

De novo Genome Assembly

Qi Zhou

Doris Bachtrog lab

CGRL seminar

zhouqi@berkeley.edu



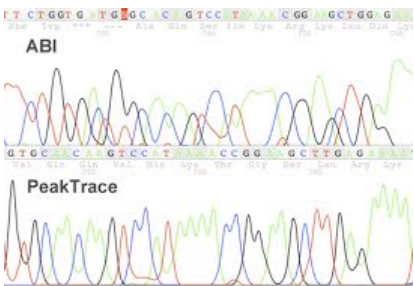
Platform summary

	Amplification	Sequencing	Read	Gbp/day	Cost \$/Mb	Status
Sanger	cloning	synthesis	1 kb	0.006	~500	Standard
454	emPCR	synthesis (multi-nt)	450 bp	0.5	~20	Standard
Solexa	PCR on slide	synthesis (single-nt)	2 x 100 bp	25	~0.5	Standard
SOLiD	emPCR	ligation	2 x 50 bp	10	~0.5	Standard
Ion Torrent	emPCR	semiconductor	100 bp	0.1?	~5?	Market 2011
PacBio	none	synthesis (real-time)	>10 kb	>100?	NA	First results

Characteristics of major NGS platforms as of February 2012

	Platform	Read length (bp)	Throughput & time per run	Technique	Dominant
Systems Genomics Sciences	HiSeq 2000 ^a	36 501 00	105–600 Gb 2–11 days	Reversible terminator	Substituted
	5500 SOLiD™ System ^b	356 075	7–9 Gb/day	Sequencing by ligation	–
		35	–	Ligation based	–
Sciences	HeliScope SMS ^c	25–55	21–35 Gb –	Single molecule sequencing	Insertion
Sciences	GS FLX Titanium XL+ ^d	≤1000	700 Mb 23 h	Sequencing by synthesis	Insertion
	Ion PGM Sequencer 318 ^e	>200	>1 Gb 2 h	Ion semiconductor sequencing	Insertion
Sciences	PacBio RS	1 k–10 k	–	Single molecule sequencing	Insertion

^aPerformance and Specifications. <http://www.illumina.com/systems/hiseq2000/performance-specifications.ilmn> (11 February 2012, date last accessed). ^b5500 SOLiD™ System: Specifications. <http://media.invitrogen.com.edgesuite.net/solid/pdf/COI8235-5500-Series-1> (11 February 2012, date last accessed). ^ctSMS™ Performance. <http://www.helicosbio.com/Technology/TrueSingleMoleculeSequencing/tabid/151/Default.aspx> (11 February 2012, date last accessed). ^dGS FLX Titanium XL+. <http://my454.com/products.aspx> (11 February 2012, date last accessed). ^eIon Personal Genome Machine™ Sequencer: Performance. <http://www.ion Torrent.com/how-does-it-perform/> (11 February 2012, date last accessed).



Sequencing info
 Nucleotide sequence
 Quality score in ASCII



Reads

Assembler

Genome



```
>gi|15611071|ref|NC_000921.1| Helicobacter pylori J99, complete genome
AGTGATTAGTGATTACAGCATCATTTTTTAAATTTAGGCATAAAACGCCCTTAAATCAAGGGTTTTTGAG
CGAGCTTTTTGCTCAAAGAATCCAAGATAGCGTTTAAAAATTTAGGGGTGTTAGGCTCAGCGTAGAGTTT
GCCAAGCTCTATGCATTCATTGATGATGATAGGGTTTTGTGTGGGCGTGAAAGCCAATTCATACGCTCCT
AAGCGTAAAAATCGCCTTTTCCATGCTCCCTAATCGCTTGAAATCCCAAGTCTTTTAAATGCGGTTTCGATGA
GAGCGTCAATCTCATTTGATTTTTTCTAACACGCCATTAAAAAGGCTTAAAGCGAAAGCGAGCTGGTTGTT
TTTAATCTTTTTTTCTTCAACATGCTGGAAGCGATTTTTTTAAATTTCTTCATTACCGCTCTCAAAACGCA
TACAACAATCAACCACAGCCCCCTGGGCTTGAGTTCGTGTCGCCATTTTAAATCCTTGAGAGTTTGGCAC
AAGCTCAATAATTCAATGAGGGTGCTCATCGCTTCAAAGCCCTTATTACCGGCTTTACTGCCGCTCTTT
CAATCGCTTGTTCAATATTGTCTGTGTCAGCACGCCAAAAGCTTACCGGCTGCTGTATTTTATGATCGC
ACTAGCAATGCCCTTAGTCGCTTCCGGCTCAGTAGTCAAAATGCGGAGTCCCCCTCTAATGATCGCT
CCTAAAAACGCACAGCCATCGTATTTTCCGCTCTCTAACAATTTGTCTAAAAATCAAAAGGCAATTCATAAG
CCCCAGGCATAGCACGAGATCTAAAAGCTCCTCATCGCCCCATGCTTTTAAAGCAATCCATCGCCCC
TTCTTTTAATCTGCTGTGATGATATGGTTGAAAGCGCGATGTTAAAAATAGCGATTTTTTTCATTCCCTTGT
AATTGCAATTTCCCTTCTATGATTTGCGATGAAATTCCTTTAAAAATAAATTTGGATTTTTAATATGTCGG
TTACTAAGTGTTCTAGCTCGTCAGGTTTTAGCATGTTTGTCTCCATCGCTTAGGGCGTTTTTAGGATCAAT
ATGCGTTTACGCGAACAGCCCATCAATCCCCACGCCGCCGCGCTCTGGGTAAAAATAGGGGAAAAAGAG
```

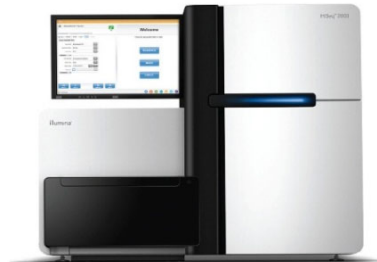
Get low amount (1ug) of DNA sample of your interest



Library prep (1~2 days), Quality check



Sequencing 1~2 months



assembly:1 week



2 months or so, get a genome

How to get a good genome in a cost-effective way?

- What do you mean by 'good genome'?
one chromosome one sequence?
- How much shall I sequence?
~30 fold illumina data, well it also depends on the assembler you want to use
- Platform to use
- Data quality
- Assembler to choose
- A reference genome available?

- **What kind of data are you getting?**
- Read quality and error correction
- Assembly
- How to assess the draft genome
- Mapping and chromosomal build
- SNP calling, RNA-seq analysis etc.

- 454/Sanger : longer reads, more accurate at repeat region; cost much more

Overlap-layout-consensus:

Phrap/Phred, Lasergene suite, Newbler, CABOG..

- Illumina : time/money cost-efficient

de bruijn/string graph:

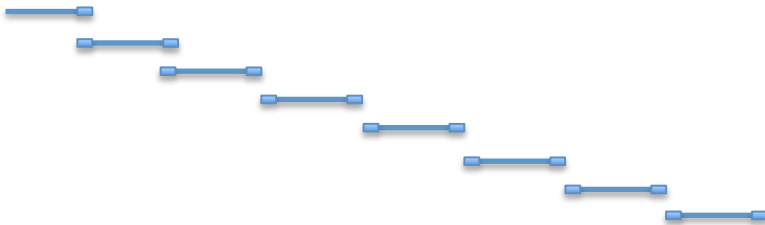
SOAPdenovo, allpaths-lg, SGA etc.

[Seqwiki](#) : a list of softwares



Sanger/454: overlap-layout-consensus algorithm

1 run of 454 reads: 500,000 reads $\rightarrow 2.5 \times 10^{11}$ pair-wised alignments



Illumina:

1 run of Hi-seq: 6×10^9 reads $\rightarrow 36 \times 10^{18}$ pair-wised alignments

Any application. Any study. Any budget.



HiSeq 2500/1500

Two run modes. High output or rapid run.

Flexibility to batch process multiple samples with high output in a single run, or get rapid results with fewer samples for time-critical studies.



HiSeq 2000/1000

One run mode. High output.

Generate the highest output and number of reads for batch processing multiple samples in a single run.

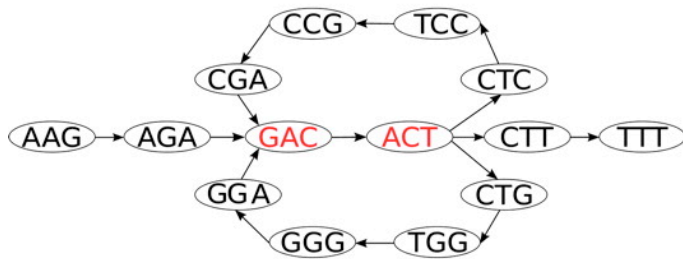
	HiSeq 2500		HiSeq 1500	
Run Mode	High Output	Rapid Run*	High Output	Rapid Run*
Output (2 x 100 bp)	600 Gb	120 Gb	300 Gb	60 Gb
Run Time (2 x 100 bp)	~11 days	~27 hours	~8.5 days	~27 hours
Cluster Generation	cBot	On board	cBot	On board
Paired-end Reads	6 Billion	1.2 Billion	3 Billion	600 Million
Single Reads	3 Billion	600 Million	1.5 Billion	300 Million
Maximum Read Length**	2 x 100 bp	2 x 150 bp	2 x 100 bp	2 x 150 bp
Bases Above Q30***	> 85% (2 x 50 bp) > 80% (2 x 100 bp)			

	HiSeq 2000	HiSeq 1000
Output (2 x 100 bp)	600 Gb	300 Gb
Run Time (2 x 100 bp)	~11 days	~8.5 days
Cluster Generation	cBot	cBot
Paired-end Reads	6 Billion	3 Billion
Single Reads	3 Billion	1.5 Billion
Maximum Read Length**	2 x 100 bp	2 x 100 bp
Bases Above Q30***	> 85% (2 x 50 bp) > 80% (2 x 100 bp)	

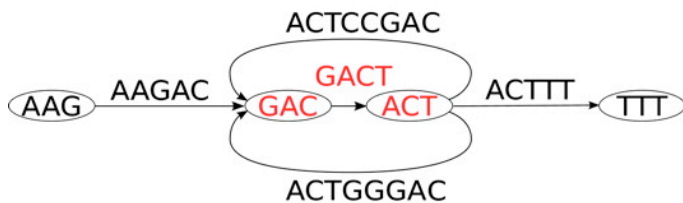
Solution?

Break the trillions of reads into zillions of even shorter fragments (kmer)!

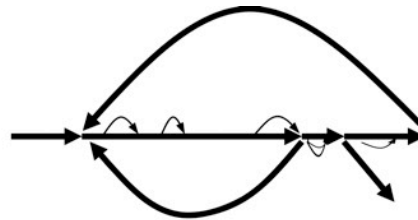
AA**GACT**CC**GACT**GG**GACT**TT



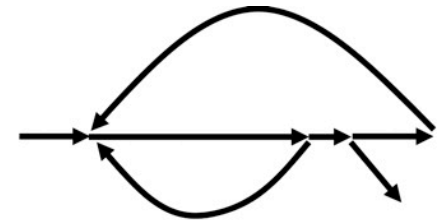
A de Bruijn graph of a sequence



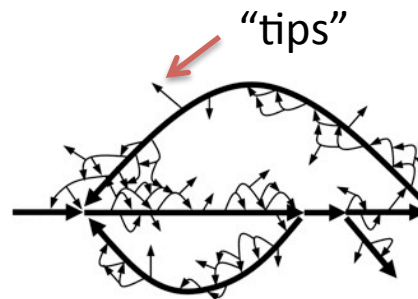
B condensed de Bruijn graph



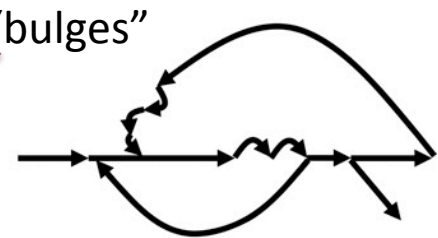
C de Bruijn graph of a genome



E repeat graph of a genome



D de Bruijn graph of a set of reads

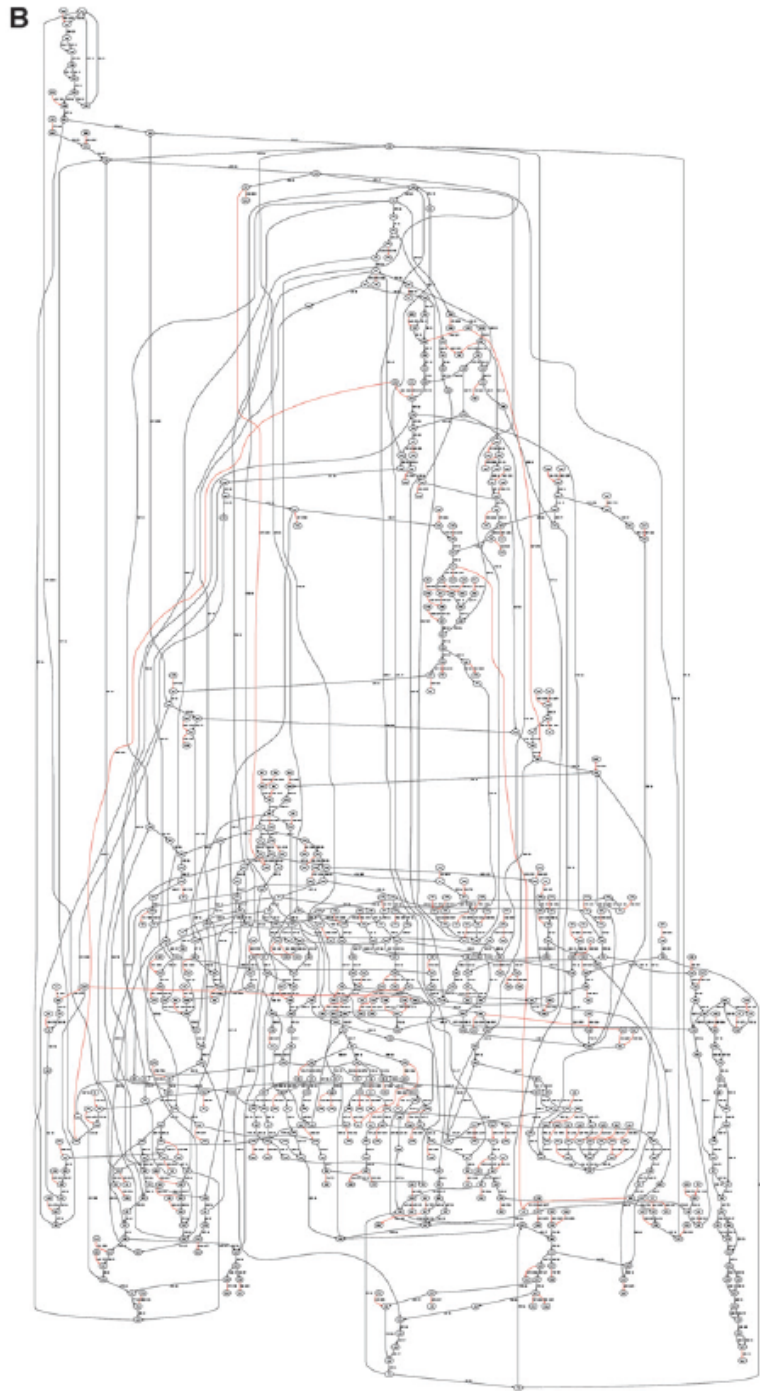


F repeat graph on a set of reads

- Error correction: most of the time we're dealing with a diploid genome
- Discriminate between sequencing errors and polymorphism
- Repeat resolution: insert-size of PE library

Something I'm perfectly happy there's no need to look into it by my eyes..

A genome-wide *de bruijn* graph!



- What kind of data are you getting?
- Read quality and error correction
- Assembly
- How to assess the draft genome
- Mapping and chromosomal build
- SNP calling, RNA-seq analysis etc.

Fasta format

```
>@HWI-ST450R:198:B00R0ACXX:3:1101:1102:2231/1
NAGTTAGCAAATCGGGTGGCCTTATTTTCAACCTGGACAACCATGTACCCCGCATCGAGCAGAGAAGACATTACATATCCCCTGGACCTGGTGCTGGAG
>@HWI-ST450R:198:B00R0ACXX:3:1101:1139:2236/1
NTTCCGATGGGGCACCGCCTTCTGGGCGGCCAACTCCCGCAGTTTCGTTTACGAGCACATGCATCTGATACTTGGAAAGATCGGAAGAGCGGTTTACGAGGAT
>@HWI-ST450R:198:B00R0ACXX:3:1101:1190:2238/1
NTCGGCAGCTGGCTTGAACACCGCTTCAACAACGTGGCTCTCGGCAGATTCTCTGATGAGCTGCGTCGTTGAGGTGGAGATCAAGCAAGCGGAACAGAAGCC
>@HWI-ST450R:198:B00R0ACXX:3:1101:1421:2224/1
NTTGGATTGGATTGGATTGGGAAGTAGAGGAAGAGTCACCAAAAATAAACGGCGAAAATGTGGCCCAACTTTTTTGGAGATCGGAAGAGCGGTTTACGAGGA
>@HWI-ST450R:198:B00R0ACXX:3:1101:1257:2227/1
NATTTTTGCGGGTAATTTATTTGCGTTTTTCAGAACAGAATTTGCGGAGATCGGAAGAGCGGTTTACGAGGAATGCCGAGACCGATCTCGTATGCCGTCTTT
>@HWI-ST450R:198:B00R0ACXX:3:1101:1265:2228/1
```

Fastq format

```
@FCC0WMIACXX:8:1101:1721:2192#TAGGAATA/1
AGGATGGTGGAAGCGTGAAGCCCGCACCACCAGTTGCAGGCAGCGACGACGAGCGGGCGCAGAAGATACTCGAAGATACTACGGTGAGAGTGGGCCAACG
+
_abeceecgegggihhadgebffhihiiiiifigeefhi`ghhhiiifeecccccc_aaaccccccbcbbc^acc_bbc_bbac`ccccS]^a^aacca
@FCC0WMIACXX:8:1101:1922:2135#TAGGAATA/1
TGGGCAATATGCAAAAATGCCAGCGGACCCAACTCAACTCCTTCTCTTTTTTCGATTCCCTTTCCCTTTTGACCATTTAGTCCCAAAATCCAAATCCCT
+
___cccdgg]acfhffhhdgfffageghidaf`ffdfhhihfffffhfh_gfYP\R^baccYZ]]_U_bbab]^`cbccb]bGKTR[[_]`bccb]^b
@FCC0WMIACXX:8:1101:1985:2180#TAGGAATA/1
CGATTTTGTGTAGAAATAAACTAAATATAAGGTCGAATTAATTAATGAGTTTGGTGCAAAAGTGTTTTTGTAAATGGTGTCGGTTTTTGGTCGATT
+
^[_eeeeega^ecgfhhhiiiihihiibgfhdffhhihhhhhiiafgihaegbfhigbb_bfgigggeeeceebdZZ`bac^aca_caW`a_ac
@FCC0WMIACXX:8:1101:1867:2225#TAGGAATA/1
AGAGCTAGAGCAACCAATTTGGTATCCACACATTTACCGTGCGAGCAGCCATCAACTCCCGCGACAGCCTGGACCTAACCTTGACATAAACATTCGAAA
+
_a_eedcggfggiihhiifhicbbaeghhhiifhiiiiicbgfhiihfhfhfhfg_gfg]ga`_aZ^`bXXX`b^bccbcccccbcb_bbb`bedcc
@FCC0WMIACXX:8:1101:1870:2225#TAGGAATA/1
```

```
@FCC0WM1ACXX:8:1101:1922:2135#TAGGAATA/1
TGGGCAATATGCAAAAATGCCAGCGGACCCAACTCAACTCCTTCTCTTTTTTCGATTCCCTTTCCCTTTTGACCATTTAGTCCCAAAATCCAAATCCCT
+
__cccdgg]acfhffhhdgfffageghidaf`ffdfhhihffffhfh_gfYP\R^baccYZ]]_U_bbab]``cbccb]bGKTR[[_]`bcccb]`b
```

- FCC0WM1ACXX: instrument name
- 8: run id
- 1101: flowcell id
- 1922: 'x'-coordinate of the cluster within the tile
- 2135: 'y'-coordinate of the cluster within the tile
- #TAGGAATA: index sequences
- /1: first end of the mate pair

Sequences

+

Quality vs. mapping quality → [An explanation here](#)

$$Q = -10\log_{10}(e)$$

The Relationship Between Quality Score and Base Call Accuracy

Quality Score	Probability of Incorrect Base Call	Inferred Base Call Accuracy
10 (Q10)	1 in 10	90%
20 (Q20)	1 in 100	99%
30 (Q30)	1 in 1000	99.9%

References

1. Ewing B, Hillier L, Wendl MC, Green P. (1998): Base-calling of automated sequencer traces using phred. I. Accuracy assessment. Genome Res. 8(3):175-185
2. Ewing B, Green P. (1998): Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Res. 8(3):186-194

1. Each base pair has a quality score
2. mapping quality/calibrated quality is more accurate than the raw quality

How to assess the read quality?



Babraham Bioinformatics



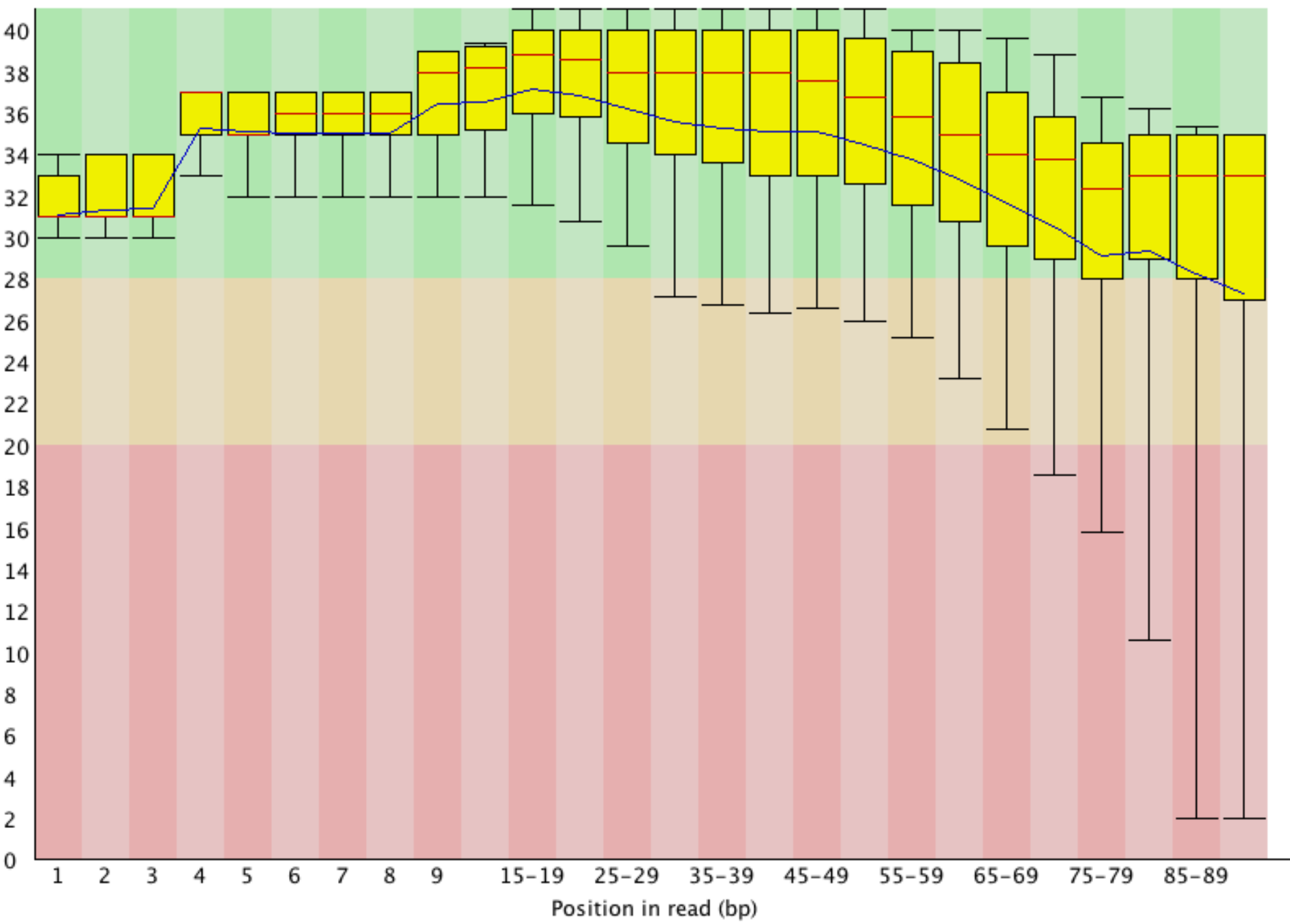
[About](#) | [People](#) | [Services](#) | [Projects](#) | [Training](#) | [Publications](#)

FastQC

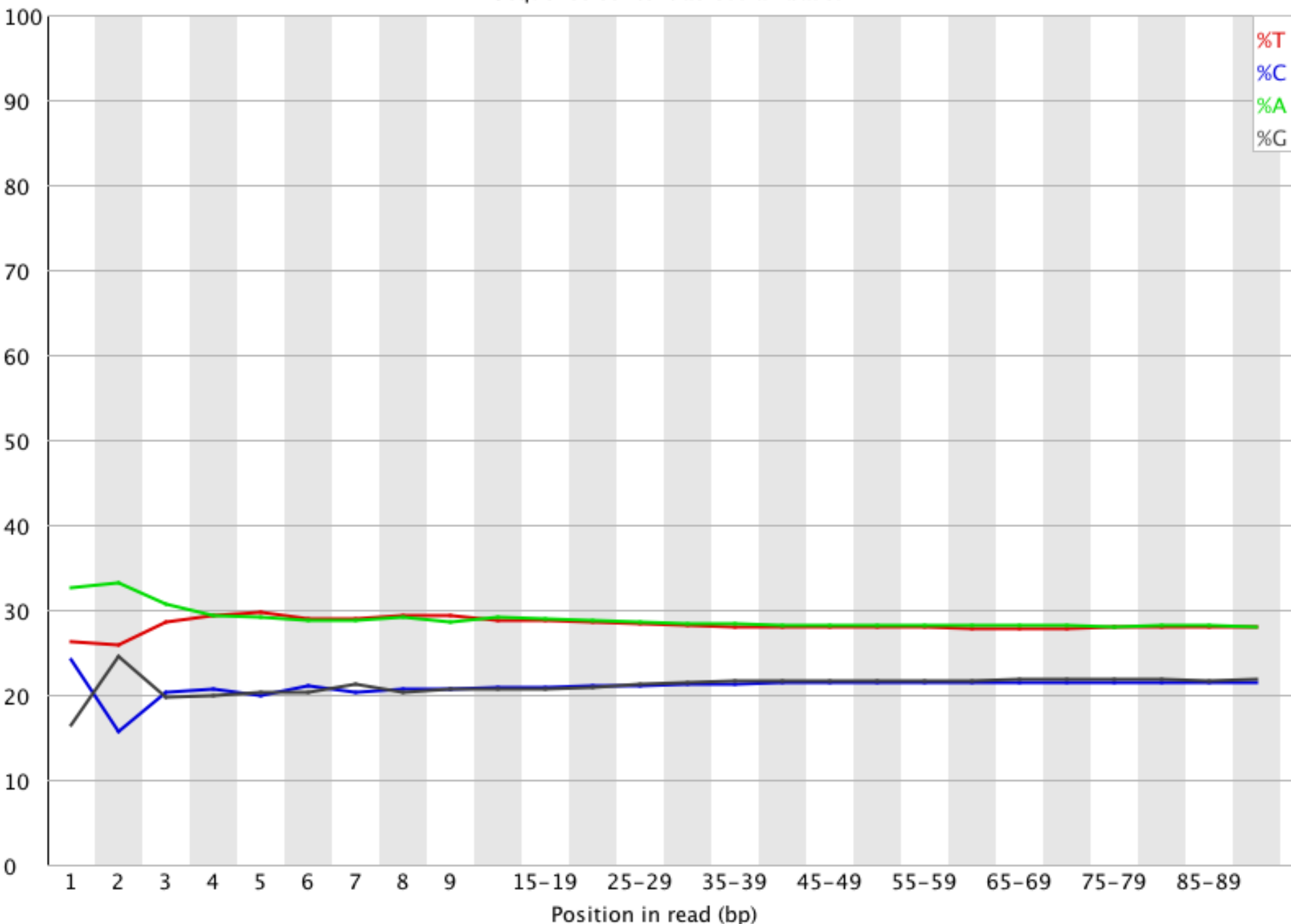
Function	A quality control tool for high throughput sequence data.
Language	Java
Requirements	A suitable Java Runtime Environment The Picard BAM/SAM Libraries (included in download)
Code Maturity	Stable. Mature code, but feedback is appreciated.
Code Released	Yes, under GPL v3 or later .
Initial Contact	Simon Andrews

[Download Now](#)

Quality scores across all bases (Illumina 1.5 encoding)



Sequence content across all bases



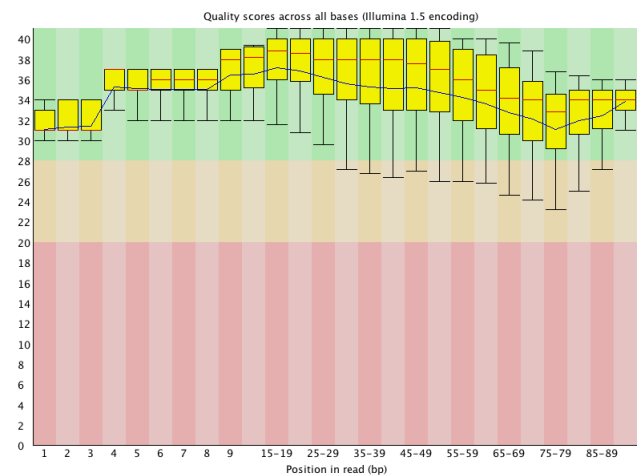
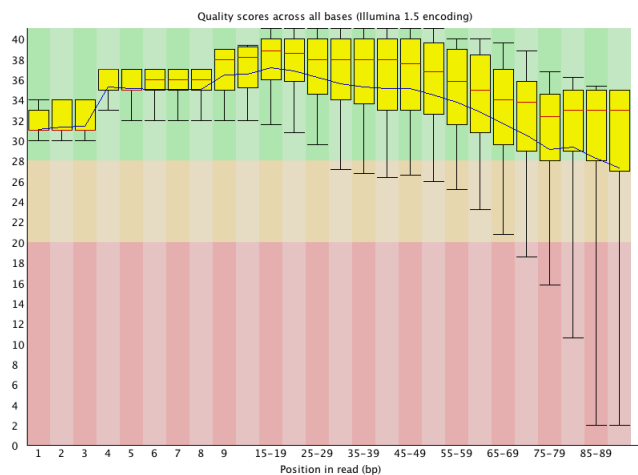
- There are a couple of reasons you may have bad quality reads:
 - 1) Bad quality of DNA/RNA template
 - 2) Something goes wrong with the sequencing reagent
 - 3) Adapter contamination
- Base composition is more informative than base quality: You'd sequence adapter sequences to very high base quality...
- Chop off those base pairs with abnormal base compositions

FASTX-Toolkit

FASTQ/A short-reads pre-processing tools

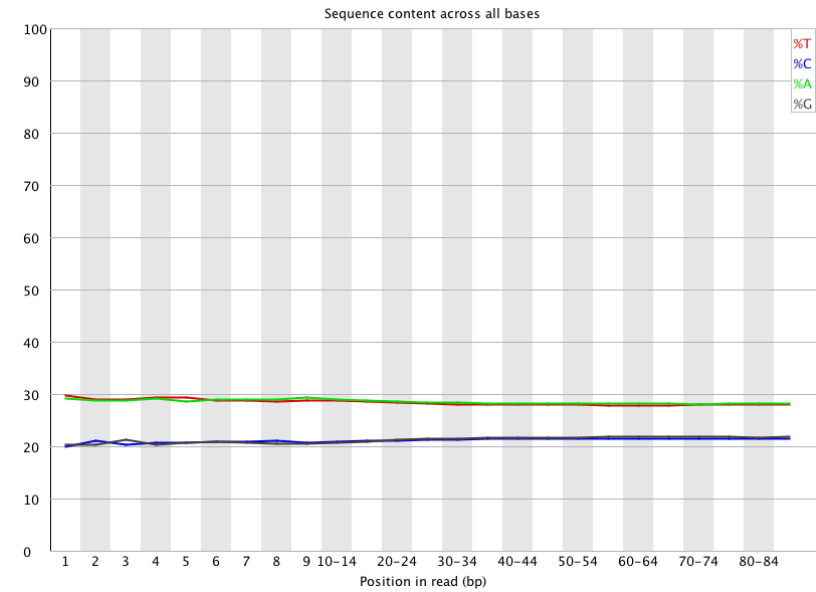
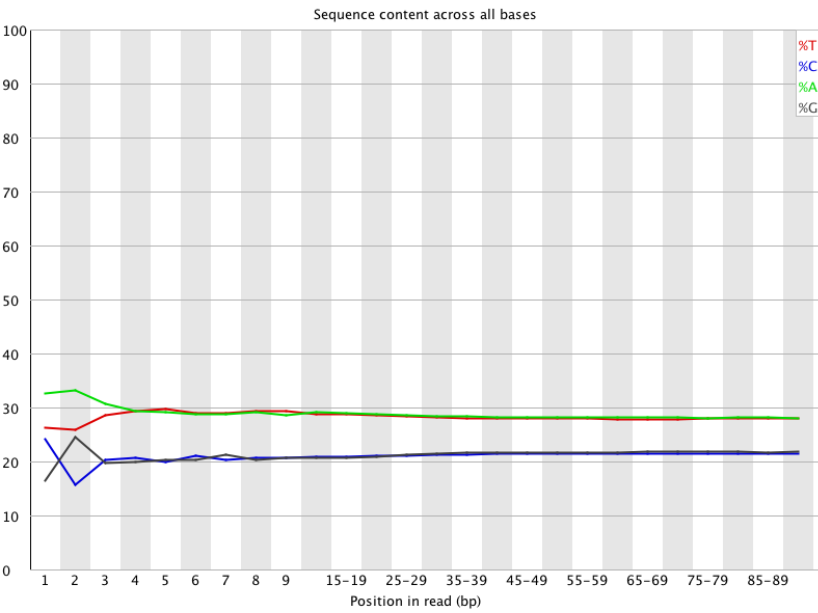
[Home](#) | [Download & Installation](#) | [Galaxy Usage](#) | [Command-line Usage](#) | [License](#) | [Useful Links](#) | [Contact](#)

```
./fastq_quality_trimmer -t 30 -l 30 -i ../test_data/test.fq1 -o test.qualtrim.fq1
```



- What kind of data are you getting?
- Read quality and error correction
- Assembly
- How to assess the draft genome
- Mapping and chromosomal build
- SNP calling, RNA-seq analysis etc.

```
./fastx_trimmer -f 5 -i ../test_data/test.fq1 -o chopped.first3bp.test.fq1 &
```



How to correct bad-quality reads?

- Lots of the 'corrector' relies on really high sequencing coverage (>50 fold) to call a non-biased distribution of k -mer so that they can correct reads.
- Not suitable for polymorphism/pool-sequence data

Some popular correctors

- Quake: [website](#)

Quake.py -f [fastq_file_list] -k 31 -p 4

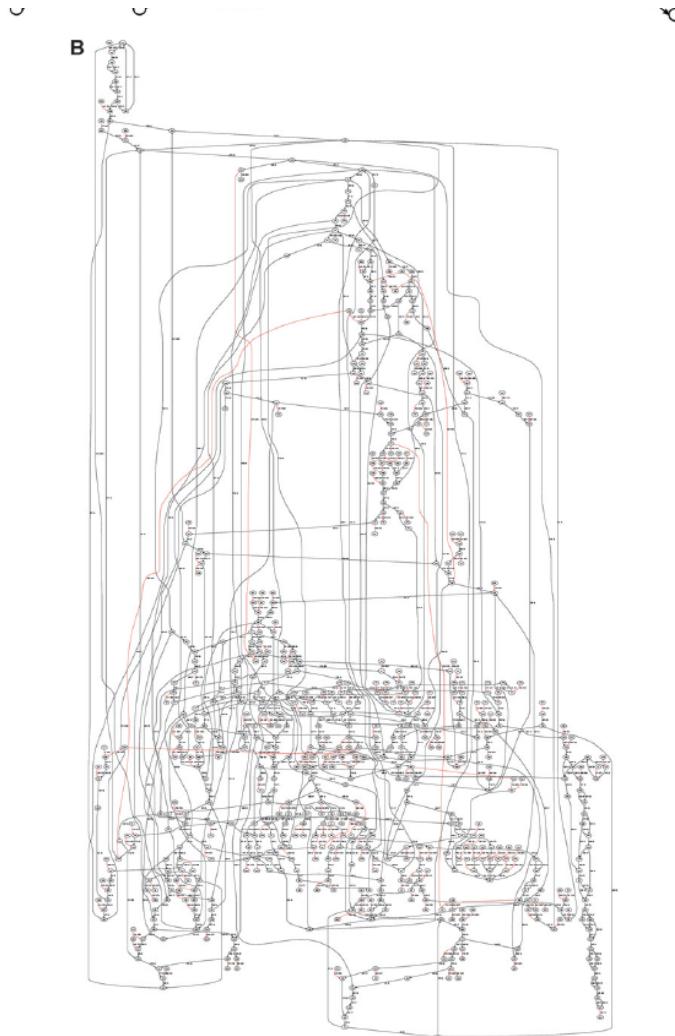
- SOAP reads corrector: [Download link](#)
- Coral: [Download link](#)

Table 4 SOAPdenovo bee assembly

Assembly	Trimmed Only	Corrected	Removed	Contigs	N50	N90	Scaffolds	N50	N90	Reads
Uncorrected Corrected	146.0 M	-	12.9 M	312,414	2,383	198	90,201	37,138	9,960	167.3 M
SOAPdenovo Corrected	134.4 M	15.7 M	15.6 M	188,480	4,051	515	36,525	36,525	9,162	164.8 M
Quake	146.9 M	16.5 M	13.0 M	189,621	4,076	514	37,279	37,014	9,255	167.3 M

SOAPdenovo assemblies of *Megachile rotundata* 124 bp paired end reads. We trimmed the reads before correcting with SOAPdenovo, which greatly improved its performance on our experiments with simulated data. The 'Trimmed only' column includes reads trimmed before and during SOAPdenovo correction. Quake trims reads automatically during correction. Correcting the reads reduces the number of contigs and scaffolds, increases the contig sizes, and allows the assembler to include more reads. Quake corrects more reads than SOAPdenovo which results in a slightly better assembly.

Now are you ready to do some assemblies?



Wait a minute, why there're so many assemblers?

[Assembler List](#)

Showing below up to **68** results starting with #1.

View (previous 250 | next 250) ([20](#) | [50](#) | [100](#) | [250](#) | [500](#))

A

- ABBA
- ABySS
- ALLPATHS
- AMOS
- ANCHOR
- Arachne

C

- CABOG
- CLCbio Genomics Workbench
- Contrail
- Cortex
- Cufflinks
- Curtain

D

- DNA Baser

E

- EDENA
- Est2assembly
- EULER

F

- FLASH
- Forge
- Fuzzypath

G

- GeeFu
- Geneious
- Genomatix Mining Station (GMS)
- GenoMiner

G cont.

- Gk arrays

I

- ICORN
- IDBA
- IOmics
- ISSAKE

L

- Lasergene
- LOCAS

M

- Mapsembler
- Mason
- Meraculous
- MetaSim
- MIRA
- MOSAIK
- MuSICA 2

N

- Newbler
- Ngs backbone
- NovelSeq

O

- Omixon Variant Toolkit

P

- Phred Phrap Consed Cross match
- PRICE

Q

- QSRA

R cont.

- RGA

S

- SCARF
- SeqCons
- SeqMan NGen
- Sequencher
- SGA
- SHARCGS
- SHORTY
- SHRAP
- SOAPdenovo
- SOPRA
- Spiral Genetics
- SR-ASM
- SSAKE
- STADEN
- Supersplat

T

- Taipan
- TASR
- Tracemblem

V

- VCAKE
- Velvet

Z

- ZORRO



Let them run!

Resource

Assemblathon 1: A competitive assessment of de novo short read assembly methods

Dent Earl,^{1,2} Keith Bradnam,³ John St. John,^{1,2} Aaron Darling,³ Dawei Lin,^{3,4} Joseph Fass,^{3,4} Hung On Ken Yu,³ Vince Buffalo,^{3,4} Daniel R. Zerbino,² Mark Diekhans,^{1,2} Ngan Nguyen,^{1,2} Pramila Nuwantha Ariyaratne,⁵ Wing-Kin Sung,^{5,6} Zemin Ning,⁷ Matthias Haimel,⁸ Jared T. Simpson,⁷ Nuno A. Fonseca,⁹ İnanç Birol,¹⁰ T. Roderick Docking,¹⁰ Isaac Y. Ho,¹¹ Daniel S. Rokhsar,^{11,12} Rayan Chikhi,^{13,14} Dominique Lavenier,^{13,14,15} Guillaume Chapuis,^{13,14} Delphine Naquin,^{14,15} Nicolas Maillet,^{14,15} Michael C. Schatz,¹⁶ David R. Kelley,¹⁷ Adam M. Phillippy,^{17,18} Sergey Koren,^{17,18} Shiao-Pyng Yang,¹⁹ Wei Wu,¹⁹ Wen-Chi Chou,²⁰ Anuj Srivastava,²⁰ Timothy I. Shaw,²⁰ J. Graham Ruby,^{21,23} Peter Skewes-Cox,^{21,22,23} Miguel Betegon,^{21,23} Michelle T. Dimon,^{21,23} Victor Solovyev,²⁴ Igor Seledtsov,²⁵ Petr Kosarev,²⁵ Denis Vorobyev,²⁵ Ricardo Ramirez-Gonzalez,²⁶ Richard Leggett,²⁷ Dan MacLean,²⁷ Fangfang Xia,²⁸ Ruibang Luo,²⁹ Zhenyu Li,²⁹ Yinlong Xie,²⁹ Binghang Liu,²⁹ Sante Gnerre,³⁰ Iain MacCallum,³⁰ Dariusz Przybylski,³⁰ Filipe J. Ribeiro,³⁰ Shuangye Yin,³⁰ Ted Sharpe,³⁰ Giles Hall,³⁰ Paul J. Kersey,⁸ Richard Durbin,⁷ Shaun D. Jackman,¹⁰ Jarrod A. Chapman,¹¹ Xiaojin Huang,³¹ Joseph L. DeRisi,^{21,23} Mario Caccamo,²⁶ Yingrui Li,²⁹ David B. Jaffe,³⁰ Richard E. Green,² David Haussler,^{1,2,23} Ian Korf,^{3,32} and Benedict Paten^{1,2,33}

¹⁻³²[Author affiliations appear at the end of the paper.]



Genome Assembly Gold-Standard Evaluations

Resource

GAGE: A critical evaluation of genome assemblies and assembly algorithms

Steven L. Salzberg,^{1,7} Adam M. Phillippy,² Aleksey Zimin,³ Daniela Puiu,¹ Tanja Magoc,¹ Sergey Koren,^{2,4} Todd J. Treangen,¹ Michael C. Schatz,⁵ Arthur L. Delcher,⁶ Michael Roberts,³ Guillaume Marçais,³ Mihai Pop,⁴ and James A. Yorke³

¹McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine, Baltimore, Maryland 21205, USA; ²National Biodefense Analysis and Countermeasures Center, Battelle National Biodefense Institute, Frederick, Maryland 21702, USA; ³Institute for Physical Sciences and Technology, University of Maryland, College Park, Maryland 20742, USA; ⁴Center for Bioinformatics and Computational Biology, University of Maryland, College Park, Maryland 20742, USA; ⁵Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, New York 11724, USA; ⁶Institute for Genome Sciences, University of Maryland School of Medicine, Baltimore, Maryland 21201, USA



Genome Assembly Gold-Standard Evaluations

[Main page](#)[Genome Assemblers](#)[Data sets](#)[Recipes](#)[Results](#)[Twitter](#)

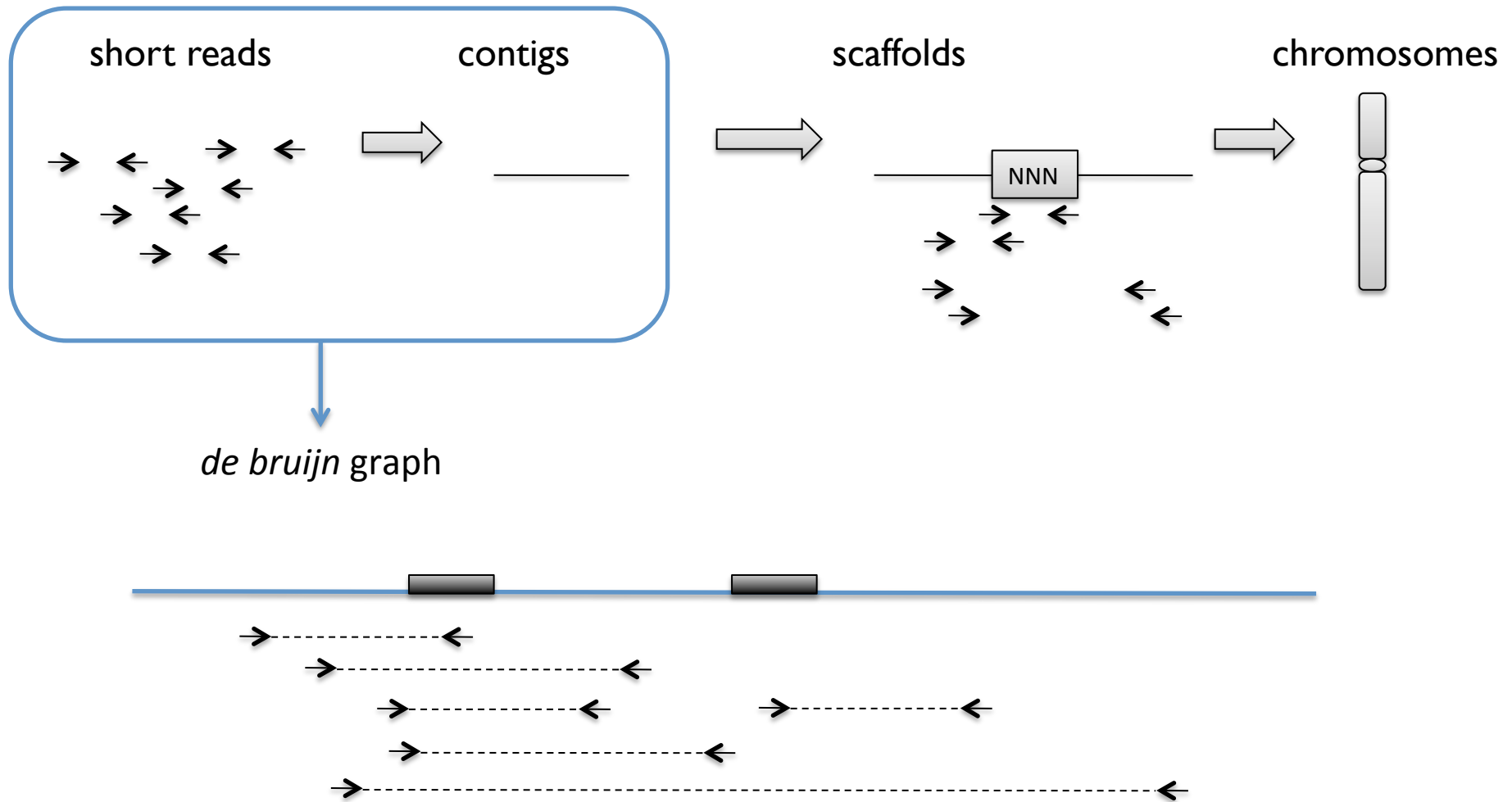
What recipes were used for each genome assembly?

Links to recipes for each genome assembler, which include instructions and all necessary scripts to reproduce the results:

1. [ABYSS recipes](#)
2. [ALLPATHS-LG recipes](#)
3. [Bambus2 recipes](#)
4. [Celera Assembler recipes](#)
5. [MSR-CA recipes](#)
6. [SGA recipes](#)
7. [SOAPdenovo recipes](#)
8. [Velvet recipes](#)

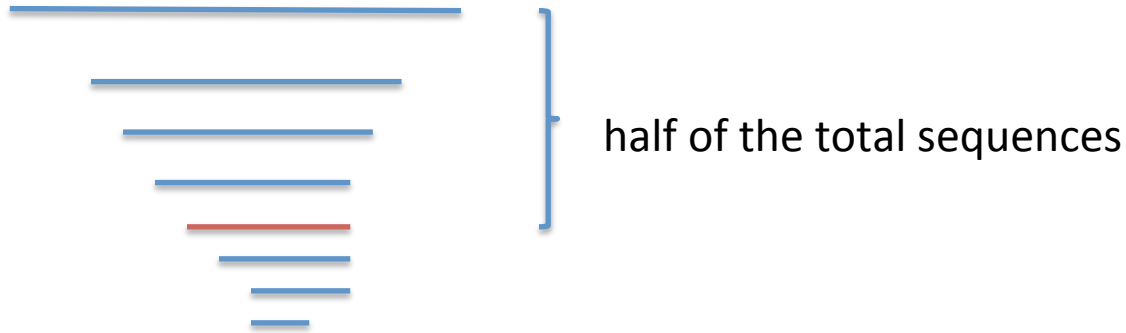
<http://gage.cbcb.umd.edu/recipes/index.html>

How does the assembler work?



Different insert-length would greatly help the scaffolding process

- What kind of data are you getting?
- Read quality and error correction
- Assembly
- How to assess the draft genome
- Mapping and chromosomal build
- SNP calling, RNA-seq analysis etc.



- Contigs: continuous sequences without any gaps
- Scaffolds: joined by contigs through mate-pair relationship
- Superscaffolds/chromosomes: joined by scaffolds by comparing to a reference genome or using other source of linkage information (physical map, FISH..etc.)
- N50 length: how continuous is the genome
the length N for which half of all bases in the sequences are in a sequence of length $L < N$.
- Gap content: integrity of the genome: gaps/kb, N%
- Accuracy: compare to a genome sequence produced by other platform or that of a related species

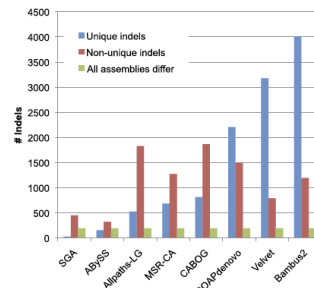
Table 3. Assemblies of *R. sphaeroides* (genome size 4,603,060)

Assembler	Contigs				Scaffolds			
	Num	N50 (kb)	Errors	N50 corr. (kb)	Num	N50 (kb)	Errors	N50 corr. (kb)
ABYSS	1915	5.9	55	4.2	1701	9	3	5
ALLPATHS-LG	204	42.5	43	34.4	34	3192	0	3192
Bambus2	177	93.2	368	12.8	92	2439	2	2419
CABOG	322	20.2	34	17.9	130	66	5	55
MSR-CA	395	22.1	42	19.1	43	2,976	5	2966
SGA	3067	4.5	8	2.9	2096	51	0	51
SOAPdenovo	204	131.7	414	14.3	166	660	3	658

Table 4. Assemblies of human chromosome 14 (ungapped size 88,289,540)

Assembler	Contigs				Scaffolds			
	Num	N50 (kb)	Errors	N50 corr. (kb)	Num	N50 (kb)	Errors	N50 corr. (kb)
ABYSS	51,924	2.0	691	2.0	51,301	2.1	9	2
ALLPATHS-LG	4529	36.5	2659	21.0	225	81,647	45	4702
Bambus2	13,592	5.9	8677	4.3	1792	324	143	161
CABOG	3361	45.3	3032	23.7	479	393	597	26
MSR-CA	30,103	4.9	4316	4.3	1425	893	1068	94
SGA	56,939	2.7	831	2.7	30,975	83	19	79
SOAPdenovo	22,689	14.7	5163	7.4	13,502	455	268	214
Velvet	45,564	2.3	4541	2.1	3,565	1190	9156	27

Columns are the same as in Table 2.

**Figure 5.** Comparison of insertion and deletion errors among all eight assemblers for human chromosome 14. (Blue) The indel errors >5 bp in length that are unique to each assembler. (Red bars) Indel errors made by at least one other assembler. (Green bars) Indels shared by all assemblers, which might represent true differences between the target genome and the reference.

- SOAPdenovo and ALLPATHS-Ig

SOAP	ALLPATHS-Ig
Quite flexible at the input data: paired-end, single-end, short insert, long insert...	Has to provide overlapping and long-insert library data
No requirement for sequencing coverage	~100 fold
Have separate modules for error correction, contiging, scaffolding, gapfilling	One button run: the program will take care of everything
At least 8CPUs, 5G memory for small genome, 150G for human genome	32Gb for small genome 512Gb for mammalian sized genome
Longer N50, but sometimes accuracy is not as good as ALLPATHS	Achieve both a good N50 and accuracy

```

SOAPdenovo127mer SOAPdenovo127mer.tal SOAPdenovo31mer SOAPdenovo31mer.tal SOAPdenovo63mer SOAPdenovo63mer.tal
ucbvnp-208-176:SOAP zhouqi$ ./SOAPdenovo31mer

Version 1.05: released on July 29th, 2010

Usage: SOAPdenovo <command> [option]
  pregraph      construction kmer-graph
  contig        eliminate errors and output contigs
  map           map reads to contigs
  scaff         scaffolding
  all           doing all the above in turn
ucbvnp-208-176:SOAP zhouqi$ ./SOAPdenovo31mer all

Version 1.05: released on July 29th, 2010

SOAPdenovo all -s configFile [-a initMemoryAssumption -K kmer -d KmerFreqCutOff -D EdgeCovCutoff -M mergeLevel -R -u -G gapLenDiff -L minContigLen -p n_cpu] -o Output
-s ShortSeqFile: The input file name of solexa reads
-a initMemoryAssumption: Initiate the memory assumption to avoid further reallocation
-K kmer(default 23): k value in kmer
-p n_cpu(default 8): number of cpu for use
-F (optional) fill gaps in scaffold
-M mergeLevel(default 1,min 0, max 3): the strength of merging similar sequences during contiging
-d KmerFreqCutoff(optional): delete kmers with frequency no larger than (default 0)
-D EdgeCovCutoff(optional): delete edges with coverage no largert than (default 1)
-R (optional): unsolve repeats by reads (default no)
-G gaplenDiff(default 50): allowed length difference between estimated and filled gap
-L minlen(default K+2): shortest contig for scaffolding
-u (optional): un-mask contigs with high coverage before scaffolding (default mask)
-o Output: prefix of output file name
ucbvnp-208-176:SOAP zhouqi$

```

Input:

- Configuration file
- Kmer size: experimental
- Cpu number

Output:

Scaffold sequences, contig sequences

Three different version of SOAPdenovo

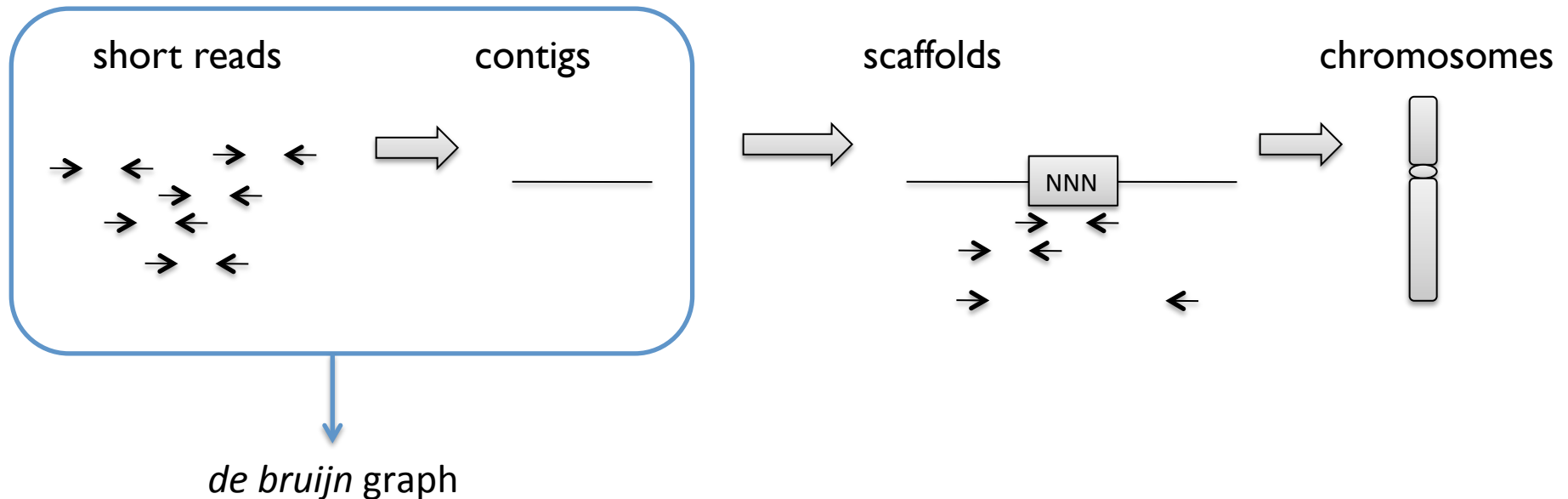
SOAPdenovo127mer, SOAPdenovo63mer, SOAPdenovo31mer

For different kmer range

An example of SOAP configuration file

```
1 #maximal read length
2 max_rd_len=100
3 [LIB]
4 #average insert size,which you'd know from the library prep
5 avg_ins=600
6 #if sequence needs to be reversed, reverse only for long-insert library
7 reverse_seq=0
8 #in which part(s) the reads are used, 0 means do nothing, 1 means do contiging, 2 means do scaffolding, 3 means do both
9 asm_flags=3
10 #in which order the reads are used while scaffolding
11 rank=1
12 #fastq file for read 1
13 q1=
14 #fastq file for read 2 always follows fastq file for read 1
15 q2=
16 #fasta file for read 1
17 #f1=/path/**LIBNAMEA**/fasta_read_1.fa
18 #fastq file for read 2 always follows fastq file for read 1
19 #f2=/path/**LIBNAMEA**/fasta_read_2.fa
20 #fastq file for single reads
21 #q=/path/**LIBNAMEA**/fastq_read_single.fq
22 #fasta file for single reads
23 #f=/path/**LIBNAMEA**/fasta_read_single.fa
24 #a single fasta file for paired reads
25 #p=/path/**LIBNAMEA**/pairs_in_one_file.fa
26 #[LIB]
27 #avg_ins=2000
28 #reverse_seq=1
29 #asm_flags=2
30 #rank=2
31 #q1=/path/**LIBNAMEB**/fastq_read_1.fq
32 #q2=/path/**LIBNAMEB**/fastq_read_2.fq
33 #q=/path/**LIBNAMEB**/fastq_read_single.fq
34 #f=/path/**LIBNAMEB**/fasta_read_single.fa
```

How does the assembler work?



Command line:

```
SOAPdenovo3 lmer all -K 31 -s test.cfg -p 32 -L 500 -o test > outlog 2> errorlog
```

(half an hour running time with 32 threads)

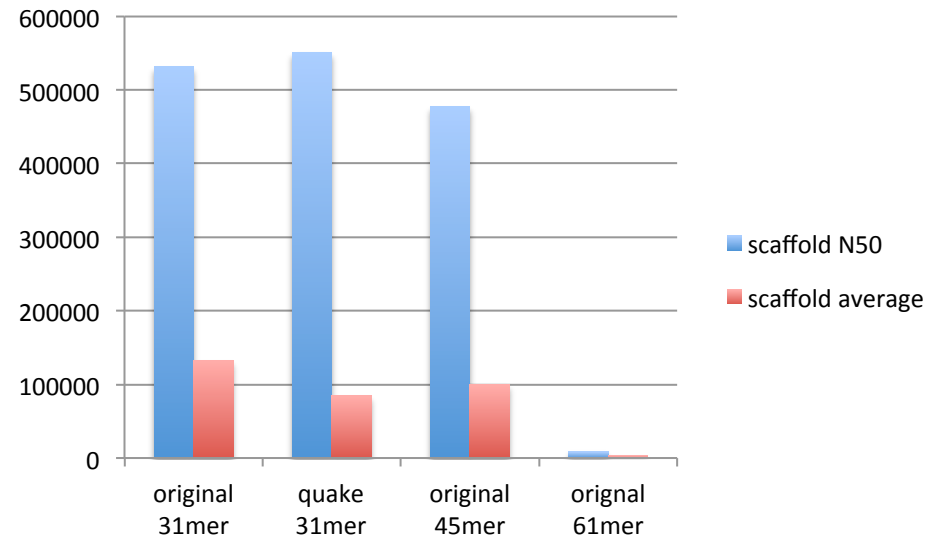
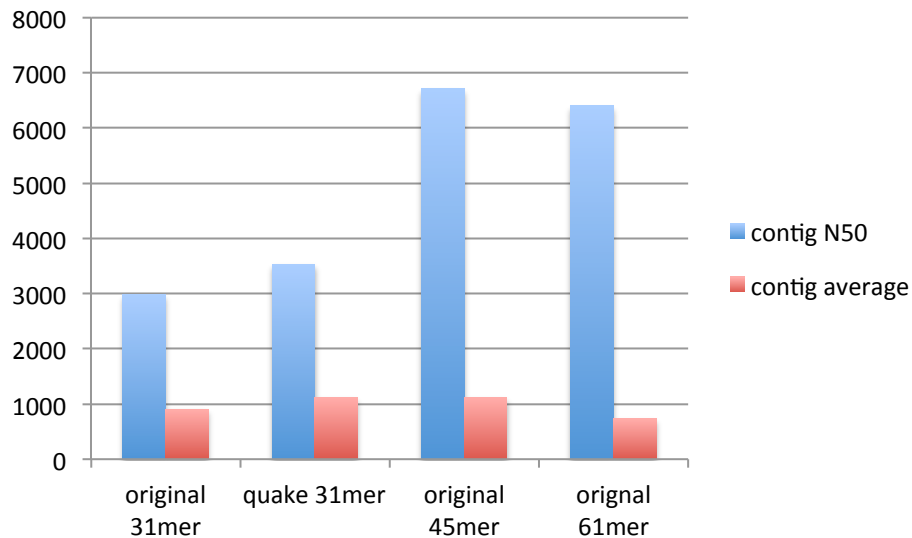
Let's see an example using 3G data of a *Drosophila* genome...

What to read from the log file

- Contig N50, average length, contig number, total size of the contigs
- Mapped reads %,
- Estimated map-pair distance: is it consistent with what you have inputted in the configuration file?
- Scaffold N50, average scaffold length, scaffold number, assembled size (genome size)

After Running SOAPdenovo..

- Adjust the *k*-mer size until getting a better N50
- Adjust the insert-length if not consistent



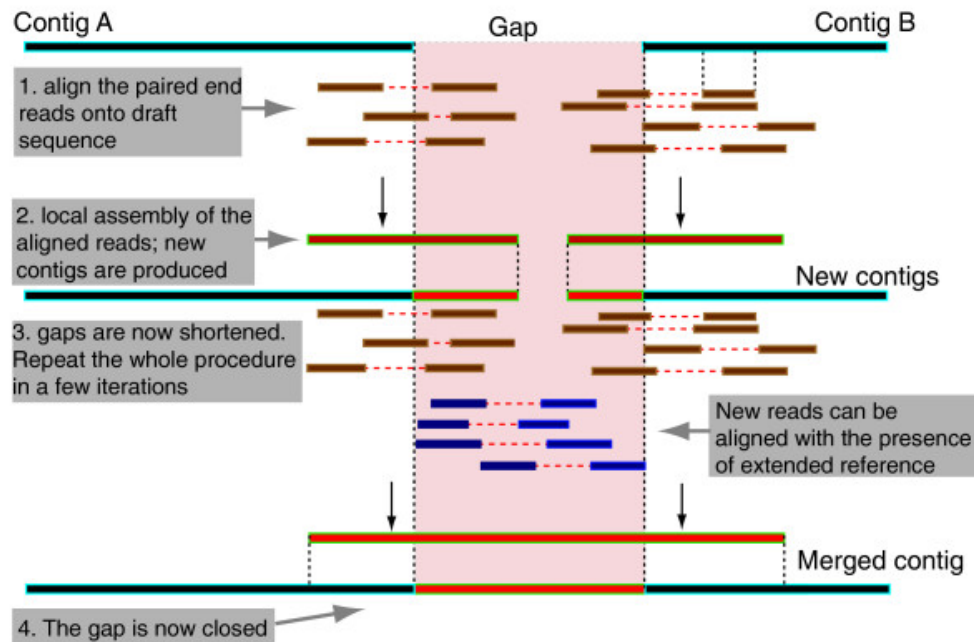
- Post-assembly gap-filling:

Popular Gap-filler: SOAP-gapcloser; IMAGE

Gapcloser -a [SOAP resulting scaffold sequences] -b [same configuration file you feed to SOAP] -o [name: Best Genome Ever] -p [kmer size] -t [thread number]

```
/jbods/data00/zhouqi/soft/GapCloser -a test.scafSeq -b test.cfg -o test.gapfilled  
-t 32 1>gapfil.out 2>gapfil.errout &
```

→example



Top
Abstract
Background
Results
Discussion
Materials and methods
Abbreviations
Authors' contributions
Acknowledgements
References

Method

Highly accessed Open Access

Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps

Isheng J Tsai*, Thomas D Otto and Matthew Berriman

* Corresponding author: Isheng J Tsai ijt@sanger.ac.uk

▼ Author Affiliations

Parasite Genomics, Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SA, UK
For all author emails, please [log on](#).

Genome Biology 2010, 11:R41 doi:10.1186/gb-2010-11-4-r41

The electronic version of this article is the complete one and can be found online at:
<http://genomebiology.com/2010/11/4/R41>

Received: 5 January 2010
Revisions received: 10 March 2010
Accepted: 13 April 2010
Published: 13 April 2010

© 2010 Tsai et al.; licensee BioMed Central Ltd.

This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Genome Biology
Volume 11
Issue 4

Viewing options

Abstract
Full text
PDF (1.3MB)
[Additional files](#)

Associated material

[PubMed record](#)
[About this article](#)
[Readers' comments](#)

Related literature

Cited by
[on Google blog](#)
[search](#)
Other articles by authors
► [on Google Scholar](#)
► [on PubMed](#)
Related articles/pages
[on Google](#)
[on Google Scholar](#)
[on PubMed](#)

Top
Abstract
Background
Results
Discussion
Materials and methods
Abbreviations
Competing interests
Authors' contributions
Acknowledgements
References

Software

Highly accessed Open Access

Toward almost closed genomes with GapFiller

Marten Boetzer and Walter Pirovano*

* Corresponding author: Walter Pirovano walter.pirovano@baseclear.com

▼ Author Affiliations

BaseClear BV, Einsteinweg 5, 2333 CC, Leiden, The Netherlands
For all author emails, please [log on](#).

Genome Biology 2012, 13:R56 doi:10.1186/gb-2012-13-6-r56

The electronic version of this article is the complete one and can be found online at:
<http://genomebiology.com/2012/13/6/R56>

Received: 10 April 2012
Revisions received: 11 May 2012
Accepted: 25 June 2012
Published: 25 June 2012

© 2012 Boetzer and Pirovano; licensee BioMed Central Ltd.

This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Genome Biology
Volume 13
Issue 6

Viewing options

Abstract
Full text
PDF (413KB)

Associated material

[PubMed record](#)
[About this article](#)
[Readers' comments](#)

Related literature

Cited by
[on Google blog](#)
[search](#)
Other articles by authors
► [on Google Scholar](#)
► [on PubMed](#)
Related articles/pages
[on Google](#)
[on Google Scholar](#)

Table 1 Gap closure results obtained on the bacterial datasets

Method	Original	IMAGE	SOAPdenovo	GapFiller	GapFiller-LC
<i>Escherichia coli</i>					
Genome size (bp)	4,478,287	4,530,961	4,490,973	4,490,638	
Scaffolds	179	179	179	179	
Gap count	544	291	16	11	
Total gap length (bp)	12,516	2,861	16	130	
Errors (SNPs)	12	40	33	22	
Errors (indels)	4	17	25	9	
Errors (misjoins)	1	1	1	1	
N50	50,557	50,558	50,558	50,558	
<i>Streptomyces coelicolor</i>					
Genome size (bp)	8,558,275	8,576,331	8,557,720	8,558,333	
Scaffolds	115	115	115	115	
Gap count	158	63	60	23	
Total gap length (bp)	9,221	4,009	1,288	806	
Errors (SNPs)	299	423	406	280	
Errors (indels)	664	677	769	686	
Errors (misjoins)	12	17	18	18	
N50	173,822	173,822	173,822	173,822	
<i>Staphylococcus aureus</i>					
Genome size (bp)	2,880,676		2,880,926	2,881,756	2,883,448
Scaffolds	19		19	19	19
Gap count	48		27	27	22
Total gap length (bp)	9,900		1,547	5,508	1,861
Errors (SNPs)	79		260	98	173
Errors (indels)	16		53	26	37
Errors (misjoins)	4		13	7	5
N50	1,091,731		1,091,333	1,092,281	1,092,421
<i>Rhodobacter sphaeroides</i>					
Genome size (bp)	4,609,785		4,609,466	4,609,596	4,610,796
Scaffolds	38		38	38	38
Gap count	170		163	161	139
Total gap length (bp)	21,409		14,166	20,667	17,625
Errors (SNPs)	218		410	230	300
Errors (indels)	187		294	190	199
Errors (misjoins)	6		10	6	7
N50	3,192,334		3,192,075	3,192,215	3,192,974

Gap closure results obtained on four bacterial datasets show that the GapFiller strategy yields the most accurate finished genomes. Also, the gap count is lower compared to the other methods. The IMAGE method significantly underperforms on all quality measures and would therefore not be the preferred method to use. Differences are smaller between GapFiller and SOAPdenovo. Interestingly, whereas the gap count after closure is generally less for GapFiller, SOAPdenovo yields in three cases a shorter total gap length. This suggests the latter method is able to close larger gaps. Strikingly, however, the amount of errors is significantly higher for SOAPdenovo regardless of the source (SNPs, indels and misjoins). Even when applying less strict settings for GapFiller (GapFiller-LC: minimum coverage $\alpha = 1$, ratio $r = 0.5$) to shorten the total gap length, our method still yields significantly less errors.

- What kind of data are you getting?
- Read quality and error correction
- Assembly
- How to assess the draft genome
- Mapping and chromosomal build
- SNP calling, RNA-seq analysis etc.

- Setup configuration file telling ALLPATHs to convert the reads format : fastq → fastb
- Runallpaths:

Log file:

Validating Fragment Libraries

Kmer Spectrum Analysis

Table 1: library names, number of pairs (N), original (L0) and new sizes (L)

Table 2: fraction of reads in each length interval

----- AllPathsReport -> assembly_stats.report

What we get from the assembler is just a “draft genome”

[illegible]

- Assess the assembly: PCR or compared to reference sequences
published EST/genome sanger sequences,
compared to a sister species' genome
- Link scaffolds into chromosomal sequences
- Gene annotation with RNA-seq data/proteins of related species
- More downstream analysis: find differential expression pattern, annotate regulatory sequences, identify copy number variation, population history analysis...

There're no existent chromosomal builder yet..

- Blast/blat your draft genome against a reference genome
- Pick up a best-aligned scaffold for each site
- Link them

A recipe of genome assembly

- Sequence the DNA to at least 30 fold coverage, preferably with different insert-lengths and long-insert libraries
- Assess the read-quality, trim them if necessary
- Error-correct the reads if there's sufficient coverage
- Assemble the genome with assembler program, test different parameter sets until getting a better N50, a lower gap content
- Gap-filling the draft genome
- Align the draft genome to other genomes to build chromosome
- Gene annotation, downstream analysis etc.

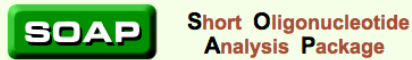
Resources



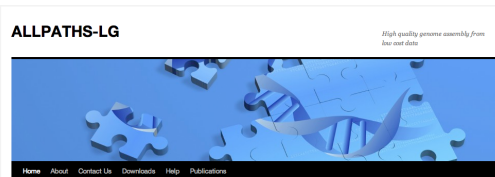
<http://gage.cbc.umd.edu>



<http://assemblathon.org>



<http://soap.genomics.org.cn> : aligner, transcriptome-assembler



<http://www.broadinstitute.org/software/allpaths-lg/blog/>



<http://seqanswers.com>

Thank you!

Bachtrog Lab

Functional and Evolutionary Genomics



- ☒ [Home](#)
 - ☐ [People](#)
 - ☐ [Research](#)
 - ☐ [Publications](#)
 - ☐ [Data & Software](#)
 - ☐ [Pictures](#)
 - ☐ [Events](#)
-

University of California, Berkeley
[Department of Integrative Biology](#)
[Center for Theoretical Evolutionary Genomics](#)
3060 VLSB
Berkeley, CA 94720-3140
office # 4087
phone (lab): (510)-642-9271