Genome Assembly: Background and Strategy

Monday, February 8, 2016

BIOL 7210: Genome Assembly Group

Aroon Chande, Cheng Chen, Alicia Francis, Alli Gombolay, Namrata Kalsi, Ellie Kim, Tyrone Lee, Wilson Martin, Tannishtha Som, Peijue Zhang

Outline

1. Illumina Sequencing

2. Genome Pipeline
   a. Quality Control and Pre-processing
   b. Genome Pipeline Steps/Description
   c. Read assembly:
      - *de novo*, reference-guided, hybrid
   d. Assembly improvement
   e. Assembly quality assessment

Objectives

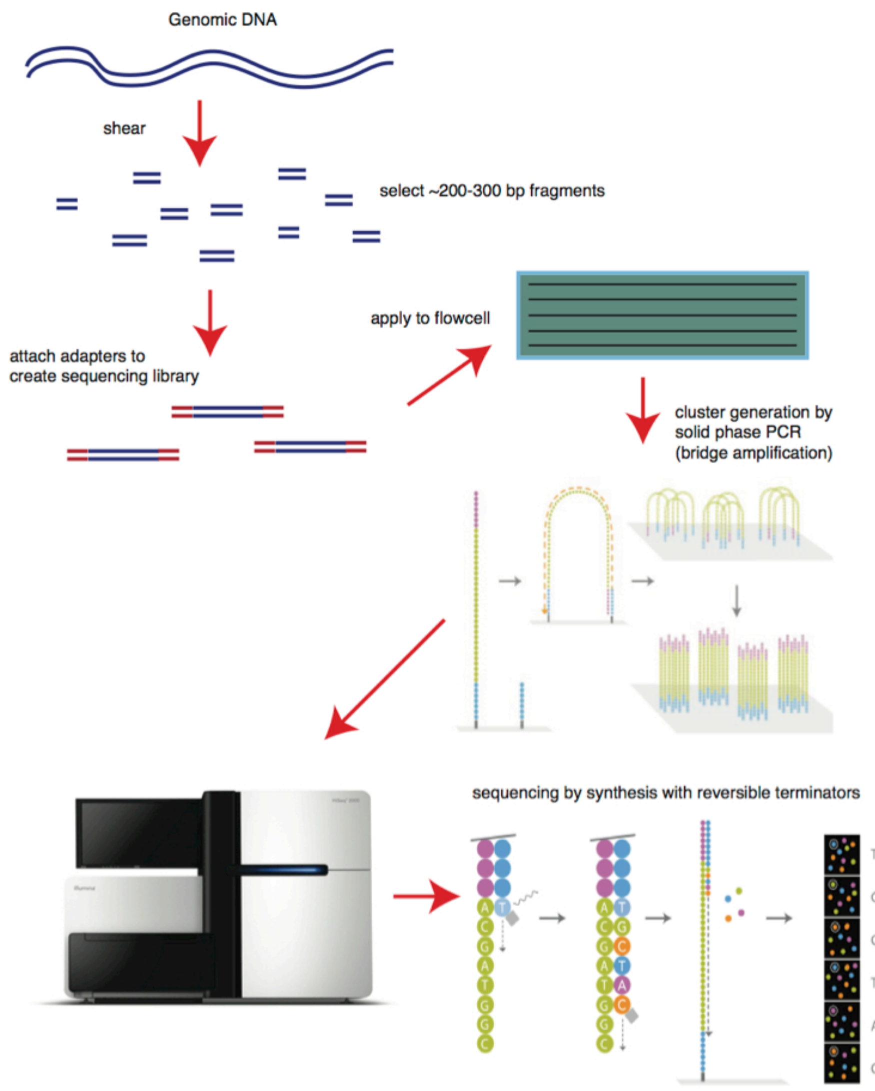
1. Research classical and new tools to use
2. Evaluate and compare these available tools
3. Choose and combine best assemblers to utilize
4. Create wrapper for pipeline to increase efficiency

Illumina Sequencing

Illumina Sequencing

- "Sequence by synthesis"
    - Sample DNA is sheared and ligated to primers
    - Bridge amplification
    - Sequencing reaction

- Each insert has a unique tag, barcode
    - Barcode is unique for each DNA sample in run
    - Barcode read step of synthesis reaction

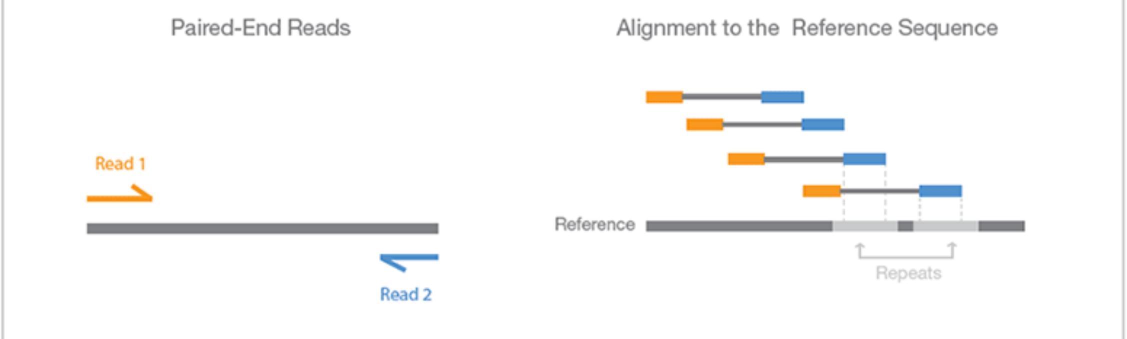

Genomic DNA
shear
select ~200-300 bp fragments
attach adapters to create sequencing library
apply to flowcell
cluster generation by solid phase PCR (bridge amplification)
sequencing by synthesis with reversible terminators

http://bitesizebio.com/13546/sequencing-by-synthesis-explaining-the-illumina-sequencing-technology/

Paired-end Sequencing

- Reads from both ends of the insert
- Insert is of unknown size
  - Estimated average provided by sequencing facility
- PE help orient reads during assembly
  - Spatial constraints for how far apart pairs can be placed

Figure 4. Paired-End Sequencing and Alignment



Paired-end sequencing enables both ends of the DNA fragment to be sequenced. Because the distance between each paired read is known, alignment algorithms can use this information to map the reads over repetitive regions more precisely. This results in much better alignment of the reads, especially across difficult-to-sequence, repetitive regions of the genome.

http://www.illumina.com/technology/next-generation-sequencing/paired-end-sequencing_assay.html

Project Data

- *Haemophilus influenzae*

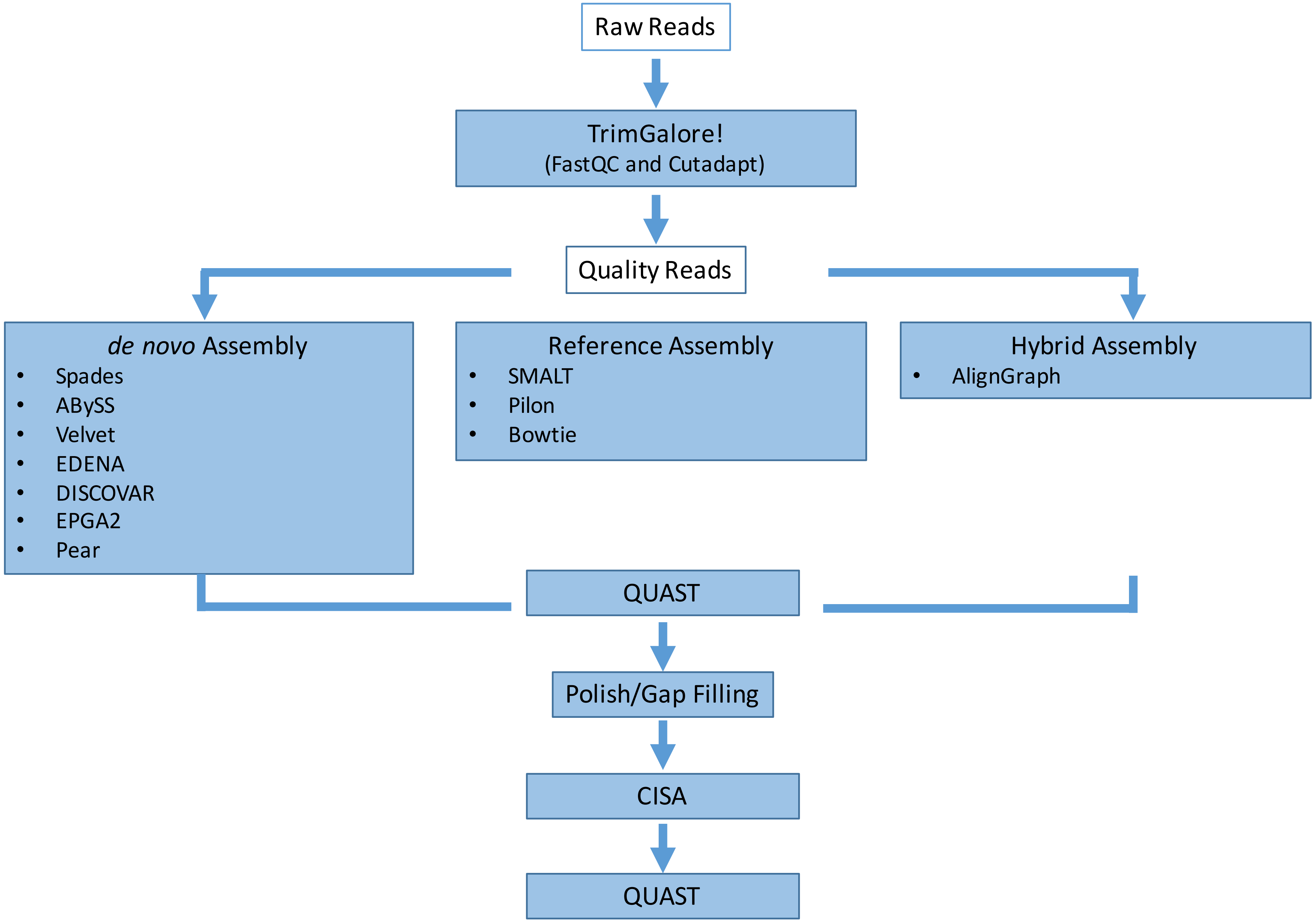
- Sequencing data:
  - Paired-end, 250bp reads
  - From Illumina HiSeq2500
  - Already demultiplexed

http://www.denniskunkel.com/DK/Bacteria/97500E.html

Genome Pipeline

Genome Pipeline

- Quality Control and pre-processing
- Read assembly:
  - *de novo*, reference-guided, and hybrid
  - Assembly programs chosen for testing
- Assembly improvement
  - Polishing/Gap filling
  - Contig integration
  - Automated SNP calling for reference-guided assemblies
- Assembly quality assessment


Raw Reads
TrimGalore!
(FastQC and Cutadapt)
Quality Reads
de novo Assembly
• Spades
• ABySS
• Velvet
• EDENA
• DISCOVAR
• EPGA2
• Pear
Reference Assembly
• SMALT
• Pilon
• Bowtie
Hybrid Assembly
• AlignGraph
QUAST
Polish/Gap Filling
CISA
QUAST

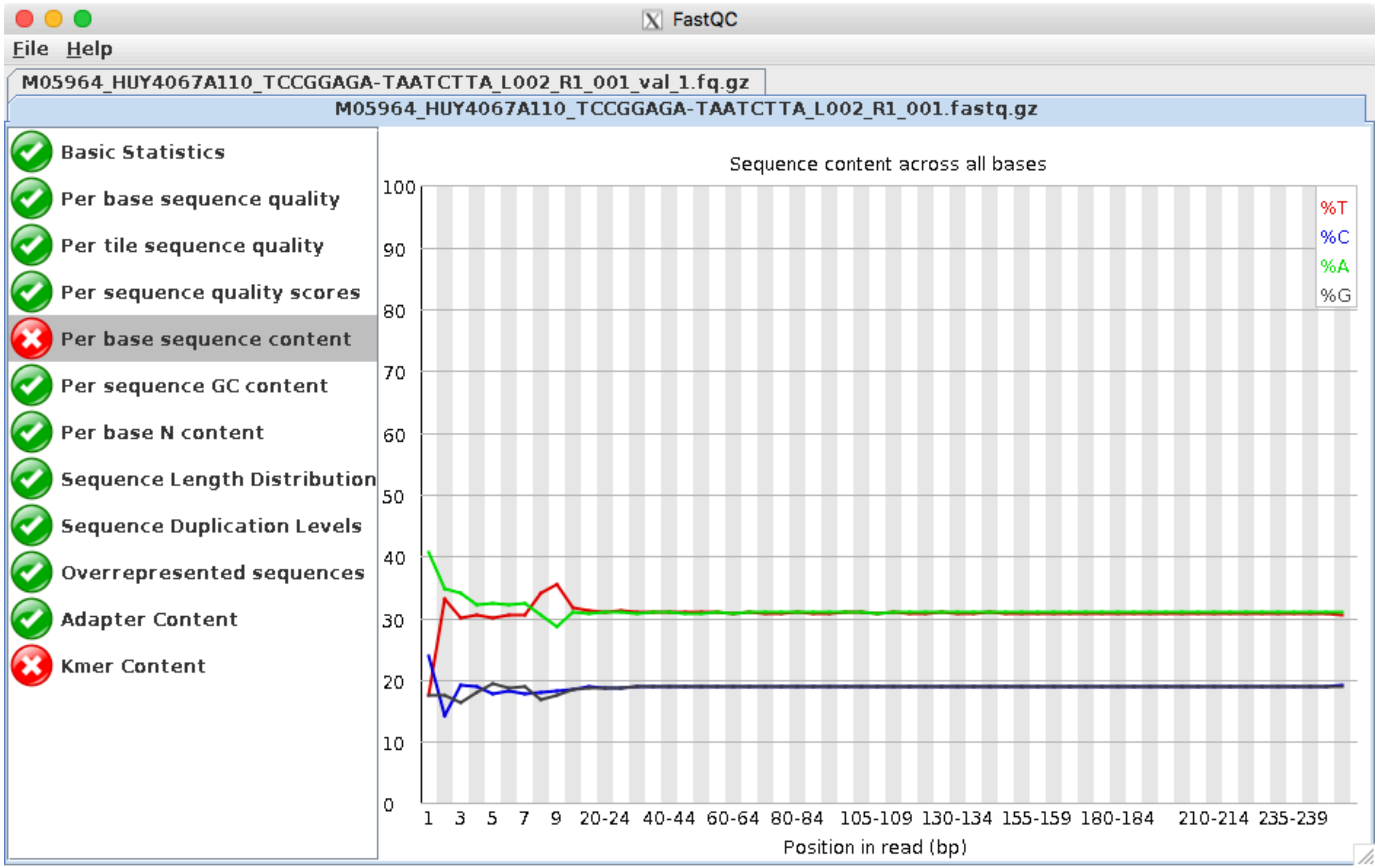
Quality Control and Pre-processing

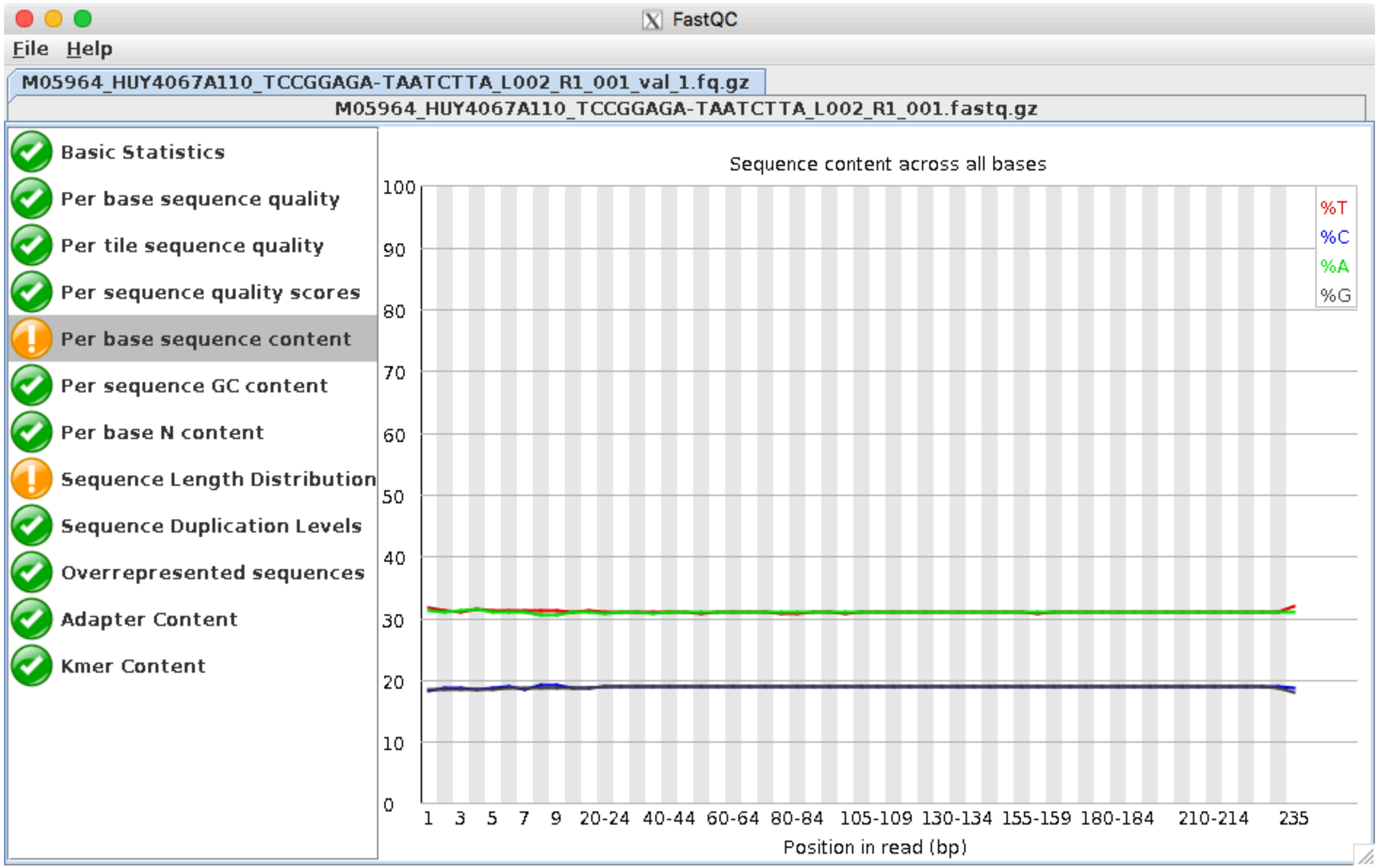
Quality Control and Pre-processing

- Remove barcodes
- Remove poor quality reads
- Filter duplicates (Typically for speed considerations)

Quality Control and Pre-processing

- Sickle
  - Sliding window trimming based on quality and length
- Quake
  - Kmer based trimming and miscall correction
  - Useful to preprocess files when using assemblers without baked in read correction
- Prinseq
  - Sliding window trimming and cleaning based on statistical modeling
- FastQC
  - Trim Galore! – Our pick
  - Uses cutadapt (adapter/barcode cleanup) plus sliding window trimming
  - Summary statistics provided by FastQC


FastQC
File Help
M05964_HUY4067A110_TCCGGAGA-TAATCTTA_L002_R1_001_val_1.fq.gz
M05964_HUY4067A110_TCCGGAGA-TAATCTTA_L002_R1_001.fastq.gz
✓ Basic Statistics
✓ Per base sequence quality
✓ Per tile sequence quality
✓ Per sequence quality scores
✗ Per base sequence content
✓ Per sequence GC content
✓ Per base N content
✓ Sequence Length Distribution
✓ Sequence Duplication Levels
✓ Overrepresented sequences
✓ Adapter Content
✗ Kmer Content
Sequence content across all bases
%T
%C
%A
%G
Position in read (bp)


FastQC

File Help

M05964_HUY4067A110_TCCGGAGA-TAATCTTA_L002_R1_001_val_1.fq.gz

M05964_HUY4067A110_TCCGGAGA-TAATCTTA_L002_R1_001.fastq.gz

✓ Basic Statistics
✓ Per base sequence quality
✓ Per tile sequence quality
✓ Per sequence quality scores
! Per base sequence content
✓ Per sequence GC content
✓ Per base N content
! Sequence Length Distribution
✓ Sequence Duplication Levels
✓ Overrepresented sequences
✓ Adapter Content
✓ Kmer Content

Read Assembly

Read Assembly

- *De novo*: from new
  - SPAdes, ABySS, Velvet, EDENA, DISCOVAR, etc.
  - Use only sequence information to order reads

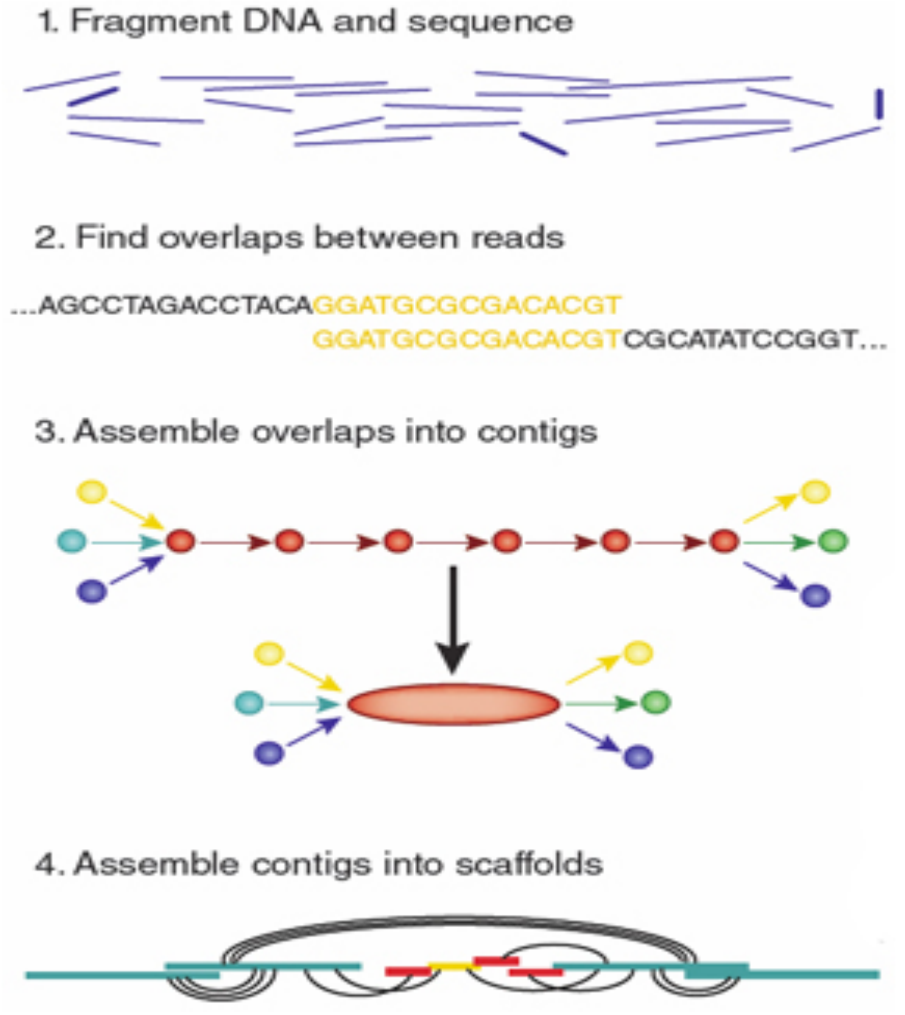
- Reference-guided
  - bwa/bowtie, SMALT PILON, etc.
  - Reference must be closely related
  - "fast"
  - SNP detection
  - Lots of "left over" reads

De novo Assembly

De novo Assembly

- Attempts to order reads using their sequence

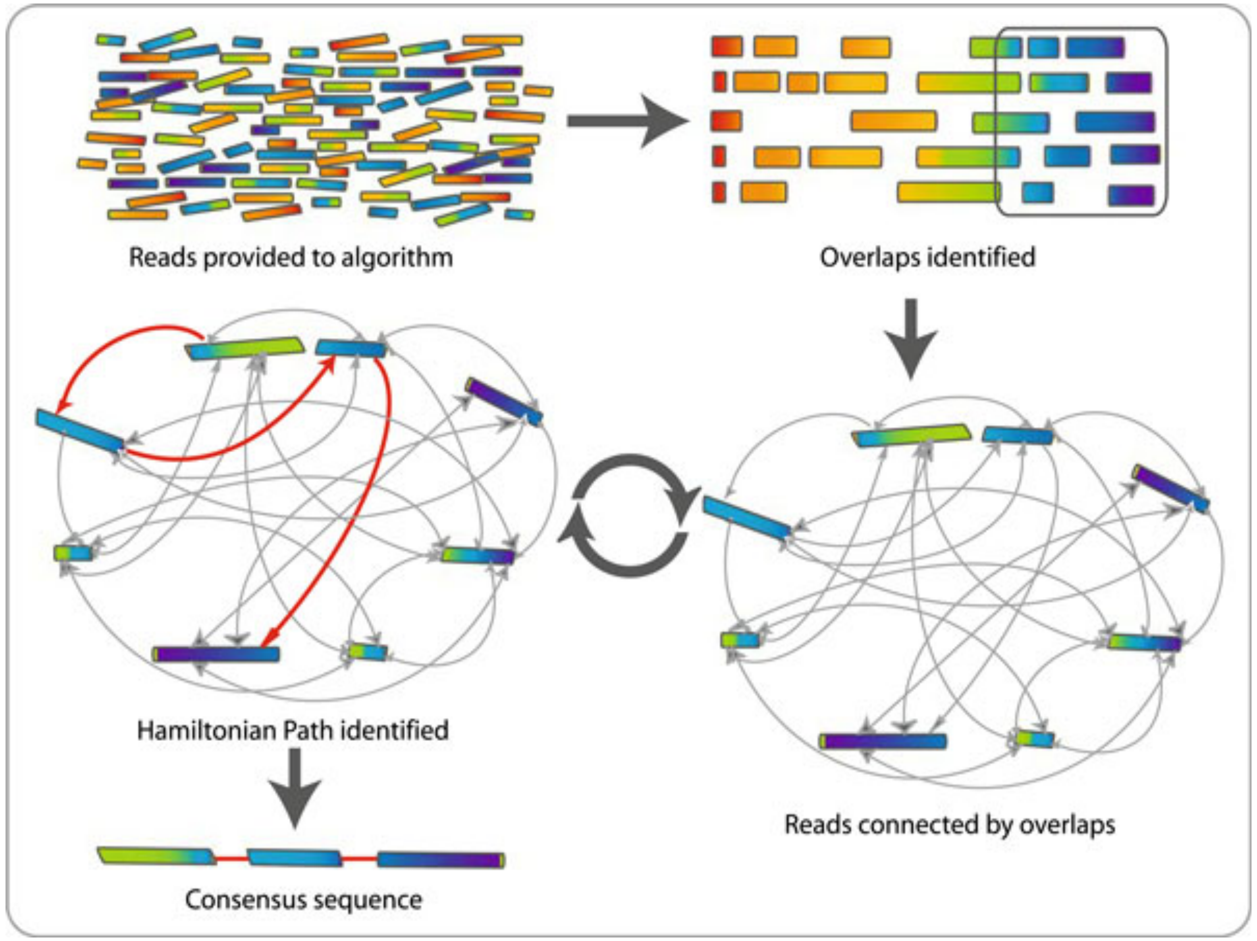
- Two methods:
    1. Overlap Layout Consensus
    2. de Bruijn graphs


1. Fragment DNA and sequence
2. Find overlaps between reads
...AGCCTAGACCTACAGGATGCGCGACACGT
GGATGCGCGACACGTCGCATATCCGGT...
3. Assemble overlaps into contigs
4. Assemble contigs into scaffolds

Overlap Layout Consensus

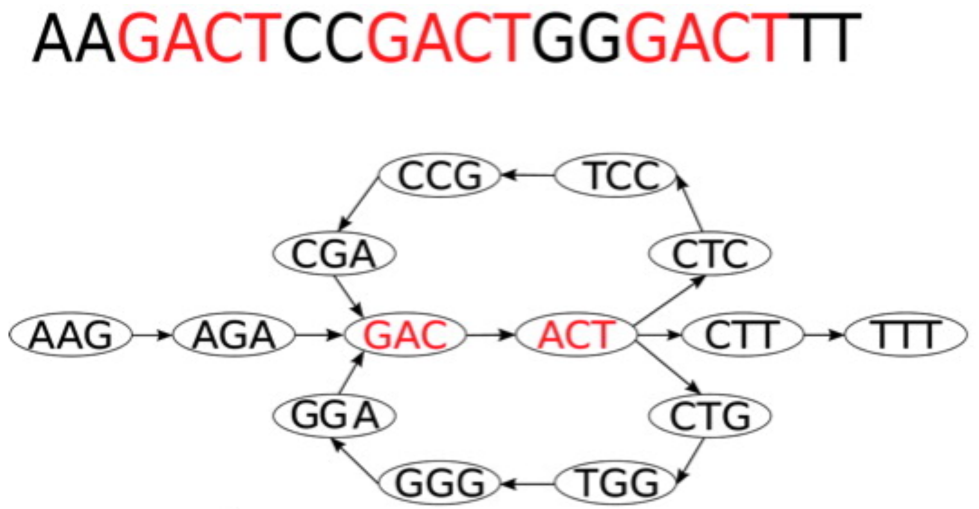
- "Old," computationally expensive

- **Overlap**: Build overlap graph
- **Layout**: Bundle stretches of the overlap graph into contigs
- **Consensus**: Pick most likely nucleotide sequences for each contig

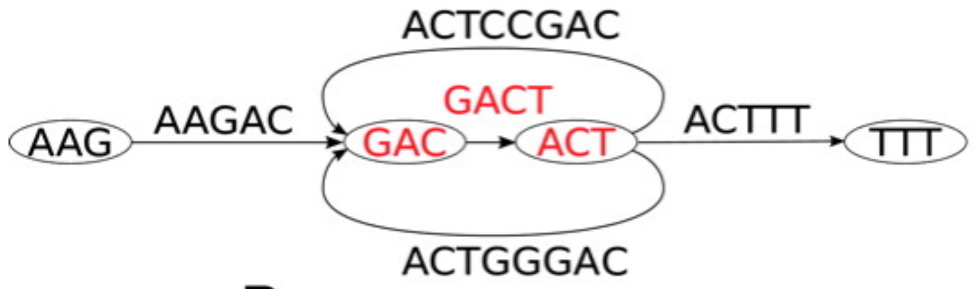

Reads provided to algorithm
Overlaps identified
Hamiltonian Path identified
Reads connected by overlaps
Consensus sequence

de Bruijn Graphs

- Fast, memory and space efficient
- Resolution of ambiguous read placement
- Assembly method:
  - Break reads into mers of size k (smaller than read length)
  - Find overlaps of kmers (computationally easier than OLC)
  - Resolve ambiguities using estimated genomic distances



A de Bruijn graph of a sequence



B condensed de Bruijn graph

SPAdes

- Short read assembler, takes single and paired ends

- High level view of SPAdes assembly:
  i. Assembly graph construction with multi-sized de Bruijn graphs (dBG) and bulge resolution
  ii. Integration of Pair-end data with dBG-derived paths to determine genomic distance
  iii. Paired assembly graph
  iv. Contig reconstruction

SPAdes improvements over other assemblers

1. SPAdes's assembly model uses multi-sized Paired de Bruijn graphs (PDBGs)

2. Bulges, tips and other ends are resolved using the paired end metadata

3. PDBGs are more space/memory/computationally efficient for building and unwinding

4. Built in read correction (BayesHammer), and chimera/miscall correction

ABySS

- Assembly By Short Sequencing
- Amounts of data generated from large-scale sequencing projects
- 3.5 billion paired-end reads from the genome of an African male publicly released by Illumina

ABySS: Algorithmic approach

1. All possible substrings of length k (termed k-mers) are generated from the sequence reads. The k-mer data set is then processed to remove read errors and initial contigs are built

2. Mate-pair information is used to extend contigs by resolving ambiguities in contig overlaps

Evaluation of ABySS performance

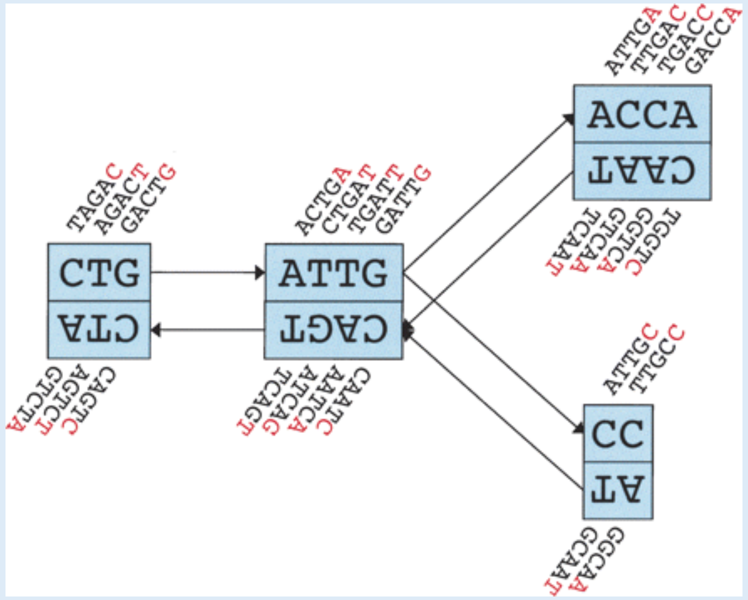
- Simulated data
- Experimental sequence data
- Polymorphic and novel human sequence identified in the assembled experimental data
- Comparison of short read assemblers

Velvet (Daniel Zerbino and Ewan Birney)

- Designed to deal with short read sequencing alignments
- Has 4 stages

Hashing reads into k-mers

Construction of the graph



Error Removal

- **Tips** due to errors at edges of reads
- **Bulges** due to internal read errors removed by TourBus algorithm
- **Erroneous connections** due to cloning errors

Simplification of repeats by separating paths sharing local overlaps

Velvet

- Pros
  - Easy to install, stable
  - Easy to run
  - Fast (multithreading)
  - Can take in long and short reads
  - Can use a reference genome to anchor reads which normally map to repetitive regions (Columbus module)
- Cons
  - Might need large amounts of RAM for large genomes, potentially **> 512 GB** for a human genome if at all possible.

EDENA

- EDENA is high-throughput novel software for de novo assembly of accurate contigs. It is suitable to characterize a bacterial genome which containing data sets with very short reads of the same length. This application is based on the classical assembly approach where all overlaps are computed and structured in a graph. EDENA has two features, exact matching and detection of spurious reads, to improve the assembly of very short sequences:

- There are four steps in this algorithm:

1. Remove redundant information
2. Overlapping phase
3. Cleaning and removing transitive and spurious edges
4. All contigs of a minimum size that are unambiguously represented in the graph are provided as output

DISCOVAR *de novo*

- At the time to Discovar’s paper (Weisenfeld et al., 2014), the methods available to investigate GV did a good job in 90% of most cases.
- Calling variants was a challenging in the 10% remaining of the genome, specifically those occurring in low-complexity sequence, segmental duplications and high GC content regions.
- Discovar was specifically designed to address challenging variant types using a modified ALLPATHS assembly algorithm that identifies variants as it assemble the genome
- Compared with GATK, Discovar provides a better coverage of challenging variants and excellent coverage of ordinary variants
- The standalone de novo assembly algorithm(DISCOVAR de novo) was released later by popular demand.

DISCOVAR *de novo*

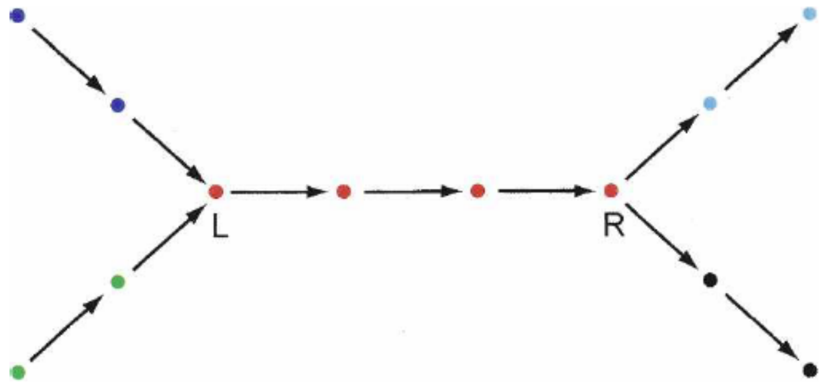
- Currently it takes as input Illumina reads of length 250 or longer — produced on MiSeq or HiSeq 2500 (exclusively) — and from a single PCR-free library.

- Yet, with adjustments, reads as short as 150bp may work with Discovar de novo, depending on fragment size and other factors.

DISCOVAR *de novo*

Phase 1: Error correction in initial graph:

- Reads are error corrected
- Pairs are closed
- Pair closures are merged into unipaths
- Formation of graph is similar to ALLPATHS
    - Compute approx graph by walking each path
    - Select ideal paths for seeding
    - Assemble neighbors around seed
    - Find allpaths
    - Glue together



Unipath: maximal unbranched sequence

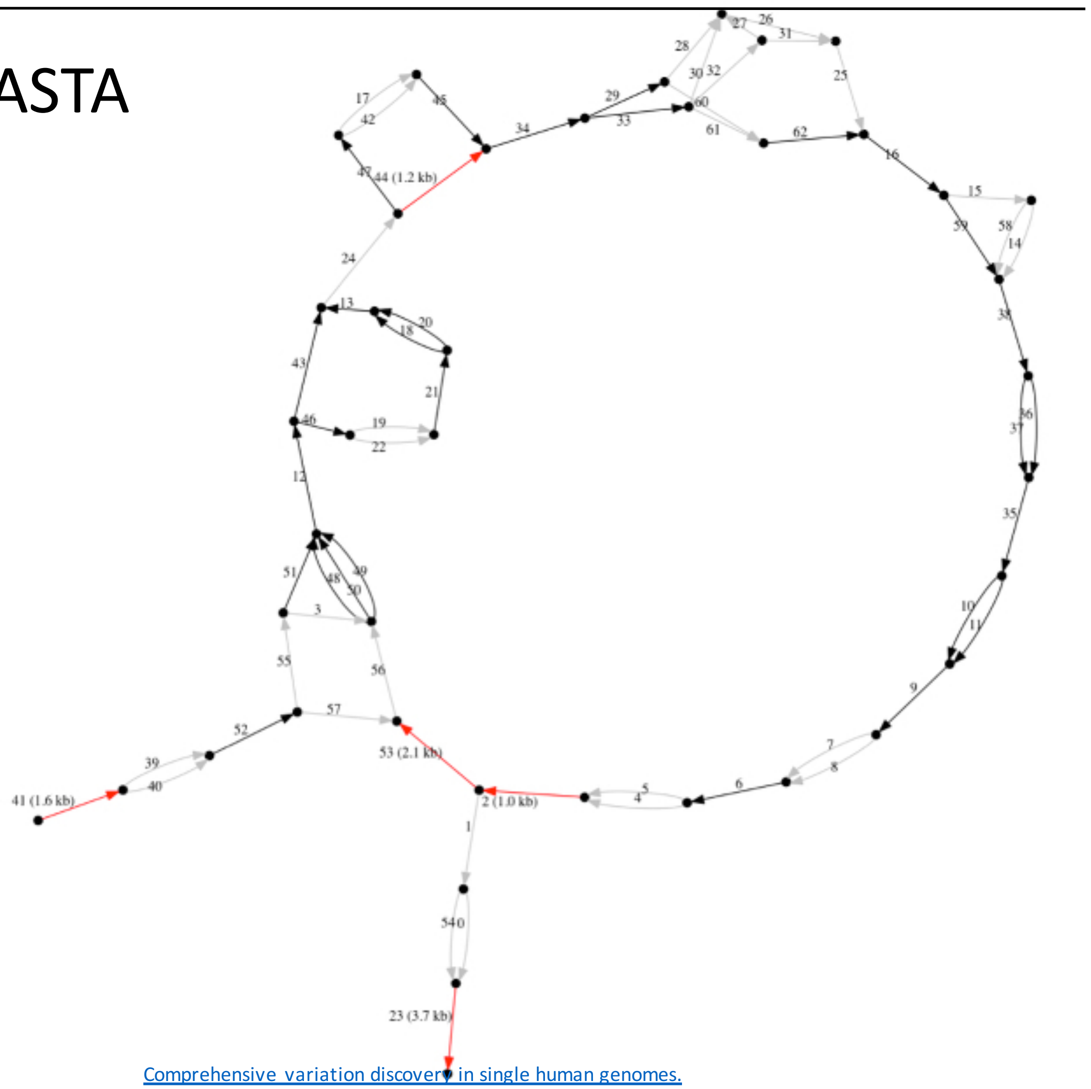
DISCOVAR *de novo*

Phase 2: Optimization of graph:

- Graph is simplified and improved for accuracy
- Pull apart simple branches
- Pull apart complex branches
- “Bubble popping” (cleaning up variants)
- Graph reconstruction
- Orient assembly to reference

DISCOVAR *de novo*

- Formats assembly result into FASTA
- Graph can be viewed at end of assembly
- Variant paths (GVs) can be highlighted



Comprehensive variation discovery in single human genomes.
Weisenfeld NI, Yin S, Sharpe T, Lau B, Hegarty R, Holmes L, Sogoloff B, Tabbaa D, Williams L, Russ C, **Nusbaum C**, Lander ES, MacCallum I, Jaffe DB.
Nat Genet. 2014 Dec;46(12):1350-5. doi: 10.1038/ng.3121. Epub 2014 Oct 19.

EPGA2: Overview

- EPGA2 is an updated version from EPGA that does not require vast amounts of memory to handle large genome sequences as most assemblers do. It is able to be memory efficient and still display improved and higher coverage for contigs and scaffolds.

- As a first step, read errors are removed to help refine the precision of De Bruijn Graph and provide less work when assembling the contigs.

- Secondly, unlike EPGA which uses a hash table to store K-mers and its counts, EPGA2 has the user define the k-mer size(<32) and keeps k-mers that have a frequency greater than one.

- Lastly, to perfect EPGA2 it parallels the contigs so that if a portion of a contig has overlapping regions those sections will merge together.

EPGA2: Algorithm

1. Error Correction of Read Endings Using BLoom-filter-based Error correction Solution for high-throughput Sequencing reads (BLESS)
2. Uses Disk Streaming of K-mers (DSK) to Count K-mers and Partition Reads from the corrected sequences given from BLESS
3. Uses BCALM to Construct De Bruijn Graph -Stores k-mers in memory as clusters
4. Contig Assembly: starts with long nodes from the De Bruijn Graph to find the nodes before and after it
5. Parallel Contigs Merging
6. Scaffolding: order is determined by paired end reads
7. Gap Filling

PEAR (Paired-End reAd mergeR)

- A memory-efficient and accurate pair-end read merger that can generate reads from both ends of the target DNA fragments.
- It can assemble 95% of the reads with 35 bp mean overlap, with a false positive rate of 0.004.
- This assembler is accurate on datasets with:
    - short overlaps, and
    - DNA target fragment sizes that are smaller than single-end read lengths.
- Does not require the target fragment size as input.
- Does not require preprocessing of the raw data or specifying the fragment size.

- PEAR scores all possible overlaps for each pair of corresponding paired-end reads to determine the overlap with the highest Assembly Score (AS).

$$\sum_{i=1\ldots|C|} (\Pr[X'_i = Y'_i | X_i = Y_i] \cdot \alpha)^{\delta_i} (\Pr[X'_i \neq Y'_i | X_i \neq Y_i] \cdot \beta)^{1-\delta_i}$$

- Where:

$$\delta_i = \begin{cases} 1 & : \text{A match is observed } (X_i = Y_i) \\ 0 & : \text{A mismatch is observed } (X_i \neq Y_i) \end{cases}$$

- A scoring matrix that penalizes mismatches with a negative value β and rewards matches with a positive value α. Empirical tests using simulated data showed that setting and α =1.0 and β = -1.0 yield the best results.
- For the merged reads, PEAR computes the overlap that maximizes the AS. We denote the overlap that maximizes the AS by C*.

- PEAR generates four FASTQ output files:
  - Successfully merged reads
  - Forward and reverse unmerged reads
  - Discarded reads.
- The basic command to run PEAR is:
  -
    ```
    ./pear -f forward_read.fastq -r reverse_read.fastq -o output_prefix
    ```
- Requirements:
  - GNU Autotools and GNU Libtool for compiling the source code.
- Repository:
  - https://github.com/xflouris/PEAR

Reference-guided Assembly

Reference-guided Assembly

- Reconstruct genome of interest using:
  i. Genome of a closely related organism
  ii. Raw reads of interest mapped for the above genome

- Genome is reconstructed by taking the consensus call for a give base

- Much faster than *de novo*, especially for strains of the same species

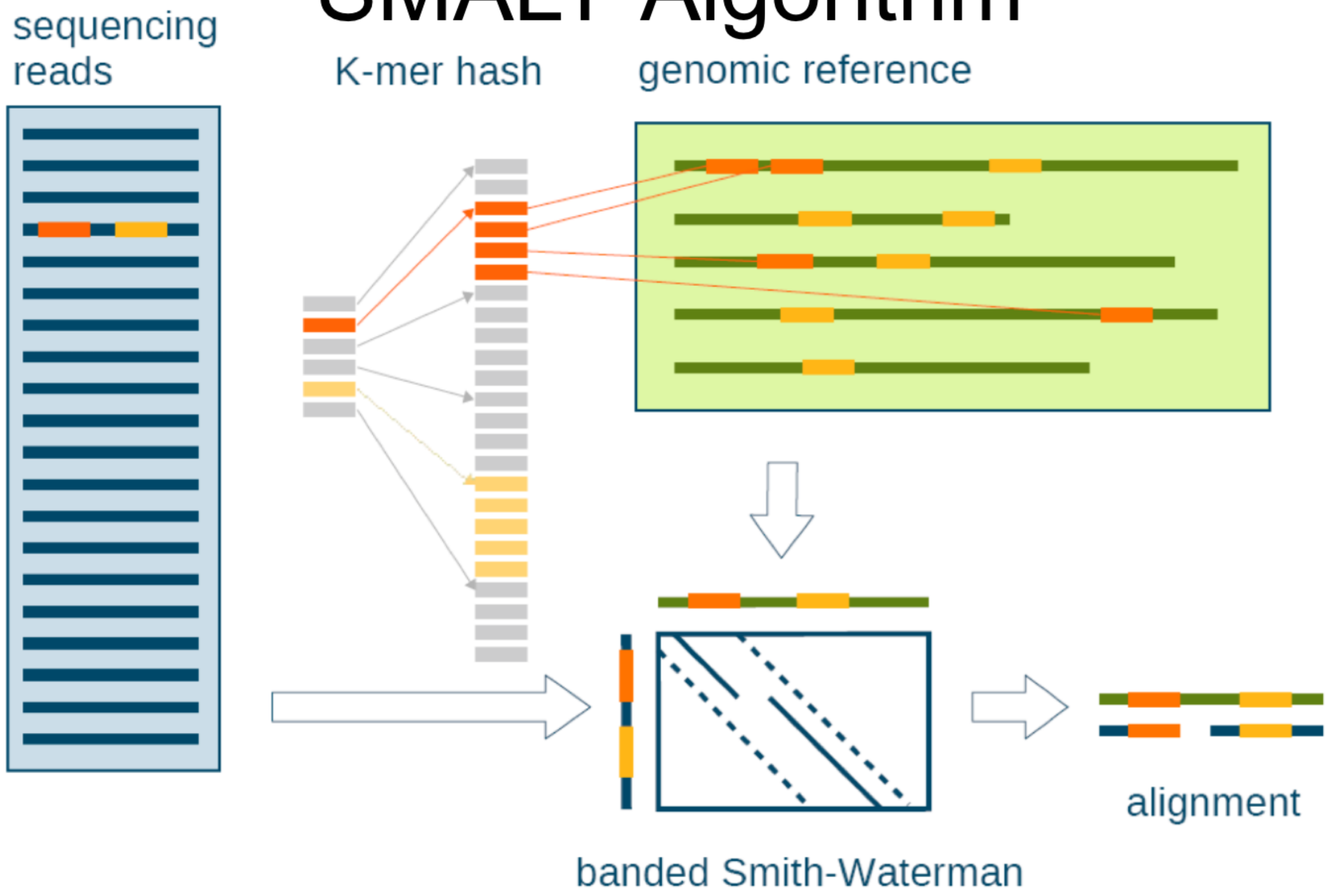
- Variant calling using same mapped data

SMALT

- SMALT is a pairwise sequence alignment program designed for the efficient and accurate mapping of DNA sequencing reads onto genomic reference sequences.

- The software employs a perfect hash index of short words, less than 15 nucleotides long, sampled at equidistant steps along the genomic reference sequences. For each read, potentially matching segments in the reference are identified from seed matches in the index and subsequently aligned with the read using a banded Smith-Waterman algorithm.

SMALT Algorithm



SMALT

- Paired-end 2 × 100 bp reads produced by Illumina sequencing platforms for the human genome can be mapped at a rate of 1.2 × 10^6 pairs per hour. Simulations suggest error rates below 0.03% when 96.7% of the reads are mapped.

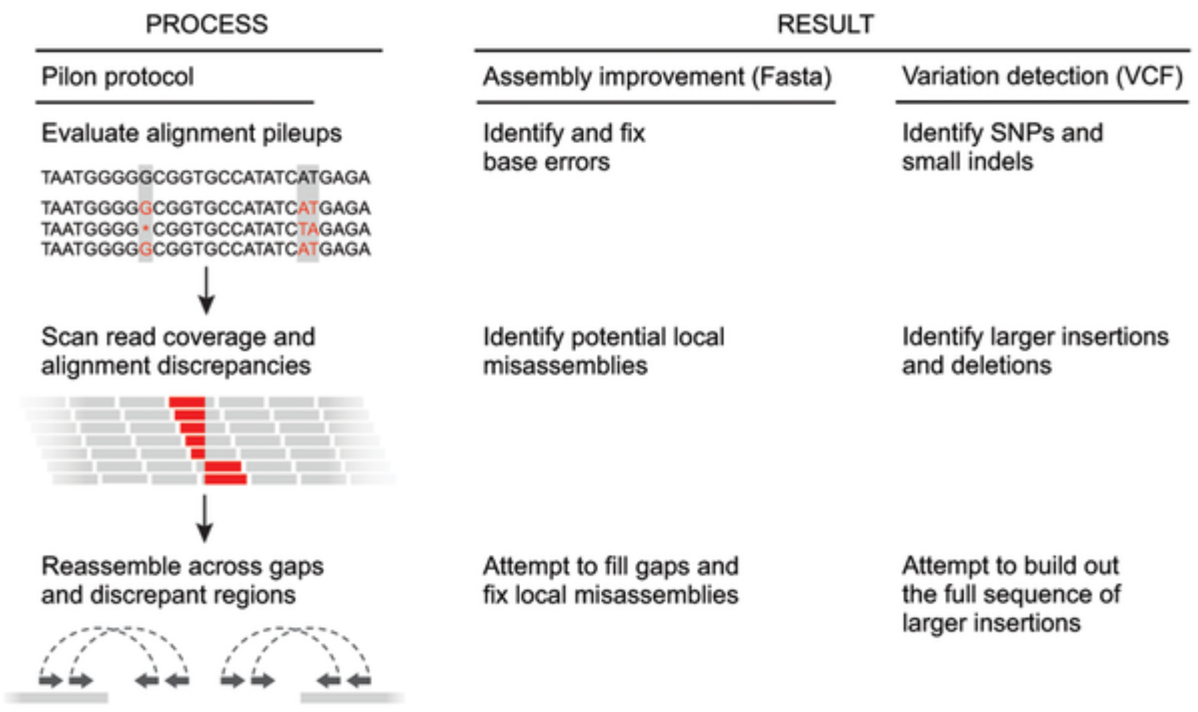
- The user can adjust the tradeoff between sensitivity and speed by tuning the length and spacing of the hashed words.

Pilon

- Ostensibly not an assembler but designed as a tool to combine and automate the tasks of correcting of draft assemblies and variant calling.
- Pilon requires as input a FASTA file of the genome - either an existing draft assembly or a reference assembly from another strain - along with one or more BAM files of reads aligned to the input FASTA file. It then attempts to make improvements to the input reference using evidence from the reads.
- Pilon then outputs a FASTA file containing an improved representation of the genome from the read data and an optional VCF file detailing variation seen between the read data and the input genome. There is also an option to produce tracks that can be displayed in genome viewers such as IGV and GenomeView.

Pilon

- Correcting single base differences
- Resolving repeats using Jump Pairs
- Identifying and aligning small indels
- Fixing larger indel or block substitution events
- Gap filling
- Identification of local misassemblies, including optional opening of new gaps


PROCESS
RESULT
Pilon protocol
Assembly improvement (Fasta)
Variation detection (VCF)
Evaluate alignment pileups
Identify and fix base errors
Identify SNPs and small indels
TAATGGGGGCGGTGCCATATCATGAGA
TAATGGGGGCGGTGCCATATCATGAGA
TAATGGGG*CGGTGCCATATCTAGAGA
TAATGGGGGCGGTGCCATATCATGAGA
Scan read coverage and alignment discrepancies
Identify potential local misassemblies
Identify larger insertions and deletions
Reassemble across gaps and discrepant regions
Attempt to fill gaps and fix local misassemblies
Attempt to build out the full sequence of larger insertions

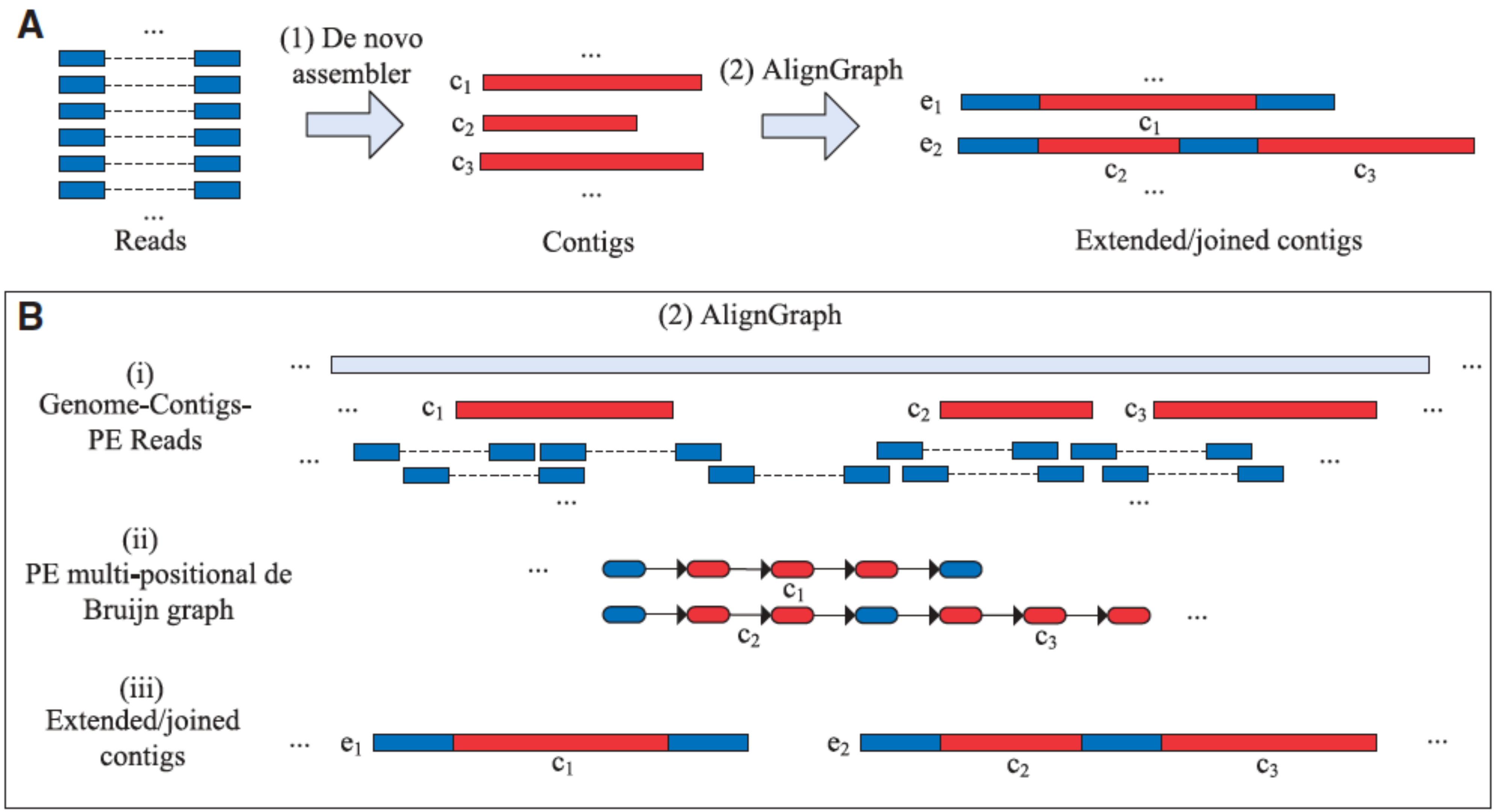
Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement.
PLoS One. 2014 Nov 19;9(11):e112963. doi: 10.1371/journal.pone.0112963. eCollection 2014.
Walker BJ[1], Abeel T[2], Shea T[1], Priest M[1], Abouelliel A[1], Sakthikumar S[1], Cuomo CA[1], Zeng Q[1], Wortman J[1], Young SK[1], Earl AM[1].

Hybrid Assembly

AlignGraph

- AlignGraph is a "reference-assisted assembly approach" (1).
- Guided by a reference genome of a closely related organism, AlignGraph algorithm extends and joins *de novo*-assembled contigs or scaffolds
- This approach will likely become the default option for future projects
- Inputs:

1. Paired-end DNA reads in FASTA format
2. *De novo* contigs or scaffolds assembled by:
    - Velvet, ABySS, ALLPATHS-LG, SOAPdenovo, etc.
3. Reference genome from a closely related organism



http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4058956/pdf/btu291.pdf

AlignGraph: Advantages

- Performance tests show AlignGraph is able to considerably improve the contigs and scaffolds from several assemblers (Velvet, ABySS, etc.)

- For example:
  - AlignGraph allowed 28.7–62.3% of the contigs of *A. thaliana* and human to be extended => improvements of assembly metrics

Assembly Improvement

Assembly Improvement

- Polishing - Fixing short indels and miscalls
- Gap filling - Remove N's and extending contig ends
- Contig integration - Merging two assemblies
- Automated SNP calling for reference guided assemblies

Polishing and Gap Filling

- Polishing - PILON, IMAGE2
    - Map raw reads to check for consistency
    - SPAdes does this automatically using bayeshammer during assembly

- Gap Filling - Sealer (ABySS), PILON, GMcloser, IMAGE2
    - Map raw reads to assembly and use low-coverage information
    - Does not always work, some regions are just not sequenced

Contig Integration

- Attempt to build better assemblies by combining multiple assemblies
  - Typically from different assemblers
  - Some assembly algorithms are strict and produce shorter, low error contigs
  - Some are greedy and produce longer contigs with a higher error rate

- Contig integration requires that input assemblies have high homology

Automated SNP Calling during Refinement

- Part of PILON's internal pipeline
- Uses the SNP calling pipeline from samtools on alignment generated during polishing and gap closing
- Applies some statistical models to suppress low quality SNP calls

Assembly Quality Assessment

Assembly Quality Assessment

- Quast
  - Gold standard
  - Produces summary statistics and graphics
- SuRankCo
  - Machine-learning based approach
  - Can be ”trained” using assemblies of interest
- qaTools
  - Uses SAM/BAM mapping files
  - Computes many of the same stats as Quast

Broad Typing using MLST

- SRST2 tool
- Short Read Sequence Typing
- Used to sort reads into Hi and Hhae

References

- ABySS: http://www.bcgsc.ca/platform/bioinfo/software/abyss

- AlignGraph:

Bao E, et al. (2014) AlignGraph: algorithm for secondary de novo genome assembly guided by closely related references, Bioinformatics, 30, i319-i328. (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4058956/pdf/btu291.pdf)

https://github.com/baoe/AlignGraph

- DISCOVAR:

Weisenfeld NI et al. Comprehensive variation discovery in single human genomes. Nat Genet. 2014 Dec; 46(12):1350-5.

- EDENA: http://www.genomic.ch/edena.php

- EPGA2: https://github.com/bioinfomaticsCSU/EPGA2

- PEAR: https://github.com/xflouris/PEAR

References

- Pilon:

Walker BJ, et al. Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS One. 2014 Nov 19; 9(11):e112963.

- SPAdes: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3342519/

- SMALT: http://wiki.hpc.ufl.edu/doc/SMALT

- Trim Galore!: http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/

- Velvet:

1. Zerbino DR, Birney E. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs.*Genome Research*. 2008;18(5):821-829. doi:10.1101/gr.074492.107.

2. https://en.m.wikibooks.org/wiki/Next_Generation_Sequencing_(NGS)/De_novo_assembly

- Images:

http://www.denniskunkel.com/DK/Bacteria/97500E.html

http://bitesizebio.com/13546/sequencing-by-synthesis-explaining-the-illumina-sequencing-technology/

http://www.illumina.com/technology/next-generation-sequencing/paired-end-sequencing_assay.html

Thanks! Questions?