Bioinformatics Seminars Series: Assembly Validation

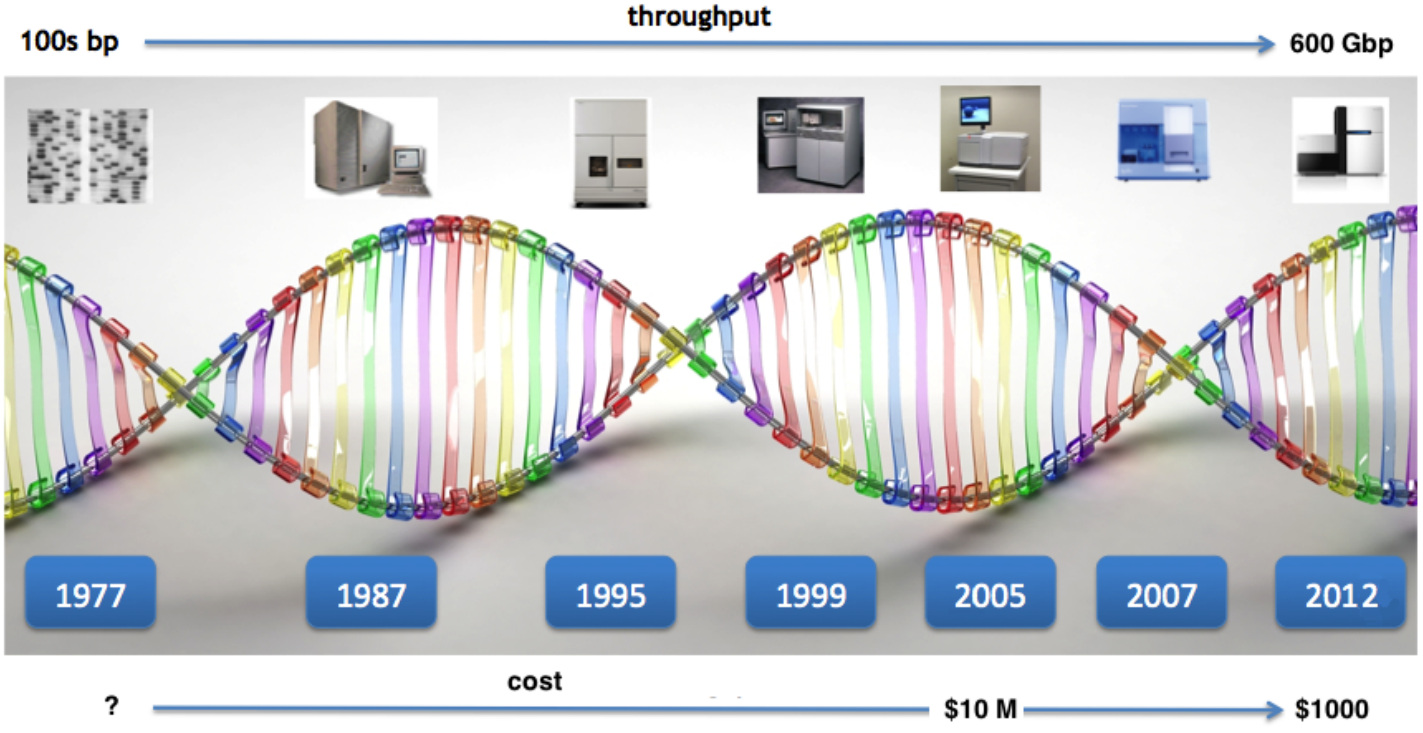
Francesco Vezzi

KTH: Royal Institute of Technology
SciLife Lab Stockholm

ROYAL INSTITUTE
OF TECHNOLOGY

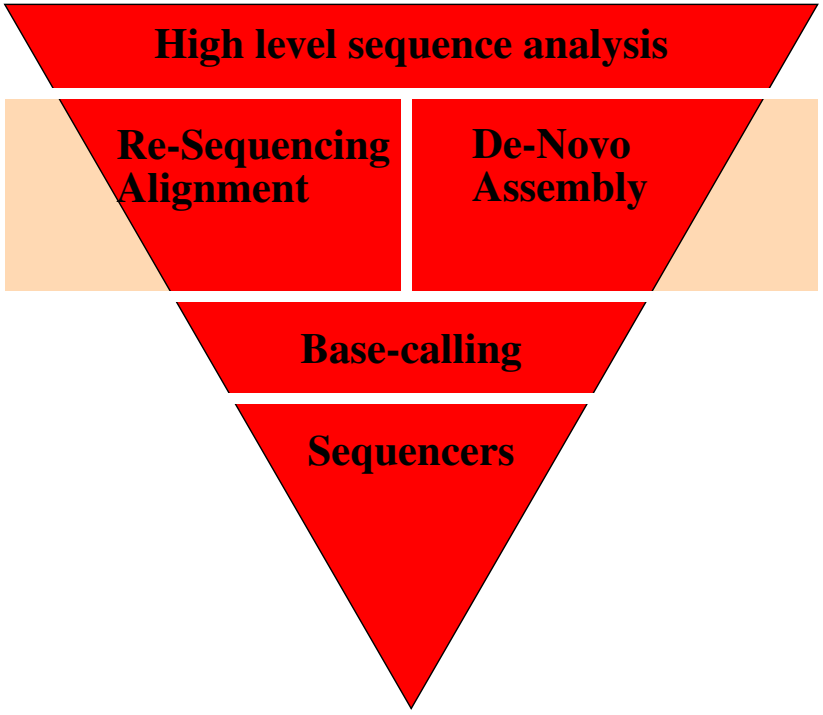
Summary

The Sequencing (R)*evolution*



In 2012 Illumina will release a new instrument able to sequence an individual Human genome for **1000$**

Genome Analysis Pyramid



Every step needs validation procedures and quality controls.

The need of evaluation

J.R. Miller

No algorithm or implementation solves the WGS assembly problem. Each of the various software packages was published with claims about its own superiority.

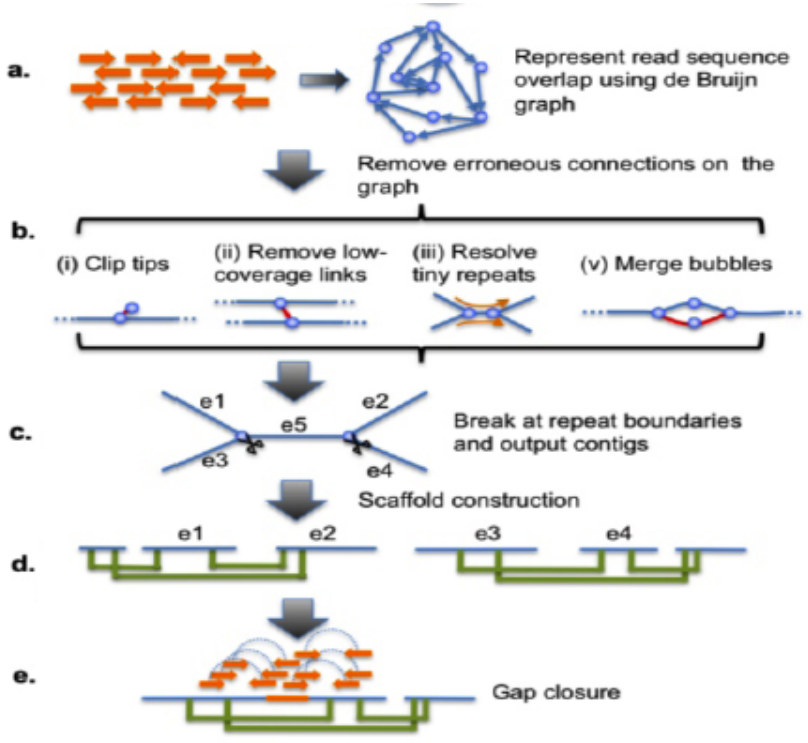
Recent Critics

- Beware of mis-assembled genomes (Sanger *et al.* 2005)
- Limitations of NGS genome sequence assembly (Alkan *et al.* 2011)
- Assembly: the good, the bad, the ugly (Birney *et al.* 2011)

Evaluation efforts:

- Assemblathon 1, 2 (maybe 3?)
- GAGE: benchmark dataset

De Novo Assembly: The Problem



Solving Strategies

- Hash Based Method
- Overlap Layout Consensus (OLC)
- De-Bruijn Graph (DBG)

Why so difficult?

- NP complete;
- Short reads;
- Repeats;

Available Assemblers

Name	Algorithm	Author	Year
Arachne WGA	OLC	Batzoglou, S. et al.	2002 / 2003
Celera WGA / CABOG	OLC	Myers, G. et al.; Miller G. et al.	2004 / 2008
Minimus (AMOS)	OLC	Sommer, D.D. et al.	2007
Newbler	OLC	454/Roche	2009
Edena	OLC	Hernandez D., et al.	2008
MIRA, miraEST	OLC	Chevreux, B.	1998 / 2008
TIGR	Greedy	TIGR	1995 / 2003
Phusion	Greedy	Mullikin JC, et al.	2003
Phrap	Greedy	Green, P.	2002 / 2003 / 2008
CAP3, PCAP	Greedy	Huang, X. et al.	1999 / 2005
Euler	DBG	Pevzner, P. et al.	2001 / 2006
Euler-SR	DBG	Chaisson, MJ. et al.	2008
Velvet	DBG	Zerbino, D. et al.	2007 / 2009
ALLPATHS	DBG	Butler, J. et al.	2008
ABySS	DBG	Simpson, J. et al.	2008 / 2009
SOAPdenovo	DBG	Ruiqiang Li, et al.	2009
SUTTA	B&B	Narzisi, G, Mishra B.	2010
SHARCGS	Greedy	Dohm et al.	2007
SSAKE	Greedy	Warren, R. et al.	2007
VCAKE	Greedy	Jeck, W. et al.	2007
QSRA	Greedy	Douglas W. et al.	2009
Sequencher	-	Gene Codes Corporation	2007
SeqMan NGen	-	DNASTAR	2008
Staden gap4 package	-	Staden et al.	1991 / 2008
NextGENe	-	Softgenetics	2008
CLC Genomics Workbench	-	CLC bio	2008 / 2009
CodonCode Aligner	-	CodonCode Corporation	2003 / 2009

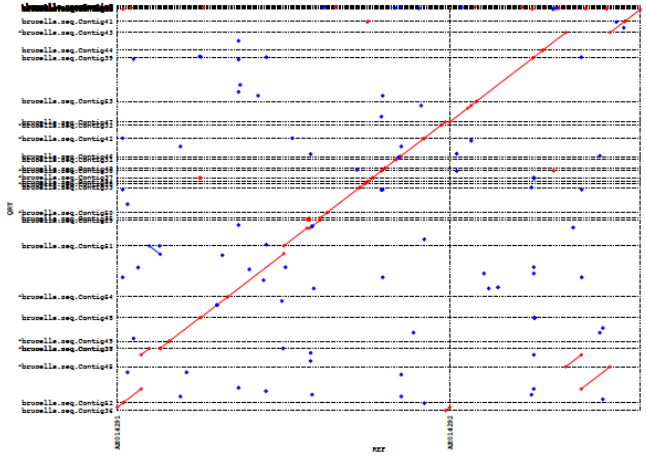
Short Reads Assemblers

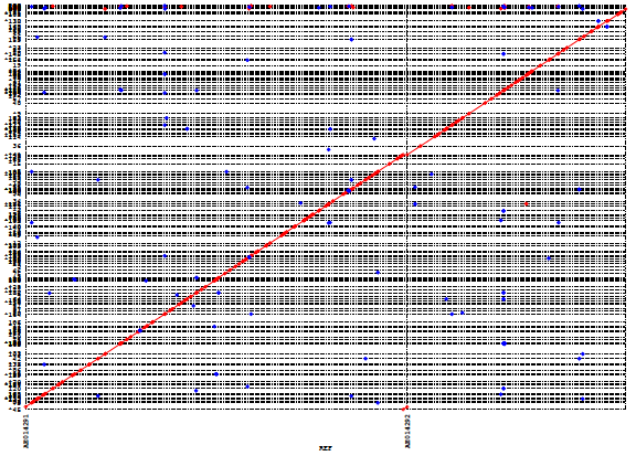
More than 20 published assemblers:

- How can we judge assembly quality?

N50 and Contig size

Given M contigs of size $c_1, c_2, ..., c_M$, N50 is defined as the largest number L such that the combined length of all contigs of length $\geq L$ is at least 50% of the total length of all contigs.





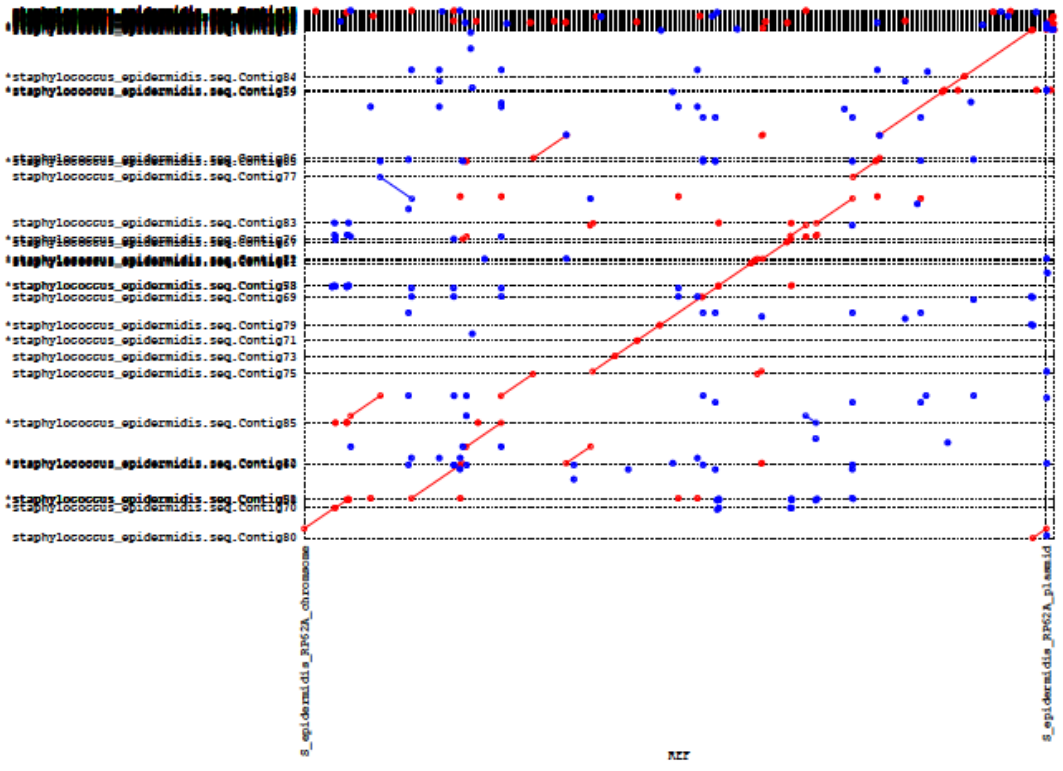
Few very long contigs: useless if mis-assembled.

Many short contigs: too short for annotation efforts.

Problem

Emphasize only size without capturing quality!!!

Counting errors



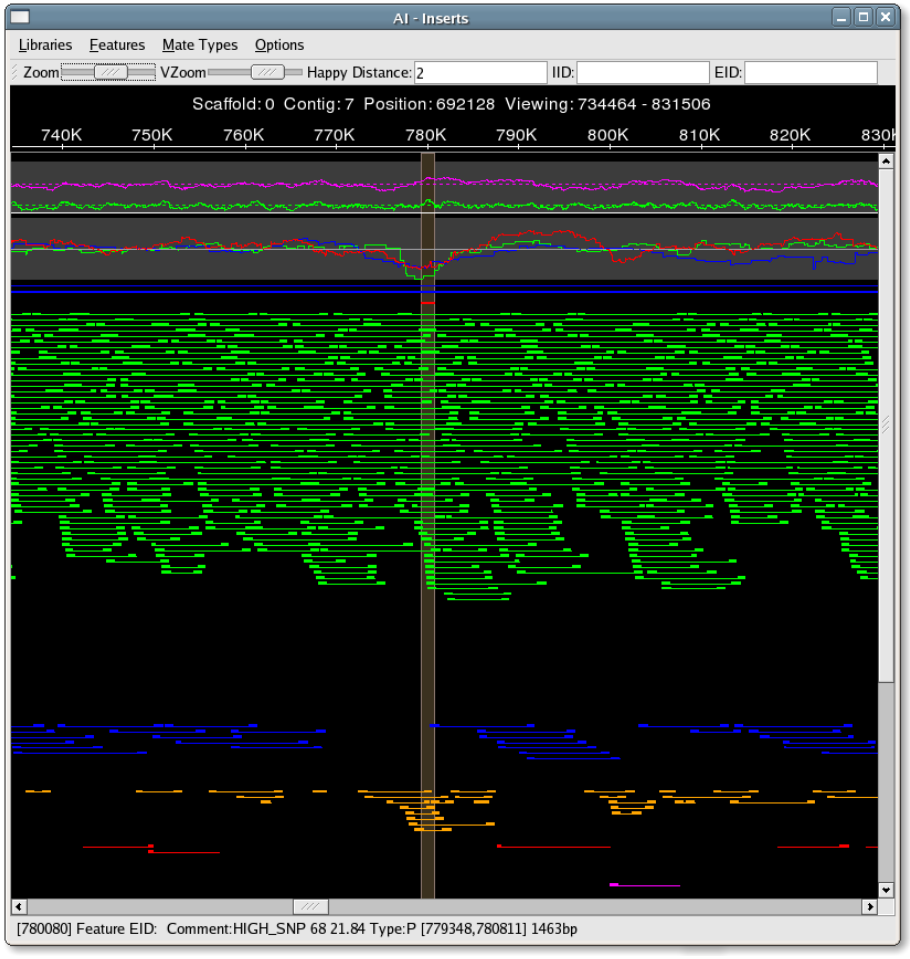
- Typically used for NGS data;
- Count the number of mis-assembled contigs by alignments to the reference genome;
- Problem: error types are not weighted accordingly

Visualization tools

- Hawkeye: Schatz et al., Genome Biology 2007;
- Good for inspection;

PROBLEM

Lack of automation!!



A wish list...

Ideal Metric

- A single value or function;
- Capture trade-off between quality and contiguity;
- Use long-range data (mate pairs, physical maps, *etc.*);
- No need for a reference;
- Easy to understand;

Features

N50, mean contig, max contig

Emphasize only size, while nothing (or almost nothing) is said about how correct the assemblies are.

Philippy et al.

Genome assembly forensics: finding the elusive mis-assembly

Features

amosvalidate pipeline returns for each contig its "features" – contigs or contig's fragment containing several different features suggest their "mis-assemblies" (i.e., errors).

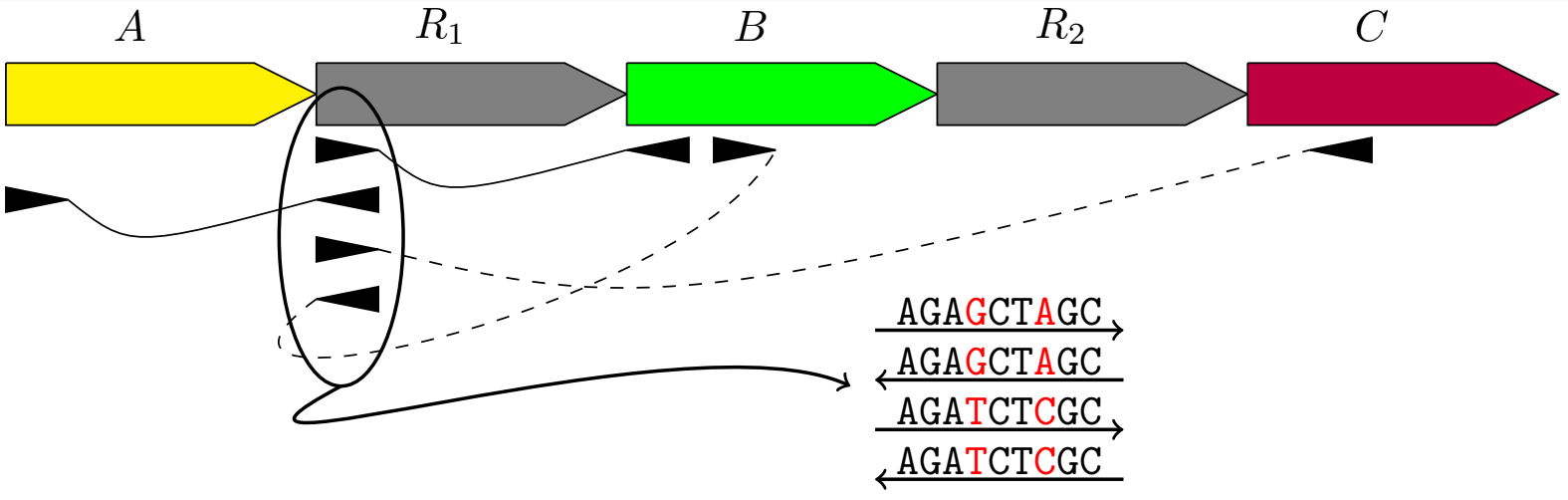
Features: One by One... (Philippy et al. 2008)

1. BREAKPOINT: left over reads partially align;
2. COMPRESSION: possible repeat collapse;
3. STRETCH: possible repeat expansion;
4. LOW_GOOD_CVG: normal oriented reads but at low coverage;
5. HIGH_NORMAL_CVG: normal oriented reads but at high coverage;
6. HIGH_LINKING_CVG: reads with mate in another scaffold;
7. HIGH_SPANNING_CVG: mate in another contig;
8. HIGH_OUTIE_CVG: incorrectly oriented mates ($\rightarrow\rightarrow$, $\leftarrow\rightarrow$);
9. HIGH_SINGLEMATE_CVG: single reads (mate not present anywhere);
10. HIGH_READ_COVERAGE: unexpected high local read coverage;
11. HIGH_SNP: SNP with high coverage;
12. KMER_COV: Problematic k-mer distribution.

If a contig is found to contain several features, then a likely explanation could be found in the contig's mis-assemblies.

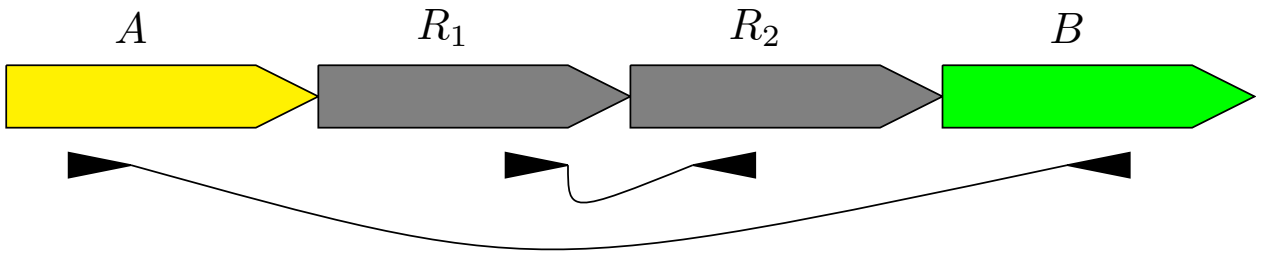
Assembly Features

SNPs as collapse indicators

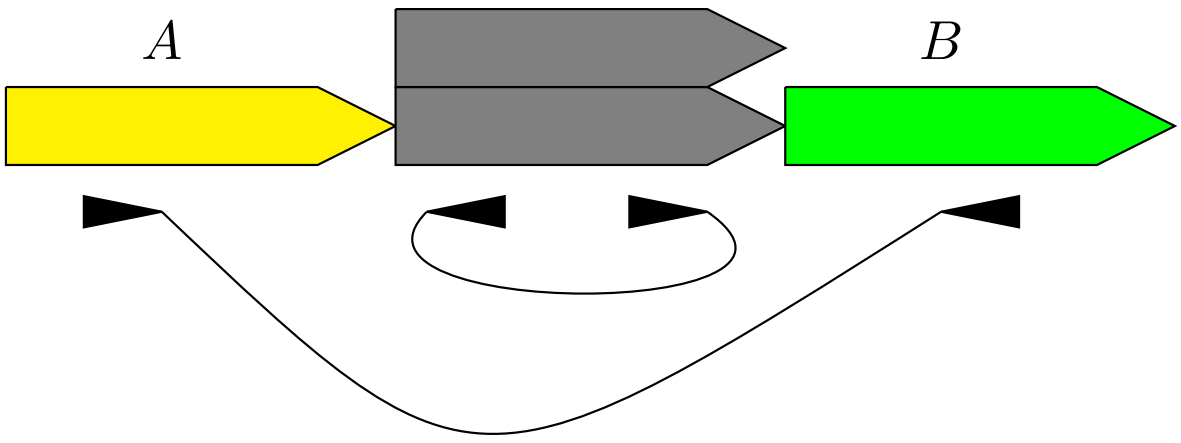


Assembly Features

Paired read suggesting errors (1)



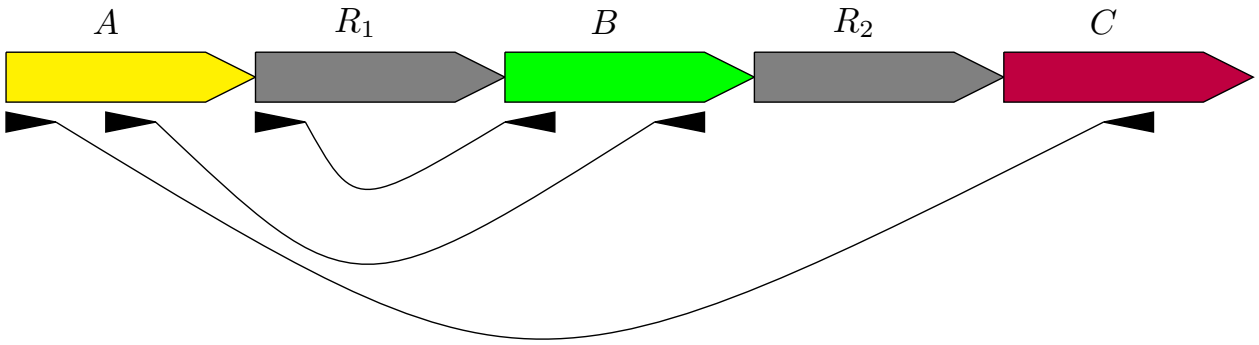
Correct Assembly



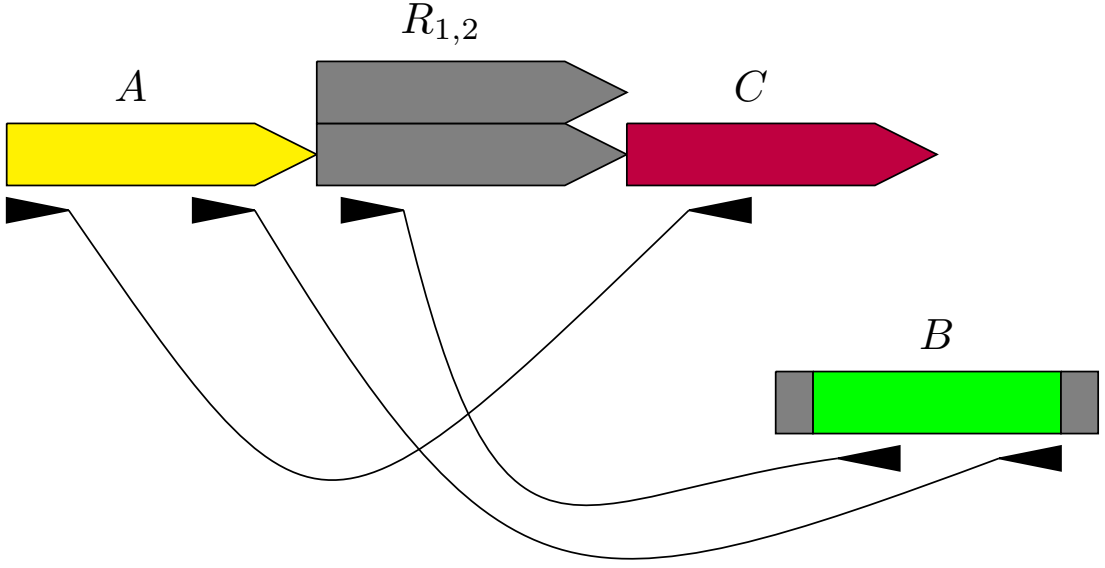
Misassembly

Assembly Features

Paired read suggesting errors (2)



Correct Assembly



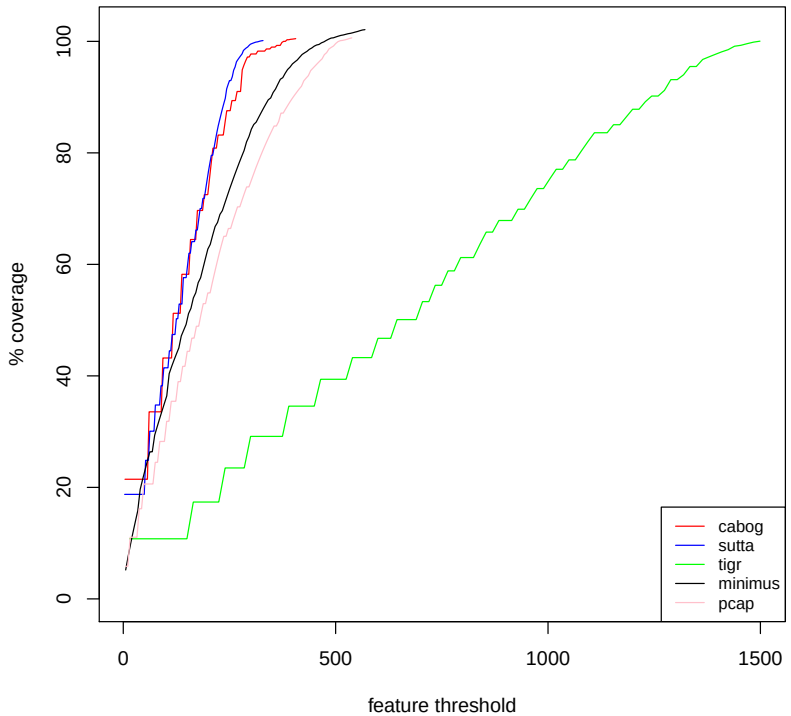
Misassembly

FRCurve (Narzisi and Mishra, 2011)

How can the feature counting allow us to compare and judge different assemblies/assemblers?

FRCurve (Narzisi and Mishra, 2011)

How can the feature counting allow us to compare and judge different assemblies/assemblers?



The Feature Response Curve (FRCurve) characterizes the sensitivity (*coverage*) of the sequence assembler as a function of its discrimination threshold (*number of features*).

Studying the Features

- A lot of features, are all necessary?
- Some features are deeply correlated
- In general features have high Sensitivity but low Specificity
- Are features "more informative" than standard measures?

PCA and ICA

Use multivariate techniques to understand how features are correlated (PCA) and what are the most important (independent) ones (ICA).

Experiments

20 genomes, 10 assemblers, real and simulated data:
more than 500 assemblies

PCA and ICA

Sanger/Illumina

1. Sanger
   - 20 real projects assembled with 5 different assemblers
   - 20 simulated coverages assembled with 4 different assemblers
2. Illumina:
   - 5 real projects assembled with 5 different assemblers
   - 20 simulated genomes assembled with 4 different assemblers

- PCA and ICA on 11 features plus N50 and NUM_CTG
- Easy work with Sanger... a nightmare with Illumina:
  - afg/bank is required to compute features
  - some tool perform scaffolding, others not
  - no standard datasets, assemblers highly dependent on parameters

PCA: Real Datasets

	Long Reads			Short Reads		
FEATURES	PC1	PC2	PC3	PC1	PC2	PC3
BREAKPOINT	0.29	-0.14	-0.21	-	-	-
COMPRESSION	0.32	0.22	0.35	-0.28	-0.15	0.24
STRETCH	-0.06	0.08	0.27	-0.3	-0.11	0.32
HIGH_NORMAL_CVG	-0.1	0.4	0.21	0.12	0.44	-0.09
HIGH_OUTIE_CVG	-0.07	0.56	-0.09	-0.32	-0.33	-0.29
HIGH_READ_COVERAGE	0.36	0.1	-0.13	-0.26	-0.3	-0.41
HIGH_SINGLEMATE_CVG	-0.01	0.27	-0.53	0.23	-0.26	-0.37
HIGH_SNP	0.05	-0.23	-0.13	-0.19	-0.05	-0.38
HIGH_SPANNING_CVG	0.28	0.38	0.31	-0.07	-0.38	0.12
KMER_COV	-0.03	0.37	-0.48	-0.08	-0.22	0.47
LOW_GOOD_CVG	0.5	-0.04	-0.02	0.41	-0.32	0.09
N50	-0.23	0.09	0.2	-0.48	0.08	0.1
NUM_CONTG	0.5	-0.03	-0.02	0.36	-0.41	0.12
cumulative variation	27%	44%	55%	26%	50%	63%

PCA: Simulated Datasets

	Long Reads			Short Reads		
FEATURES	PC1	PC2	PC3	PC1	PC2	PC3
BREAKPOINT	0.26	-0.38	-0.04	-	-	-
COMPRESSION	-	-	-	0.32	0.20	0.33
STRETCH	0.22	0.42	0.12	0.2	0.37	0.26
HIGH_NORMAL_CVG	0.02	0.2	-0.44	0.1	0.13	-0.62
HIGH_OUTIE_CVG	0.12	0.46	0.01	0.19	0.15	-0.536
HIGH_READ_COVERAGE	0.36	0.21	-0.19	0.35	0.09	-0.01
HIGH_SINGLEMATE_CVG	0.04	-0.07	-0.76	-0.11	-0.5	0.15
HIGH_SNP	0.3	0.02	-0.18	0.37	0	-0.06
HIGH_SPANNING_CVG	0.41	0.04	0	0.36	-0.24	-0.16
KMER_COV	0.24	0.37	0.16	0.31	0.28	0.28
LOW_GOOD_CVG	0.41	-0.28	0.04	0.34	-0.35	0.09
N50	-0.27	0.01	-0.3	-0.19	0.25	0.02
NUM_CONTG	0.39	-0.31	0.02	0.3	-0.42	0.03
cumulativevariation	36%	59%	70%	43%	62%	75%

ICA

Sanger (Real) ICA-Features

COMPRESSION, HIGH_OUTIE_CVG, HIGH_SINGLEMATE_CVG, HIGH_READ_COVERAGE, KMER_COV, LOW_GOOD_CVG

Illumina (Real) ICA-Features

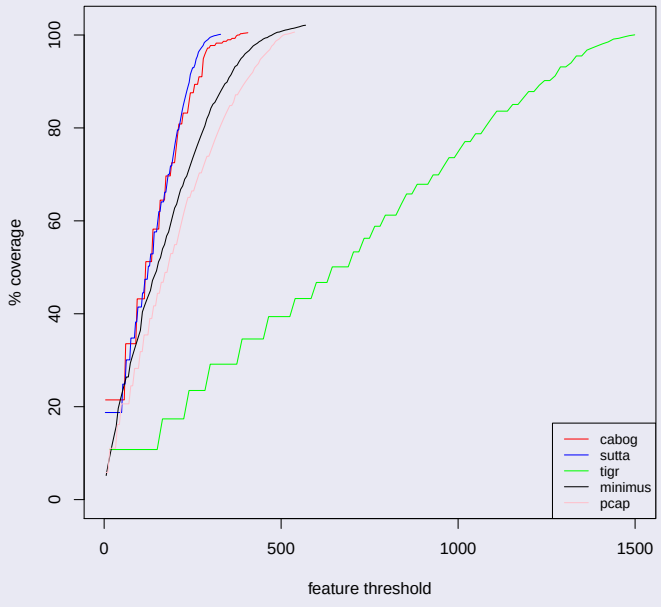
COMPRESSION, LOW_GOOD_CVG, KMER_COV, HIGH_SPANNING_CVG, HIGH_OUTIE_CVG, CE_STRETCH

Illumina (Simulated) ICA-Features

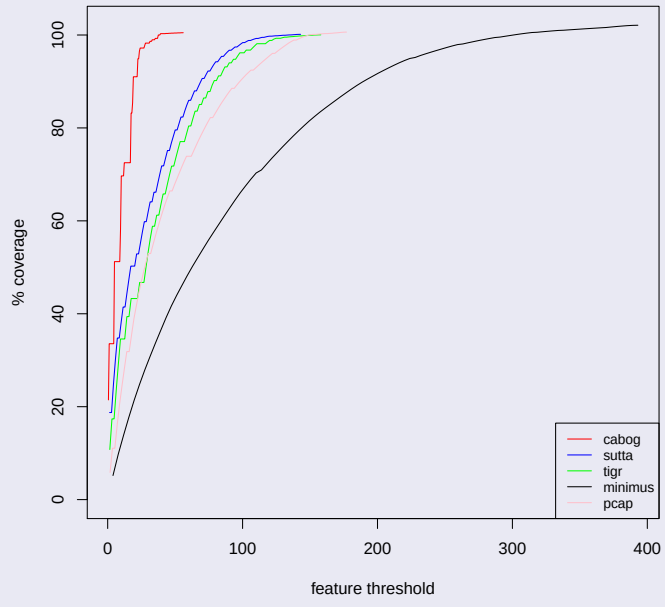
HIGH_READ_COVERAGE, HIGH_SNP, HIGH_NORMAL_CVG, HIGH_SPANNING_CVG, KMER_COV, CE_STRETCH

Long real reads: Brucella Suis

Feature Space



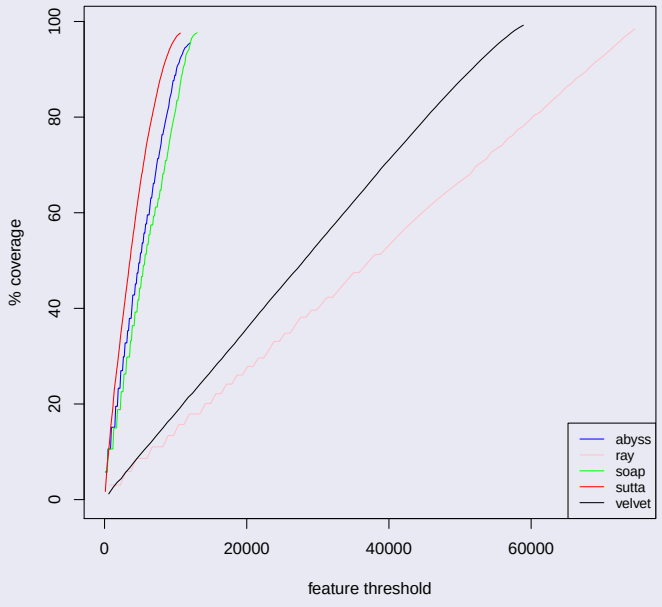
ICA space



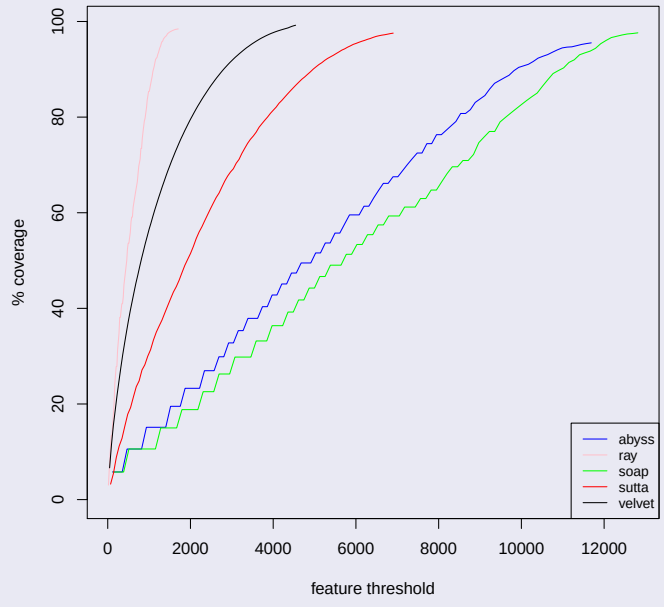
Assembler	# Ctg	N50 (Kbp)	Max (Kbp)	Errs	# Feat	# Feat corr	# ICA	# ICA corr
cabog	41	265	711	24	375	24	45	18
minimus	205	31	89	44	382	37	208	36
pcap	91	69	194	50	455	57	94	41
sutta	72	93	621	45	261	23	75	22
tigr	69	111	357	31	1281	24	134	20

Short real reads: E. Coli (130×)

Feature Space



ICA space



Assembler	# Ctg	N50 (Kbp)	Max (Kbp)	Errs	# Feat	# Feat corr	# ICA	# ICA corr
abyss	113	97	268	11	11804	119	11475	105
ray	194	58	140	17	74565	52	1701	30
soap	125	109	267	62	12254	174	12053	140
sutta	690	11	41	56	7949	140	5528	114
velvet	65	142	428	136	2156	26	131	2

PCA and ICA results

PCA analysis

- Feature space redundant.
- Lack of precise read simulators.
- N50 bad quality predictor!!

ICA analysis

- Possibility to reduce feature space.
- Improved accuracy (less false positive).

Problems

- FRC included in *AMOS* package:
  - based on *amosvalidate* package;
  - needs a **bank**, or **afg** output file
  - tool compatible with few (maybe 2) assemblers
- Features designed for Sanger data (*i.e.* leftovers);
- Features have high *Sensitivity* but low *Specificity*

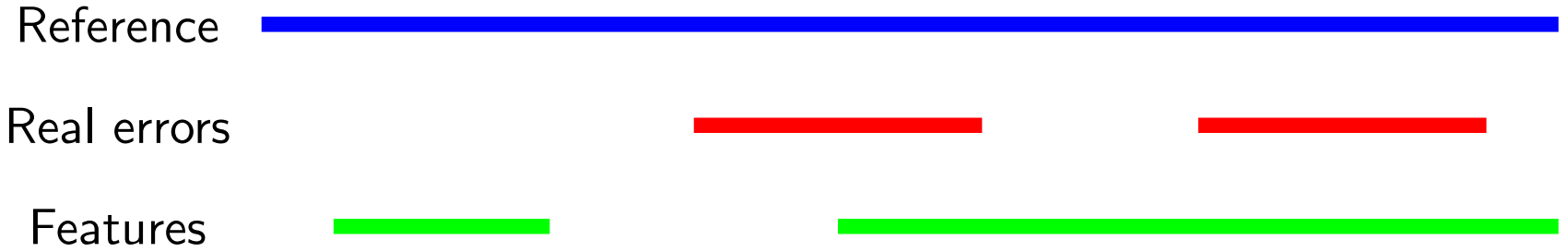
Sensitivity and Specificity

Sensitivity

$$Sensitivity = \frac{True\ \ Positives}{True\ \ Positives + False\ \ Negatives}$$

Specificity

$$Specificity = \frac{True\ \ Negatives}{True\ \ Negatives + False\ \ Positives}$$



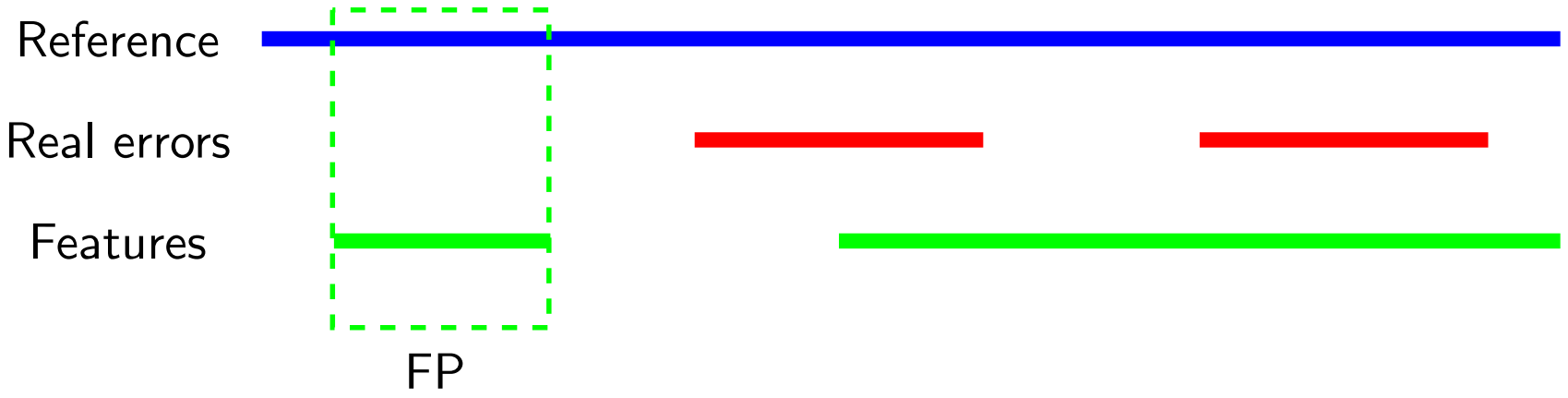
Sensitivity and Specificity

Sensitivity

$$Sensitivity = \frac{True\ \ Positives}{True\ \ Positives + False\ \ Negatives}$$

Specificity

$$Specificity = \frac{True\ \ Negatives}{True\ \ Negatives + False\ \ Positives}$$



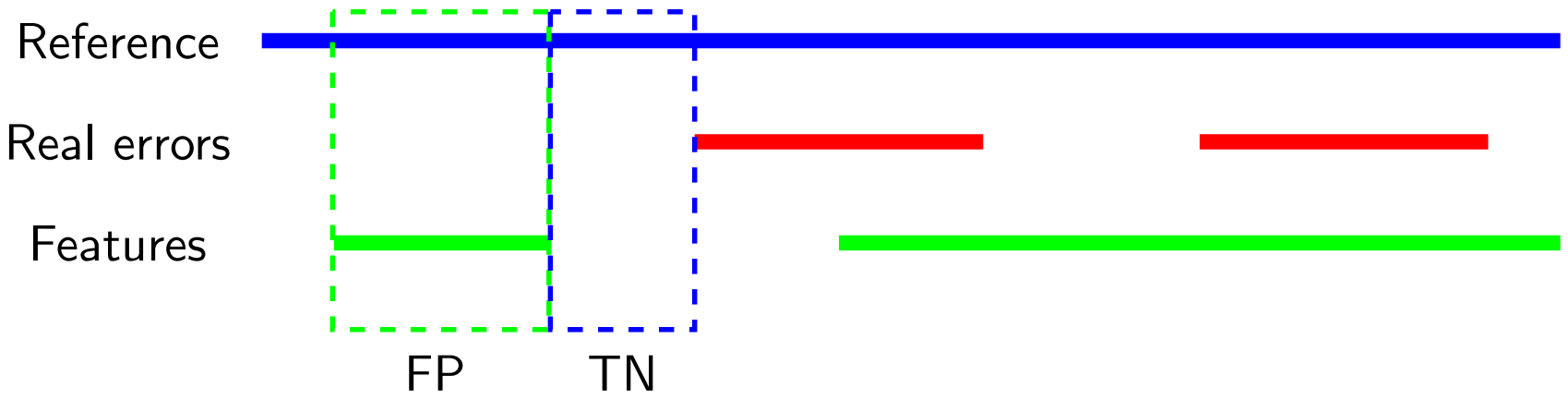
Sensitivity and Specificity

Sensitivity

$$Sensitivity = \frac{True\ \ Positives}{True\ \ Positives + False\ \ Negatives}$$

Specificity

$$Specificity = \frac{True\ \ Negatives}{True\ \ Negatives + False\ \ Positives}$$



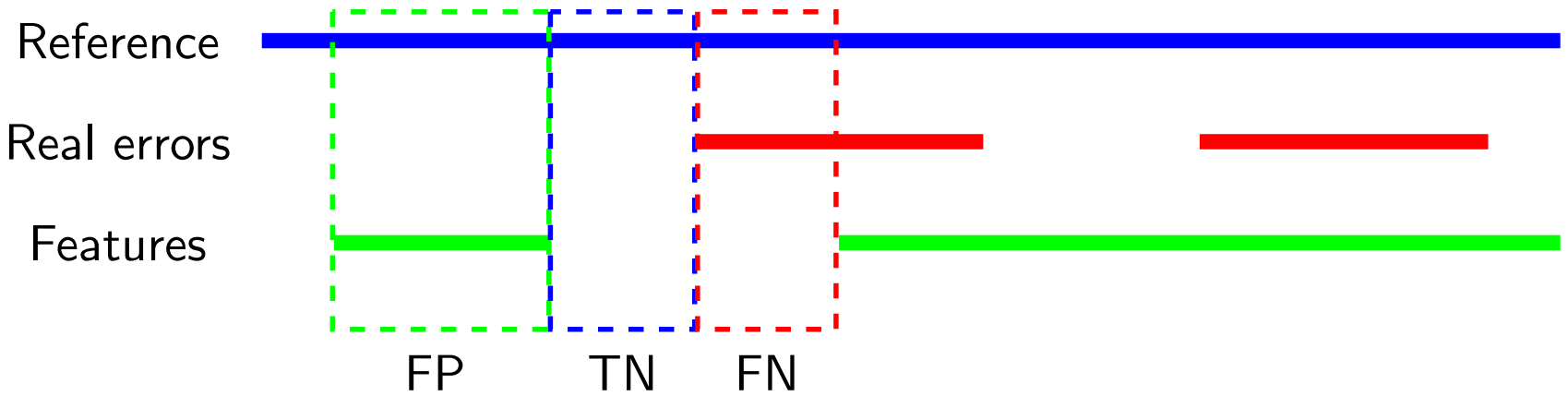
Sensitivity and Specificity

Sensitivity

$$Sensitivity = \frac{True\ \ Positives}{True\ \ Positives + False\ \ Negatives}$$

Specificity

$$Specificity = \frac{True\ \ Negatives}{True\ \ Negatives + False\ \ Positives}$$



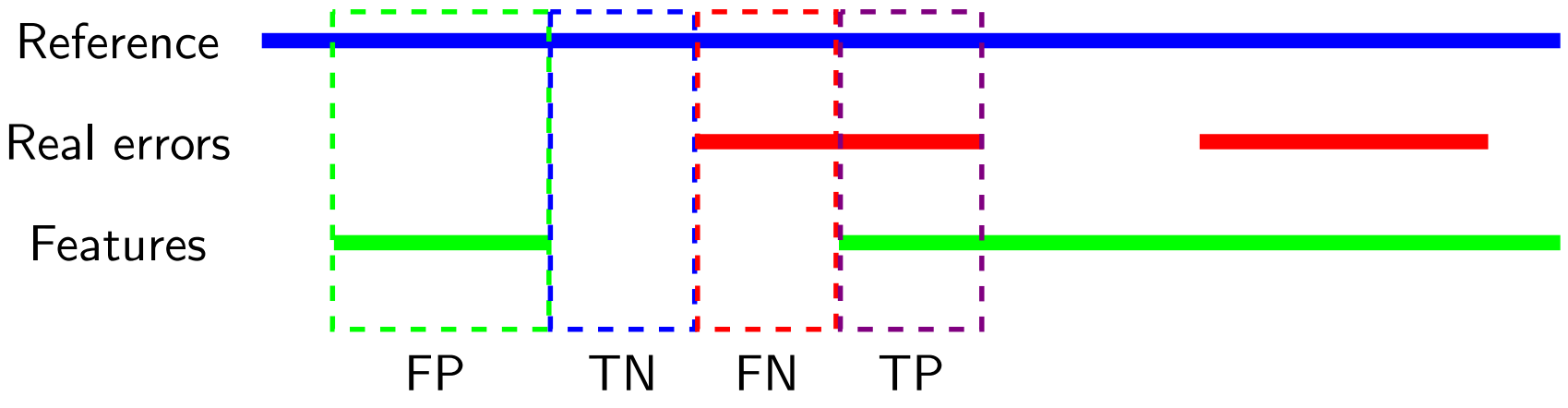
Sensitivity and Specificity

Sensitivity

$$Sensitivity = \frac{True\ \ Positives}{True\ \ Positives + False\ \ Negatives}$$

Specificity

$$Specificity = \frac{True\ \ Negatives}{True\ \ Negatives + False\ \ Positives}$$



Features from alignment

- NGS-based *de novo* assembler do not output *layout*
- Alignment only way to obtain an **approximate layout**:
  - alignment is a typical post-assembly procedure;
  - allows to design NGS-specific features (PE, MP)

FRC^{bam}

Read alignments (SAM/BAM format) and computes most important (ICA-independent) features:

- LOW_COV_AREA and HIGH_COV_AREA
- LOW_NORMAL_AREA and HIGH_NORMAL_AREA
- HIGH_SPANNING_AREA
- HIGH_SINGLE_AREA
- HIGH_OUTIE_AREA
- COMPRESSION and EXPANSION (CE statistics, Zimin *et al.*)

How to test?

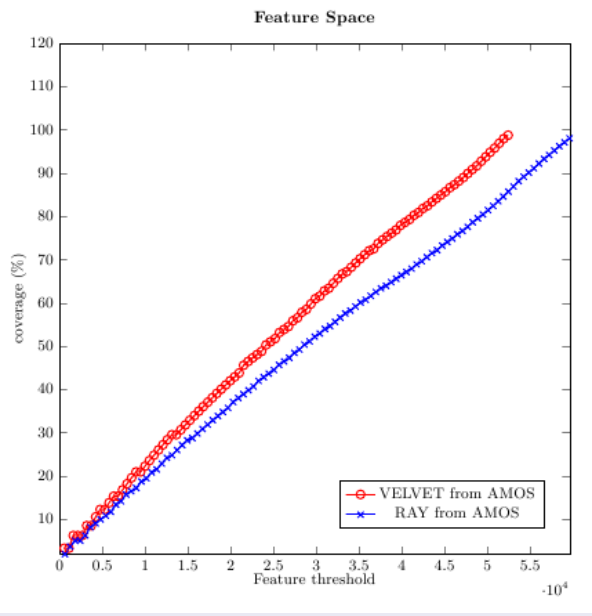
- Need of data and references;
  - Which datasets can we use?
- Relationship between *amos*-based features and *alignment*-based features:
  - can we trust *alignment*-based features?
  - need of *AMOS*-compatible assemblers
- Test *alignment*-based features on new data:
  - Sensitivity/Specificity
  - Comparison with alignment based validation

Genome Assembly Gold-Standard Evaluations

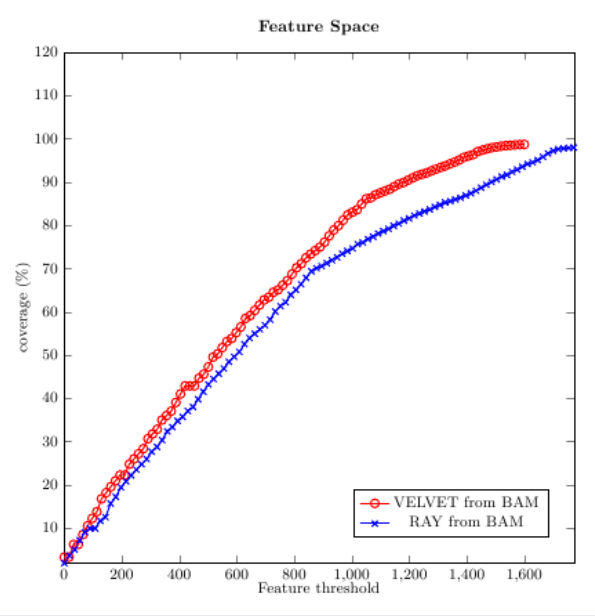
THE ASSEMBLATHON

GAGE: Staphylococcus aureus

AMOS Features



Alignment Features

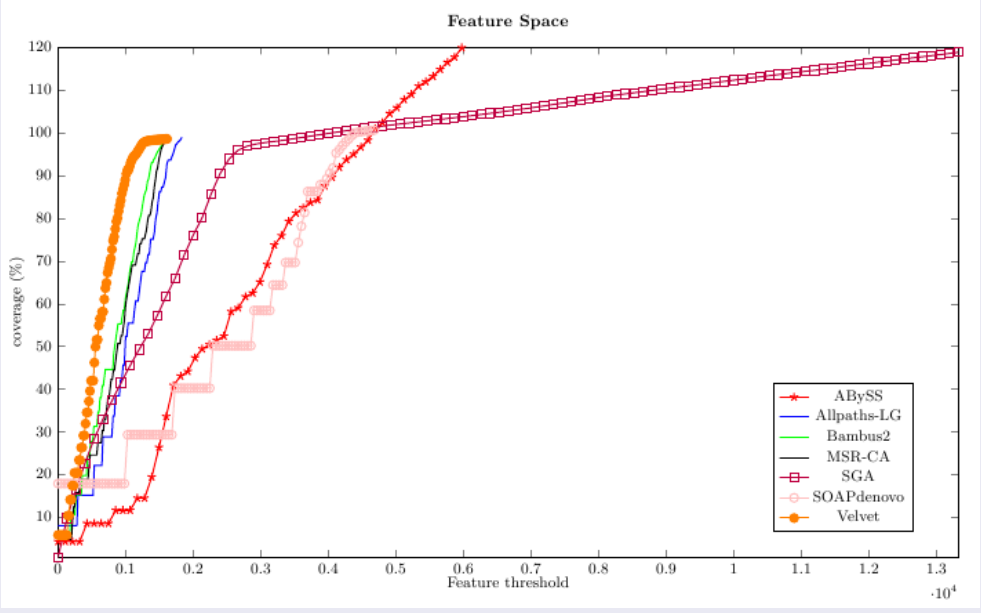


	# Ctg	N50 (Kbp)	ERRORS inser	trans	breakpoints	AMOS sens	AMOS spec	BAM sens	BAM spec
Ray	303	21.6	295	288	830	0.91	0.36	0.93	0.56
Velvet	438	10.9	270	441	1106	0.99	0.22	0.90	0.47

	% Real Errors	% AMOS feat	% BAM feat
Ray	2.5%	65.7%	45%
Velvet	1.4%	78.0%	53.4%

GAGE: Staphylococcus aureus

Alignment Features



	# Ctg	N50 (Kbp)	ERRORS Misjoin & Indels > 5	Chaff (%)	Dupl. Ref (%)	SNPs & Indels < 5	BAM sens	BAM spec
ABySS	302	29.2	19 (10+9)	66.00	23.30	278	0.91	0.32
ALLPATHS	60	96.7	20 (8+12)	0.03	0.03	83	0.88	0.52
BAMBUS2	109	50.2	190 (26+164)	0	0.01	84	0.90	0.53
MSR-CA	94	59.2	34 (24+10)	0.02	0.83	214	0.87	0.56
SGA	252	4.0	10 (8+2)	21.38	0.03	34	0.95	0.20
SOAP	107	**288.2**	65 (34+31)	0.35	1.44	271	0.96	0.22
Velvet	162	48.4	42 (28+14)	0.45	0.10	223	0.88	0.61

Conclusions

Features and FRCurve

- Features important instrument for assembly/assemblers evaluation.
- FRCurve useful instrument to gauge assembler performances:
  - one "simple" function;
  - reference free;
  - easy to improve

FRC^{bam}

- overcomes FRCurve/AMOS limits;
- possibility to develop NGS-based features;

What's next?

- improve features sensitivity and specificity;
- design application specific features (Fosmid pools, metagenomics, *etc.*);
- (sequencing) technology agnostic features (physical maps);

That's all Folks

Many Thanks to

- Prof. Lars Arvestad
- Prof. Bud Mishra
- PhD Giuseppe Narzisi

Thanks For The Attention!