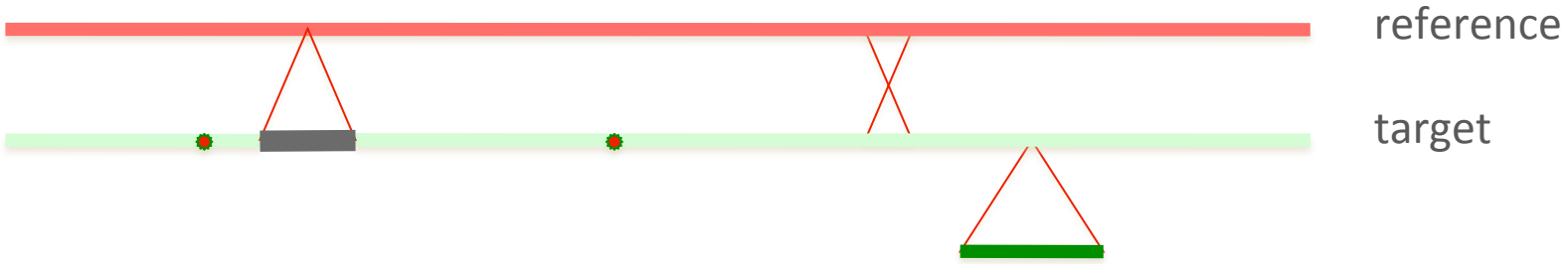
- 454 reads
  - Trim 3' ends of reads according to quality
  - Remove reads with average low quality
  - If possible correct for long mononucleotide repeats

- For mapping some people don't clean the data…

Mapping = alignment of reads on a reference mostly for Ultra High Throughput (re)Sequencing

- Simpler by mapping reads onto an existing genome
  - User must select the most appropriate reference
  - Success depends on the degree of similarity of the reference

- Variations detectable: SNPs and deletions
- Variations difficult to guess: insertions and inversions

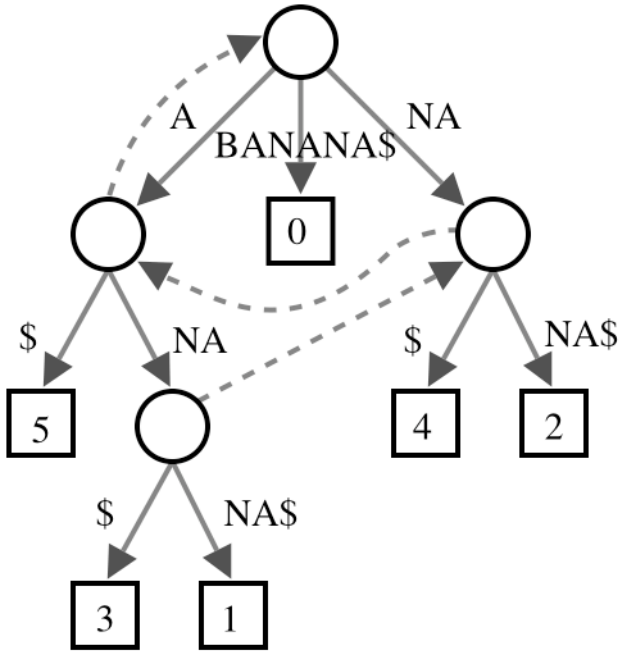


Mapping methods

- By sequence comparison with Smith-Waterman
  - much too slow
- By sequence indexing (e.g., BLAST or BLAT)
  - Conventional tools like Blast or Blat do not work well with short sequence reads.
  - > Modification of existing alignment algorithms to handle short reads.
- Indexing methods
  - Suffix tree
  - Suffix array
  - Seed hash tables
  - BWT (Burrows-Wheeler Transform)

Suffix tree

- The suffix tree for a string S is a tree whose edges are labelled with strings. Suffix trees also provided one of the first linear-time solutions for the longest common substring problem. These speedups come at a cost: storing a string's suffix tree typically requires significantly more space than storing the string itself.



35Gb for the human genome

Suffix array

- Consider the string BANANA$ of length 7. It has 7 suffixes:

index	suffix
0	BANANA$
1	ANANA$
2	NANA$
3	ANA$
4	NA$
5	A$
6	$

sort →

index	suffix
6	$
5	A$
3	ANA$
1	ANANA$
0	BANANA$
4	NA$
2	NANA$

The suffix array is the array of indices: {6,5,3,1,0,4,2}

12Gb for the human genome

Seed hash table

- Given the string ACGTACGTAAG of length 10, extract all substrings length 4 (seeds) and store their starting positions.

index	seed
0,4	ACGT
1,5	CGTA
2	GTAC
3	TACG
6	GTAA
7	TAAG

sort →

index	seed
0,4	ACGT
1,5	CGTA
6	GTAA
2	GTAC
7	TAAG
3	TACG

The size of the hash table depends on the length of the seed and the complexity of the input string

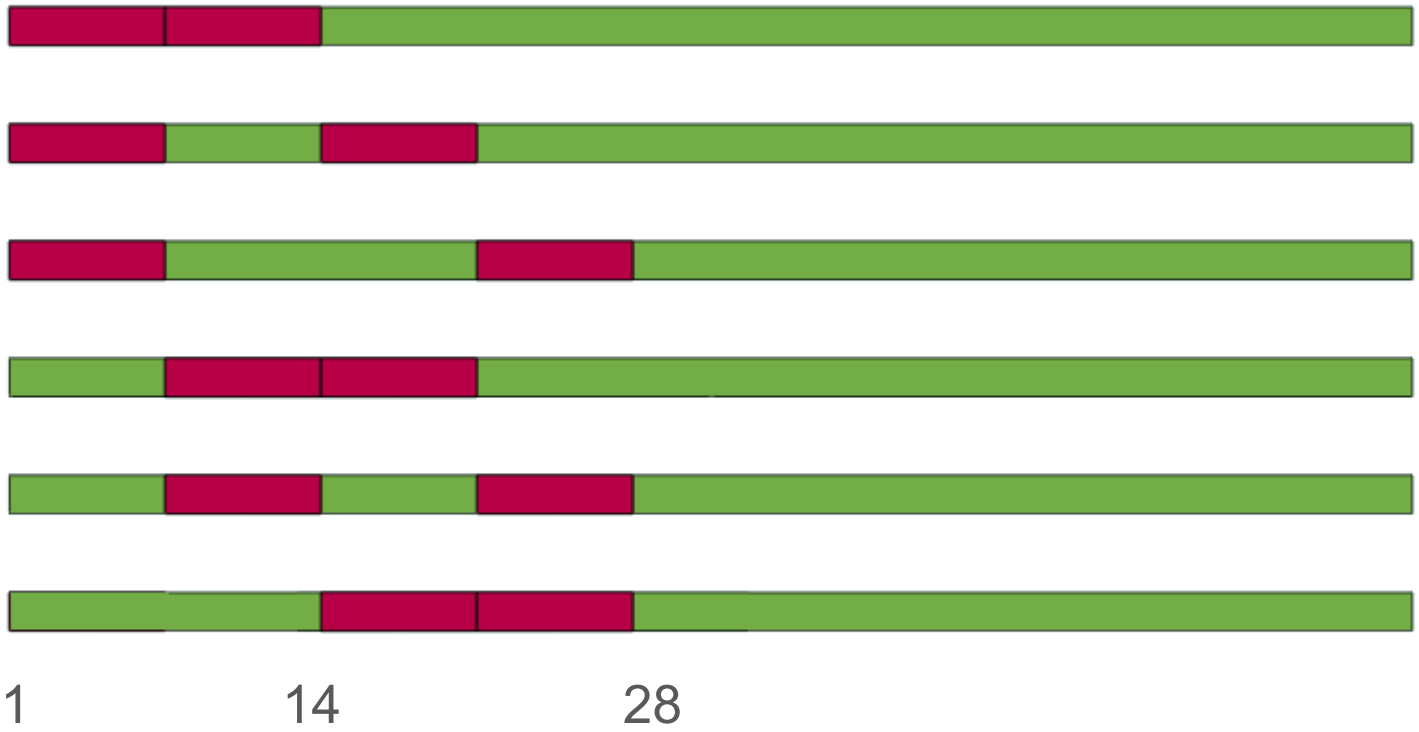
12Gb for the human genome

Seed hash table

- Hash tables can be generated according to different hash functions and using various seeds. Example for the sequence AGTGACAGT
  - Continuous seed (length 4): AGTG, GTGA, TGAC…
  - Non continuous seed (length 4): AGTG, ACAG
  - Spaced seed (length 4, weight 3, path1101): AG*G, GT*A, TG*C…
  - Periodic spaced seed: path=n*(1101)
- Hash tables have been extensively used in mapping programs.

Spaced seed hash table indexing (MAQ)

- MAQ build 6 hash tables, each indexing 14 of the first 28 bases

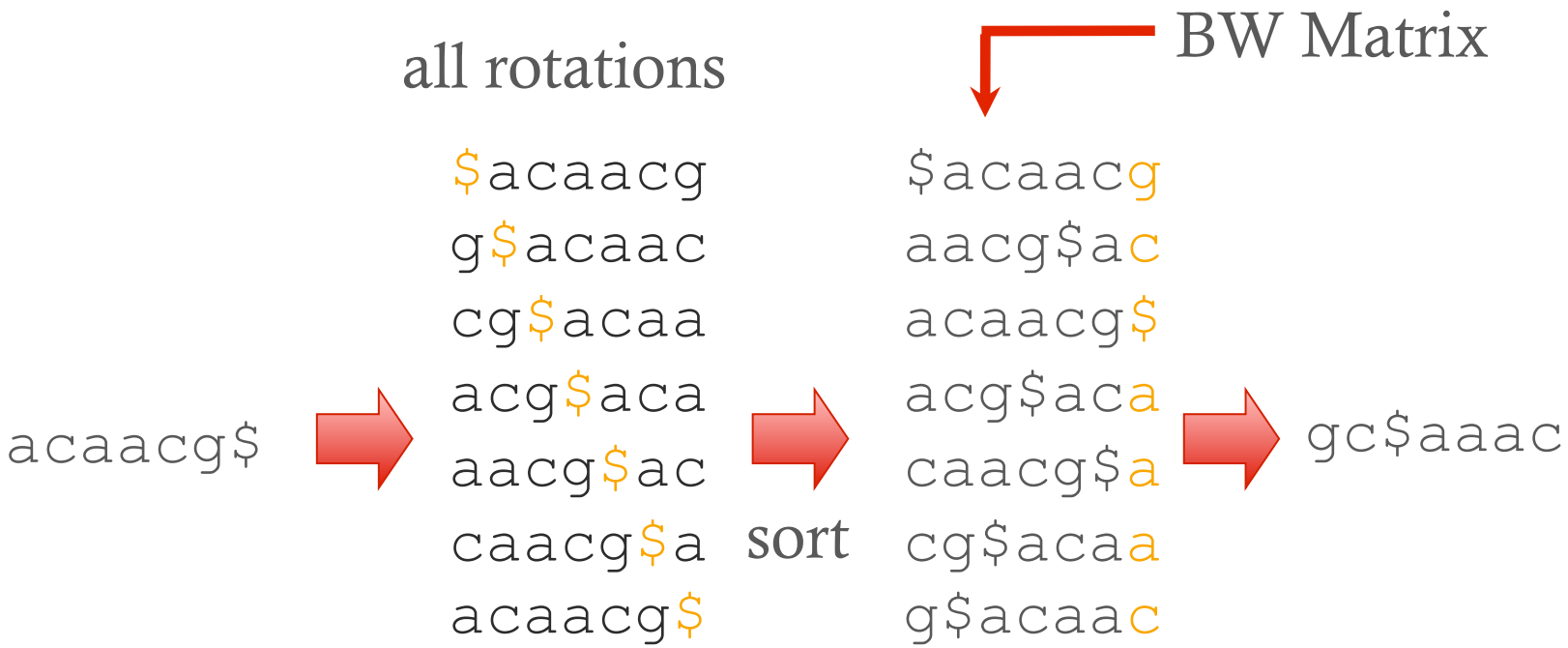


Hence, Maq finds all alignments with at most 2 mismatches in the first 28 bases.

Why Burrows-Wheeler?

- BWT very compact:
  - Approximately ½ byte per base
  - As large as the original text, plus a few “extras”
  - Can fit onto a standard computer with 2GB of memory
- Linear-time search algorithm
  - proportional to length of query for exact matches

Burrows-Wheeler Transform (BWT)



Langmead et al. 2009 Genome Biology

Burrows-Wheeler Matrix

$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac

See the hidden suffix array?

File formats

- Input
  - FASTA
  - FASTQ (various versions)
  - csFASTA
  - QSEQ
- Paired-end
  - 2 files
  - crossbow style

- Output
  - map
  - bwt
  - pileup
  - SAM
  - BAM

Example of FASTQ Illumina 1.5

```
@C3PO_0001:2:1:17:1499#0/1
TGAATTCATTGACCATAACAATCATATGCATGATGCAAATTATAATATCATTTTTAGTGACGTCGTGAATCGTTT
+C3PO_0001:2:1:17:1499#0/1
abaaaaaaaaaaa`a`aa_aaaaaaaaaaaaaaa_a__aaa`aaaaa^aaaaa`a]^`a__YZYZ^`NJDJ\_Z
@C3PO_0001:2:1:17:1291#0/1
TGTTTGAGCAAATGATTCATAATAATGTATTTCAATATTTTTAGGAATATCTCCCAATATTGCGCGTGCTGAATT
+C3PO_0001:2:1:17:1291#0/1
a`_`_\a_aaaa_a^Z^^a[a^aa]a_^_a_``aa__`aa`X^X^^`aa_\_]VR`\a_]W\_`_a]a]][\RZV
@C3PO_0001:2:2:1452:1316#0/1
GTCCATCCGCAGCAGCGAATTTTTGACGTCCCCCCCCCGAANGGANGNGANNNNGNNGNNNTNTNNAAANGNNNNN
+C3PO_0001:2:2:1452:1316#0/1
_U__a\__`]_`ZP\\_Z^[]aa^a_]XNBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
…
```

Warning: various FASTQ formats…

```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
.........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................
.............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.....................
.............................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
|                         |    |        |                              |                     |
33                       59   64       73                            104                   126
```

S – Sanger Phred+33, raw reads typically (0, 40)

X – Solexa Solexa+64, raw reads typically (-5, 40)

I – Illumina 1.3+ Phred+64, raw reads typically (0, 40)

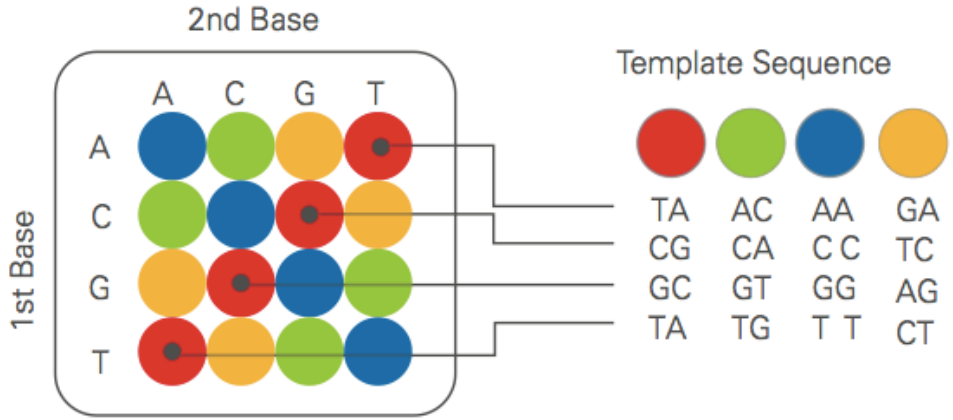
J – Illumina 1.5+ Phred+64, raw reads typically (3, 40) with 0 and 1=unused, 2=Read Segment Quality Control Indicator (**bold**)

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

SOLiD color space FASTA format

```
>1_51_64_F3
T1030103123033323320333300002112222 3
>1_51_127_F3
T2010323233203132310110100200310310 2
```

Each number can be replaced according to this table



MAQ Pileup example

```
…
emb|BA000018.3 36129  A   102  @...,,,..,,,,,,,,,,,.,...,,,.,,,..,,.,.,,,,....,.,,,,,.,,,,,,,,,.,,,,,.,....,.,,,,........,.....,,,,,.,,,,..
emb|BA000018.3 36130  A   103  @,,,..,,,,,,,,,,.,....,,,.,,,,..,,,..,,.,,,....,.,,,,,..,,,,,,,,,.,,,,,.,....,.,,,,..........,.....,,,,,.,,,,......
emb|BA000018.3 36131  T   100  @,,,,,,,,,.,....,,,.,,,..,,,..,.,,,....,.,,,,,.,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,...g.,,,,,.,,,,.......,.
emb|BA000018.3 36132  T   93   @,.,....,.,,,..,,,..,.,,,....,.,,,,,..,,,,,,,,.,,,,,.,,,,,.,....,.,,,,...........,.....,,,,,.,,,,......,..
emb|BA000018.3 36133  A   95   @,...,,.,,,,..,,,..,,.,,,....,.,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,
emb|BA000018.3 36134  G   98   @,...,,.,,,,..,,,..,,.,,,....,.,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,...
emb|BA000018.3 36135  T   99   @,...,,.,,,,..,,,..,,.,,,....,.,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,.....G,G,,.,,,,......,..,,,,,...,
emb|BA000018.3 36136  C   97   @,,.,,,..,,..,.,,,....,.,,,,,..,,,,,,,,,t,....,.,,,,,,.,.,,,,,...........,........,,,,,.,,,,......,..,,,,,..,,.
emb|BA000018.3 36137  T   96   @,,.,,,..,,.,,,,,....,.,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,
emb|BA000018.3 36138  A   96   @,,..,,,,....,,,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,,...
emb|BA000018.3 36139  T   93   @.,,,,....,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,....
emb|BA000018.3 36140  C   94   @,,,....,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....
emb|BA000018.3 36141  A   97   @,,,....,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.
emb|BA000018.3 36142  A   100  @....,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,...
emb|BA000018.3 36143  A   102  @...,.,,,,,..,,,,,,,,.,,,,,.,....,.,,,,.........,,,,C,......,..,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,...
emb|BA000018.3 36144  A   102  @,.,,,,..,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,.,.
emb|BA000018.3 36145  G   102  @TTTTTtTTTTTTTTTtTTTTTtTttttTtTTTTttttttttttTtttttTTTTTtTTTtttttttTttTTTTtttTTtTTTTttttTttTTtTTtttTttTTtTT
emb|BA000018.3 36146  A   103  @,,..,,,,,,,,,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,.,,,,
emb|BA000018.3 36147  A   105  @,,..,,,,,,,,,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,..g.,..,,,,,.,,,,
emb|BA000018.3 36148  A   108  @,..,,,,,,,,,,,,..,,,,,,,,.,,,,,.,.........,C,,.,,,,,.......,..,,,,,.,,,,......,,t.,,,,.......,,,,,..,,,,,,,,,,,.
emb|BA000018.3 36149  G   110  @..,,,,,,,,,,,,.,....,.,,,,,.........c,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,.,,,,,,.,
emb|BA000018.3 36150  G   113  @,,,,,,,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,.,,,,,,,,,.,.
emb|BA000018.3 36151  G   109  @.,,,,.,....,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,...,.
emb|BA000018.3 36152  G   110  @,,,,.,....,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,...,,
emb|BA000018.3 36153  T   111  @,.,....,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,.......,,,,,,,C.,,,,,...,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,...,,,,
emb|BA000018.3 36154  T   110  @,....,,,,,,..,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,...,,,..
emb|BA000018.3 36155  G   111  @,.,,,,........,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,...,,,,,
emb|BA000018.3 36156  G   116  @,,,,.........,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,,...,,,,,
emb|BA000018.3 36157  G   112  @.....,.......,,,,,,,,.,,,,,.,....,.,,,,...........,,,,,,.,t.,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,...,,...
emb|BA000018.3 36158  A   108  @,.........,,,,,,,,.,,,,,.,....,.,,,,.........,,g,,....C..,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,.......,,..
emb|BA000018.3 36159  C   111  @.......,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,,.......,,,,,,,
emb|BA000018.3 36160  T   113  @...,,,,,,,,.,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,.N.......,,,,,,,,,,
emb|BA000018.3 36161  G   114  @,,,,,,,.........,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,,.......,,,,,,,
emb|BA000018.3 36162  T   116  @,,,,,.........,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,,.........,,,,,,,
emb|BA000018.3 36163  T   120  @,,,.......,,,,,.,....,.,,,,...........,........,,,,,.,,,,......,..,,,,,..,,,,.....,,.,,,,,,,,,,,,.........,,,,,,,...
…
```

SAM/BAM formats

- Here is an example of an SAM file:

```
@HD VN:1.0
@SQ SN:chr20 LN:62435964
@RG ID:L1 PU:SC_1_10 LB:SC_1 SM:NA12891
@RG ID:L2 PU:SC_2_12 LB:SC_2 SM:NA12891
read_28833_29006_6945 99  chr20 28833 20 10M1D25M = 28993 195 AGCTTAGCTAGCTACCTATATCTTGGTCTTGGCCG
     <<<<<<<<<<<<<<<<<<<<:<9/,&,22;;<<< NM:i:1 RG:Z:L1
read_28701_28881_323b 147 chr20 28834 30 35M      = 28701 -168 ACCTATATCTTGGCCTTGGCCGATGCGGCCTTGCA
     <<<<<;<<<<7;:<<<6;<<<<<<<<<<<7<<<< MF:i:18 RG:Z:L2
```

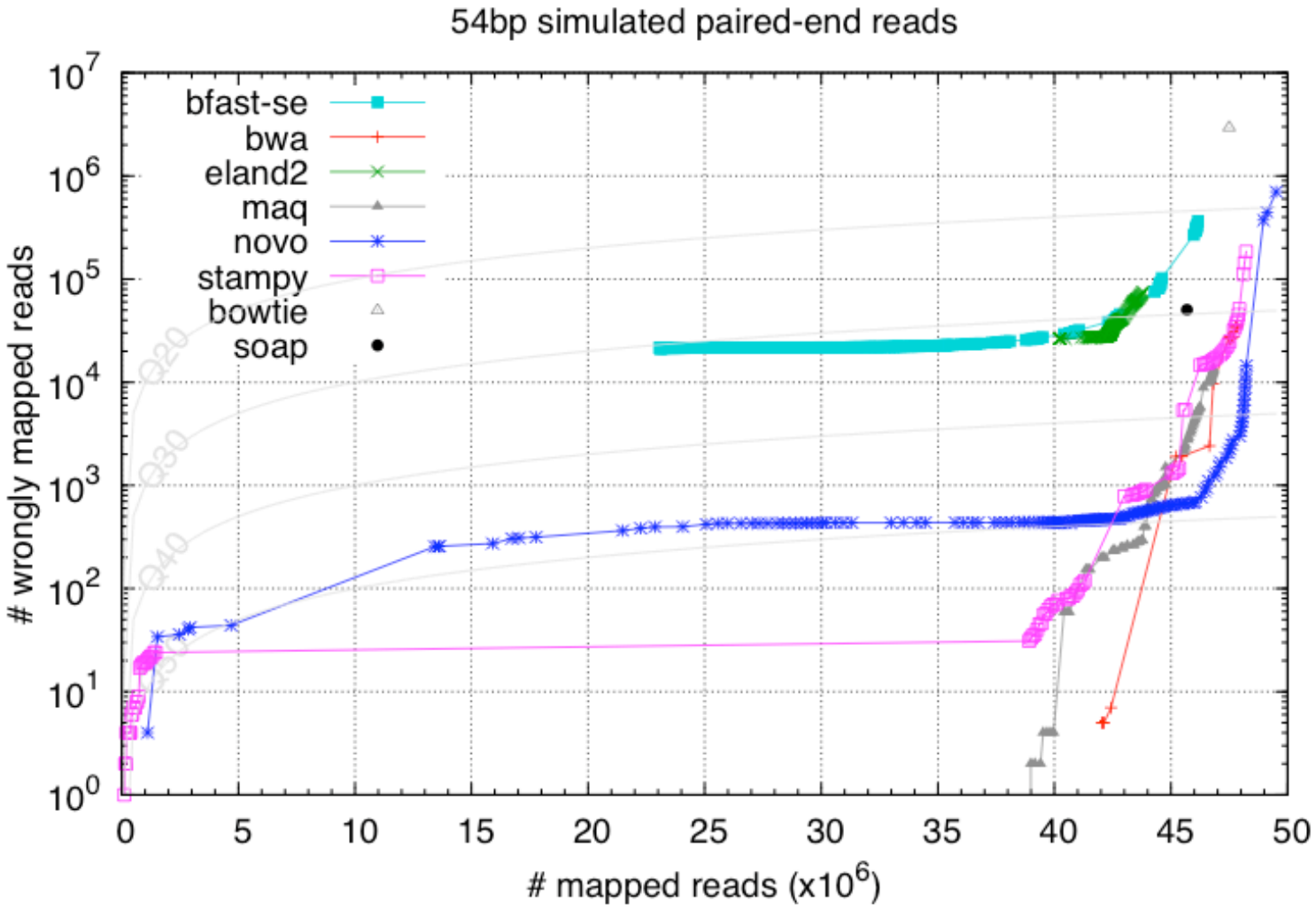
- BAM is the binary version of the same data

http://samtools.sourceforge.net/SAM1.pdf

Mapping Tools list (non-exhaustive)

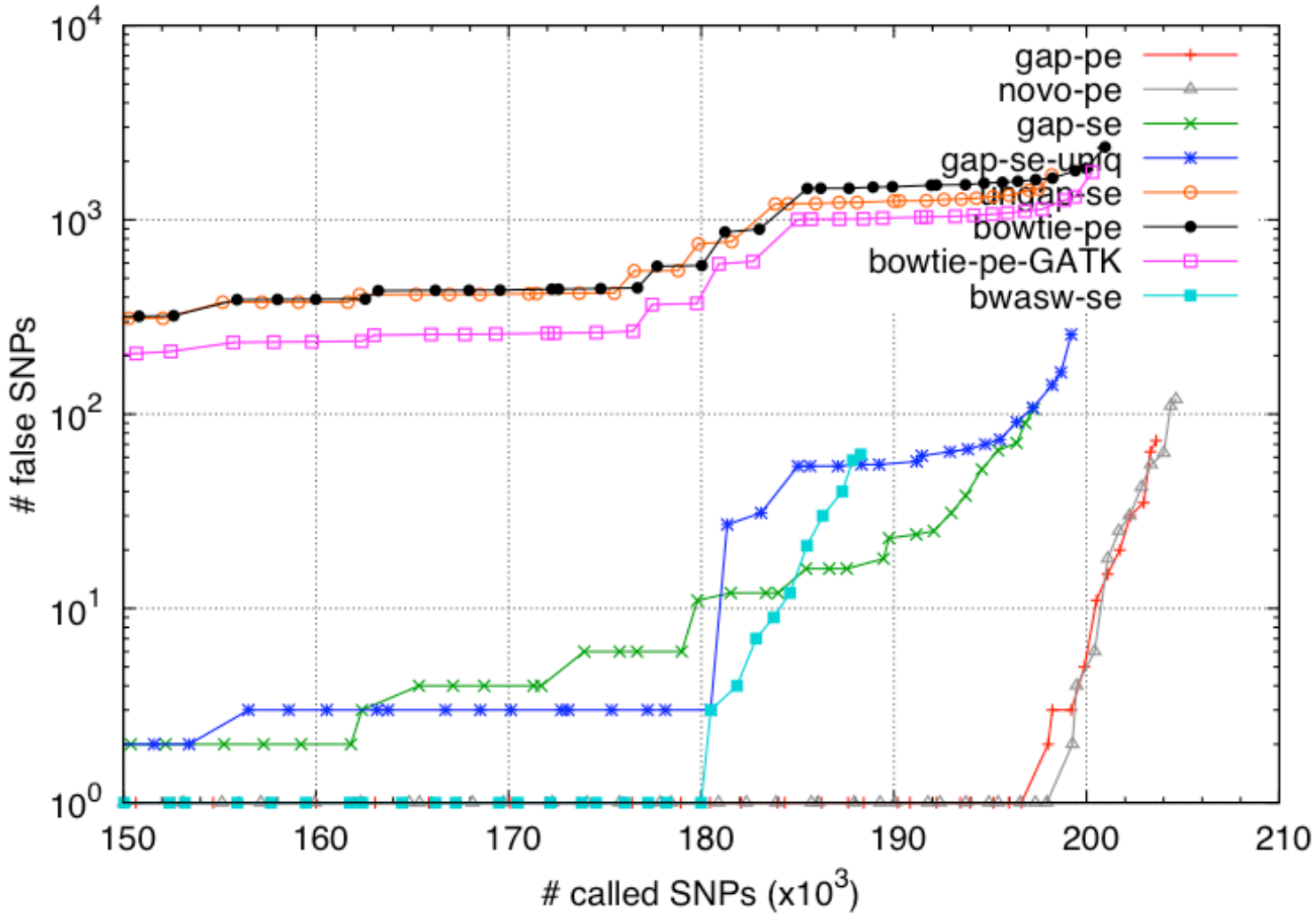
Tool	Open Source	Max read Length	Algorithm	SOLiD colorspace?
BFAST	Yes		Hash genome + SW	Yes
Bowtie	Yes		BWT	Yes
BWA	Yes	200 + more	BWT + SW	Yes
ELAND	Com.		Hash reads	No
MAQ	Yes	127	Hash reads	Yes
Mosaik	Yes		Hash genome + SW	Yes
Novoalign	Com.		Hash reads	No
RMAP	Yes	64	Hash reads	No
SHRiMP	Yes		Hash reads + SW	Yes
SOAP2	No	60	2way BWT	No
Zoom!	Com.	240	Hash reads	Yes

Accuracy varies…



Bala et al.

Alignment strategy and SNP calling



Choosing aligners

- There are many mapping aligners and they vary in performance (speed and memory usage)
- They also vary in accuracy
- In SNP calling, effective paired-end mapping and gapped alignment are essential to obtain high SNP accuracy

- A good compromise: BWA

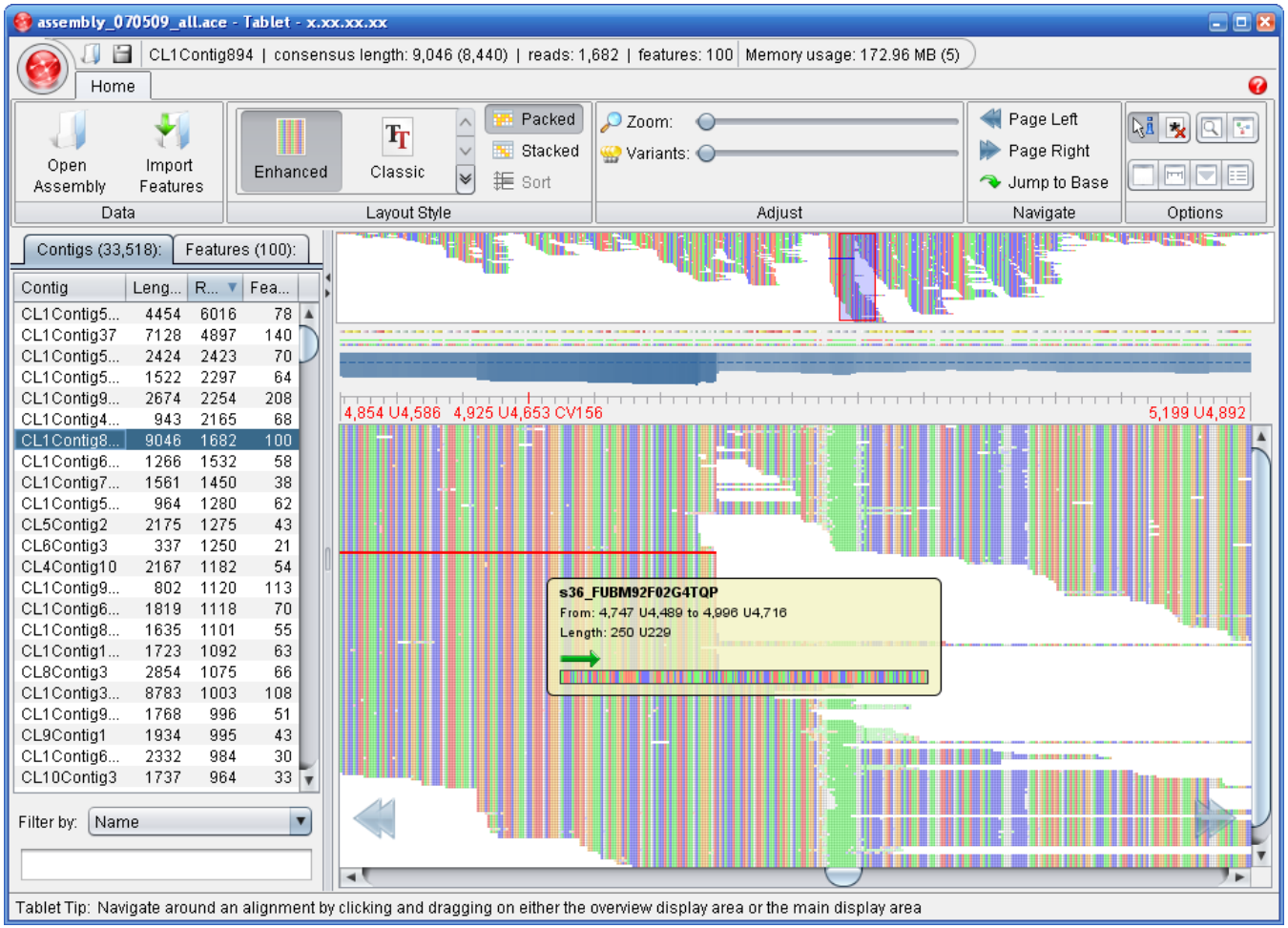
Software issues

- File formats jungle
  - Each software has its own internal formats, few comply with the emerging standards (BWA, Bowtie)
- Often single threads
  - Some software are multithreaded
- Difficult to identify insertions/deletions/inversions (> 10bp)
- Unfinished beta software or not maintained
- Poor visualization tools

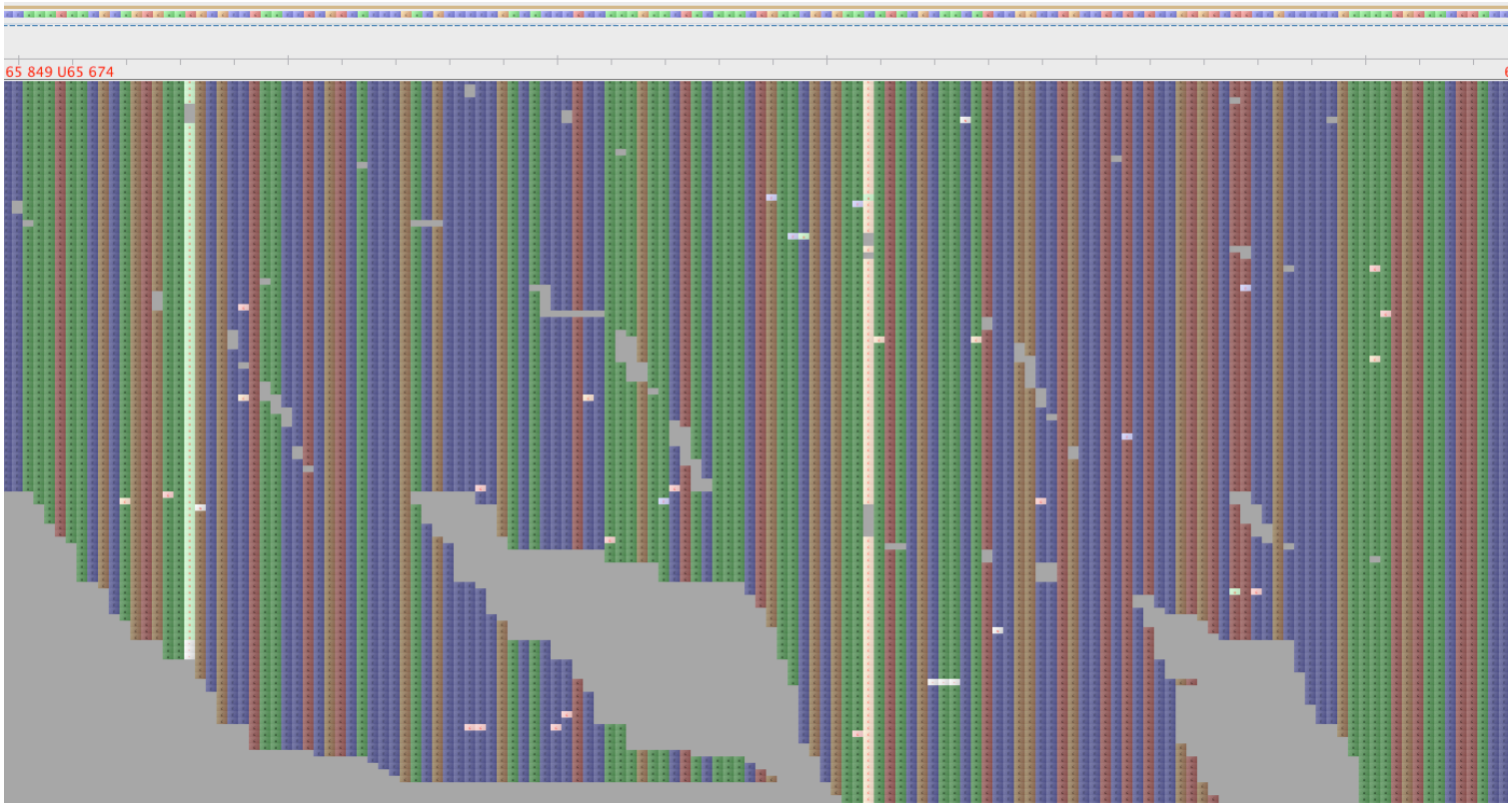
Visualization tools for assemblies

Tool	Windows	Linux	Mac	Input format
BAMview	Y	Y	Y	BAM
Consed/Gap5	N	Y (X11)	Y (X11)	ACE, MAQ, BAM
Eagleview	Y	Y	Y	ACE
Gambit	Y	Y	Y	BAM
Hawkeye	Y (cigwin)	Y	(Y)	afg (AMOS)
IGVviewer	Y	Y	Y	BAM, SAM, …
Tablet	Y	Y	Y	ACE, MAQ, BAM, afg, SAM, …

Tablet interface



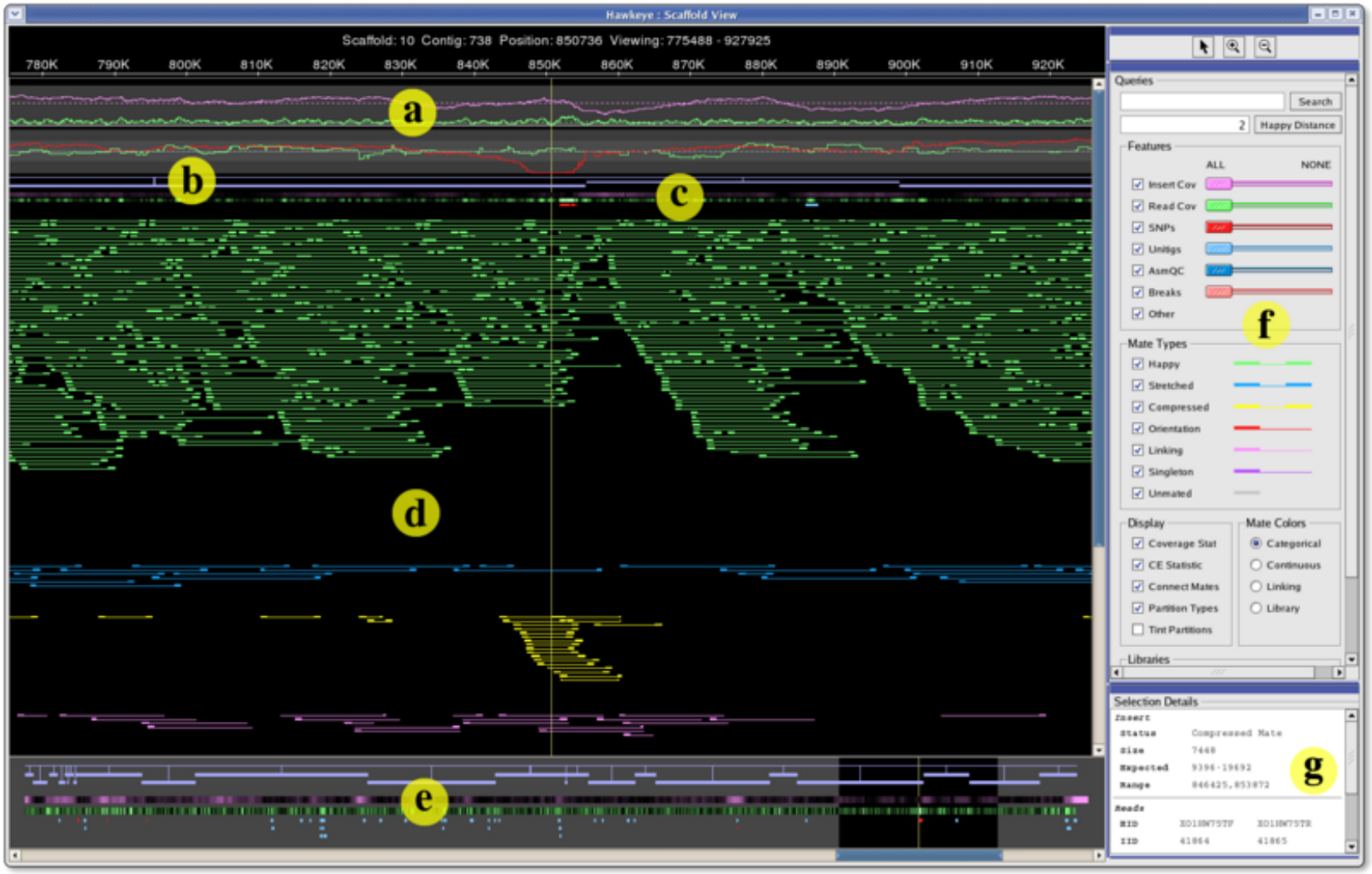
Tablet
visualization of the mapping and the SNPs



Mapping of the reads of a Staphylococcus aureus sequencing, showing 2 SNPs vs the reference genome.

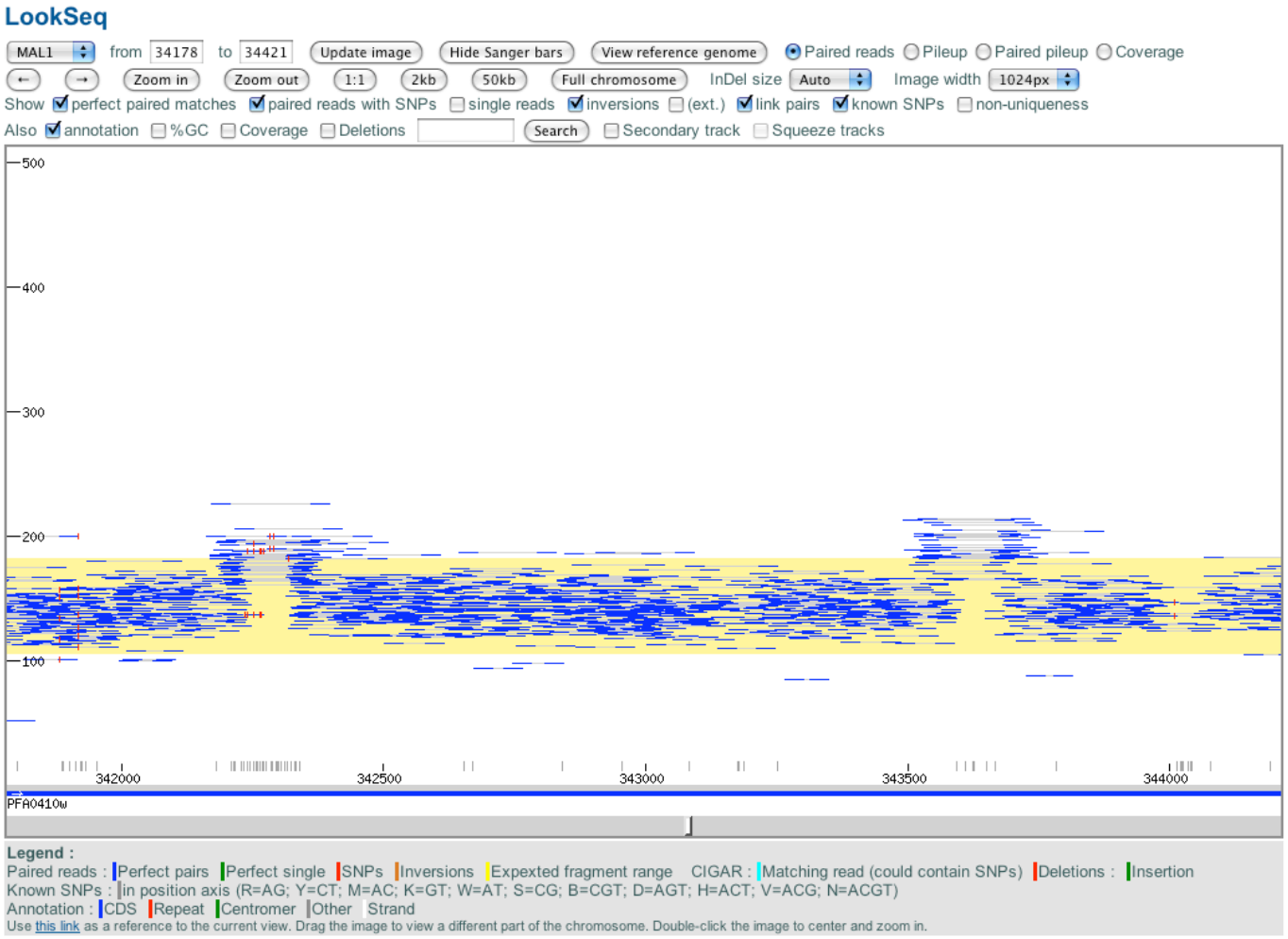
SIB Swiss Institute of Bioinformatics

Hawkeye scaffold view

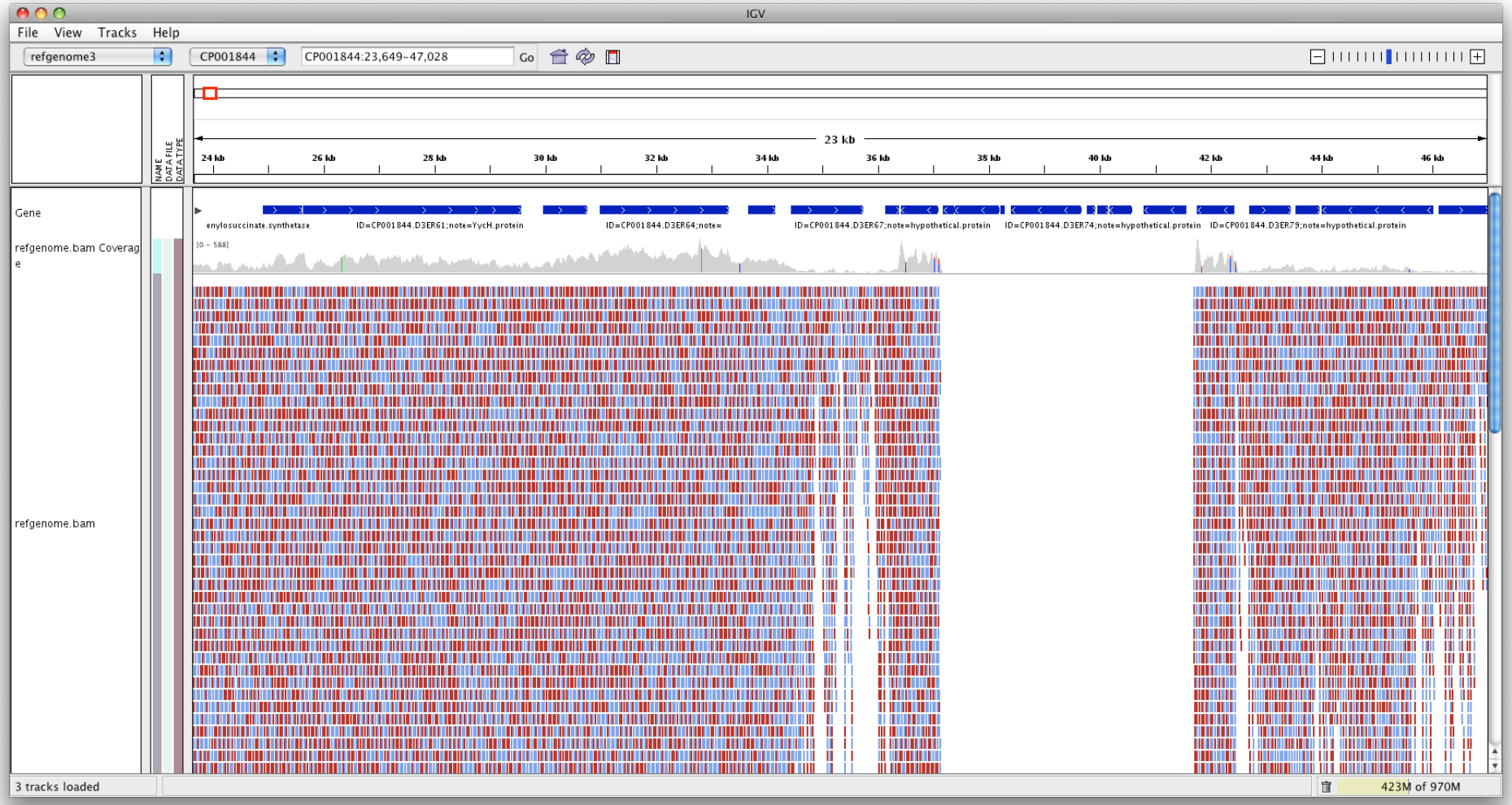


SIB Swiss Institute of Bioinformatics

Best view: LookSeq (also in BAMview)

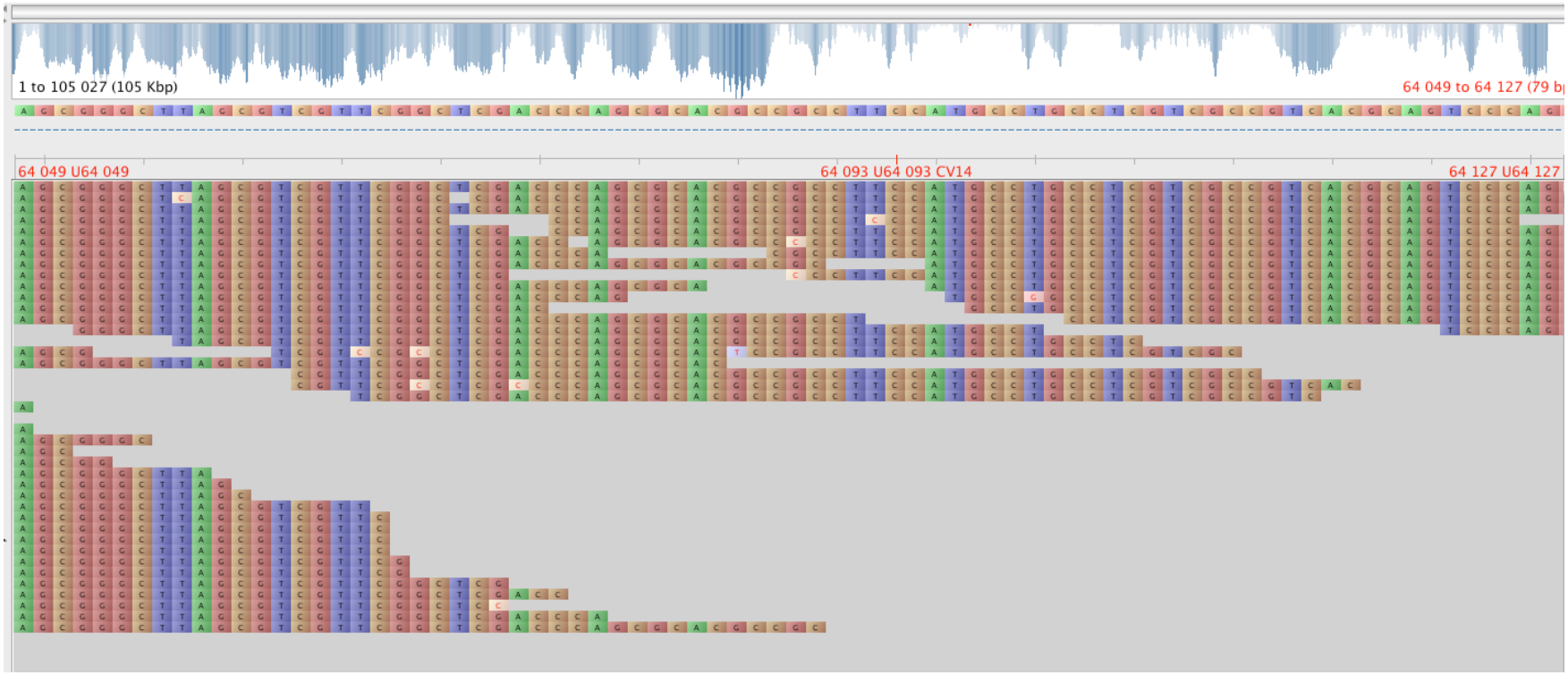


IGV (integrated genome viewer)

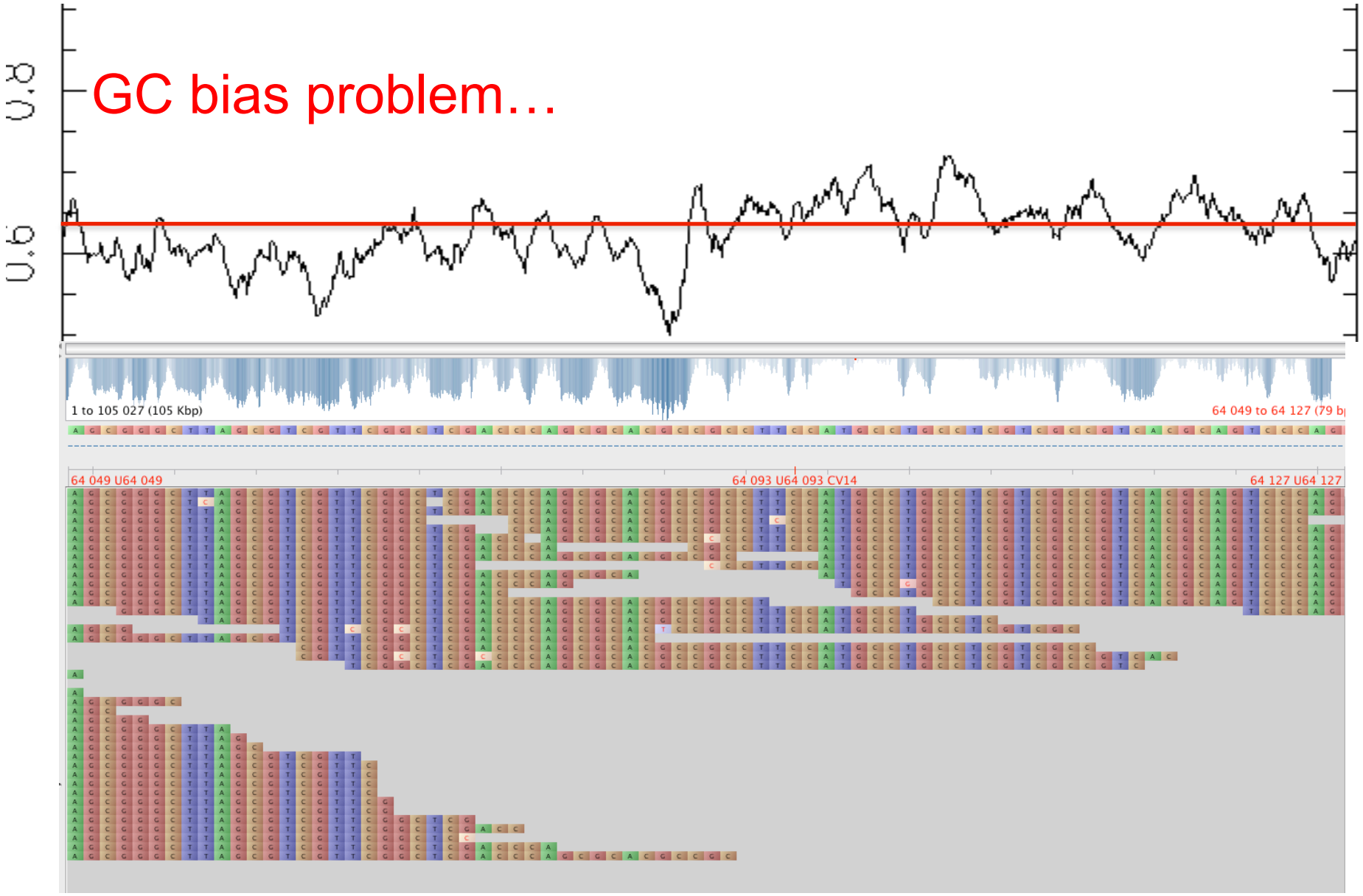


SIB Swiss Institute of Bioinformatics

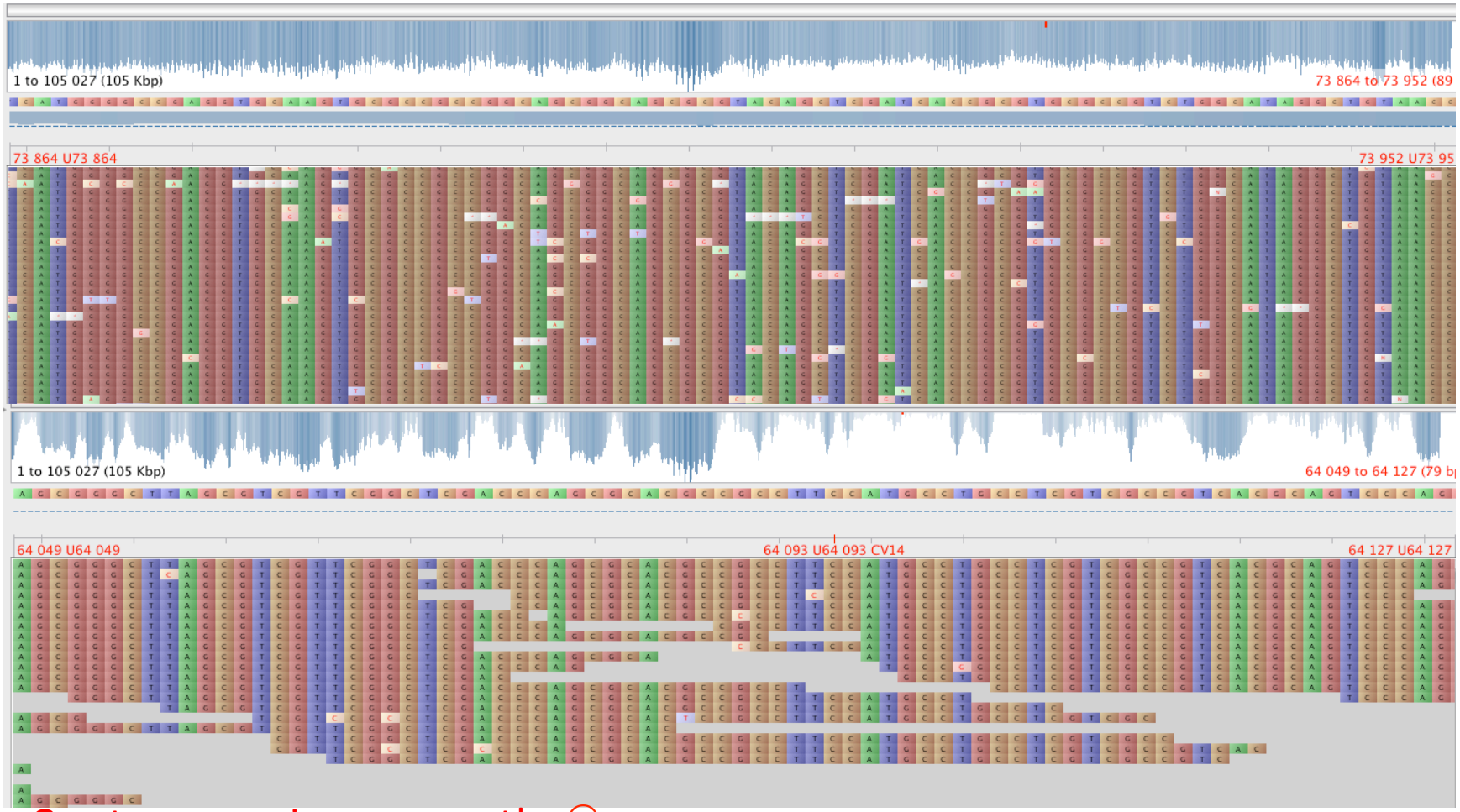
Coverage problem ? (mapping on genomic island ICEclc 100kbp)



GC bias problem…



New sequencing coverage of CLC 100kbp for P.knackmussii 2762 (top) vs B13 (bottom)



Great coverage improvement!... ☺

SIB Swiss Institute of Bioinformatics

The practicals

- http://edu.isb-sib.ch

- select «workshops» on the left menu
- select «Bogota NGS workshop»

- Login to bioinf-hpc server, then follow the instructions of the exercises

Thank you

Swiss Institute of
Bioinformatics

BWA

```
# index reference
bwa index reference.fa
# index reads
bwa aln -t 4 mybest.fa ../saureus_1.fq > saureus_1.sai
bwa aln -t 4 mybest.fa ../saureus_2.fq > saureus_2.sai
# map reads
bwa sampe -a 600 -P reference.fa saureus_1.sai saureus_2.sai ../
   saureus_1.fq ../saureus_2.fq > mybest.sam

# index reference
samtools faidx reference.fa
# convert SAM to BAM
samtools view -T reference.fa -b mybest.sam > mybest.bam
# sort BAM
samtools sort mybest.bam mybest.sorted.bam
# index BAM
samtools index mybest.sorted.bam
```

MAQ

```
#index the reference
maq fasta2bfa reference.fa genomeref.bfa
#index the reads
maq fastq2bfq S6out_1.fastq S6out_1.bfq
maq fastq2bfq S6out_2.fastq S6out_2.bfq
#map the reads in paired-end
maq map -a 600 -1 36 -2 36 S6.map genomeref.bfa S6out_1.bfq S6out_2.bfq
# get the consensus
maq assemble -p S6.cns genomeref.bfa S6.map
maq cns2fq S6.cns > S6consensus.fq  ## warning need to convert to fasta
### Find SNPs
maq cns2snp S6.cns > S6.snp
## filter quality
maq.pl SNPfilter -d 50 -w 20 -q 40 S6.snp > S6.fil.snp
### pileup the reads
maq pileup -p -m 2 genomeref.bfa S6.map > S6.pileup
```

Bowtie

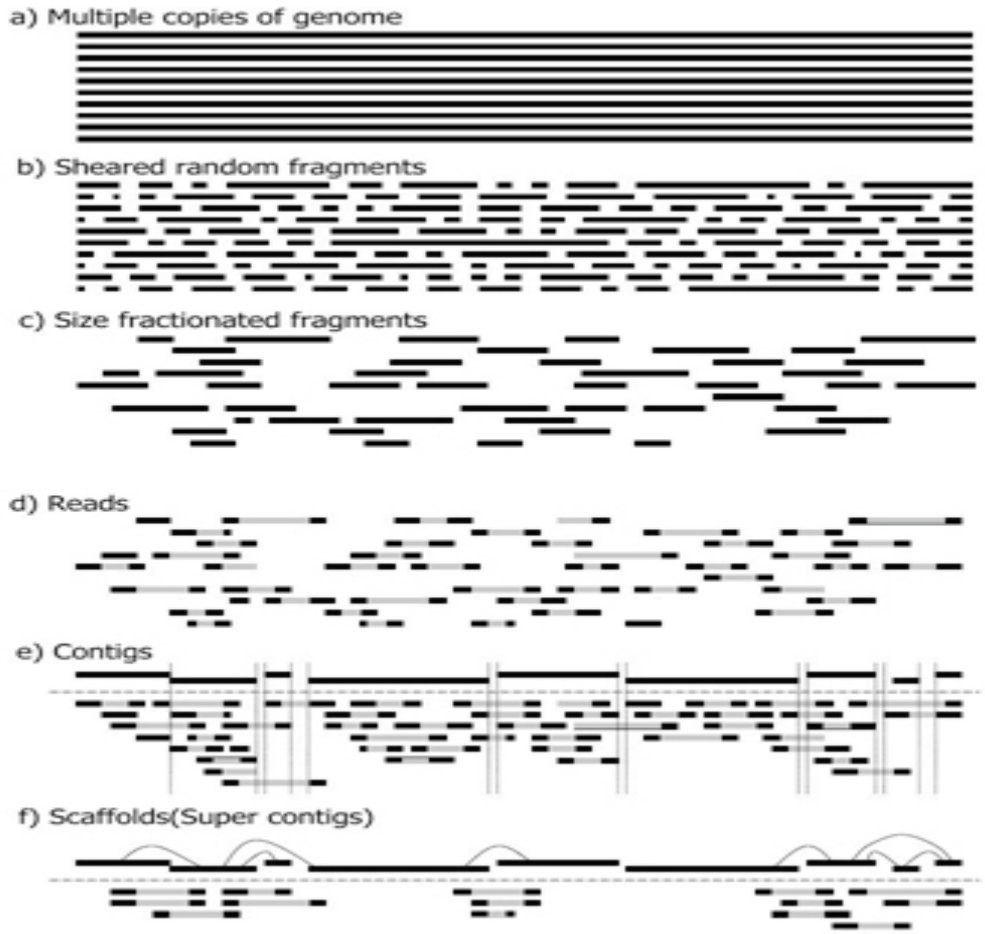
```
# index reference
bowtie-build –f reference.fa Saureus
# map reads
bowtie -n 1 -l 36 -I 400 -X 700 –un unmapped -p 10 Saureus -1
   s_1_1_sequence.txt -2 s_1_2_sequence.txt > mybest.bwtmap

# convert to SAM & BAM
# index reference
samtools faidx reference.fa
# convert bowtie to SAM
bowtie2sam.pl mybest.bwtmap > mybest.sam
# convert SAM to BAM
samtools view -T reference.fa -b mybest.sam > mybest.bam
# sort BAM
samtools sort mybest.bam mybest.sorted.bam
# index BAM
samtools index mybest.sorted.bam
```

Next Generation Sequencing: *de novo* Genome Assembly

Laurent Falquet, Vital-IT
Bogota, March 22, 2011

Swiss Institute of Bioinformatics

Ultra High Throughput Sequencing (WGS)



- http://www.k.u-tokyo.ac.jp/pros-e/person/shinichi_morishita/shinichi_morishita.htm

Ultra High Throughput Sequencing and Genome Assembly: a Simple Jigsaw Puzzle?

- Yes, but you must deal with
  - Millions of pieces
  - Lots of malformed pieces
  - Often missing pieces
  - Pieces mixed from another puzzle
  - Lots of identical blue sky pieces…
  - If *de novo* you…

Genome assembly, deep blue…

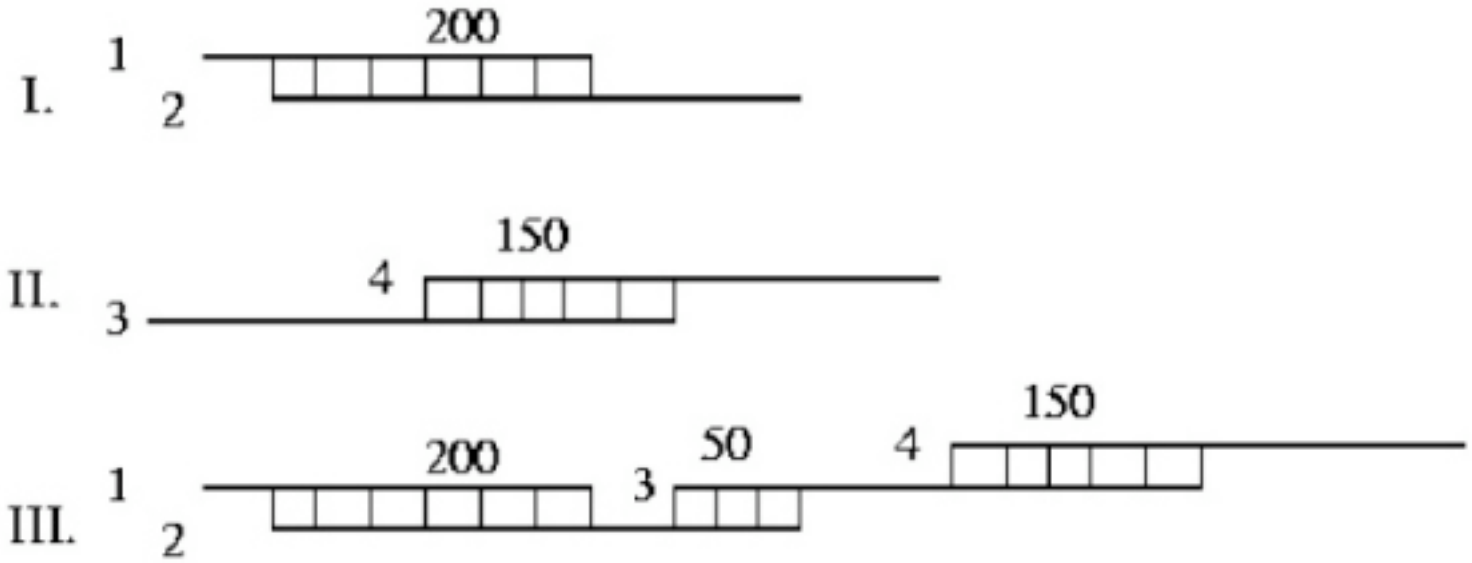
…don't even know the final picture…

Algorithms for assembly

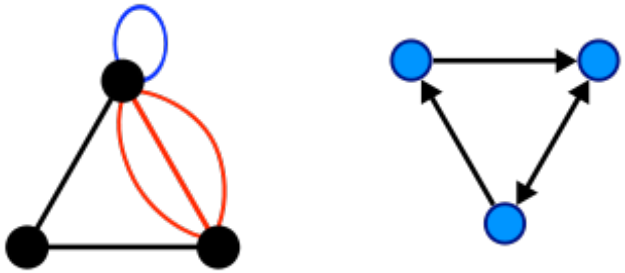
- Greedy
  - Phrap, Cap3, TIGR assembler, …
- Overlap-layout-consensus
  - Celera wgs Assembler, Phusion, MIRA3, Edena …
- Eulerian path
  - Euler-SR, Velvet, ABySS, SOAPdenovo, VCAKE, …
- Align-layout-consensus (mapping)
  - Projector2, Mozaik, MAQ, Bowtie, BWA, SOAP2, Novoalign, ELAND, MUMmer, …
- Bac-by-Bac
  - Atlas, …

Greedy

- Greedy assemblers - The first assembly programs followed a simple but effective strategy in which the assembler greedily joins together the reads that are most similar to each other.
- An example is shown below, where the assembler joins, in order, reads 1 and 2 (overlap = 200 bp), then reads 3 and 4 (overlap = 150 bp), then reads 2 and 3 (overlap = 50 bp) thereby creating a single contig from the four reads provided in the input. One disadvantage of the simple greedy approach is that because local information is considered at each step, the assembler can be easily confused by complex repeats, leading to mis-assemblies.



Graph theory

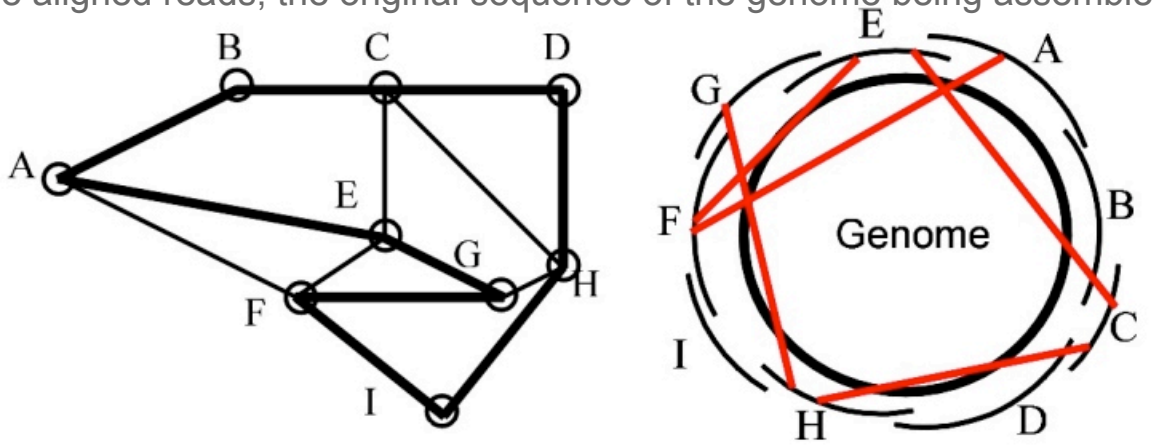


- A graph refers to a collection of **vertices** (or '**nodes**') and a collection of **edges** (or '**vectors**') that connect pairs of vertices.
- A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another (digraph).

http://en.wikipedia.org/wiki/Graph_(mathematics)

Overlap-layout-consensus

- Overlap-layout-consensus - The relationships between the reads provided to an assembler can be represented as a graph, where the nodes represent each of the reads and an edge connects two nodes if the corresponding reads overlap. The assembly problem thus becomes the problem of identifying a path through the graph that contains all the nodes - a Hamiltonian path (Figure below). This formulation allows researchers to use techniques developed in the field of graph theory in order to solve the assembly problem.
- An assembler following this paradigm starts with an **overlap stage** during which all overlaps between the reads are computed and the graph structure is computed. In a **layout stage**, the graph is simplified by removing redundant information. Graph algorithms are then used to determine a layout (relative placement) of the reads along the genome. In a final **consensus stage**, the assembler builds an alignment of all the reads covering the genome and infers, as a consensus of the aligned reads, the original sequence of the genome being assembled.

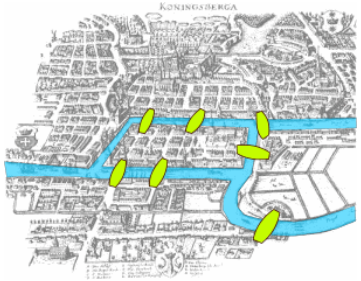


Overlap graph for a bacterial genome. The thick edges in the picture on the left (a Hamiltonian cycle) correspond to the correct layout of the reads along the genome (figure on the right). The remaining edges represent false overlaps induced by repeats (exemplified by the red lines)

Leonhard Euler
1707 - 1783

- Swiss mathematician
- Euler's identity, the most famous formula!

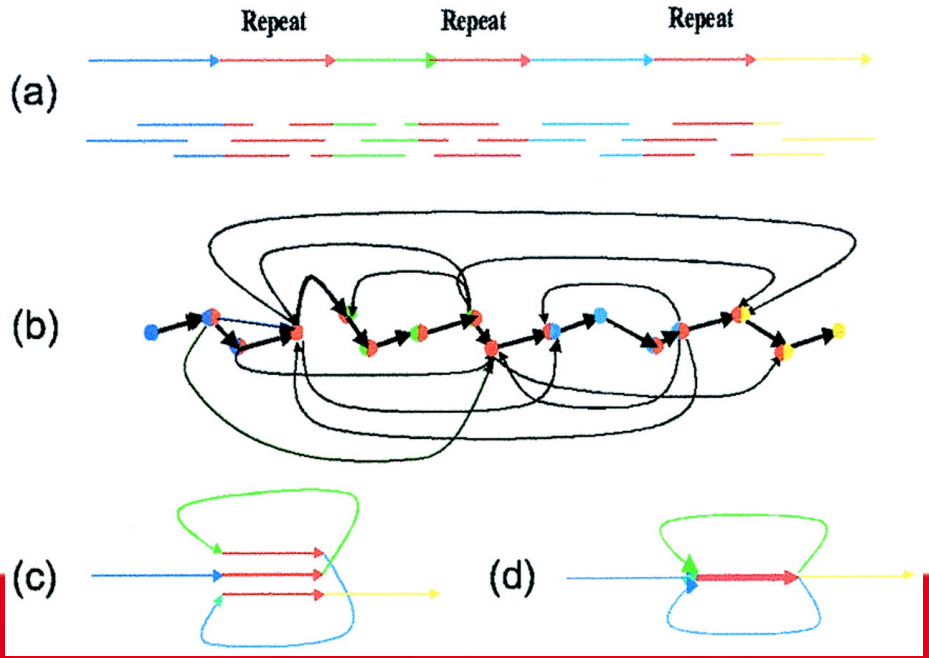
$$e^{i\pi} + 1 = 0$$



- Graph theory
  - In 1736, Euler solved the problem known as **the Seven Bridges of Königsberg**. The city of Königsberg, Prussia was set on the Pregel River, and included two large islands which were connected to each other and the mainland by seven bridges.
  - The problem is to decide whether it is possible to follow a path that crosses each bridge exactly once and returns to the starting point. It is not: there is no Eulerian circuit. This solution is considered to be the first theorem of graph theory, specifically of planar graph theory.

Eulerian path

- **Eulerian path** approaches are based on early attempts to sequence genomes through a technique called sequencing by hybridization. In this technique, instead of generating a set of reads, scientists identified all strings of length k (k-mers) contained in the original genome.
- This approach, also based on a graph-theoretic model, breaks up each read into a collection of overlapping k-mers. Each k-mer is represented in a graph as an edge connecting two nodes corresponding to its k-1 bp prefix and suffix respectively. It is easy to see that, in the graph containing the information obtained from all the reads, a solution to the assembly problem corresponds to a path in the graph that uses all the edges - an Eulerian path.
- One advantage of the Eulerian approach is that repeats are immediately recognizable while in an overlap graph they are more difficult to identify.



Eulerian *vs* Hamiltonian path ?

- Both definitions are very similar:
  - a **Hamiltonian** path visits every vertex exactly once.
  - an **Eulerian** path visits every edge exactly once.
  - a **de Bruijn** graph is Eulerian **and** Hamiltonian.

- In practice, however, it is much more difficult to construct a Hamiltonian path or determine whether a graph is Hamiltonian, as that problem is NP-complete.

http://en.wikipedia.org/wiki/Hamiltonian_path
http://en.wikipedia.org/wiki/Eulerian_path

Limitations of the sequence

- Repeats
  - Transposases, IS-Elements, retroviruses, duplications, etc.
- Polymorphisms
  - SNPs, CNVs, multiploids, sample mixture, etc.
- Sequence bias
  - %GC

Repeats are a major issue for all assemblers

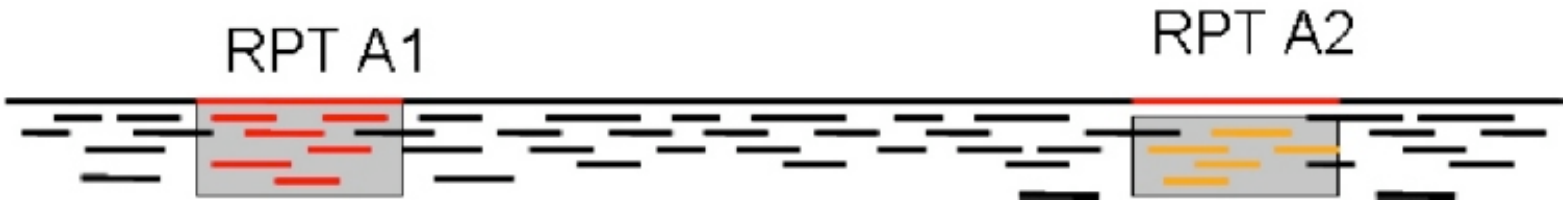


Figure top. Two copies of a repeat along a genome. The reads colored in red and those colored in yellow appear identical to the assembly program.

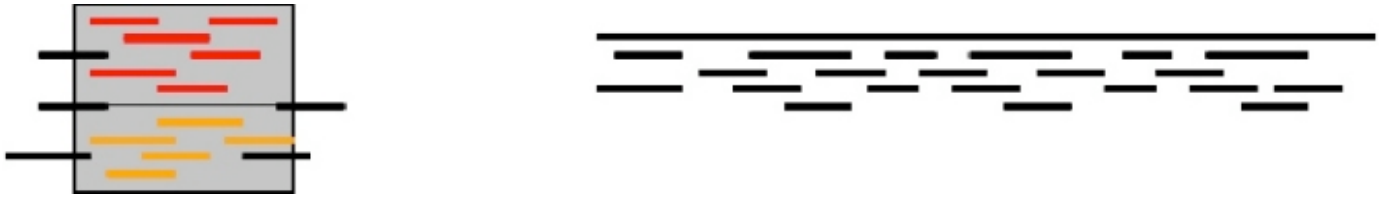


Figure bottom. Genome mis-assembled due to a repeat. The assembly program incorrectly combined the reads from the two copies of the repeat leading to the creation of two separate contigs.

Helping the assembly with linked reads

- When the **distance** and the **orientation** between 2 reads is known
- First proposed by
  - Edwards, A; Caskey, T (1991). "Closure strategies for random DNA sequencing". Methods: A Companion to Methods in Enzymology 3 (1): 41–47. doi:10.1016/S1046-2023(05)80162-8.
- Also called
  - Double-barreled
  - Mate-pairs
  - Paired-ends

Mate pairs validation example

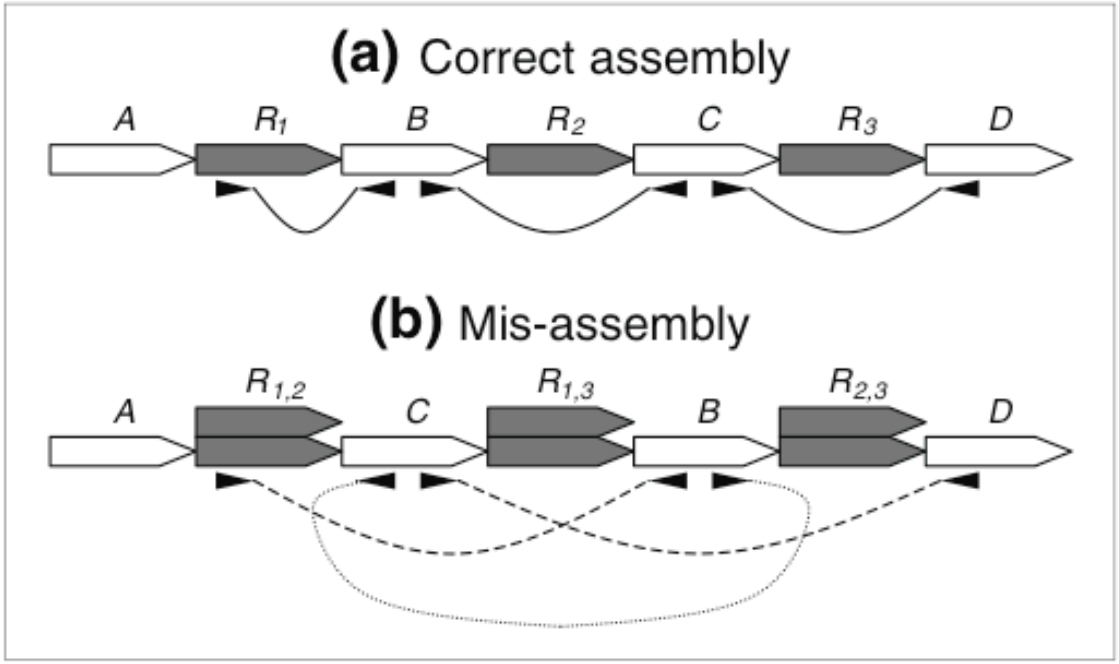
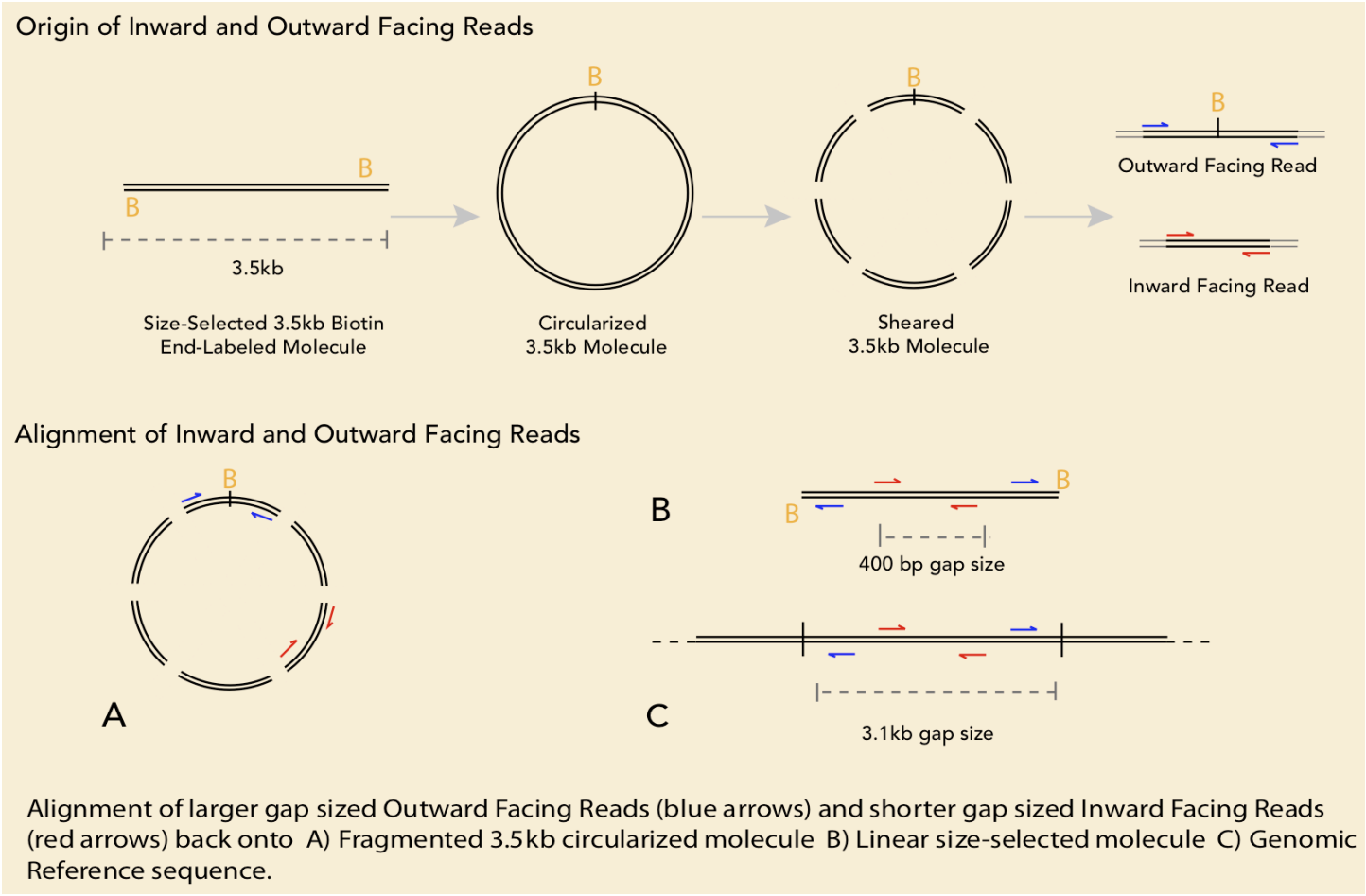


Figure 3
Mate-pair signatures for rearrangement style mis-assemblies. **(a)** Three copy repeat *R*, with interspersed unique sequences *B* and *C*, shown with properly sized and oriented mates. **(b)** Mis-assembled repeat shown with mis-oriented and expanded mate-pairs. The mis-assembly is caused by co-assembled reads from different repeat copies, illustrated by the stacked repeat blocks.

- 4 main criterias
  - Mates too close to each other
  - Mates too far from each other
  - Mates with same orientation
  - Mates pointing away from each other
- Other criterias
  - Mates not present on the assembly (singletons)
  - Mates on different contigs

Mate-pair Illumina

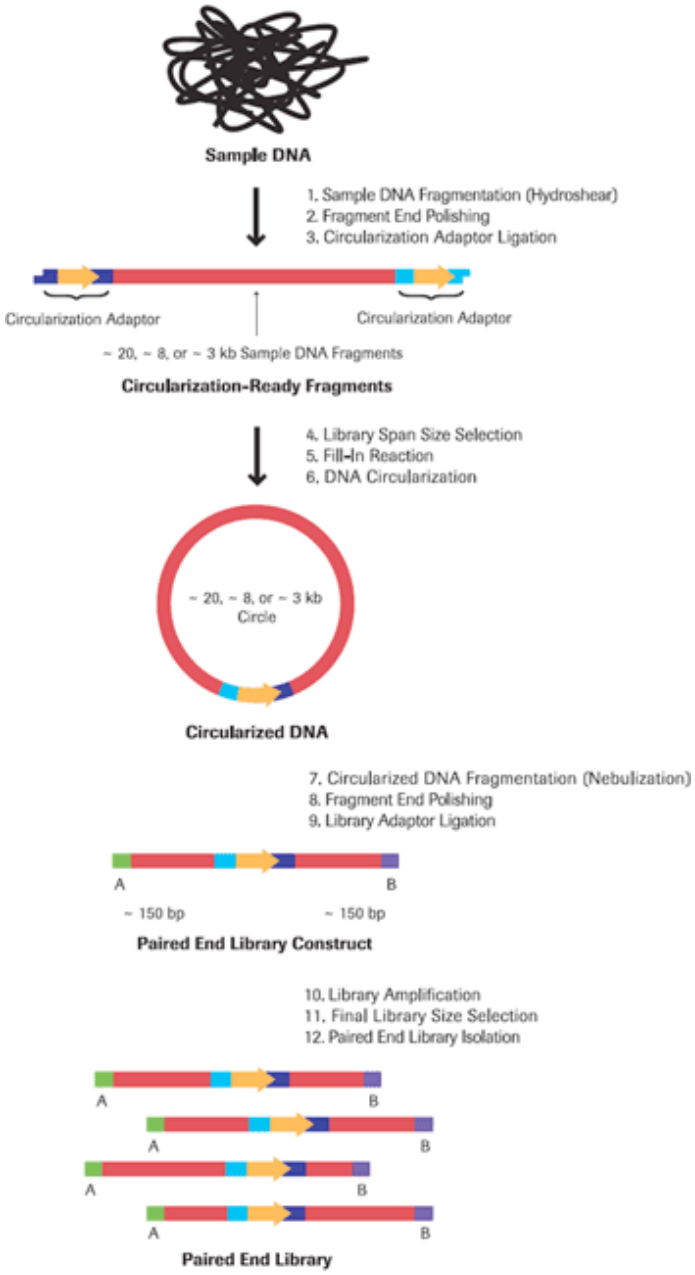


Alignment of larger gap sized Outward Facing Reads (blue arrows) and shorter gap sized Inward Facing Reads (red arrows) back onto A) Fragmented 3.5kb circularized molecule B) Linear size-selected molecule C) Genomic Reference sequence.

- Chimeric sequences undetectable + mixture of PE and MP
- High bias due to size selection -> clonal sequencing

Mate-pair 454

- Advantage
  - Known adaptor sequence
  - Little bias due to size



Paired-end summary

- Illumina: paired-end 200-500bp
- Illumina: mate-pair 3Kbp
- 454: mate pair 140bp, insert any size (3Kbp, 8Kbp, 20Kbp)
- SOLiD: paired-end 50+25bp, insert 200-600bp
- SOLiD: mate pair 50+25bp, insert 600bp-10Kbp

Tools for *de novo* sequence assembly (non-exhaustive list)

- ABySS
- Velvet
- SOAPdenovo
- Euler
- Edena
- Newbler
- MIRA
- WGS(Celera)
- Amos
- Phusion
- Phrap
- Cap3

- Mummer
- SAMtools
- BreakDancer
- Eagleview
- Hawkeye
- Tablet

Software issues

- File formats jungle
  - Each software has its own internal formats, few comply with the emerging standards
- Parameters tuning
  - Several parameters must be tuned, in particular the Kmer
- Large memory requirements
  - Some software might require hundreds of Gbytes
- Often single threads
  - Few of the software are multithreaded
- Unfinished beta software
- Poor visualization

File formats jungle

- **.fasta**
- .qual (phred quality file)
- .fastq
- .sff (454 binary data file, Standard Flowgram Format)
- .srf (sequence read format) plateform independent format
- .txt (Illumina/Solexa files) (FastQ-like)
- .csfasta (SOLiD color space)

- Paired-end
  - 2 files
  - crossbow style

- Output
  - fasta
  - SAM/BAM
  - afg
  - Other files (stats)

De novo assembly

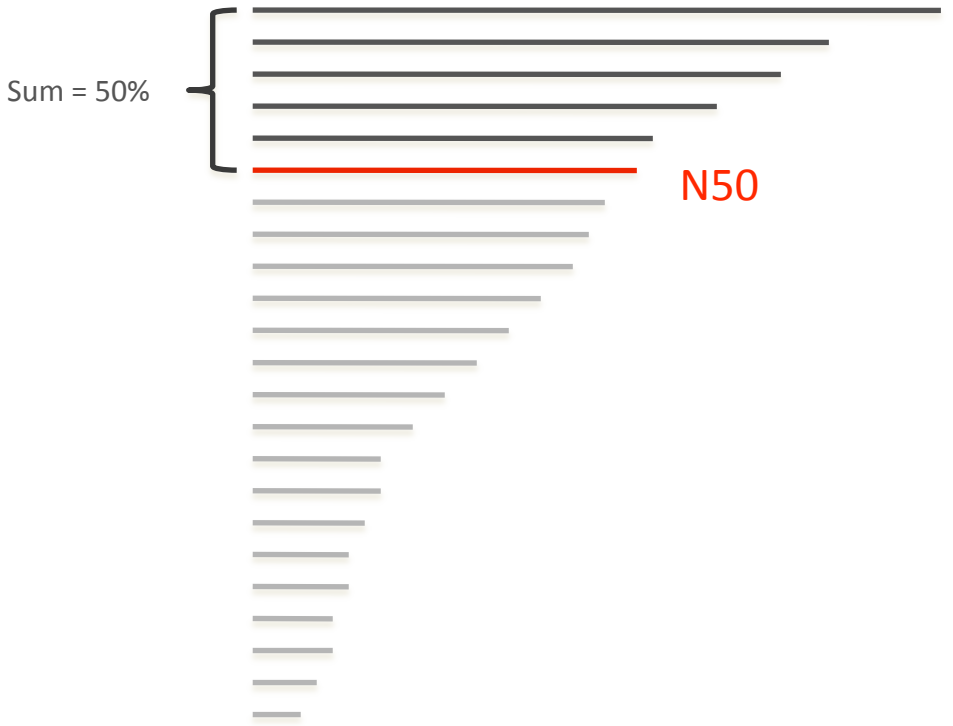
- Current trend: start with small inserts paired-end and add larger inserts sequentially
- Do not mix all reads (454, Illumina, SOLiD, etc…)
- Assemble them separately or sequentially
  - 454 with newbler
  - Illumina with SOAPdenovo, ABySS or Velvet
  - SOLiD with Velvet or ABySS
- Combine assemblies (not recommended)
  - With newbler, SOAPdenovo, CAP3, Phrap, etc…

Assembly quality measurements

- Number of contigs
  - Ideally 1 for a bacterial genome…, but the lower the better
- Contig sizes
  - The larger the better (up to the size of the genome), usually given in maximum, minimum and average lengths.
- Correctness
  - Difficult to assess for a new genome
- N50
  - **The most used quality value for *de novo* assembly**
  - The N50 is the size of the smallest of all the large contigs covering 50% of the genome

N50 what's that?

- Sort the contigs by size
- Sum them starting with the largest until you reach 50% of the estimated genome size
- Last contig added = N50



SIB Swiss Institute of Bioinformatics

Velvet for S5

K =	21	23	25	27	29	31
Nr contigs	7012	1906	767	420	325	252
Consensus size bp	3103135	2918437	2875773	2863169	3619536	2936521
N50	12182	43427	67361	66898	66306	107440
Min	41	45	49	53	57	61
Max	161171	201664	201396	201389	238872	369778

ABySS for S5

K =	21	23	25	27	29	31
Nr contigs	4318	2891	2123	1636	1339	1113
Consensus size bp	3220552	3127361	3088161	3049081	3078819	3052504
N50	15928	25693	29334	30241	31596	29797
Min	21	23	25	27	29	31
Max	63797	132812	132816	132992	132996	122383

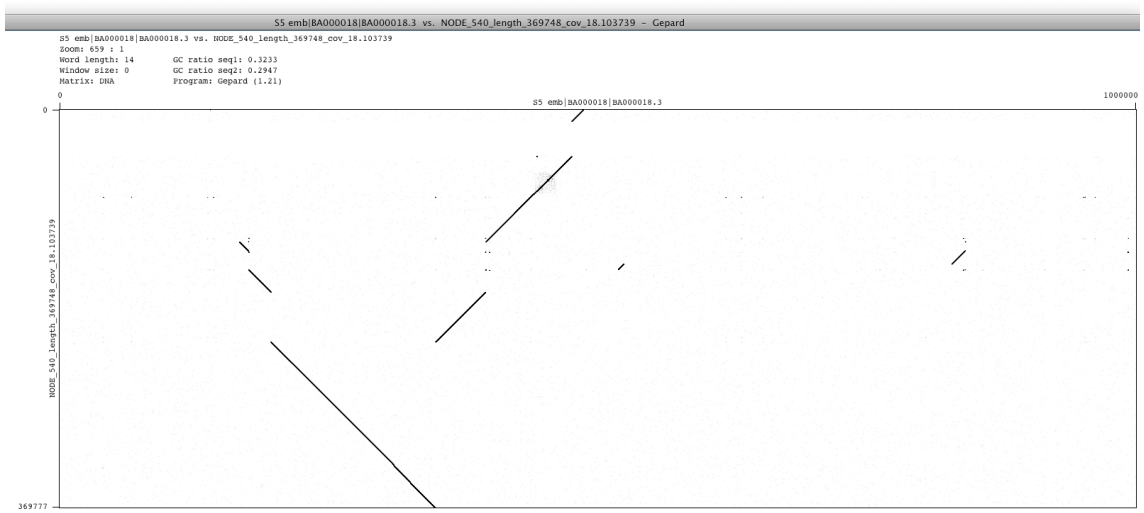
SOAPdenovo for S5

K =	21	23	25	27	29	31
Nr contigs	247	213	234	248	280	362
Consensus size bp	2818333	2825424	2831502	2833334	2828297	2785031
N50	98956	99286	82319	82910	84517	52098
Min	100	100	100	100	100	100
Max	253458	252985	181975	182194	182217	141106

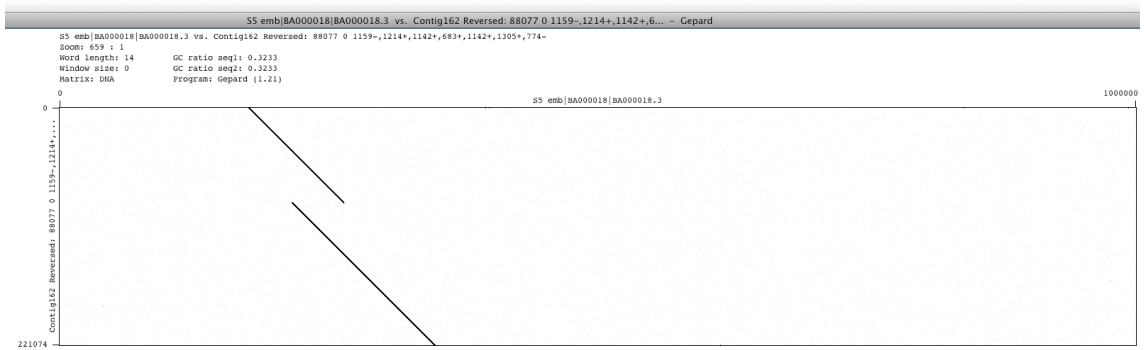
Best scores

K =	Velvet31	ABySS29	SOAPdenovo23
Nr contigs	252	1339	213
Consensus size bp	2936521	3078819	2825424
N50	107440	31596	99286
Min	61	29	100
Max	369778	132996	252985

However longest contig is not always the best…



- Velvet: 369’778bp



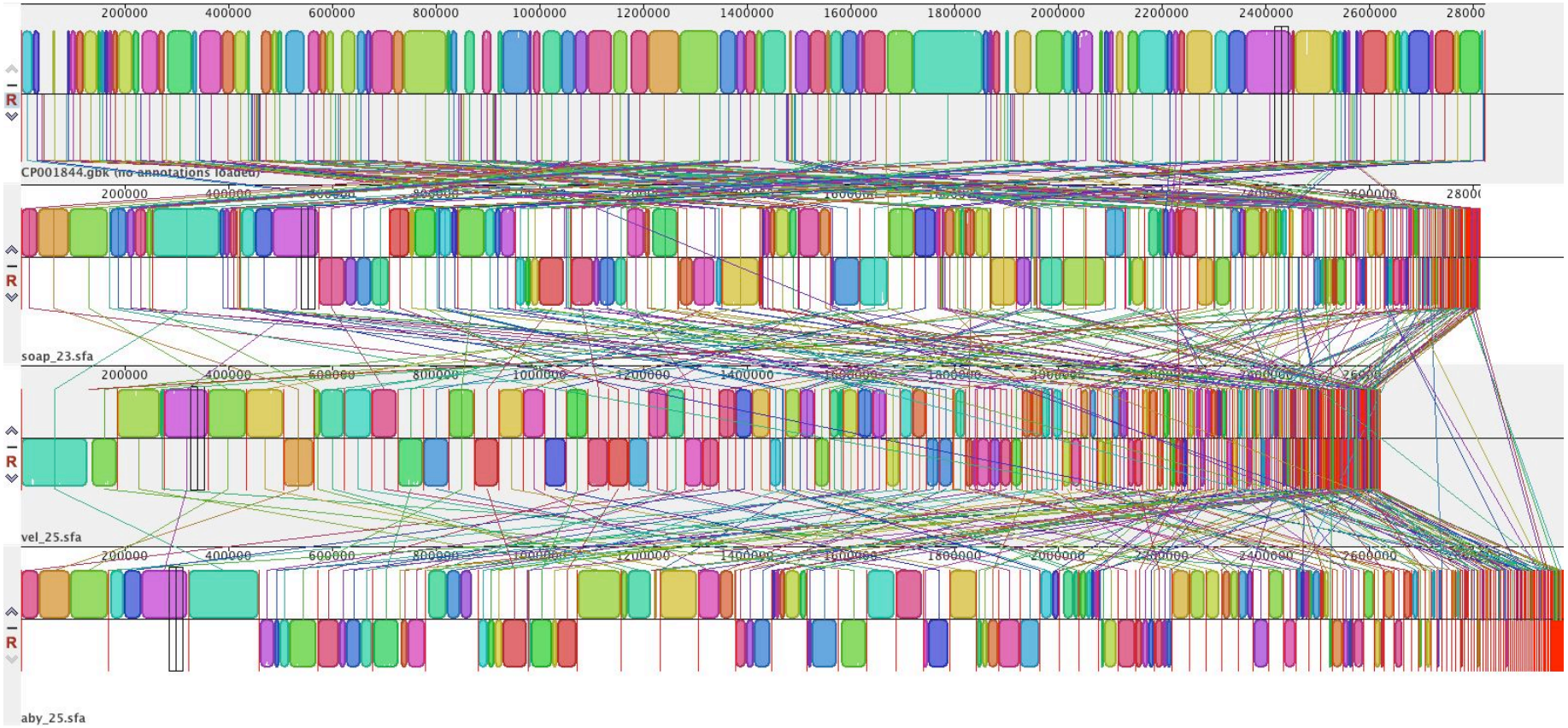
- ABySS: 88’077bp and 132’996bp

Beware of unwanted options…

- E.g., Velvet scaffolding (ON by default in the last version 0.7.55)…

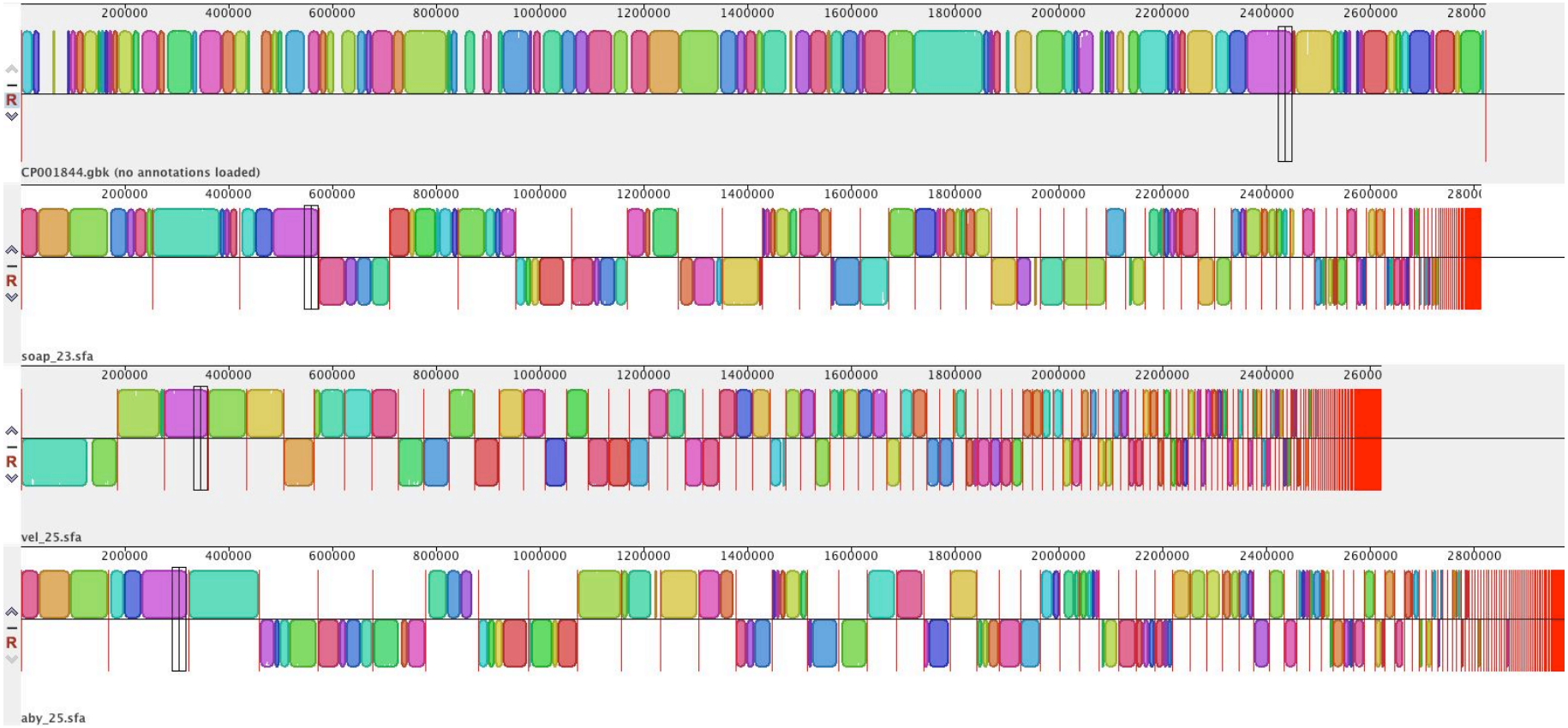
```
>NODE_1931_length_5525_cov_11.776470
TTTTTAAGTGCATGTGTATAATTTTCTACTGGGATAGGATCTGATGTTGCTGAACCTTCA
AATATAGTTATTTCTGGCAATCTTTCCTCTGCATAGTTAAAAGCTTTATTTAAAATTTCA
TCTATGTCTACATATATTTTTGTATACAGTCTCTTACCTAATTGAGTATTTAAATAACAA
TATTCACACATCCCCATGCACCCACTAACTAAAGGTAACTGATAATGTGCGGATGGTTTN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
…
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNTATACATCTGTAGATATCCATTCTCTAGAAACTTCT
ACTCCATTTTCATATGTTATTTGGTAAGAACTTGCCTTATAGCCTGGCATACCATTAATT
TTTACTTTTTGCTCTCCTTCTTTTAAAGCTGGATTATCAATATATTCAACTTCTGGATTA
TATTTTTCGTGTACTTCATTTACTAGTTGATATGTCTTTCCACTTAAAGCCTCTTTATTT
CCGTAAATATTAAATCCTACAGTTCCTCCACCAATATATCCTTCTATATATACTGGAAAA
TCATATGAATTAACAAATTTATAATCTACTACTCCATAAGCTACTGTAGCATCTAAACCT
GGATCTGAATATGAAACTGTTAATGTGTGGTTAGCTCTTTCTGTTGATTTTAAATTAGCA
```

Viewing assemblies with MAUVE

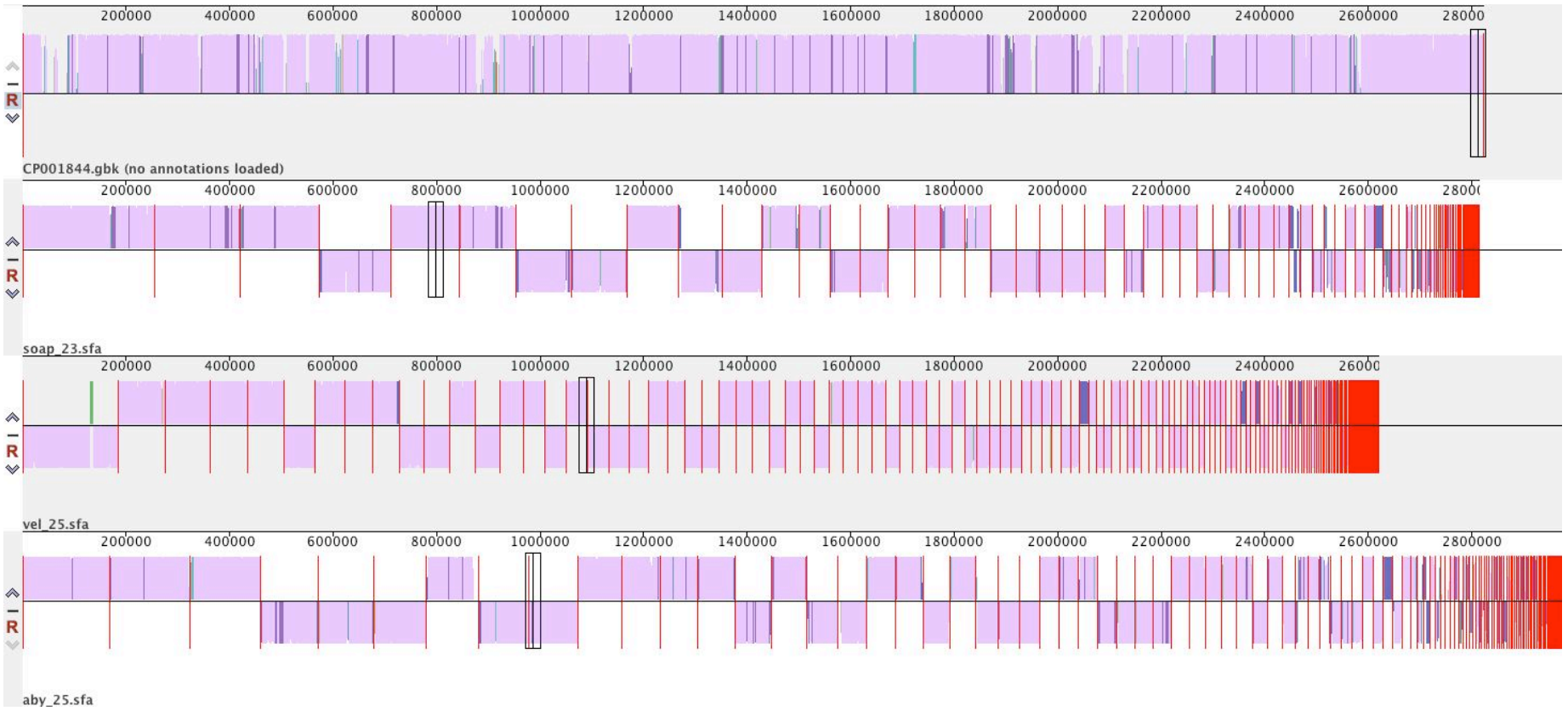


SIB Swiss Institute of Bioinformatics

Viewing assemblies with MAUVE



Viewing assemblies with MAUVE



Viewing assemblies with MAUVE

Viewing assemblies with Nucmer

```
show-coords -L 5000 -lro soapvsref.delta

    [S1]     [E1]  |     [S2]     [E2]  |  [LEN 1]  [LEN 2]  |  [% IDY]  |  [LEN R]  [LEN Q]  | [TAGS]
==========================================================================================================
       1    36398  |    95684   132081  |    36398    36398  |    99.98  |  2821452   132081  | CP001844   scaffold27    [BEGIN]
   89112   103612  |    14554        1  |    14501    14554  |    99.44  |  2821452    14554  | CP001844   scaffold15    [CONTAINS]
  106042   122142  |     2211    18327  |    16101    16117  |    99.80  |  2821452    48924  | CP001844   scaffold7
  121939   152241  |    18601    48924  |    30303    30324  |    99.84  |  2821452    48924  | CP001844   scaffold7
  152839   161164  |     8345       20  |     8326     8326  |   100.00  |  2821452     8345  | CP001844   scaffold76    [CONTAINS]
...
 2676049  2725173  |      626    49750  |    49125    49125  |    99.98  |  2821452    49750  | CP001844   scaffold28    [CONTAINS]
 2725745  2821452  |        1    95683  |    95708    95683  |    99.95  |  2821452   132081  | CP001844   scaffold27    [END]

show-tiling -c soapvsref.delta
>CP001844 2821452 bases
-95707  36373    52738   132081 100.00  99.96    +    scaffold27
89112   103665   -327    14554  100.00  99.44    -    scaffold15
103339  152262   576     48924  98.81   99.63    +    scaffold7
152839  161183   -95     8345   99.77   100.00   -    scaffold76
161089  232922   -580    71834  98.63   99.86    +    scaffold12
232343  331334   80242   98992  99.96   99.85    +    scaffold38
411577  448614   2014    37038  98.54   99.94    -    scaffold39a
450629  454689   -219    4061   98.65   99.13    +    scaffold57
454471  503676   5826    49206  99.64   99.69    +    scaffold3
...
```

Summary

- Lessons from the *de novo* genome assembly
  - Contigs obtained must be verified
  - Repeats are a nightmare in any case
  - Paired-ends help!

Genome assembly short reminder

- 1st step: Assembly
  - consists mainly in building contigs from short reads
- 2nd step: Scaffolding
  - where contigs are ordered and oriented
- 3rd step: Finishing (also called closing)
  - where gaps are closed

Closing or finishing?

- Finishing efforts are usually directed at closing gaps, not at fixing mis-assemblies, and therefore **'finished'** genomes are very likely to contain errors!
- A better term for such genomes is **'closed'**: gaps are closed but sequence is not confirmed. Many of the already-published finished genomes in the databases today contain assembly errors.

Salzberg SL, Yorke JA: **Beware of mis-assembled genomes.** ***Bioinformatics 2005, 21:4320-4321.***

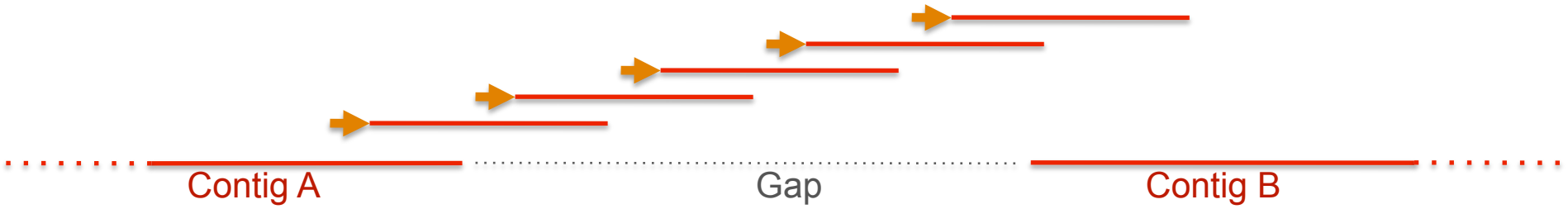
Existing methods for scaffolding and closing

- More Next Generation Sequencing (long range Mate-pairs)
  - 454 protocol
  - Illumina protocol
- Other methods
  - Chromosome walk
  - Multiplexed PCR
- Optical maps

Example of tools capable to join contigs into scaffolds with Mate-Pairs

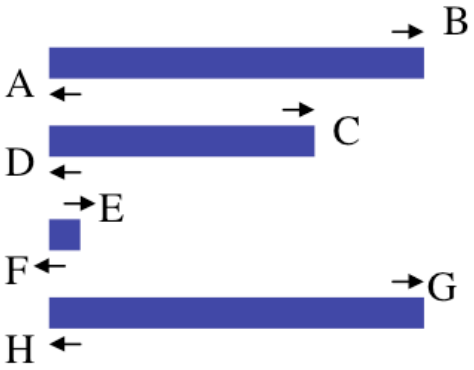
- General Assemblers (usually better with increasing insert sizes)
  - Newbler (only with 454 data)
  - Velvet (with Colombus module)
  - ABySS (not iteratively)
  - SOAPdenovo
- Other specialized
  - Bambus (AMOS package)
  - SuperContigs (old not maintained)
  - SSPACE (can also extend contigs)

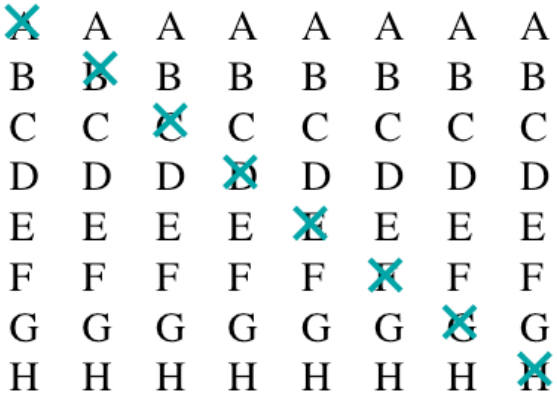
Primer “chromosome” walk

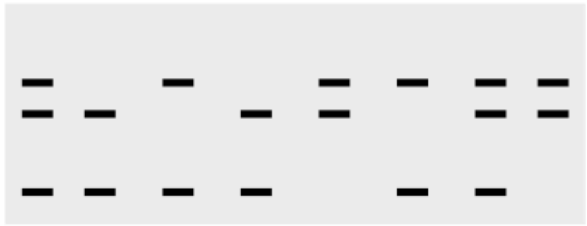


- Long and boring
- Often blocked for unknown reasons
- Requires more DNA

Multiplexed (or combinatorial) PCR



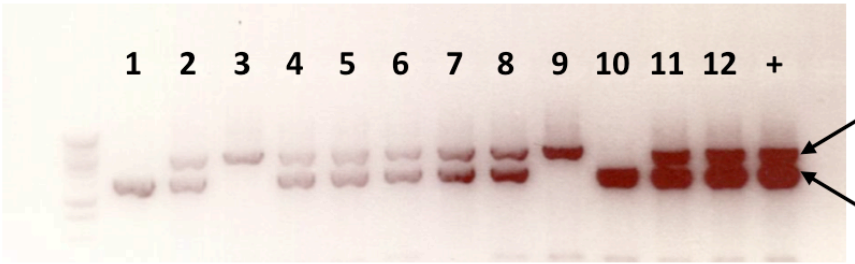




Only 8 PCR reactions required instead of 8*(8-1)/2 = 28

Multiplexed PCR for more contigs

- Primers are divided in pools of 4 to 6 primers
- In the first step: PCRs are performed using two pools of primers
- In the second step: when a PCR is positive, a new PCR is performed and each primer of the two pools is eliminated one at a time



this amplicon is generated by primers 1 and 10

this amplicon is generated by primers 3 and 9

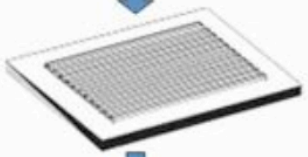
71 contigs → 24 pools of 6 → ~1152 PCRs

(276 PCRs for 1st step and 876 PCRs for 2nd step)

Only 1'152 PCR reactions required instead of 142*(142-1)/2 = 10'011

Optical maps (restriction maps)

Cells gently lysed to release genomic DNA

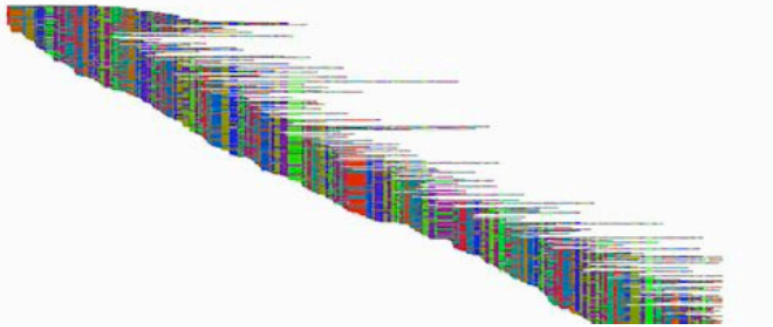


DNA captured in parallel arrays of long single DNA molecules using a microfluidics device

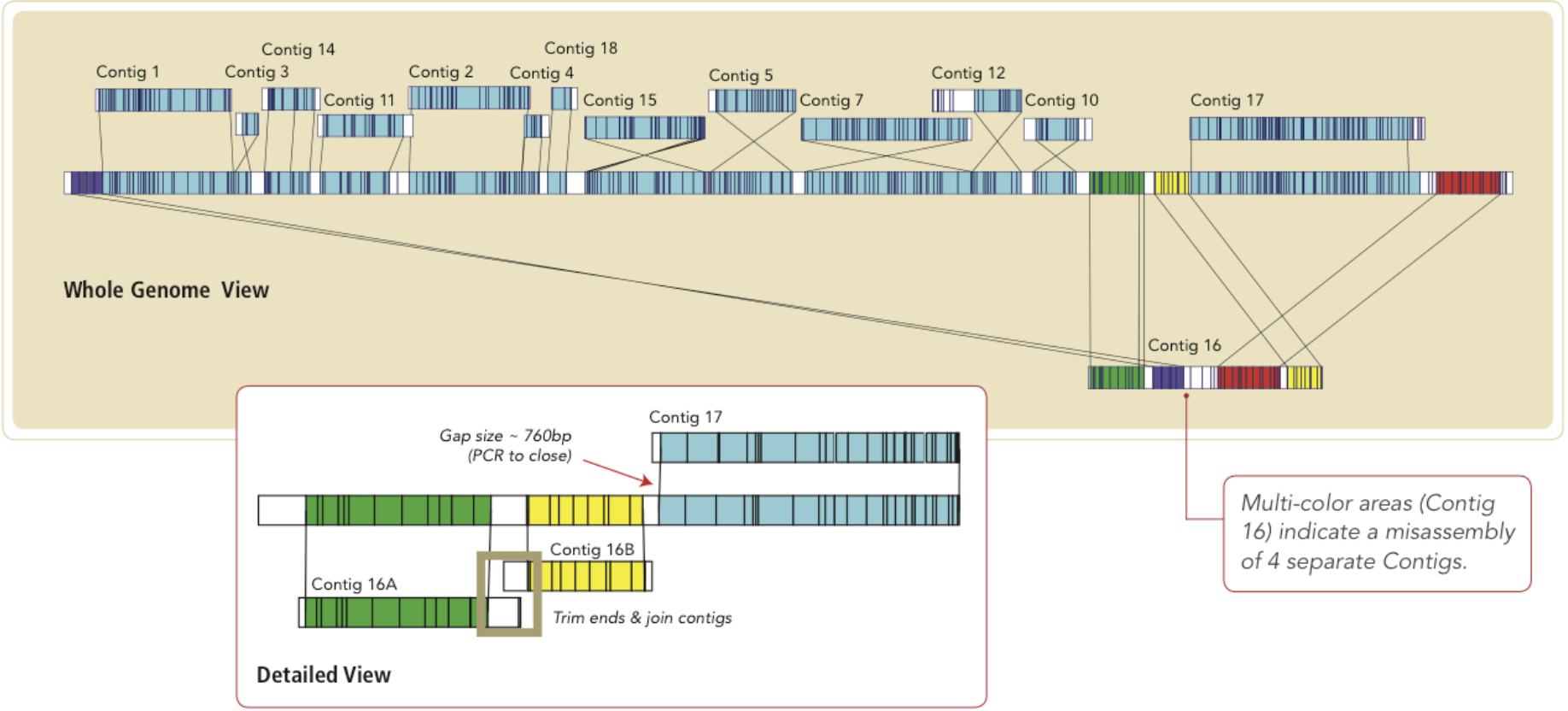
Restriction enzymes cleave the DNA at specific sites

Gaps in the DNA indicate location of cut sites

The DNA is stained with fluorescent dye and image analysis measures the size and order of each fragment to create "Single Molecule Maps".



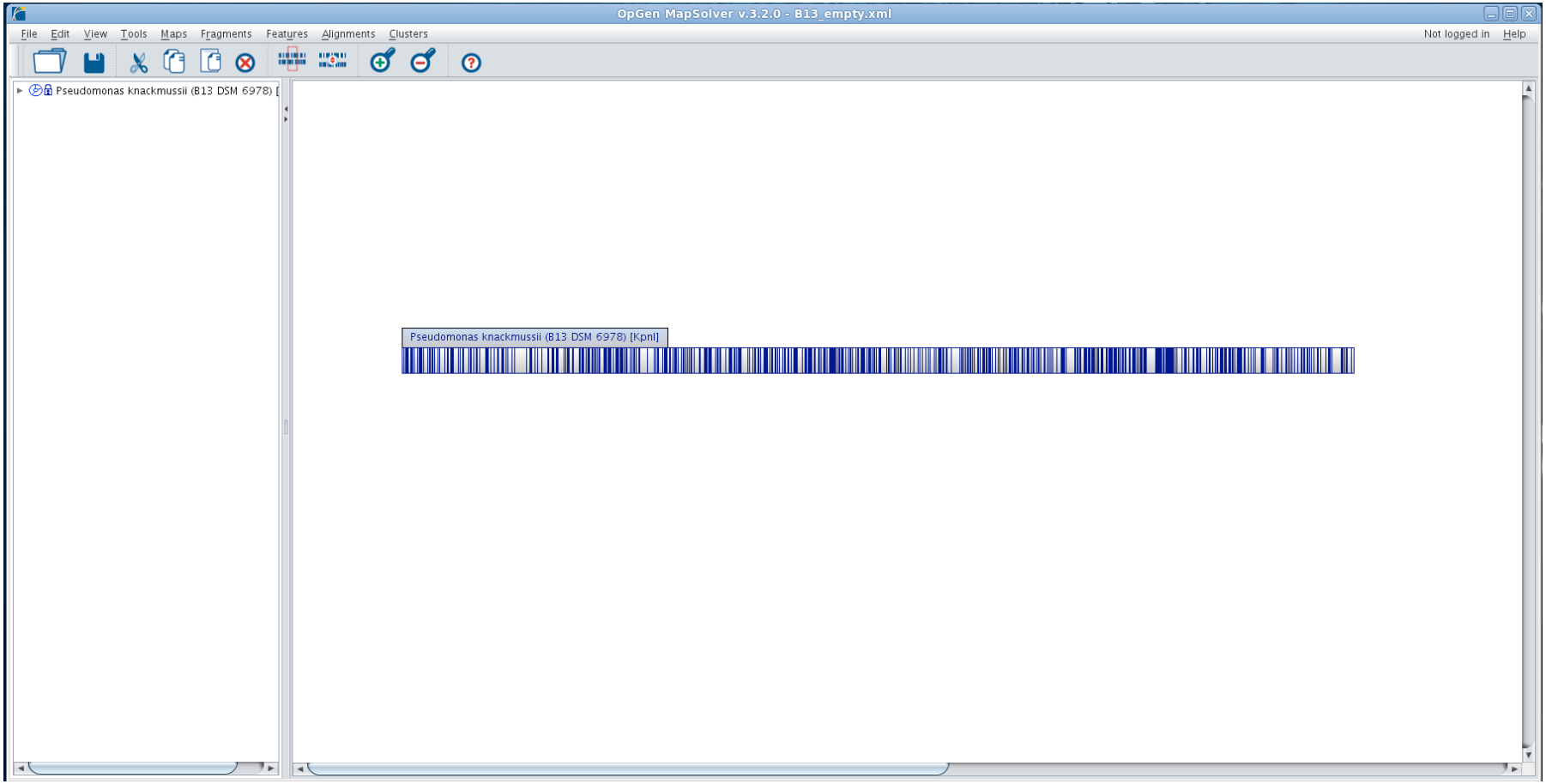
Optical maps: virtual contig scaffolding



Optical maps software and companies

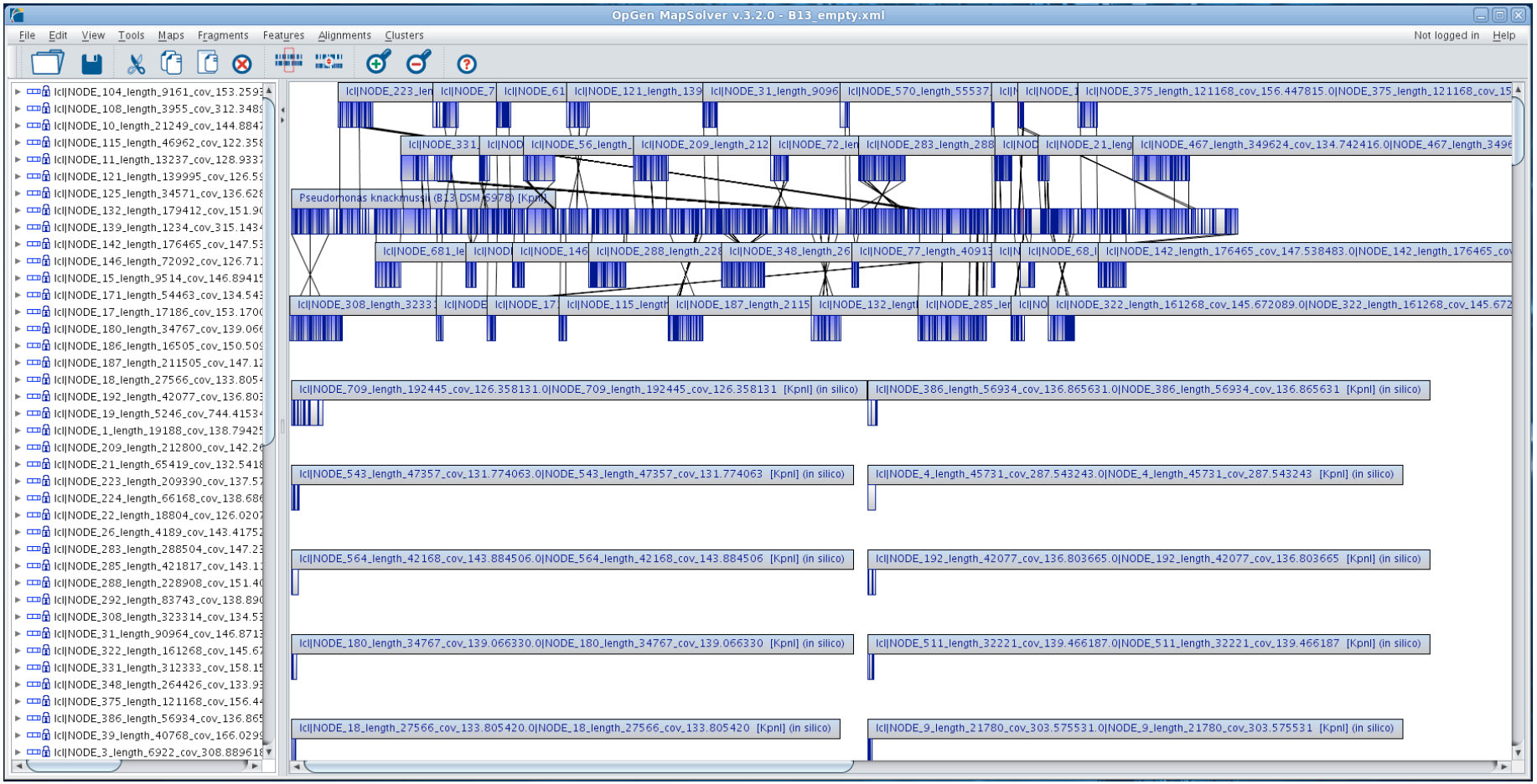
- OpGen (commercial www.opgen.com)
  - Offers service plus software MapSolver (limitation 6Mbp)
- Software only
  - Gentig (Bayesian inference) not available
  - BACop (graph based) not available
  - SOMA (Open Source: http://www.cbcb.umd.edu/soma)
- Other new or future techniques/companies
  - Nanoslits (http://www.pnas.org/content/104/8/2673.long)
  - BioNanomatrix (http://www.bionanomatrix.com)

OpGen results for MLS:
Example with *Pseudomonas knackmussii B13*

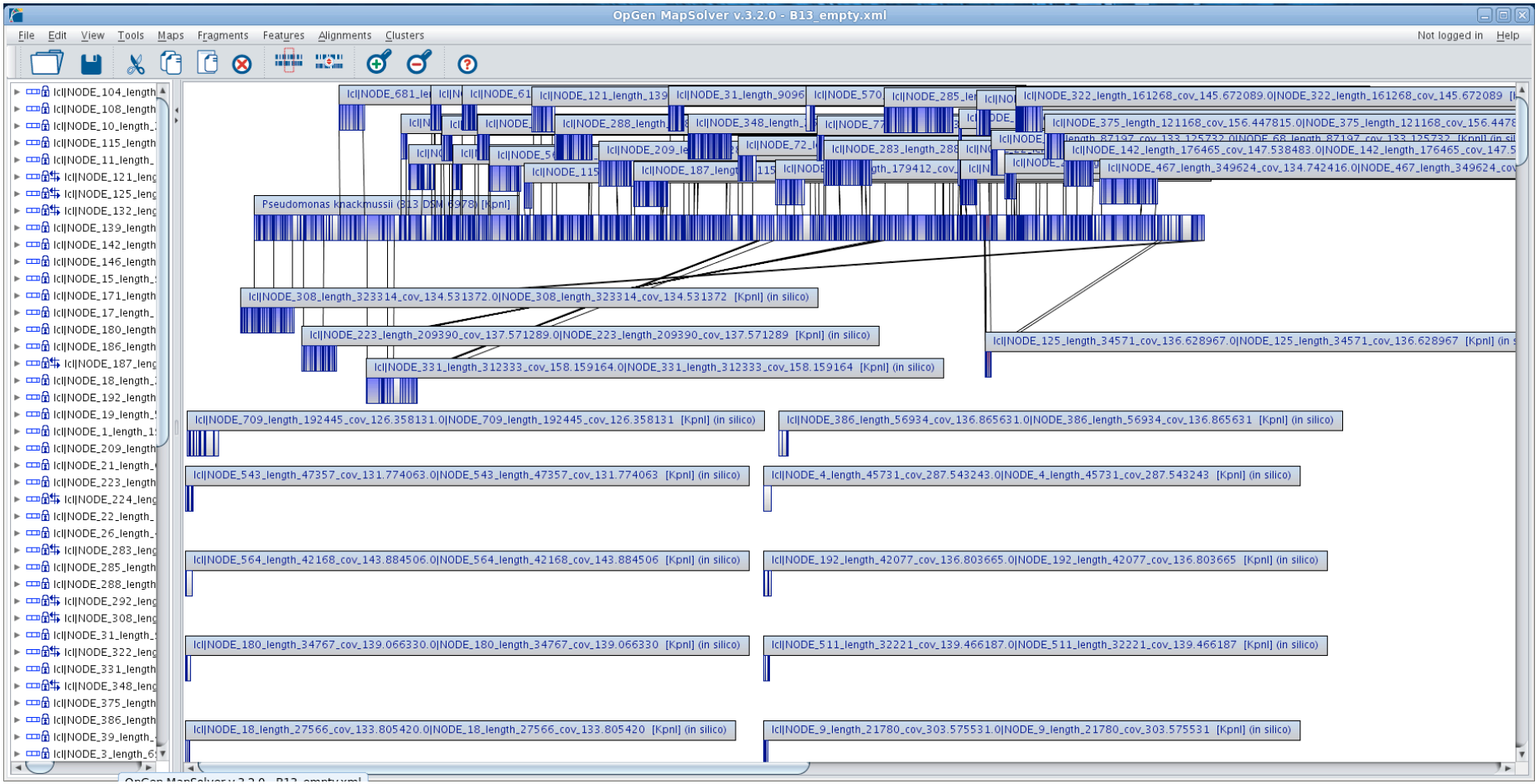


SIB Swiss Institute of Bioinformatics

Import and map B13 Velvet contigs

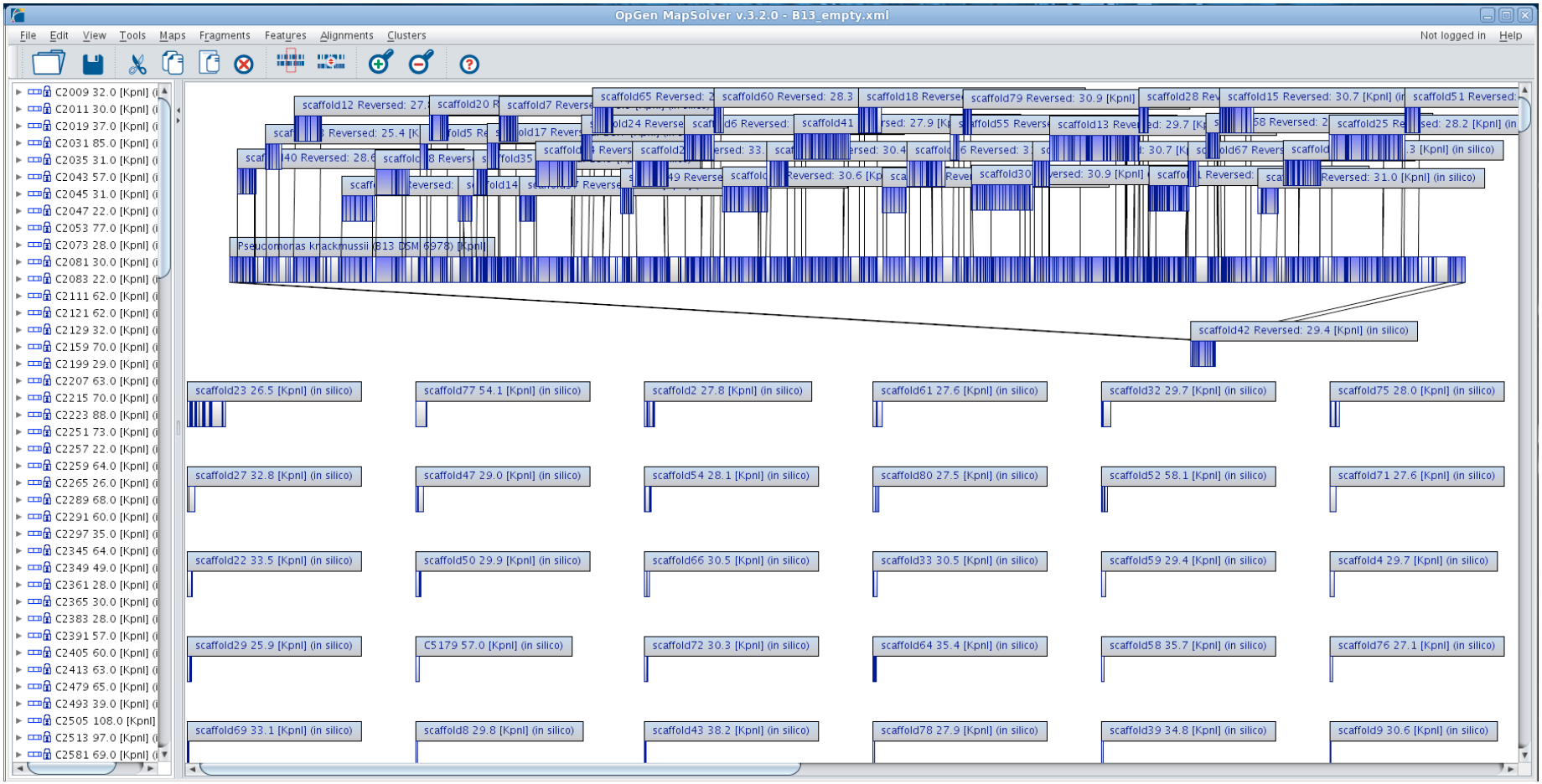


Clean B13 Velvet contigs -> misassembled contigs

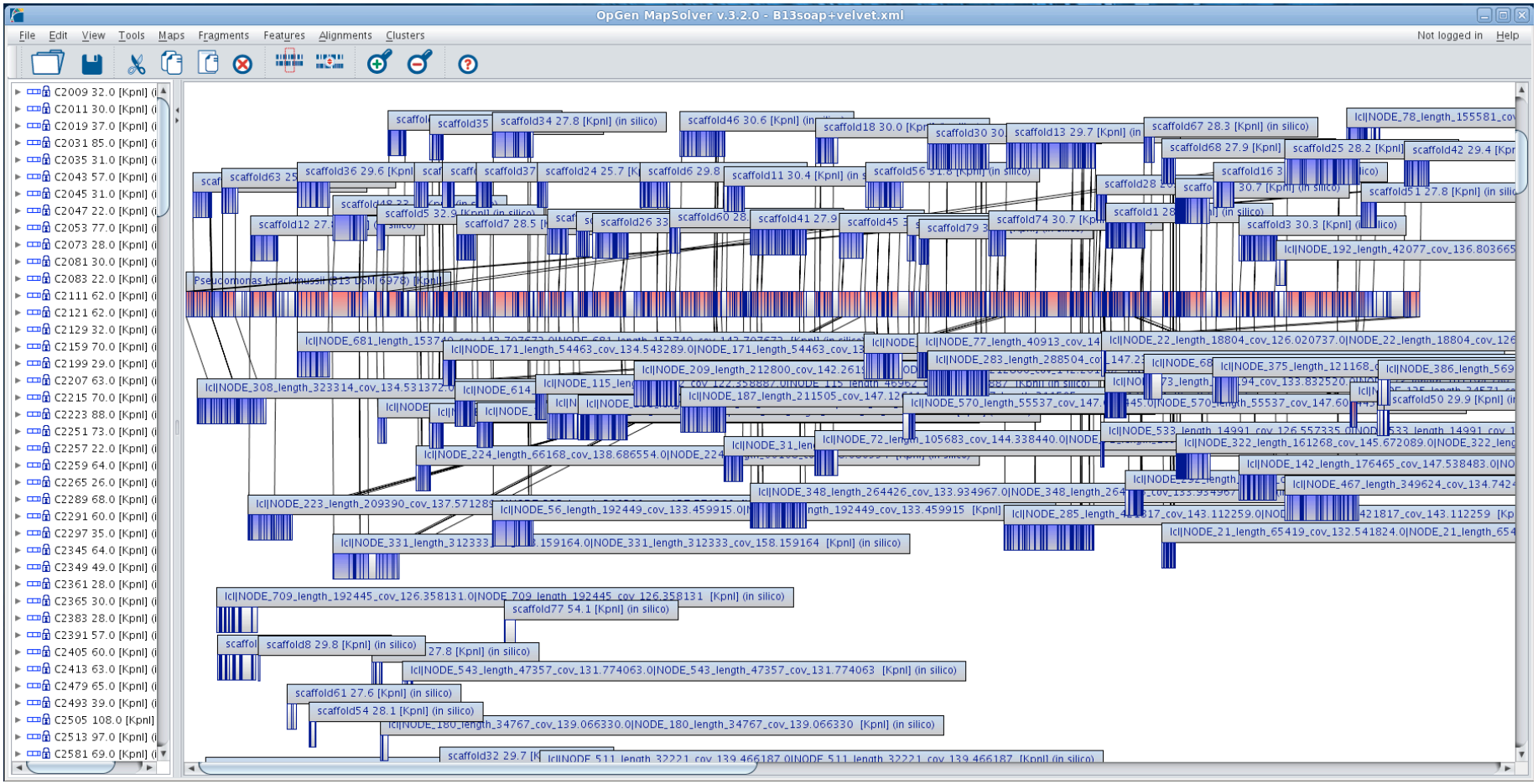


Swiss Institute of Bioinformatics

Clean B13 SOAPdenovo contigs



Combine SOAPdenovo and Velvet contigs



Swiss Institute of Bioinformatics

Output from MapSolver: preliminary scaffold

```
Chromosome     Start   End     Contig  Start   End     Orientation
Pseudomonas knackmussii (B13 DSM 6978) [KpnI]       1       36397   scaffold42 29.4 [KpnI] (in silico)    79546   115810  1
Pseudomonas knackmussii (B13 DSM 6978) [KpnI]       36398   121764  scaffold40 28.6 [KpnI] (in silico)    1       83983   1
Pseudomonas knackmussii (B13 DSM 6978) [KpnI]       172068  232821  scaffold63 25.4 [KpnI] (in silico)    10688   70779   -1
Pseudomonas knackmussii (B13 DSM 6978) [KpnI]       300907  422483  scaffold12 27.8 [KpnI] (in silico)    4149    121914  -1
…

N50 (kb):      153.908
Avg. Contig Size (kb): 42.312380281690146
% significant contigs (> 5 kb):       28.169014084507044
% contigs placed:      28.169014084507044
Total size of placed contigs: 4972722
Total size of unplaced contigs:       1035636
% genome covered:      84.80643398130894
Number of gaps over 2 kb:     25
Total number of gaps:  38
Avg. gap size: 23240.21052631579
Total number of contig overlaps:      0
% Close-able gaps:     34.21052631578947

Gaps/Overlaps:
Optical Map    Type    Start   End     Length
Pseudomonas knackmussii (B13 DSM 6978) [KpnI]       Gap     124047  170149  46103
Pseudomonas knackmussii (B13 DSM 6978) [KpnI]       Gap     243509  294529  51021
Pseudomonas knackmussii (B13 DSM 6978) [KpnI]       Gap     426632  522945  96314
…

Unplaced contigs:)
scaffold38 35.0 [KpnI] (in silico)
scaffold39 34.8 [KpnI] (in silico)
…
```

Next finishing steps

- Verify scaffolds predictions by PCR (short gaps)
- Close remaining gaps by multiplexed PCRs
- New sequences may allow to place remaining contigs

Question you should ask yourself when you plan to sequence a full genome (or more than one)…

- Why?
- Size of the genome? (eukaryote vs prokaryote)
- %GC of the genome?
- Are there any known or expected repeats?
- Is there a known genome available to serve as a reference?
- How far do you want to go?
  - Fully assembled genome
  - Comparative genomics (SNPs and/or Indels)
  - Partial assembly (draft genome)
  - Only contigs searching for specific coding regions
  - Functional annotation and Pathway classification

Why answering those questions first?

- The answers will drive the choice of the technique…
- …the amount of time required
- …and the costs!

- Yes, the sequencing costs decrease every year…
- …but don't underestimate the costs of the bioinformatics part:
  - Storage
  - Calculation
  - Expertise

Summary

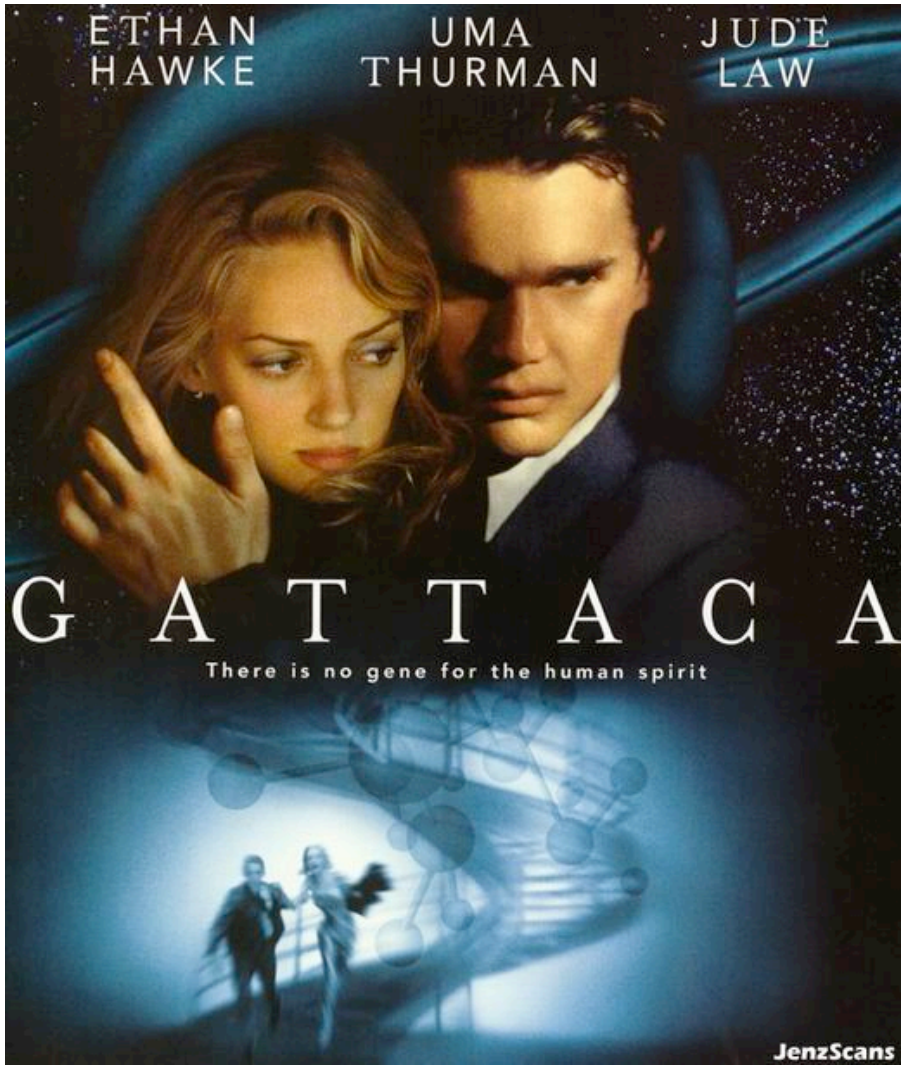
- Lessons from the genome assembly
  - Easy to map reads onto a closely related reference (always better than *de novo*)
  - Less easy to find non-matching reads and what they are (plasmids, insertion sequences, phages, virus, other)
  - Contigs obtained by de novo assembly must be verified
    - Combine RNAseq with assembly!
  - Repeats are a nightmare in any case
  - Paired-ends help especially for *de novo* assembly!
  - Finishing requires a lot of resources

- But… should we care??

Welcome to GATTACA !

- Next-gen sequencing = last year…
- Next-next-gen = this year
- Illumina HiSeq2000
- SOLiD 4hq
- IonTorrent

- Next-next-next-gen…
- Pacific Bioscience
- By 2014: up to 50Kbp per read of single molecule with 99.3% accuracy, 1 Mio ZMW -> (99.999% accurate human genome in 30 minutes!!)



The practicals

- http://edu.isb-sib.ch

- select «workshops» on the left menu
- select «Bogota NGS workshop»

- Login to bioinf-hpc server, then follow the instructions of the exercises

Thank you

Swiss Institute of
Bioinformatics

FASTA example

```
>contig_6 length=320 nReads=87 !529472 ] !2294037  ]
TAACGGTAGGCTTTTTTGACCGCTTCATCGTCGGGTGGTTCAACATTTTCTAATTGATAT
GGGATGCCTAAATTTTTCCACTTATACACGCCGAGTTGGTGATAGGGTAAGATTTCAAAT
TTTTCAACGTTATCAAGAGAATTAATAAATTCTCCAAGTTGAATGAGATCTTCTTTATCA
TCTGAGATACCTGGCACTAGGACGTGACGAATCCATACAGGTTGTTTCATATCTGACAAT
TTACGGGCAAATTTGAGTATATGTGTATTGGGTTTGCCTGTTAATTGAATATGTTTTTCA
TTATTAATATGCTTTATGTC
>contig_7 length=140 nReads=45 60537 ] 1378182  ]
TCGTTTTATAACTGAAGAAGAACTATCAAAATATATGAACGCCGATCAAAAACAACCTGA
AGAACCTGCAGCTCAAGAAATTAAACAACATCAAAATGTCGATAACCCGCGTGGTATTGA
ACAATTTAATACACACAATA
>contig_8 length=212 nReads=59 1604937 ] 1907084  ]
ATAAGTTGAATCTGTTTGATTAGCTTGAGTGATGGCATTACCATTCGACTGATGGTTAAA
ACCTTGGTCTACTTGATTATTTTCTATAGTTGCAGCTGAAGCCTCGTGATGTGATGTAAG
AAATAAAGCAGAAGTAGTGATAGTTGCGCCGATTAAGTATTTGATAGAATGATGAGTCAA
AAAAATCTCCCCTTGAATATATTTATTTATAC
```

Quality file example

```
>contig_6 base quality
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 31 21 24 23 23 22 22 22 22 30 31 20 20 20 20
20 20 30 30 31 30 33 33 34 34 34 34 31 33 31 34 34 30 30 25 25 25 26 30 30
33 37 30 33 36 35 23 23 24 24 24 24 30 30 30 30 30 30 30 30 30 33 33 33 33
30 30 33 33 33 30 33 33 29 29 29 29 33 34 33 33 21 21 21 21 23 23 23 23 23
23 24 24 26 26 26 26 26 26 26 40 33 30 30 30 30 33 33 33 33 33 30 33 33 33
33 33 33 33 31 31 31 31 34 37 40 40 45 37 52 52 52 52 52 52 55 55 59 64 65
68 58 57 49 49 49 49 49 49 49 56 60 53 60 53 49 49 49 49 49 34 45 45 45 45
30 25 25 25 28 30 45 45 49 49 49 49 49 70 60 60 59 59 53 56 53 53 53 55 53
53 60 45 49 49 42 42 42 45 45 36 33 34 49 46 46 46 54 54 59 53 49 49 42 41
43 40 44 40 49 49 49 54 58 53 52 57 51 51 41 39 15 15 35 35
>contig_7 base quality
…
```

Example of FASTQ Illumina 1.5

```
@C3PO_0001:2:1:17:1499#0/1
TGAATTCATTGACCATAACAATCATATGCATGATGCAAATTATAATATCATTTTTAGTGACGTCGTGAATCGTTT
+C3PO_0001:2:1:17:1499#0/1
abaaaaaaaaaaa`a`aa_aaaaaaaaaaaaaaa_a__aaa`aaaaa^aaaaa`a]^`a__YZYZ^`NJDJ\_Z
@C3PO_0001:2:1:17:1291#0/1
TGTTTGAGCAAATGATTCATAATAATGTATTTCAATATTTTTAGGAATATCTCCCAATATTGCGCGTGCTGAATT
+C3PO_0001:2:1:17:1291#0/1
a`_`_\a_aaaa_a^Z^^a[a^aa]a_^_a_``aa__`aa`X^X^^`aa_\_]VR`\a_]W\_`_a]a]][\RZV
@C3PO_0001:2:2:1452:1316#0/1
GTCCATCCGCAGCAGCGAATTTTTGACGTCCCCCCCCCGAANGGANGNGANNNNGNNGNNNTNTNNAAANGNNNNN
+C3PO_0001:2:2:1452:1316#0/1
_U__a\__`]_`ZP\\_Z^[]aa^a_]XNBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
…
```

Warning: various FASTQ formats…

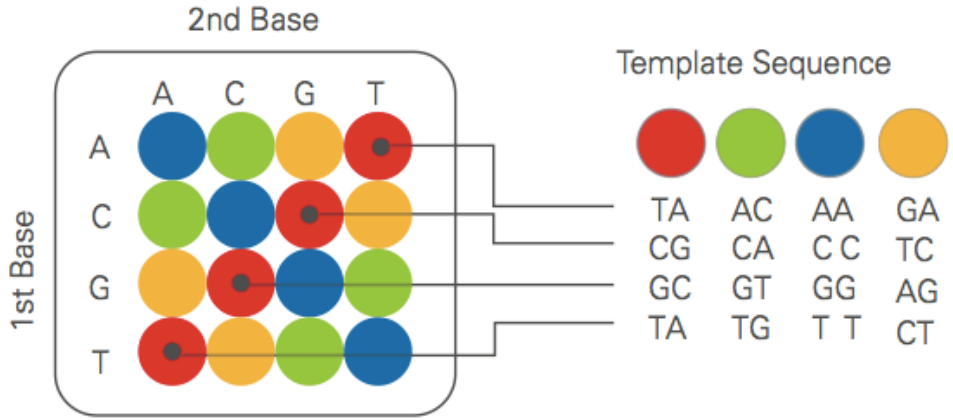
```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................
..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.....................
..............................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
|                         |    |        |                              |                     |
33                         59   64       73                             104
126
S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40) with 0 and 1=unused, 2=Read Segment
Quality Control Indicator (bold)
```

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

SOLiD color space FASTA format

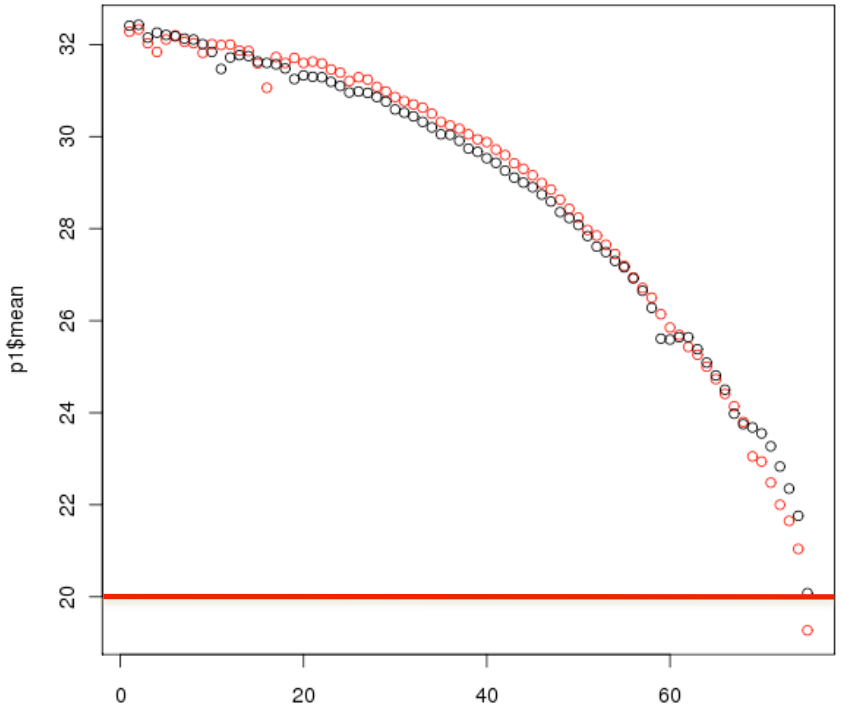
```
>1_51_64_F3
T1030103123033323320333300002112222 3
>1_51_127_F3
T2010323233203132310110100200310310 2
```

Each number can be replaced according to this table

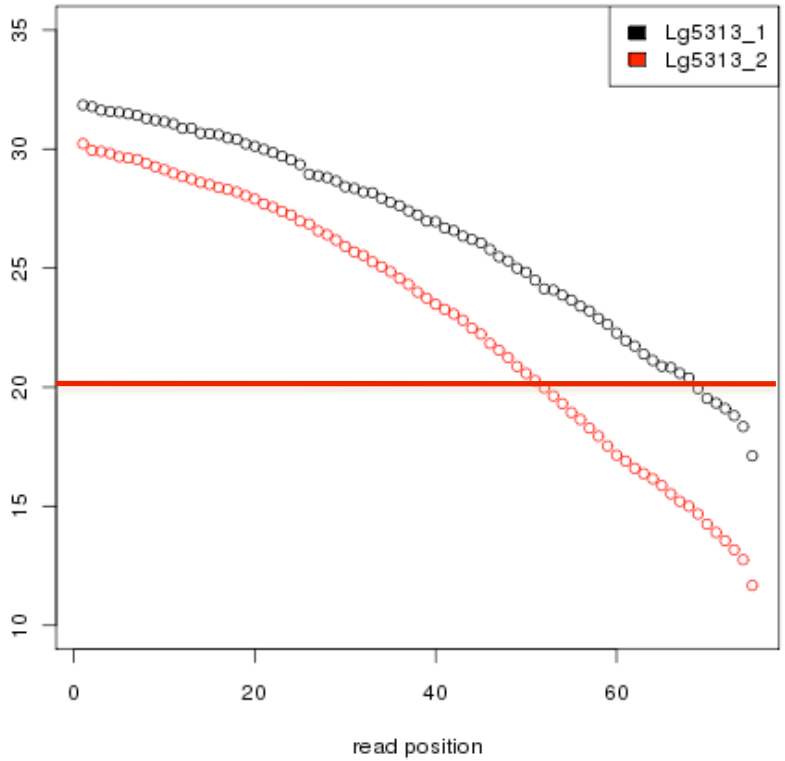


Variability in the quality (mean value per position)

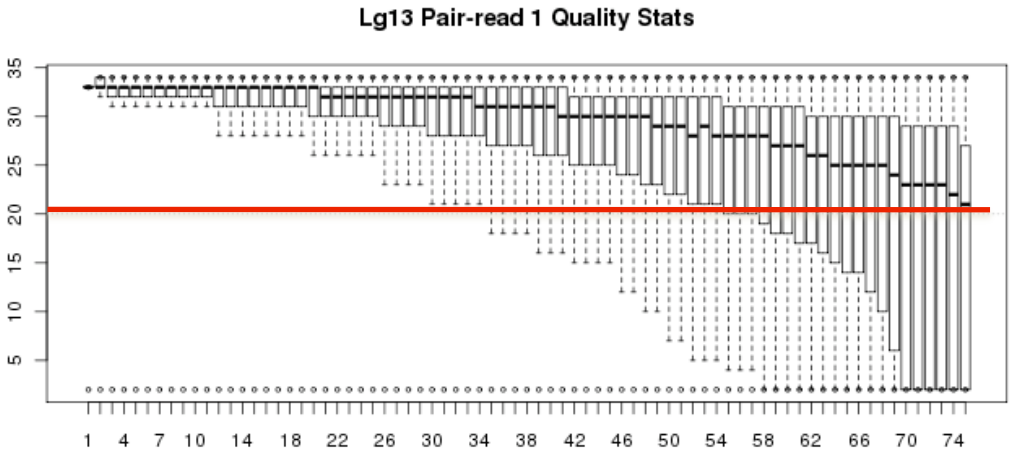
- Good example

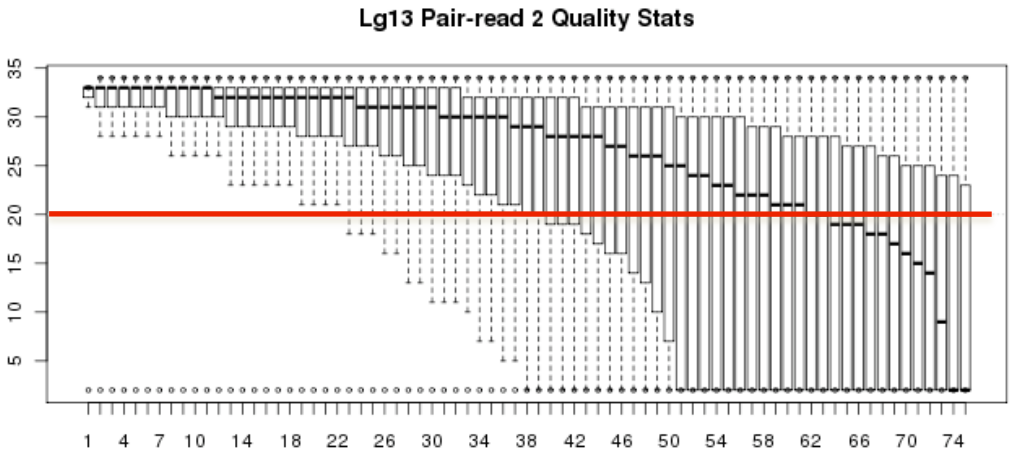


- Less good example…

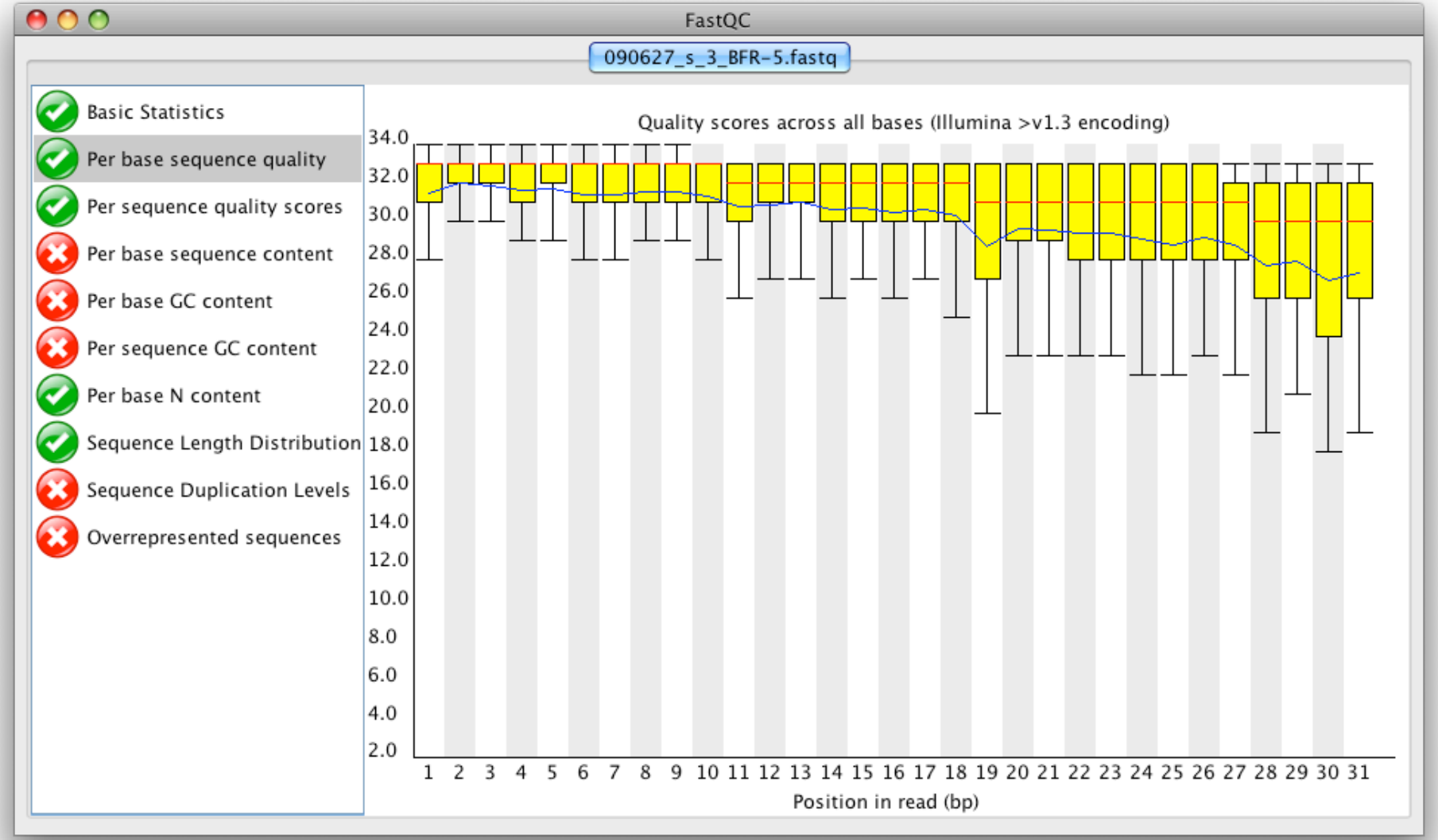


Variability in the quality (boxplot)

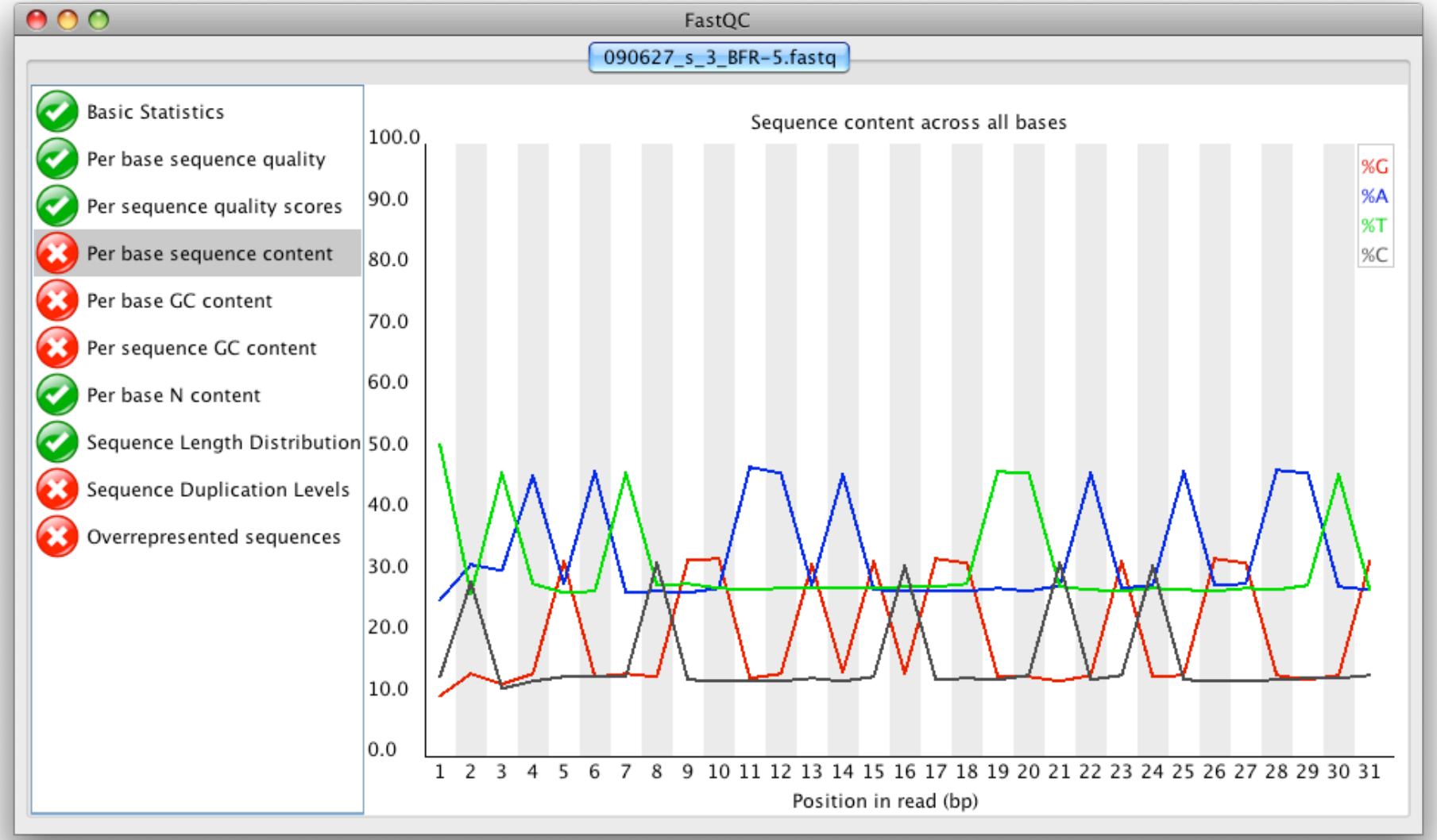




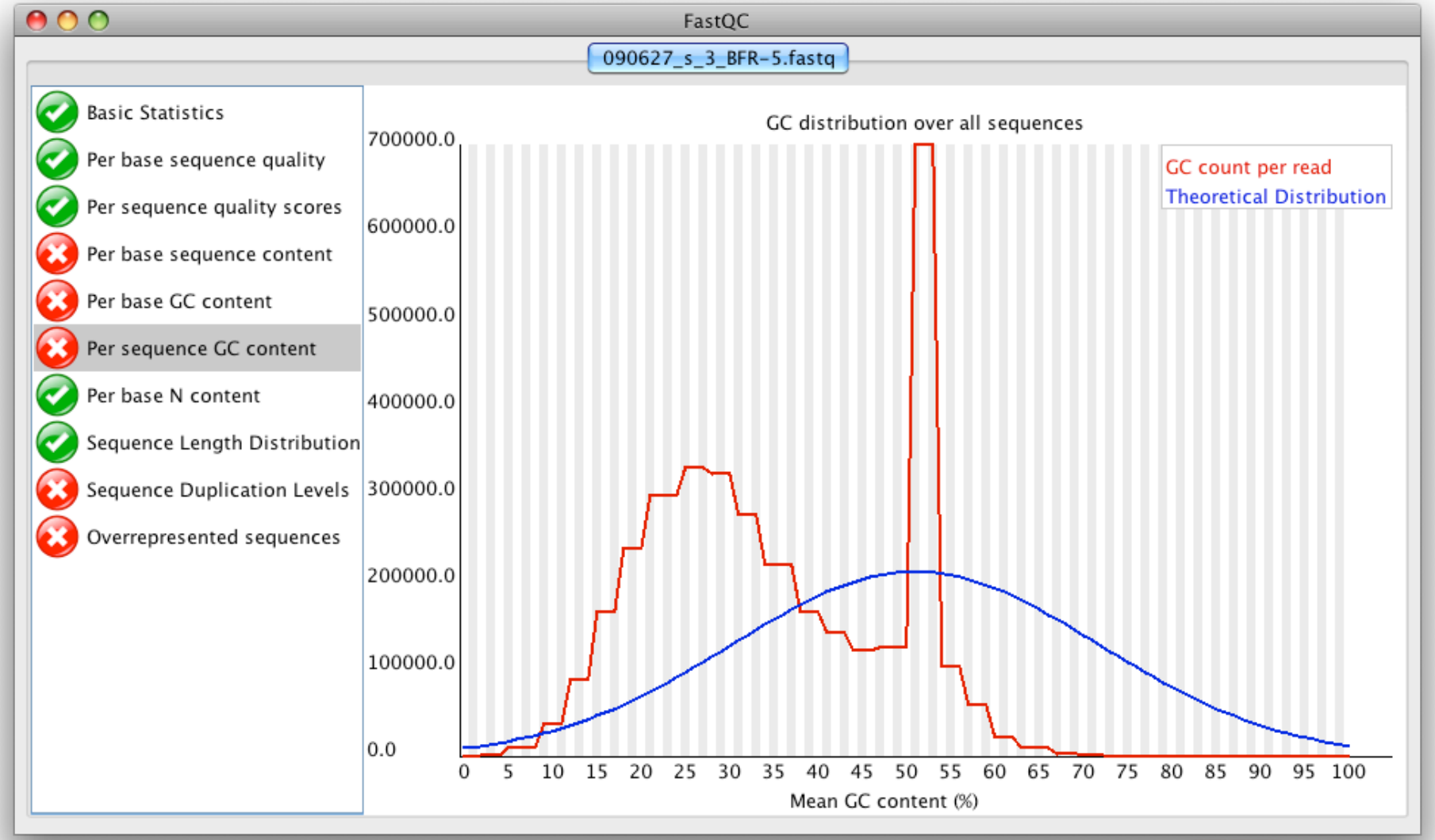
Quality Control example



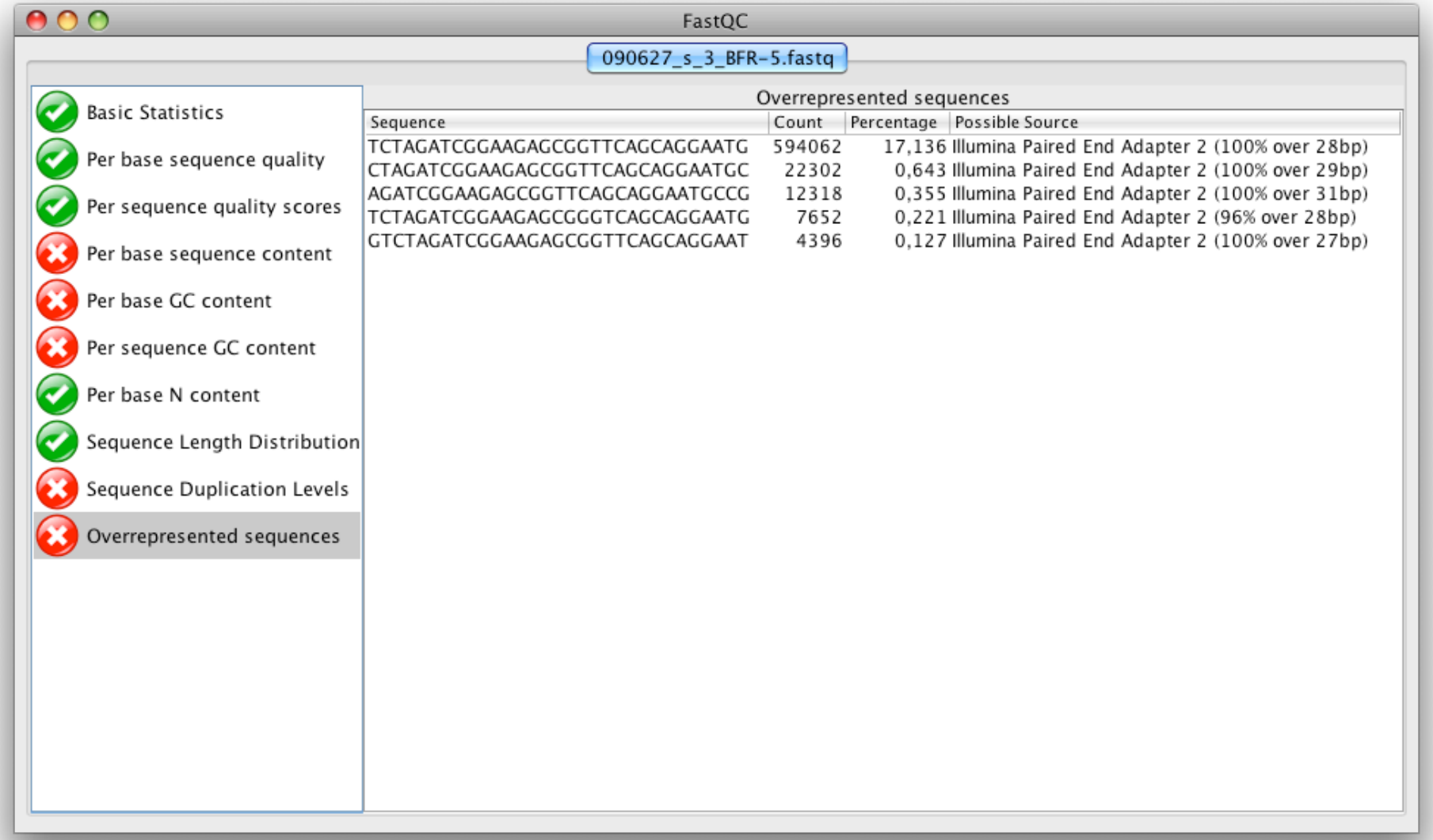
Quality Control example



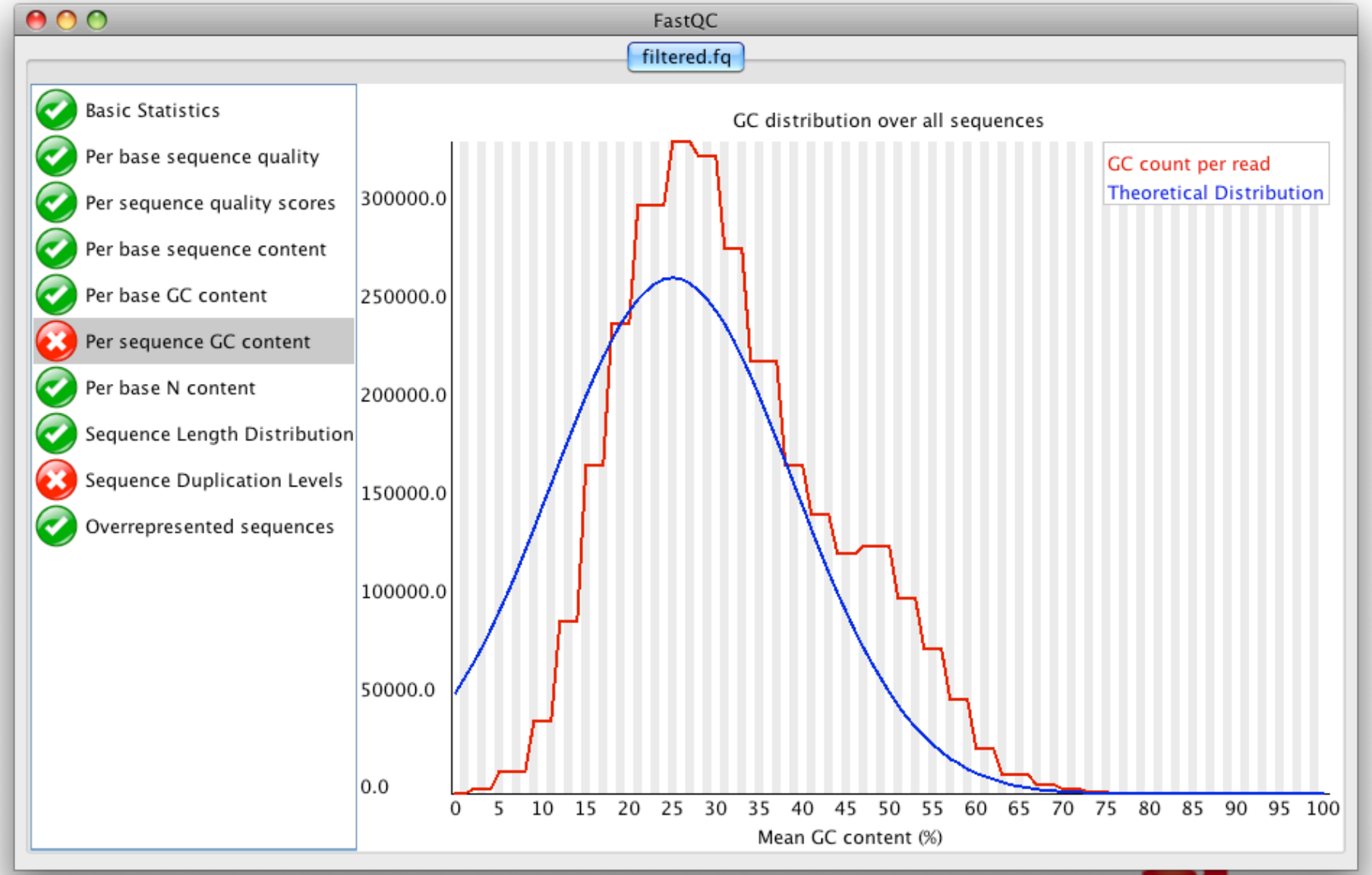
Quality Control example



Quality Control example



Quality Control example



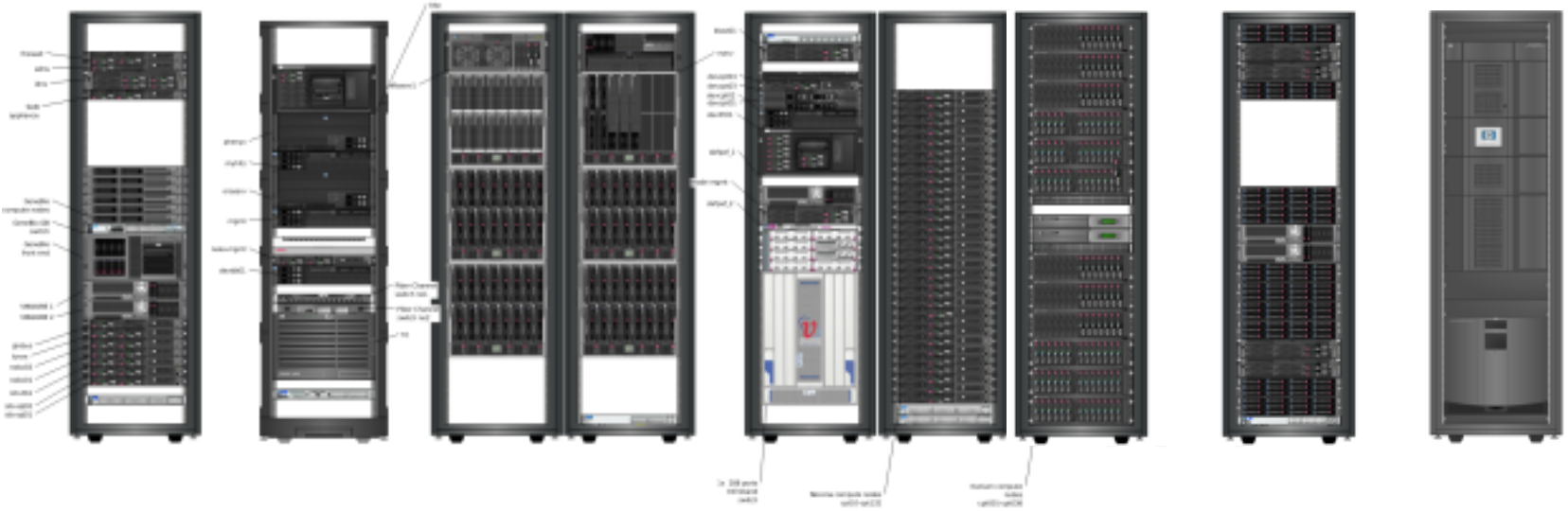
Filtering data can help

http://pathogenomics.bham.ac.uk/blog/2009/09/tips-for-de-novo-bacterial-genome-assembly/

- Illumina reads quality decrease with length
  - Trim 3' ends of reads according to quality
  - Remove reads with average low quality
  - If coverage is high, remove orphan reads
- 454 reads
  - Trim 3' ends of reads according to quality
  - Remove reads with average low quality
  - If possible correct for long mononucleotide repeats
- Check contigs by remapping reads

Vital-IT hardware

- Cluster of > 1200 nodes
- 3 dedicated machines with 8-24 CPU + 256Gb RAM
- Large storage capacity (>700Tb)



What are NextGenerationSequencing short reads data?

Sequencing platform	ABI3730xl Genome Analyzer	Roche (454) FLX	Illumina Genome Analyzer	ABI SOLiD	HeliScope
Sequencing chemistry	Automated Sanger sequencing	Pyrosequencing on solid support	Sequencing-by-synthesis with reversible terminators	Sequencing by ligation	Sequencing-by-synthesis with virtual terminators
Template amplification method	In vivo amplification via cloning	Emulsion PCR	Bridge PCR	Emulsion PCR	None (single molecule)
Read length	700–900 bp	200–500 bp	36-108 bp	35-75 bp	25–55 bp
Sequencing throughput (old numbers)	0.03–0.07 Mb/h	13 Mb/h	25 Mb/h	21–28 Mb/h	83 Mb/h
Advantage by price	700 bp / $	16'000 bp / $	500'000 bp / $	1'000'000 bp / $	1'000'000 bp / $
Nr of installed machines (estimation)	??	197	684	213	10

Limitations of the techniques

- All methods
  - Sequencing errors
  - Missing data (sampling/coverage bias)
- Roche 454
  - long (>12) mononucleotide repeats
- Illumina
  - short reads (36-150bp)
- SOLiD
  - very short reads (25-50bp)
  - biased paired-ends (50/25)