Next-generation sequencing

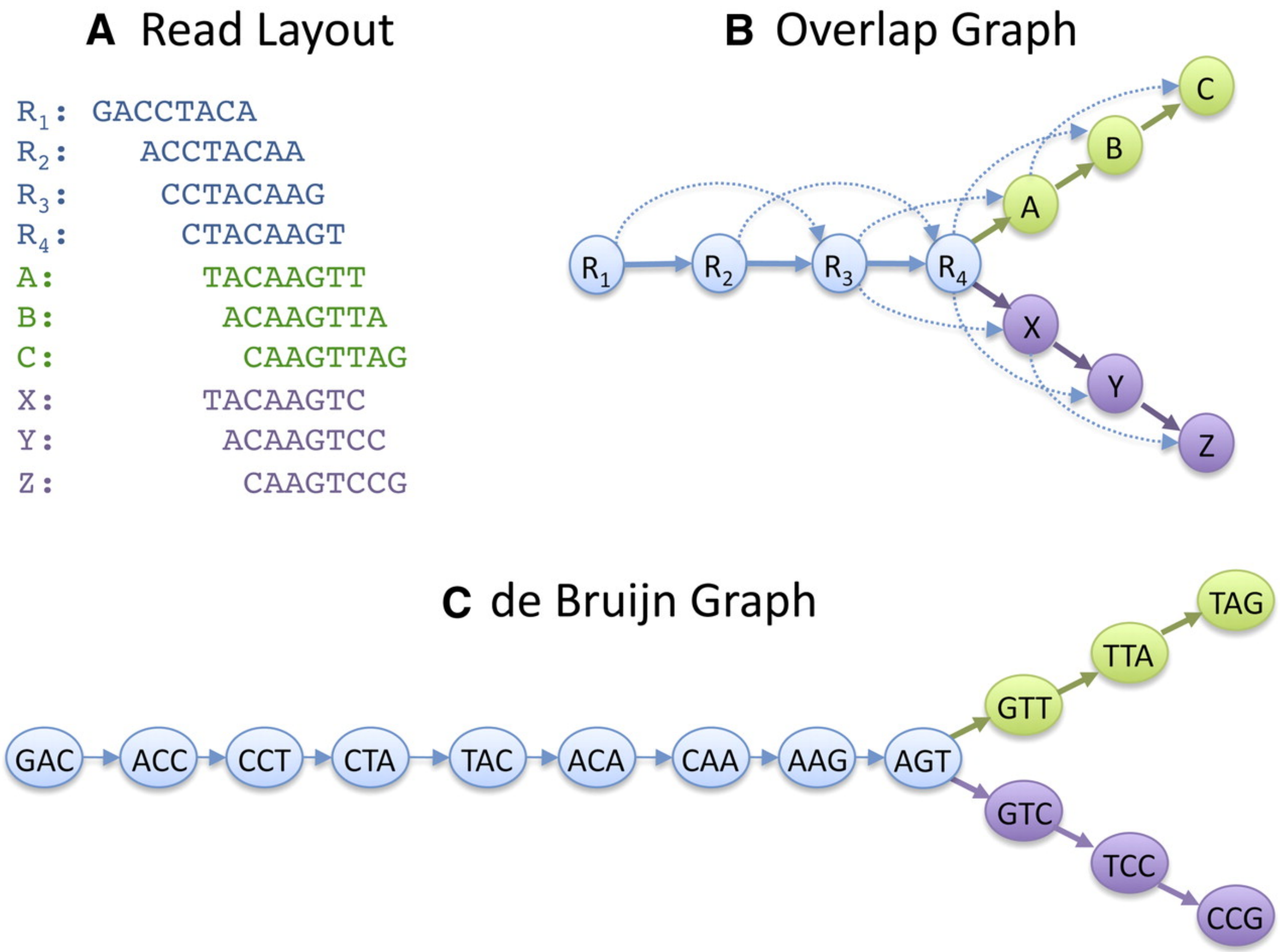
Lecture 6

Assembly Algorithms

Main algorithm used:

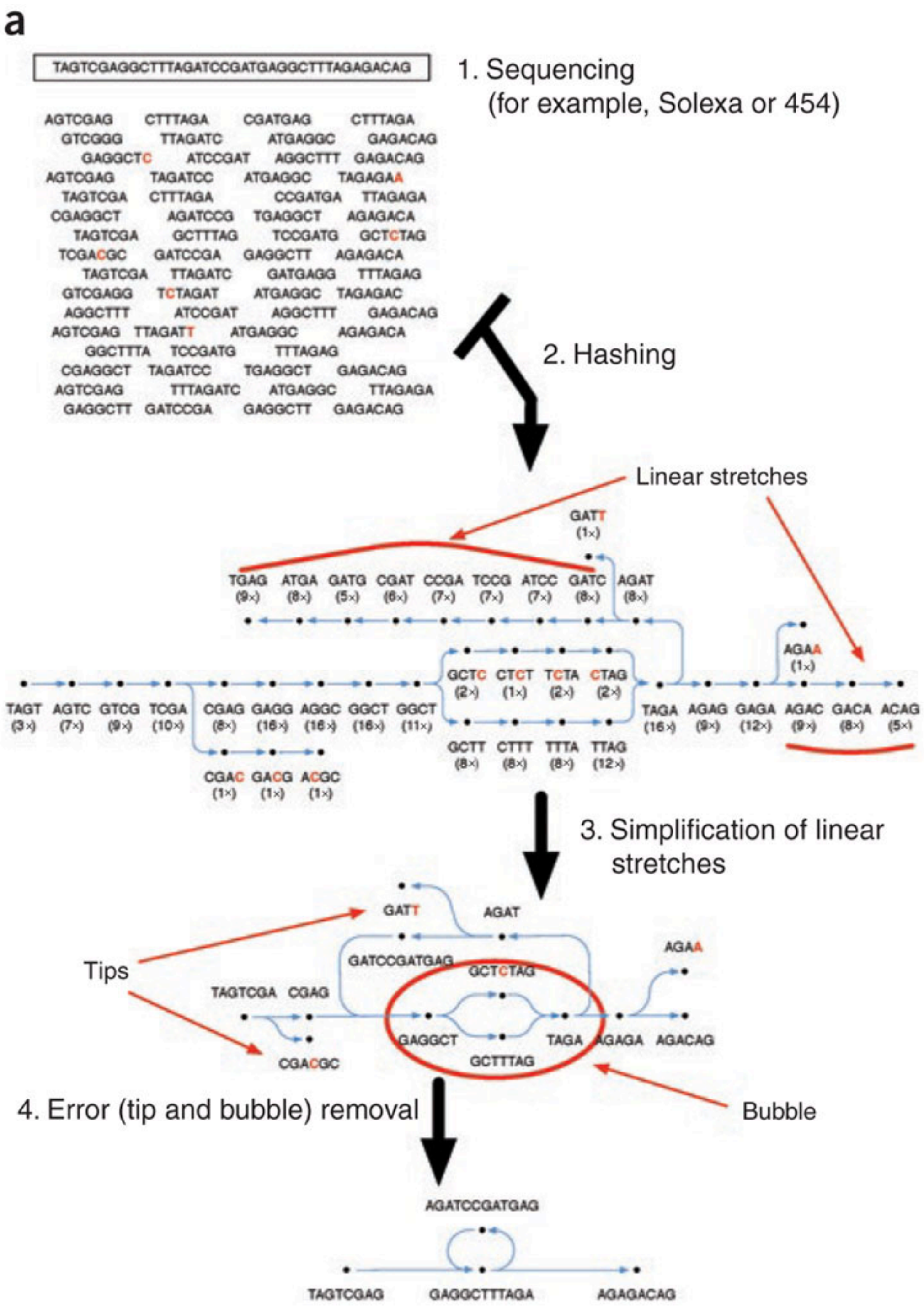
- Greedy algorithms
- Overlap Layout Consensus
- De bruijn graphs

Differences between an overlap graph and a de Bruijn graph for assembly.



Schatz M C et al. Genome Res. 2010;20:1165-1173

De Bruijn Graphs


a
TAGTCGAGGCTTTAGATCCGATGAGGCTTTAGAGACAG
1. Sequencing
(for example, Solexa or 454)
AGTCGAG CTTTAGA CGATGAG CTTTAGA
GTCGGG TTAGATC ATGAGGC GAGACAG
GAGGCTC ATCCGAT AGGCTTT GAGACAG
AGTCGAG TAGATCC ATGAGGC TAGAGAA
TAGTCGA CTTTAGA CCGATGA TTAGAGA
CGAGGCT AGATCCG TGAGGCT AGAGACA
TAGTCGA GCTTTAG TCCGATG GCTCTAG
TCGACGC GATCCGA GAGGCTT AGAGACA
TAGTCGA TTAGATC GATGAGG TTTAGAG
GTCGAGG TCTAGAT ATGAGGC TAGAGAC
AGGCTTT ATCCGAT AGGCTTT GAGACAG
AGTCGAG TTAGATT ATGAGGC AGAGACA
GGCTTTA TCCGATG TTTAGAG
CGAGGCT TAGATCC TGAGGCT GAGACAG
AGTCGAG TTTAGATC ATGAGGC TTAGAGA
GAGGCTT GATCCGA GAGGCTT GAGACAG
2. Hashing
Linear stretches
GATT (1×)
TGAG (9×) ATGA (8×) GATG (5×) CGAT (6×) CCGA (7×) TCCG (7×) ATCC (7×) GATC (8×) AGAT (8×)
AGAA (1×)
GCTC (2×) CTCT (1×) TCTA (2×) CTAG (2×)
TAGT (3×) AGTC (7×) GTCG (9×) TCGA (10×) CGAG (8×) GAGG (16×) AGGC (16×) GGCT (16×) GGCT (11×)
TAGA (16×) AGAG (9×) GAGA (12×) AGAC (9×) GACA (8×) ACAG (5×)
GCTT (8×) CTTT (8×) TTTA (8×) TTAG (12×)
CGAC (1×) GACG (1×) ACGC (1×)
3. Simplification of linear stretches
GATT
AGAT
GATCCGATGAG
GCTCTAG
AGAA
Tips
TAGTCGA CGAG
GAGGCT
TAGA AGAGA AGACAG
CGACGC
GCTTTAG
4. Error (tip and bubble) removal
Bubble
AGATCCGATGAG
TAGTCGAG GAGGCTTTAGA AGAGACAG

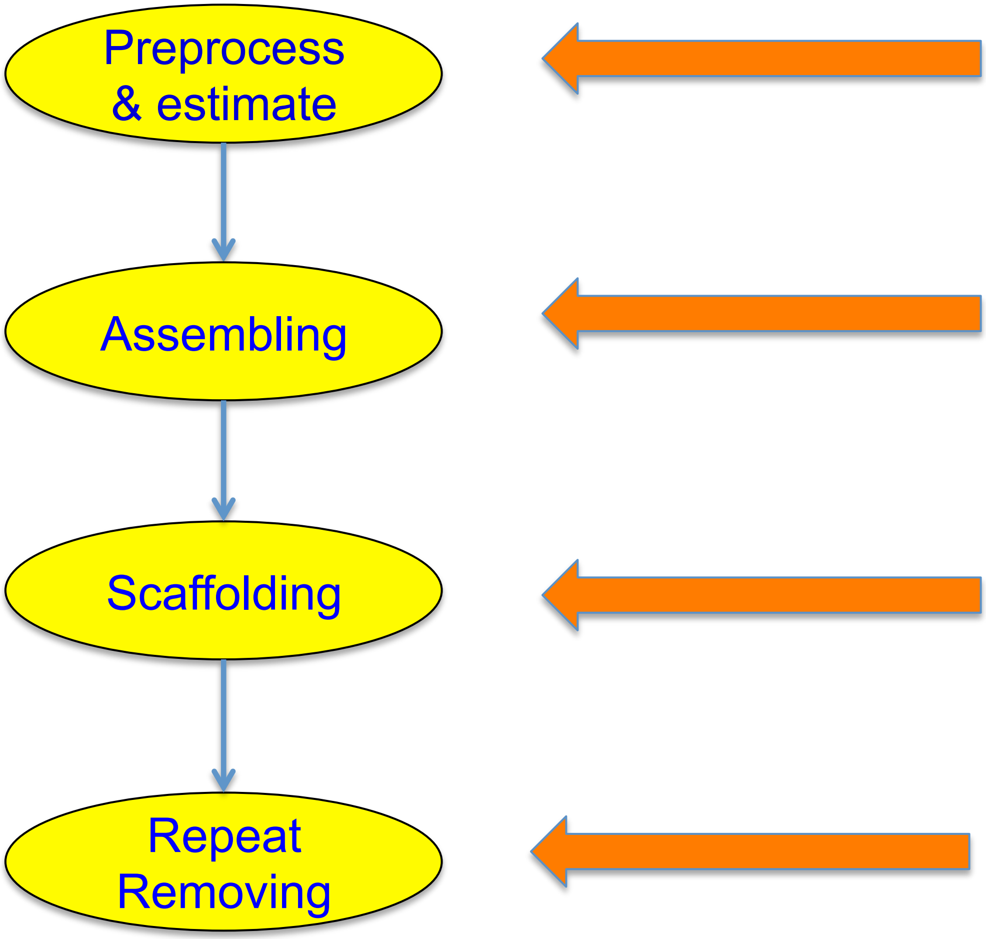
1. Get k-bp (k-mer) subsequences for reads.
2. k-mers in the reads are collected into nodes and the coverage at each node is recorded. Link two k-mer nodes if they have overlap.
3. the graph is simplified to combine nodes that are associated with the continuous linear stretches into single, larger nodes of various k-mer sizes.
4. error correction removes the tips and bubbles that result from sequencing errors and creates a final graph structure that accurately and completely describes in the original genome sequence.

Flicek, Nature Methods, 2009

De Bruijn Assemblers

- Euler: http://nbcr.sdsc.edu/euler/ , Sanger, 454, 2001-2006
- **Velvet**: http://www.ebi.ac.uk/~zerbino/velvet/, small genomes, Sanger, 454, Solexa, SOLiD, 2007-2009 (very good for small genome)
- **ABySS**: http://www.bcgsc.ca/platform/bioinfo/software/abyss, large genome, Solexa, SOLiD, 2008-2011 (for very large genome)
- **SOAP-denovo**: http://soap.genomics.org.cn/soapdenovo.html, Solexa, 2009
- ALLPATH-LG: http://www.broadinstitute.org/software/allpaths-lg/blog/, large genome, Solexa, SOLiD, 2011 (very good performance bu require 2 lib of different insert sizes)
- IDBA-UD: http://i.cs.hku.hk/~alse/hkubrg/projects/idba_ud/, Sanger, 454,Solexa, 2010 (metagenomic, doesn't rely on coverage to remove error)

Assembly Pipeline



Preprocess & estimate

Assembling

Scaffolding

Repeat Removing

30-50X coverage, paired-end sequencing, insert size 1k-100kbp, multiple librar

- **Velvet**: small genomes
- **ABySS**: large genome

Using mate pairs. BAMBUS, SSPACE, GRASS

Need to deal with repeat at any steps during assembly. Assemblers can do some work for repeat control

Assessing Assembly Quality

- Why do we need QC?
  - Misassembly correction is expensive
  - some assemblers have a simple quality-control method that does not capture larger errors

- Common measures of quality:
  - number and sizes of contigs (N50)
    - Assumption: few large contigs is better than many small contigs.
    - True because there are less gaps in the former, but, does not account for the possibility of misassemblies.
  - And more ..
  - Compare with a complete sequence

Assembly validation

N50 is the most commonly used metric:

Weighted median such as 50% of your assembly is contained in contigs with length >=N50

1. Make a list L of positive integers (contig lengths).
2. Create another list L', which is identical to L, except that every element n in L has been replaced with n copies of itself.
3. The median of L' is the N50 of L.

Assembly validation

For example:

L = {2, 2, 2, 3, 3, 4, 8, 8},

L’={2,2, 2,2, 2,2, 3,3,3, 3,3,3 ,4,4,4,4, 8,8,8,8,8,8,8,8, 8,8,8,8,8,8,8,8}

N50 of L is the median of L’.

N50=(4+8)/2 = 6.

Assembly validation

While the N50 value thus quantifies the ability of the assembly algorithm to combine reads into large seamless blocks, it fails to capture all aspects of assembly quality.

For example, artificially high N50 values can be obtained by lowering thresholds for amalgamating smaller blocks of contiguous reads, resulting in misassembled contigs.

N50 values fail to reflect fine-scale inaccuracies, such as substitution and indel errors.

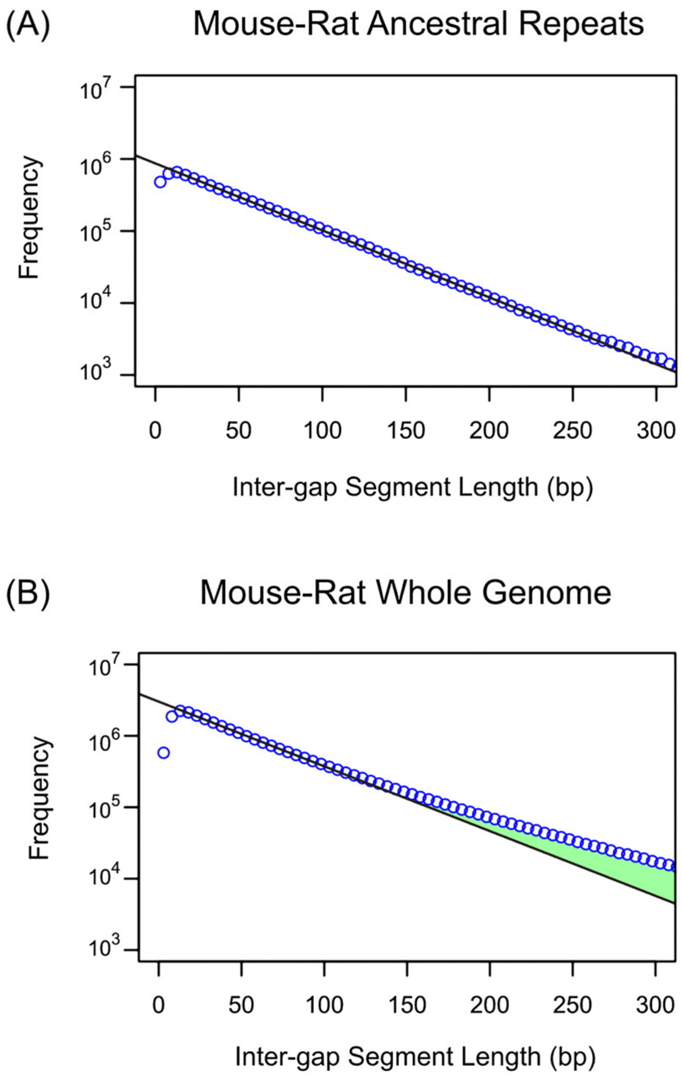
Assembly validation

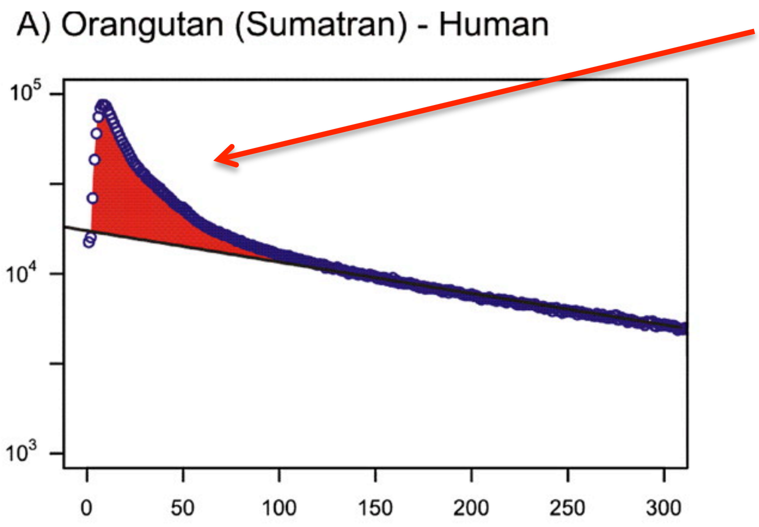
- Quality at the nucleotide level for contigs can be used to detect fine-scale inaccuracies, such as substitution and indel errors.
- Method 1: Once assembled, a base is assigned a consensus quality score (CQS) depending on its read depth and the quality of each base contributing to that position. (Huang and Madan 1999, Genome Research, 9: 868–877).
- Method 2: A multiple sequence alignment of reads is constructed and a consensus sequence along with a quality value for each base is computed for each contig.

Assembly validation

- Method 3: a statistical and comparative genomics method that quantifies the fine-scale quality of a genome assembly and that has the merit of being complementary to the aforementioned approaches.
- This approach estimates the abundance of indel errors between aligned genome pairs, by separating these from true evolutionary indels.
- indel mutations leave a precise and determinable fingerprint on the distribution of ungapped alignment block lengths. These block lengths, which represent distances between successive indel mutations are intergap segment (IGS) lengths.

Assembly validation





errors

Under the **neutral indel model**, these inter-gap segment (IGS) lengths are expected to follow a geometric frequency distribution.

Meader et al., Genome Research, 2010, 20(5):675

Assembly validation

Compare with existing genes.

CEGMA: Core Eukaryotic Genes Mapping Approach

- Looks in your assembly for genes that should be there
- Usually best assembly have best CEGMA score

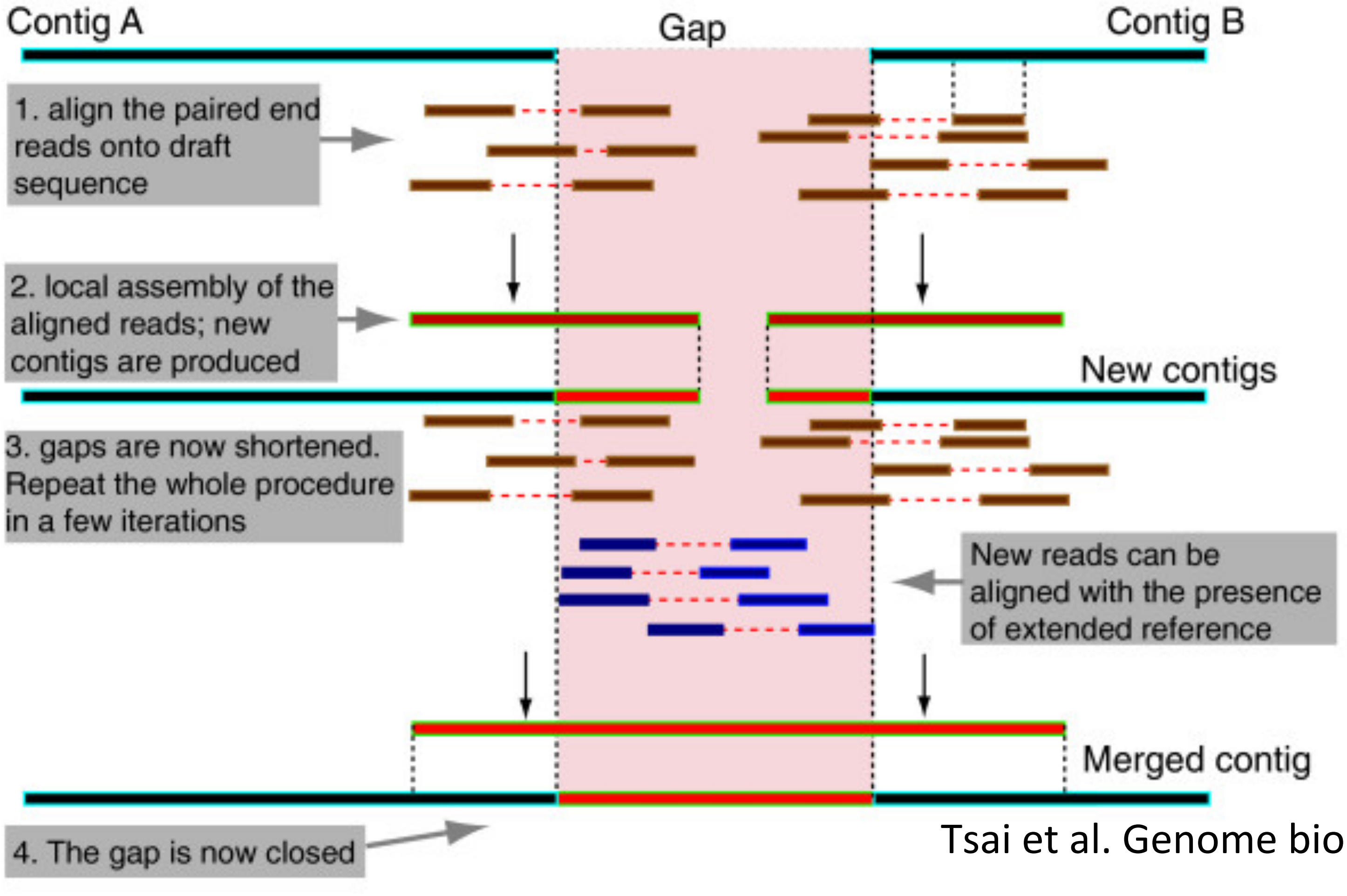
http://korflab.ucdavis.edu/datasets/cegma/

What makes an assembly good?

- High coverage: 50 to 100X
- Different but precise insert size libraries (Paired end from different library sizes will allow you to stitch across several repeat type.)
- Avoid large number of variant.
- Sequencing errors will increase the size of the graph before correction and will sometime create branches that look real
- Error Correction: Correct the read before assembly

What makes your assembly better?

IMAGE: Gap Filling. improve draft genome assemblies by aligning sequences against contig ends and performing local assemblies to produce gap-spanning contigs.



Contig A
Gap
Contig B
1. align the paired end reads onto draft sequence
2. local assembly of the aligned reads; new contigs are produced
New contigs
3. gaps are now shortened. Repeat the whole procedure in a few iterations
New reads can be aligned with the presence of extended reference
Merged contig
4. The gap is now closed

Tsai et al. Genome biology 2010

Whole genome sequencing

- *De Novo* sequencing
- Mapping assembly (Reference-guided assembly) (Resequencing)

"DNA resequencing is the task of sequencing a DNA region for an individual given that a reference sequence for this region is already available for the specific species. "

✓ Whole genome assembly

✓ Variant discovery

Discover or quantitate rare sequence variants
HIV mutants within a single patient
Scan for mutations in tumor samples