GENOME ASSEMBLY

CHUMPOL NGAMPHIW, PH.D.

NATIONAL CENTER FOR GENETICS ENGINEERING AND BIOTECHNOLOGY (BIOTEC)
NATIONAL SCIENCE AND TECHNOLOGY DEVELOPMENT AGENCY (NSTDA)

"GENOME ASSEMBLY AND ANNOTATION" : AUGUST 6-9, 2018 @ KUPS

OUTLINE

- How to start genome assembly ?
- Different types of genome assembly
- Assembly algorithms
- Assembler software

SO YOU WANT TO START A DE NOVO GENOME ASSEMBLY PROJECT

Assuming you have a good reason to sequence and assemble a genome.

1. What is the size of the genome?
2. What will be your sequencing "recipe"?
3. Do you have the computational resources?
   –i.e. a machine with 32 processors, 512GB RAM
4. Do you have the time? Personnel? Bioinformatics experience?

Marc Tollis, Ph.D. : *De Novo* Genome Assembly Using Next Generation Sequence Data, 2016

GLOSSARY

Assembly : Computational reconstruction of a longer sequence from smaller sequence reads

***De novo* Assembly** : Refers to the reconstruction of contiguous sequences without making use of any reference sequence

Contig : A contiguous linear stretch of DNA or RNA consensus sequence. Constructed from a number of smaller, partially overlapping, sequence fragments (reads)

Scaffold : Two or more contigs joined together using read-pair information

REVIEWS AND SYNTHESIS

A field guide to whole-genome sequencing, assembly and annotation

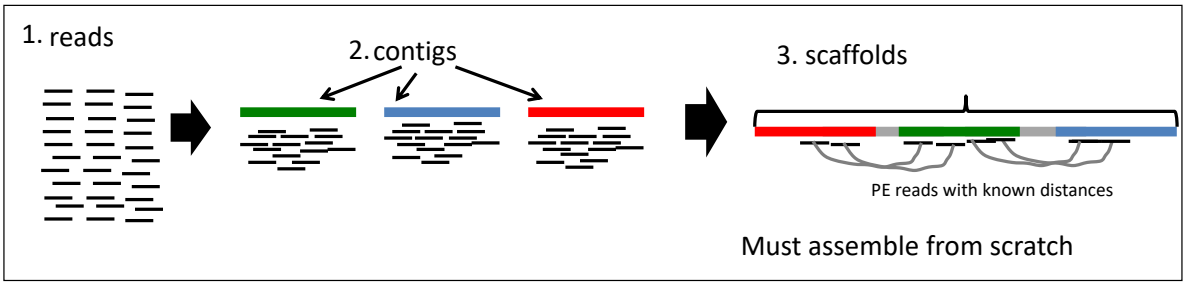
Robert Ekblom and Jochen B. W. Wolf

Department of Evolutionary Biology, Uppsala University, Uppsala, Sweden

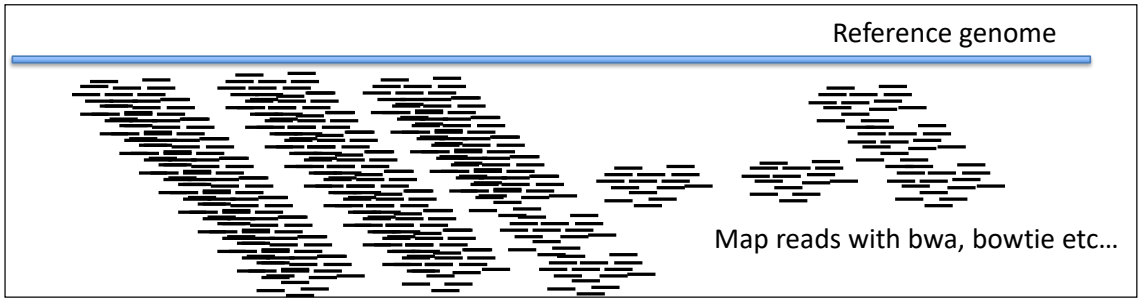
DE NOVO SHORT READ ASSEMBLY VS. SHORT READ MAPPING ASSEMBLY

In sequence assembly, two different types can be distinguished:

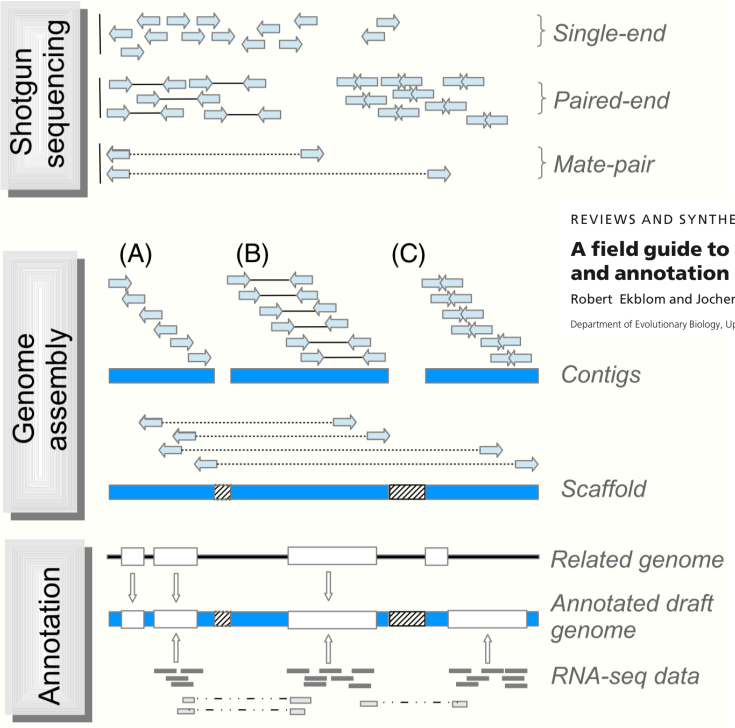
***de novo* assembly**



Reference-based assembly



ASSEMBLY PROCESS



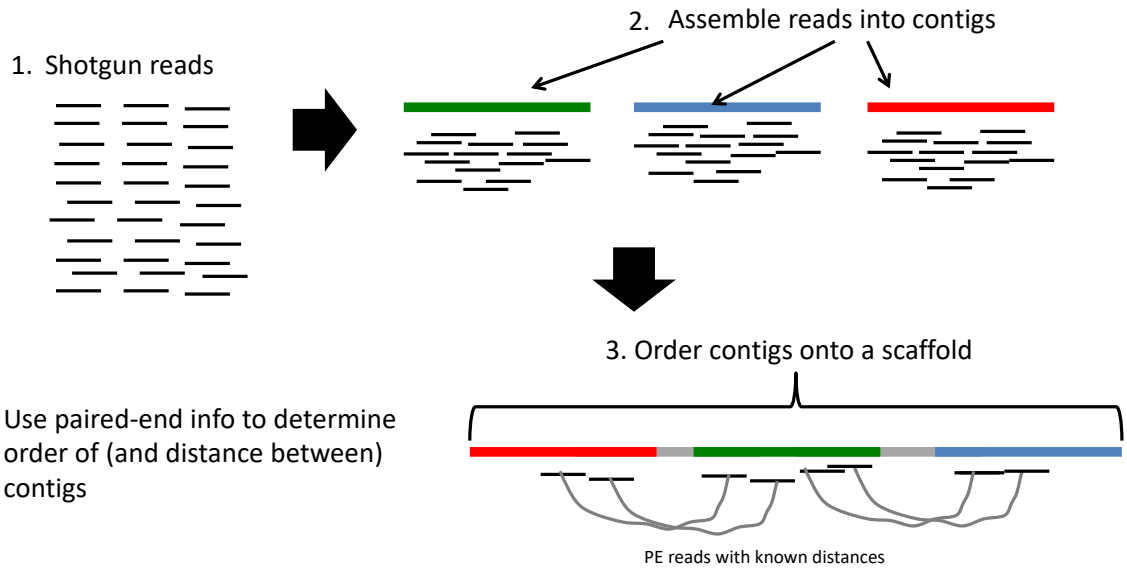
REVIEWS AND SYNTHESIS

A field guide to whole-genome sequencing, assembly and annotation

Robert Ekblom and Jochen B. W. Wolf

Department of Evolutionary Biology, Uppsala University, Uppsala, Sweden

DE NOVO ASSEMBLY BASIC



Use paired-end info to determine order of (and distance between) contigs

4. (optional) gaps between contigs are filled in by mapping reads back to the scaffolds

ASSEMBLY ALGORITHMS

- Overlap-Layout-Consensus (OLC)
- Eulerian / de Bruijn Graph (DBG)

"K-MER" CONCEPT

- A **k-mer** is a sub-string of length **k**
- A string of length L has (L - k + 1) **k-mers**
- Sequencing reads must be sub-sampling into **k-mers**
- Example read L=8 has 5 **k-mers** (8 - 4 + 1) when **k**=4
  - **AGATCTGA**
  - AGAT
  - GATC
  - ATCT
  - TCTG
  - CTGA

Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

OVERLAP - LAYOUT - CONSENSUS (OLC)

- **Overlap**
  - All against all pair-wise comparison
  - Build graph : nodes=reads, edges=overlaps
- **Layout**
  - Analyse/simplify/clean the overlap graph
  - Determine Hamiltonian path (NP-hard)
- **Consensus**
  - Align reads along assembly path
  - Call bases using weighted voting

Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

OLC : OVERLAP EXAMPLE

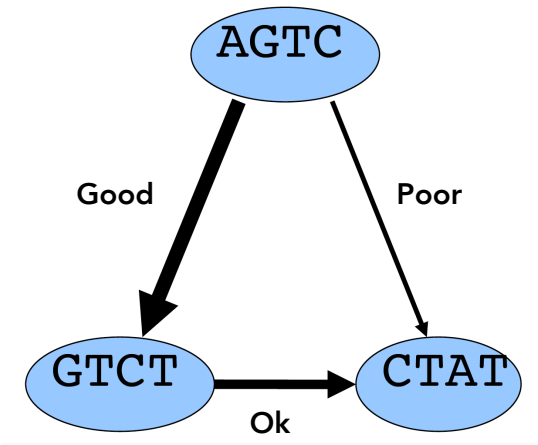
- True sequence (7 bp)
  - AGTCTAT
- Reads (3 x 4 bp)
  - AGTC, GTCT, CTAT
- Pairs to align (3)
  - AGTC + GTCT, AGTC + CTAT, GTCT + CTAT
- Best overlaps

• `AGTC-`	• `AGTC---`	• `GTCT--`
• `-GTCT`	• `---CTAT`	• `--CTAT`
• `(Good)`	• `(poor)`	• `(ok)`

Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

OLC : OVERLAP GRAPH

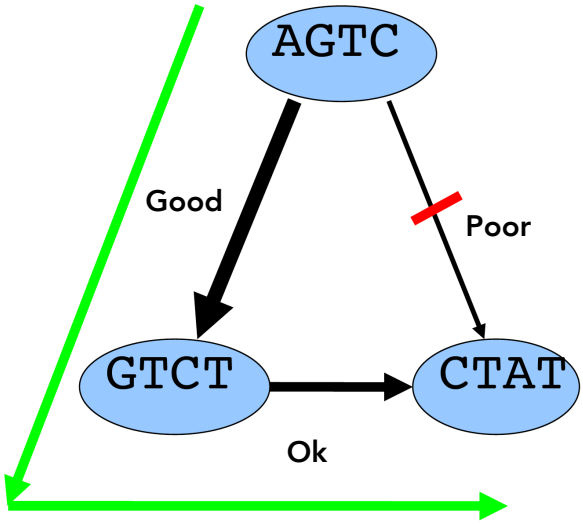
- Nodes are the 3 read sequences
- Edges are the overlap alignment with orientation
- Edge thickness represents score of overlap



Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

OLC : LAYOUT - CONSENSUS

- Optimal path shown in green
- Un-traversed weak overlap in red
- Consensus is read by outputting the overlapped nodes along the path
- aGTCTCTat



Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

OLC : SOFTWARE

- Pharap, CAP3, PCAP
  - Smaller scale assemblers
- Celera Assembler
  - Sanger-era assembler for large genomes
- Arachne, Edena, CABOG, Mira
  - Modern Sanger/hybrid assemblers
- Newbler (gsAssembler)
  - Used for 454 NGS "long" reads
  - Can be used for IonTorrent flowgrams too

Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

EULERIAN APPROACH

- Break all reads (length L) into (L - **k** + 1) **k-mers**
  - L = 50, k = 31 gives 20 **k-mers** per read
- Construct a *de Bruijn* graph (DBG)
  - Nodes = one for each unique k-mer
  - Edges = k-1 exact overlap between two nodes
- Graph simplification
  - Merge chains, remove bubbles and tips
- Find a Eulerian path through the graph
  - Linear time algorithm, unlike Hamiltonian

Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

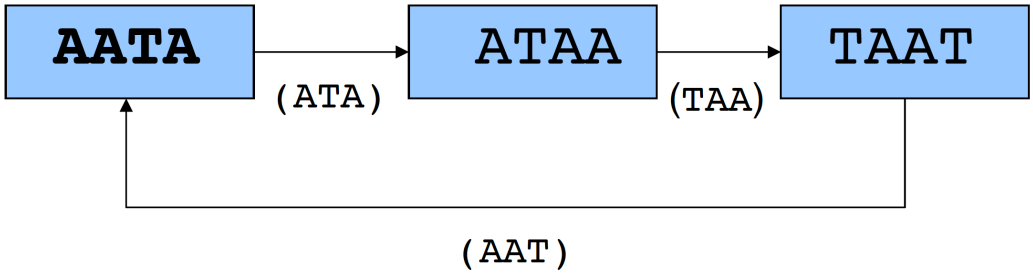
DBG : SIMPLE

- Sequence (6 bp)
  - AACCGG
- k-mers (k=4)
  - AACC ACCG CCGG
- Graph

Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

DBG : REPEATED K-MER

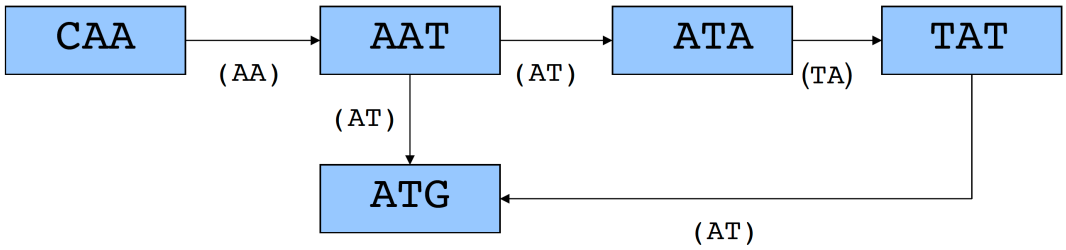
- Sequence (7 bp)
  - AATAATA
- k-mers (k=4)
  - AATA ATAA TAAT AATA (repeat)
- Graph



Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

DBG : ALTERNATE PATHS

- Sequence (7 bp)
  - CAATATG
- k-mers (k=3)
  - CAA AAT ATA TAT ATG
- Graph



Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

DBG : VARIATION IN A DE BRUIJN GRAPH

- Variation in sequence produces a bubble in a de Bruijn graph
- Sequences
  - AATCGACAGCCGG
  - AATCGATAGCCGG

Modified form "Genome Assembly Using de Bruijn graph" pdf by Biostatistics 666

DBG : SOFTWARE

- Velvet
  - Fast, relatively easy to use, multi-threaded
- AllPaths-LG
  - Designed for larger genomes, robust
- AbySS
  - Runs on cluster to get around RAM issues, integrates well with cluster job schedulers
- Ray
  - Designed for MPI/SMP clusters

Modified form "*De Novo* Genome Assembly of NGS data" pdf by Torsten Seeman

BRIEFINGS IN FUNCTIONAL GENOMICS. VOL 11. NO 1. 25–37 doi:10.1093/bfgp/elr035

Comparison of the two major classes of assembly algorithms: overlap–layout–consensus and de-bruijn-graph

Zhenyu Li, Yanxiang Chen*, Desheng Mu*, Jianying Yuan, Yujian Shi, Hao Zhang, Jun Gan, Nan Li, Xuesong Hu, Binghang Liu, Bicheng Yang and Wei Fan*

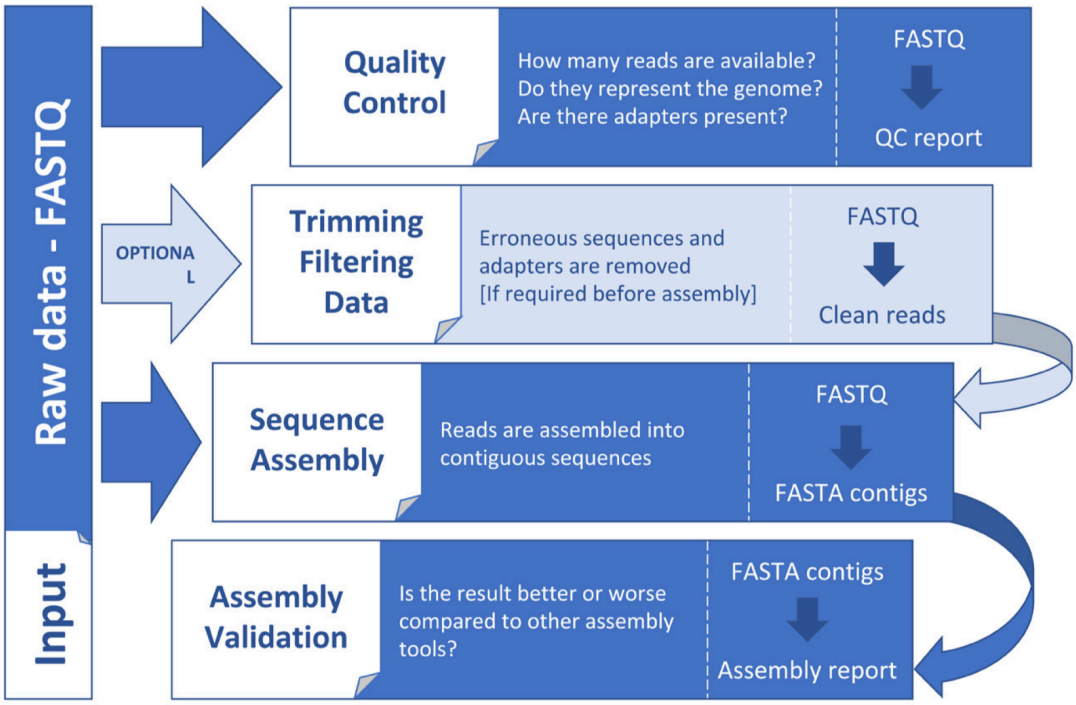
Advance Access publication date 19 December 2011

OLC (Overlap-layout-consensus) algorithm is more suitable for the low-coverage long reads, whereas the DBG (De-Bruijn-Graph) algorithm is more suitable for high-coverage short reads and especially for large genome assembly

Key Points

- High-quality genome sequences for many species are still strongly desired by the genomics community. With the rapid development of sequencing technologies and assembly algorithms, we have seen practical improvements and a bright future lies ahead.
- There are two major types of assembly algorithms: OLC and DBG; both of them are in accordance with Lander–Waterman model, but suit the assembly of different read lengths and sequencing depths, and have significant differences in computational efficiency.
- How well a genome can be assembled depends not only on sequencing technologies such as read length and sequencing error rate, but also on the characteristics of the genome, including repeat and the heterozygosity rate of the sequenced sample.

GENERAL STEPS IN A GENOME ASSEMBLY WORKFLOW



DAV Dominguez, et al., "Ten steps to get started in Genome Assembly and Annotation", F1000Res. 2018 Feb 5;7

ASSEMBLER SOFTWARE

Name	Type	Technologies	Author	Presented / Last updated	Licence*
AFEAP cloning Lasergene Genomics Suite	a precise and efficient method for large DNA sequence assembly	two rounds of PCRs followed by ligation of the sticky ends of DNA fragments	AFEAP cloning	2017 / 2018	C
DNASTAR Lasergene Genomics Suite	(large) genomes, exomes, transcriptomes, metagenomes, ESTs	Illumina, ABI SOLiD, Roche 454, Ion Torrent, Solexa, Sanger	DNASTAR	2007 / 2016	C
Newbler	genomes, ESTs	454, Sanger	454/Roche	2004/2012	C
Phrap	genomes	Sanger, 454, Solexa	Green, P.	1994 / 2008	C / NC-A
SPAdes	(small) genomes, single-cell	Illumina, Solexa, Sanger, 454, Ion Torrent, PacBio, Oxford Nanopore	Bankevich, A et al.	2012 / 2017	OS
Velvet	(small) genomes	Sanger, 454, Solexa, SOLiD	Zerbino, D. et al.	2007 / 2011	OS
HGAP	Small genomes	PacBio reads	Chin et al.[6]	2011 / 2015	OS
Falcon	Diploid genomes	PacBio reads	Chin et al.[7]	2014 / 2017	OS
Canu	Small and large, haploid/diploid genomes	PacBio/Oxford Nanopore reads	Koren et al.[8]	2001 / 2018	OS
MaSuRCA	Any size, haploid/diploid genomes	Illumina and PacBio/Oxford Nanopore data, legacy 454 and Sanger data	Zimin A, et al	2011 / 2018	OS
Hinge	Small microbial genomes	PacBio/Oxford Nanopore reads	Kamath et al.[9]	2016 / 2018	OS

***Licences:** OS = Open Source; C = Commercial; C / NC-A = Commercial but free for non-commercial and academics

https://en.wikipedia.org/wiki/Sequence_assembly

VELVET: USING DE BRUIJN GRAPHS FOR DENOVO SHORT READ ASSEMBLY

Resource

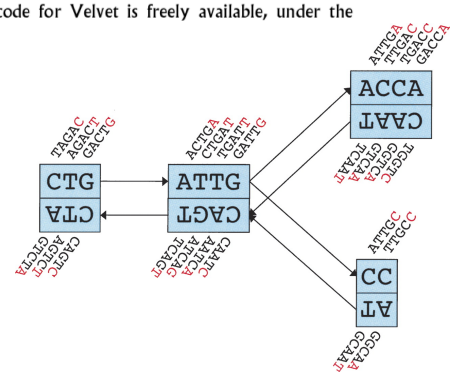
Velvet: Algorithms for de novo short read assembly using de Bruijn graphs

Daniel R. Zerbino and Ewan Birney[1]

EMBL-European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom

We have developed a new set of algorithms, collectively called "Velvet," to manipulate de Bruijn graphs for genomic sequence assembly. A de Bruijn graph is a compact representation based on short words (*k*-mers) that is ideal for high coverage, very short read (25–50 bp) data sets. Applying Velvet to very short reads and paired-ends information only, one can produce contigs of significant length, up to 50-kb N50 length in simulations of prokaryotic data and 3-kb N50 on simulated mammalian BACs. When applied to real Solexa data sets without read pairs, Velvet generated contigs of ~8 kb in a prokaryote and 2 kb in a mammalian BAC, in close agreement with our simulated results without read-pair information. Velvet represents a new approach to assembly that can leverage very short reads in combination with read pairs to produce useful assemblies.

[Supplemental material is available online at www.genome.org. The code for Velvet is freely available, under the GNU Public License, at http://www.ebi.ac.uk/~zerbino/velvet.]



***Velvet needs about 20-25x coverage and paired reads

VELVETOPTIMISER

Victorian Bioinformatics Consortium

VelvetOptimiser is a multi-threaded Perl script for automatically optimising the three primary parameter options (K, -exp_cov, -cov_cutoff) for the Velvet *de novo* sequence assembler.

- http://www.vicbioinformatics.com/software.velvetoptimiser.shtml
- Dependencies
  - Velvet => 1.1
  - Perl => 5.8.8
  - BioPerl => 1.4
  - GNU utilities : grep sed free cut

SPADES

Journal of Computational Biology, Vol. 19, No. 5 | Original Articles

SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing

Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A. Gurevich, Mikhail Dvorkin, Alexander S. Kulikov, Valery M. Lesin, Sergey I. Nikolenko, Son Pham, Andrey D. Prjibelski, Alexey V. Pyshkin, Alexander V. Sirotkin, Nikolay Vyahhi, Glenn Tesler, Max A. Alekseyev, and Pavel A. Pevzner

Published Online: 7 May 2012 | https://doi.org/10.1089/cmb.2012.0021

View Article Tools Share

Abstract

The lion's share of bacteria in various environments cannot be cloned in the laboratory and thus cannot be sequenced using existing technologies. A major goal of single-cell genomics is to complement gene-centric metagenomic data with whole-genome assemblies of uncultivated organisms. Assembly of single-cell data is challenging because of highly non-uniform read coverage as well as elevated levels of sequencing errors and chimeric reads. We describe SPAdes, a new assembler for both single-cell and standard (multicell) assembly, and demonstrate that it improves on the recently released E+V-SC assembler (specialized for single-cell data) and on popular assemblers Velvet and SoapDeNovo (for multicell data). SPAdes generates single-cell assemblies, providing information about genomes of uncultivatable bacteria that vastly exceeds what may be obtained via traditional metagenomics studies. SPAdes is available online (http://bioinf.spbau.ru/spades). It is distributed as open source software.

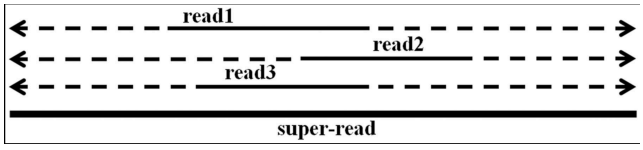
SPADES

Table 1. Assemblies of *B. cereus* (download **contigs**, **scaffolds**)

		ABySS	CABOG	MaSuRCA	MIRA	SGA	SOAPdenovo	SPAdes 3.0	Velvet
Contigs	Num	115	**78**	90	153	3335	105	53	404
	N50 (kb)	130.6	155.4	246.7	116.5	25.5	246.3	**286.8**	24.5
	Errors	2	5	9	9	17	**0**	1	3
	Errors-L	25	**6**	11	14	9	20	10	11
	N50Corr (kb)	130.6	150.5	246.7	100.0	25.5	246.3	**286.8**	24.5
	GenFrac (%)	98.6	**99.3**	99.2	99.2	98.9	98.3	98.8	97.8
	Unaligned	1	**0**	**0**	4	4	1	1	1
	Duplication	**1.0**	**1.0**	**1.0**	**1.0**	1.1	**1.0**	**1.0**	**1.0**
Scaffolds	Num	74	**33**	61	n/a	341	56	41	78
	N50 (kb)	135.6	431.5	337.9	n/a	25.5	456.6	**775.7**	247.7
	Errors	3	9	12	n/a	1	**0**	2	11
	Errors-L	29	13	13	n/a	**1**	39	11	258
	N50Corr (kb)	135.3	364.2	337.9	n/a	25.5	**456.0**	286.8	208.4
	GenFrac (%)	98.4	**99.3**	99.2	n/a	97.6	98.3	98.7	97.7
	Unaligned	**0**	**0**	**0**	n/a	**0**	1	**0**	1
	Duplication	**1.0**	**1.0**	**1.0**	n/a	**1.0**	**1.0**	**1.0**	**1.0**

MaSuRCA

- New hybrid approach
  - de Bruijn graph + Overlap-based assembly
- Transform large numbers of paired-end reads into a much smaller number of longer 'super-reads'
- Assemble combinations of illumine reads together with longer reads from 454 and Sanger sequencing technology
- Maryland Super-Read Celera Assembler



The MaSuRCA genome assembler FREE

Aleksey V. Zimin, Guillaume Marçais, Daniela Puiu, Michael Roberts, Steven L. Salzberg, James A. Yorke

Bioinformatics, Volume 29, Issue 21, 1 November 2013, Pages 2669–2677,
https://doi.org/10.1093/bioinformatics/btt476
Published: 29 August 2013 **Article history** ▾

Split View PDF Cite Permissions Share ▾

Abstract

Motivation: Second-generation sequencing technologies produce high coverage of the genome by short reads at a low cost, which has prompted development of new assembly methods. In particular, multiple algorithms based on de Bruijn graphs have been shown to be effective for the assembly problem. In this article, we describe a new hybrid approach that has the computational efficiency of de Bruijn graph methods and the flexibility of overlap-based assembly strategies, and which allows variable read lengths while tolerating a significant level of sequencing error. Our method transforms large numbers of paired-end reads into a much smaller number of longer 'super-reads'. The use of super-reads allows us to assemble combinations of Illumina reads of differing lengths together with longer reads from 454 and Sanger sequencing technologies, making it one of the few assemblers capable of handling such mixtures. We call our system the Maryland Super-Read Celera Assembler (abbreviated MaSuRCA and pronounced 'mazurka').

Results: We evaluate the performance of MaSuRCA against two of the most widely used assemblers for Illumina data, Allpaths-LG and SOAPdenovo2, on two datasets from organisms for which high-quality assemblies are available: the bacterium *Rhodobacter sphaeroides* and chromosome 16 of the mouse genome. We show that MaSuRCA performs on par or better than Allpaths-LG and significantly better than SOAPdenovo on these data, when evaluated against the finished sequence. We then show that MaSuRCA can significantly improve its assemblies when the original data are augmented with long reads.

A POST-ASSEMBLY GENOME-IMPROVEMENT TOOLKIT (PAGIT)

PROTOCOL

A post-assembly genome-improvement toolkit (PAGIT) to obtain annotated genomes from contigs

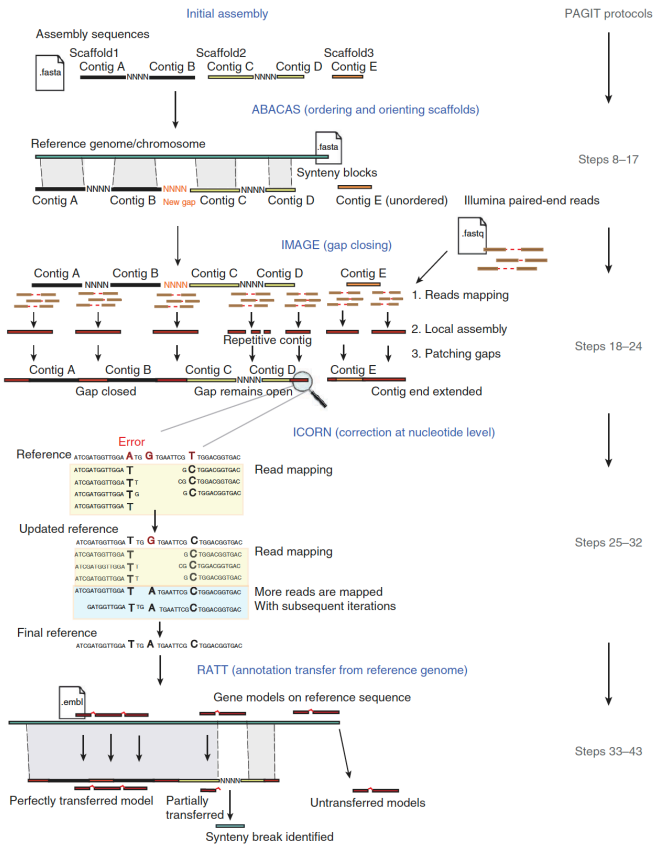
Martin T Swain[1,2], Isheng J Tsai[1], Samual A Assefa[1], Chris Newbold[1,3], Matthew Berriman[1] & Thomas D Otto[1]

[1]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Cambridge, UK. [2]Institute of Biological, Environmental and Rural Sciences, Aberystwyth University, Penglais Campus, Aberystwyth, UK. [3]Weatherall Institute of Molecular Medicine, University of Oxford, John Radcliffe Hospital, Oxford, UK. Correspondence should be addressed to T.D.O. (tdo@sanger.ac.uk).

Published online 7 June 2012; doi:10.1038/nprot.2012.068

Genome projects now produce draft assemblies within weeks owing to advanced high-throughput sequencing technologies. For milestone projects such as *Escherichia coli* or *Homo sapiens*, teams of scientists were employed to manually curate and finish these genomes to a high standard. Nowadays, this is not feasible for most projects, and the quality of genomes is generally of a much lower standard. This protocol describes software (PAGIT) that is used to improve the quality of draft genomes. It offers flexible functionality to close gaps in scaffolds, correct base errors in the consensus sequence and exploit reference genomes (if available) in order to improve scaffolding and generating annotations. The protocol is most accessible for bacterial and small eukaryotic genomes (up to 300 Mb), such as pathogenic bacteria, malaria and parasitic worms. Applying PAGIT to an *E. coli* assembly takes ~24 h: it doubles the average contig size and annotates over 4,300 gene models.

PAGIT WORKFLOW



CANU

- To specialize in assembling PacBio or Oxford Nanopore sequences

Method

Canu: scalable and accurate long-read assembly via adaptive *k*-mer weighting and repeat separation

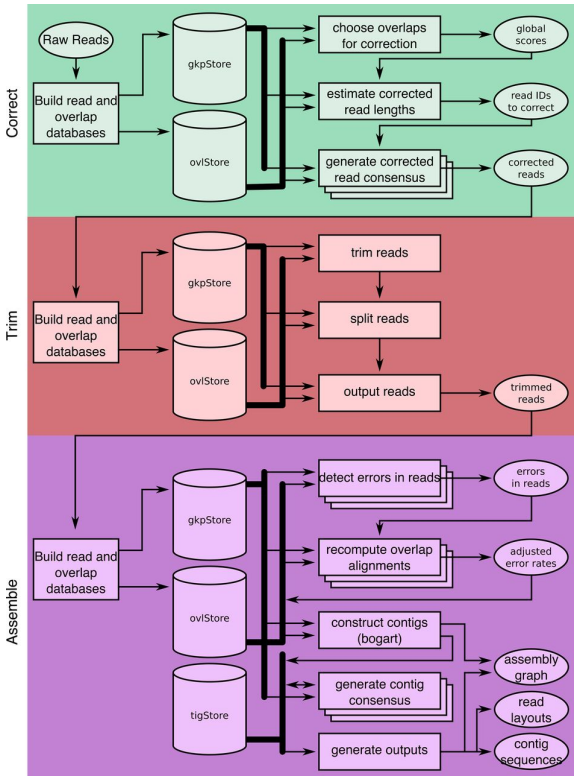
Sergey Koren,[1,5] Brian P. Walenz,[1,5] Konstantin Berlin,[2] Jason R. Miller,[3] Nicholas H. Bergman,[4] and Adam M. Phillippy[1]

[1]*Genome Informatics Section, Computational and Statistical Genomics Branch, National Human Genome Research Institute, National Institutes of Health, Bethesda, Maryland 20892, USA;* [2]*Invincea Incorporated, Fairfax, Virginia 22030, USA;* [3]*J. Craig Venter Institute, Rockville, Maryland 20850, USA;* [4]*National Biodefense Analysis and Countermeasures Center, Frederick, Maryland 21702, USA*

Long-read single-molecule sequencing has revolutionized de novo genome assembly and enabled the automated reconstruction of reference-quality genomes. However, given the relatively high error rates of such technologies, efficient and accurate assembly of large repeats and closely related haplotypes remains challenging. We address these issues with Canu, a successor of Celera Assembler that is specifically designed for noisy single-molecule sequences. Canu introduces support for nanopore sequencing, halves depth-of-coverage requirements, and improves assembly continuity while simultaneously reducing runtime by an order of magnitude on large genomes versus Celera Assembler 8.2. These advances result from new overlapping and assembly algorithms, including an adaptive overlapping strategy based on *tf–idf* weighted MinHash and a sparse assembly graph construction that avoids collapsing diverged repeats and haplotypes. We demonstrate that Canu can reliably assemble complete microbial genomes and near-complete eukaryotic chromosomes using either Pacific Biosciences (PacBio) or Oxford Nanopore technologies and achieves a contig NG50 of >21 Mbp on both human and *Drosophila melanogaster* PacBio data sets. For assembly structures that cannot be linearly represented, Canu provides graph-based assembly outputs in graphical fragment assembly (GFA) format for analysis or integration with complementary phasing and scaffolding techniques. The combination of such highly resolved assembly graphs with long-range scaffolding information promises the complete and automated assembly of complex genomes.

30x-60x coverage is the recommended minimum

CANU WORKFLOW (CONT)



- Improve the accuracy of bases in reads
- Trim reads to the portion with high-quality sequence
- Order the reads into contigs, generate consensus sequences and create graphs of alternate paths

BEST ASSEMBLY ADVICE

- Remember : your goal is to have a genome assembly
- Require more than one assembler
- In the end you will have many assemblies to choose from
- Use a lot of assembly tools for a lot of k values
  - Large k can better resolve repeats
  - Comes at coverage cost
  - The whole process should take a few months

Marc Tollis, Ph.D. : *De Novo* Genome Assembly Using Next Generation Sequence Data, 2016

Q & A