# Genome assembly


Lars Arvestad
in BB2490

1

## *Objective*: Reconstruct a molecule from parts

- (Gene)
- Bacterial genome
  Circular
- Eukaryotic genome
  Size?
  Haploid/diploid/polyploidy?
  Complexity?
- Genomes from a sample
  — metagenomics

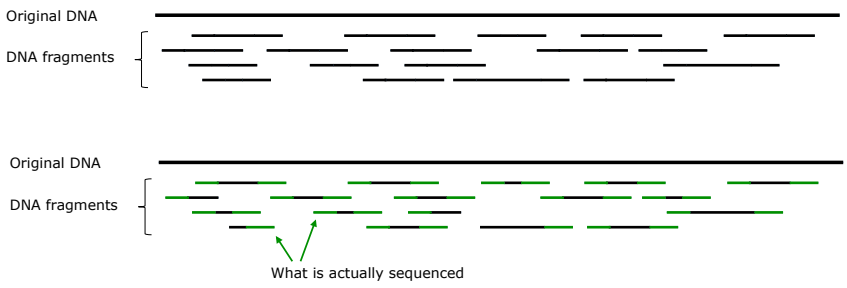
2

# Assembly applications
## — Why *you* might need to assemble reads

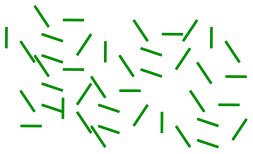
- **Get models of genomes**
  - *de novo* genome assembly
- **Fix problems** with genome models
  - When an assembly is wrong
  - When there is a region missing
- **Get models of genes (regional assembly)**
  - From ”fresh” gene sequencing
  - From hits in NCBI’s Trace Archive: sequencing projects deposit early
- **Structural variant analysis**
  - Find reads from region that may differ from reference
  - Reassemble — local assembly

3

# Shotgun sequencing




What you see:



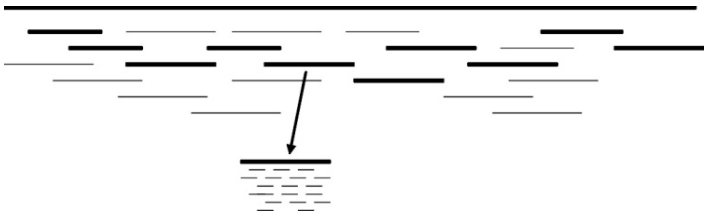
4

**Strategy:**

# BAC-to-BAC sequencing

- … *or* compartmental sequencing
- … *or* hierarchical sequencing

1. Break genome into large fragments, eg Bacterial Artificial Chromosomes (BACs)
2. Order the BACs and choose a ”tiling” of the genome. Requires a *mapping* of the genome!
3. Sequence the BACs



5

**Strategy:**

# Whole-genome shotgun


- All sequencing directly on whole genomes or whole chromosomes — avoids BACs and their mapping
- One huge computational problem instead of many small BAC problems

6

**Strategy:**

## Fosmid pool sequencing

- Like BACs, but with *fosmids,* 40kbp fragments
- Unlike BACs, fosmids are ”shotgun-constructed”
- Pool the fosmids
- Many medium computational problems instead of
  - many small BAC problems, or
  - one big WGS.
- Assemble pool-by-pool ⇒ contigs to be used as *long reads.*

7

**Strategy:**

## Long read sequencing

- Long reads from PacBio or Oxford Nanopore instruments
- Use a long-read assembler (mature technology?)
- High error rate. How correct?
  - Built into assembler, or
  - using Illumina reads

8

**Strategy:**

## Hybrid assembly

- Combine WGS with additional technology
  - Long reads (PacBio)
  - Fosmid pools
- Merging assemblies from different tools

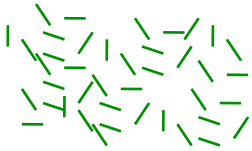
9

**Core problem:**

## Assemble the shotgun pieces

**In:**

A set of reads of unknown orientation



**Out:**

*Ideally:* a genome model

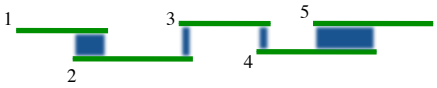
*In practice:*
A set of *contigs*

…and a lot of ”chaff”

10


# Greedy assembly

- While there are sequences with overlap:
  - Find sequences with largest overlap
  - Merge those sequences

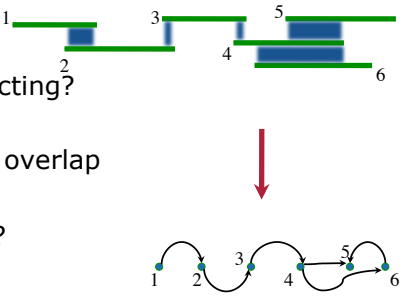



- **Advantage**:
  - Simple
- **Disadvantage**:
  - Early mistakes create bad assemblies
  - A lot of comparisons

12

# Overlap-Layout-Consensus

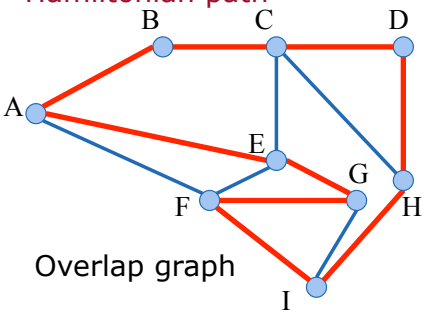
- **Clean your input**
  Remove "vector sequence", low quality, etc
- **Overlap**: What reads are intersecting?
  - Create a node for each read
  - Create directed edge for each overlap
- **Layout**: How combine the reads?
  - Simplify graph
  - Find suitable paths in the graph
- **Consensus:** Derive contigs from layout




13

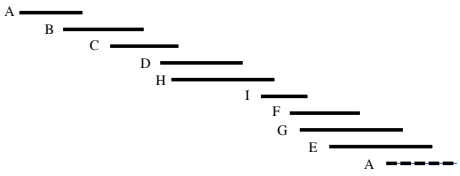
# The layout stage

Hamiltonian path




Overlap graph

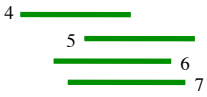
Layout




From http://www.cbcb.umd.edu/research/assembly_primer.shtml

14

# Consensus stage



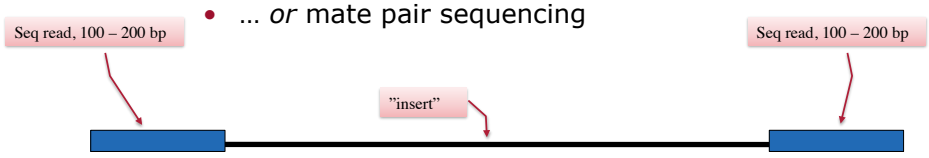

```
Seq4    TTCACACACCCTATACCAATAGTTTTCTGGCTCCTGACCATCAAACTG
Seq5                        TTTTCTGGCTCCTGACCTTCAAA-TGCCTCCATATGACTGTGCTCT
Seq6                TACCAATAGTTTACTGGCTCCTGACCCTCAAACTGCCTCC
Seq7                    ATAGTTTTCTGGCTCCTGACCGTCAAACTGCCTCCATATGA

Cons    TTCACACACCCTATACCAATAGTTTTCTGGCTCCTGACCNTCAAACTGCCTCCATATGACTGTGCTCT
```

15

# Paired ends sequencing

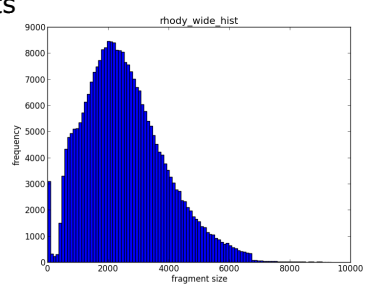
- … *or* mate pair sequencing



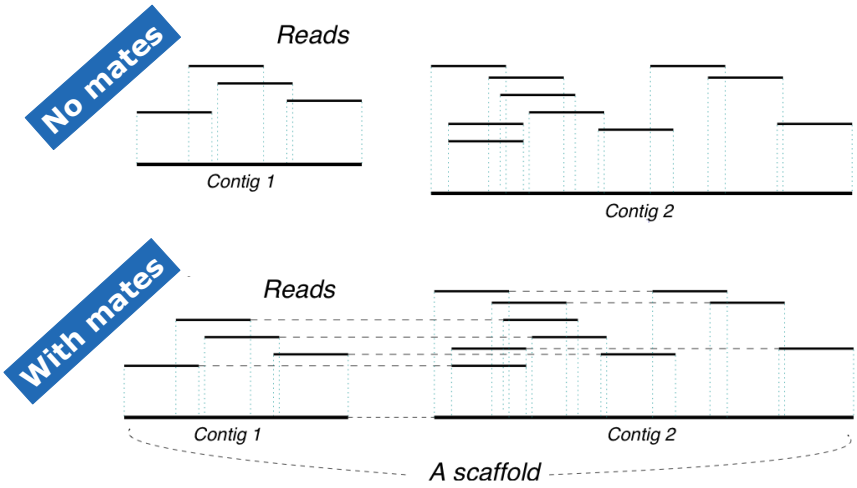

- **Advantage**: adds constraints

**Insert sizes**

- Paired ends: 30 – 700 bp
- Mate pairs: 2 kbp – 10 kbp



# Value of read/mate pairs?



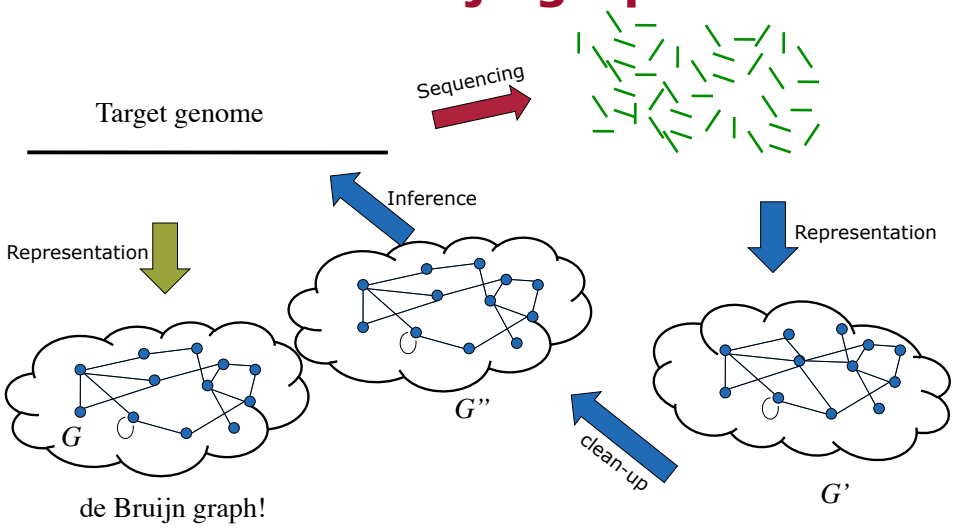

17

# Scaffolds

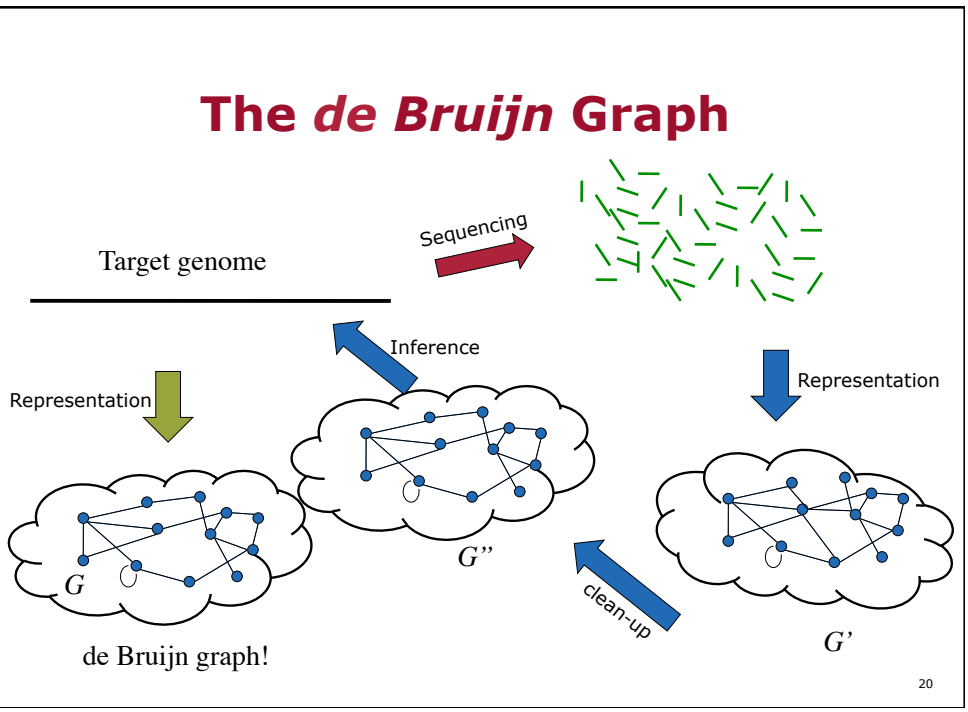
CCCAGTCCATTCTCCCACTGATGTCTGTAACTATATTCATCAATCTCTTT
GATTTCCAAAGCGCATACCATGGCCTCTGAATGTACTTTGCAGCTTGCCC
TTCACACACCCTATACCAATAGTTTTCTGGCTCCTGACCATCAAACTGCC
TCCATATGACTGTGCTCTTGTCTTTCCTTAGTTGCATGGGTGTCATCTTA
TGGGTCACGACCTTCTTAACTGGAACTTTCTTCTTATGGGAGCGATCCCA
TTTCTTCCAACCTCTCAAAAATTCACCCTCTTCAACAATTGACGCCTCCT
CCTTAAGATGCTCAAAATCAAAATAAAACCTAAAATCCTTCCCCTCCTGA
TCCTTCCCATCCGGATTATAATTACCTGCCAAGCATATACTCAAGTCCAT
GACAAATGTCCTGTTCAAATACATAGCCTCCCCCAACCCACACAAGAAAC
TCCACATGTAATGATTCATTCCCCTTGCAATAATCCCCCTCCTCTTGAATAA
TACAAGTACTTCCCTTTTCTAAAGTTCGTTTCTGATCNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNCCTCTCAAACTATCGTGAGTACGGGATTATATTTCTAAAGTAAGTAAA
TTATCCAAGATGGGCCGATCCATTACGAAAAGGAAAGATAATCCACTATT
ATTTCTTAATATAAGGGCAAATTTAAATATATAAAGATGTAATTGTTGTT
GGCGAGTGCCTCTCTTGGTTGTTGAGAGTGAAATTGACAGCAAAGTTGTA
GATTGTGACAGCCAATGTAACTTATTCACAAATTGGCCTGCCAATGGTAC
ATCATGAATCGCTATGCCACATACTTGATTATACCTTCTAAAGTACCTGT
GAATTTTATTTATTTTTTCCATTTTAAAAGTATGTTTTATTTGGAAAAAA
TATCAAATTATTATTTTACTATTATATTTTAATATATTCTTAAATAAAAA
ACACTATTAAAAAATATTATGCACCGCAATAATAAACACATATTAAACAA
ACAGATAATTTTTATATGGCATTTCACTATTGTTGTGGAATAATATCTTT

18

# The Euler approach & the de Bruijn graph

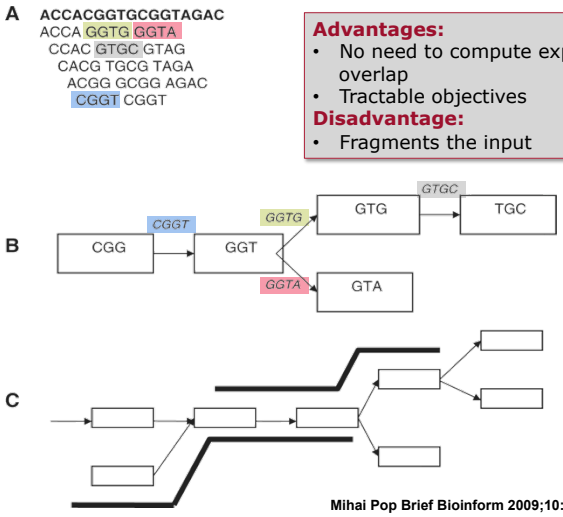



# The *de Bruijn* Graph



# The *de Bruijn* Graph

**(A) k-mer spectrum of a DNA string (bold) for k = 4; (B) Section of the corresponding deBruijn graph.**




**Advantages:**
- No need to compute explicit overlap
- Tractable objectives

**Disadvantage:**
- Fragments the input

Mihai Pop Brief Bioinform 2009;10:354-366



# Genome coverage

- … or read depth
- … or coverage depth
- … or redundancy

How many times is a position sequenced, on average?

- Drosophila:
- H
- D 2.39x
- Mouse lemur: 1.93x

Read-length matters!

**Short-read technology:**
- Panda: > 50x

# How good coverage do you need?

- High coverage good, but expensive
- What if I want at least 99 % of the genome?

## Lander-Waterman model

- **Assumption**: Reads are uniformly distributed
- Coverage $C$
- #times position $i$ sequence: $X_i$
- $X_i$ is Poission distributed

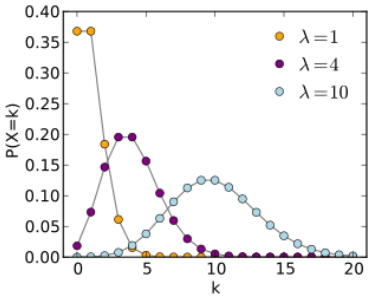
$$\Pr(X_i = k) = C^k e^{-C}/k!$$

23

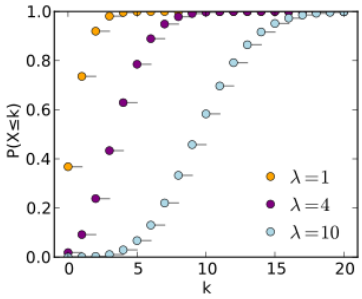
# The Poisson distribution

- You won't ever get perfect coverage!

Density function




Cumulative probability




Pics from Wikipedia

24

## More Lander-Waterman

Require $0 < \theta < 1$ overlap to join reads into a contig.

- Expected number of contigs if $N$ reads: $Ne^{-C(1-\theta)}$

  *Dog: 8x, require e.g. 10% overlap,* $32 \times 10^6$ *reads: 24 000 contigs*

- Expected contig size: $L\frac{e^{C(1-\theta)}-1}{C} + \theta$.

  *Dog, assume* $L = 500$*: contigs are* $\sim$ *83 700 bp*

25

## Lander-Waterman and reality

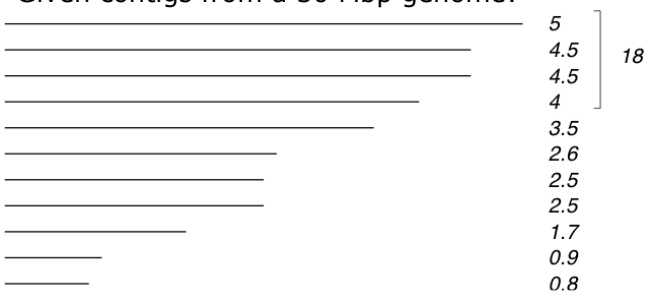
> ”For both a simulated unassisted 2x mouse genome assembly (Margulies et al. 2005) and the assisted 1.9x cat genome assembly of Pontius et al. (2007) euchromatic genome coverage by assembled contigs was only 65%, significantly less than the theoretical Poisson expectation (Lander and Waterman 1988) of 85%.”
>
> *Green, 2007*

- Why this discrepancy?

26

# Quality: N50

- Operational definition:
  1. Sort all contigs by size
  2. Add contig sizes, one by one, towards the smallest
  3. Stop when you have contigs covering half the genome
  4. The length of the last contig is the N50
- Given contigs from a 30 Mbp genome:




N50 is 4 Mbp, because 5+4.5+4.5+ 4 > 30

**N50**: "covering half the assembly"
**NG50**: "covering half the actual genome"
**Scaffold N50**: Looking scaffolds, not contigs

27

# N50 characteristics

- High N50 ⇒ Good contigs ⇒ Good assembly
- Low N50 ⇒ Many small contigs ⇒ Genome badly sequenced ⇒ Bad assembly

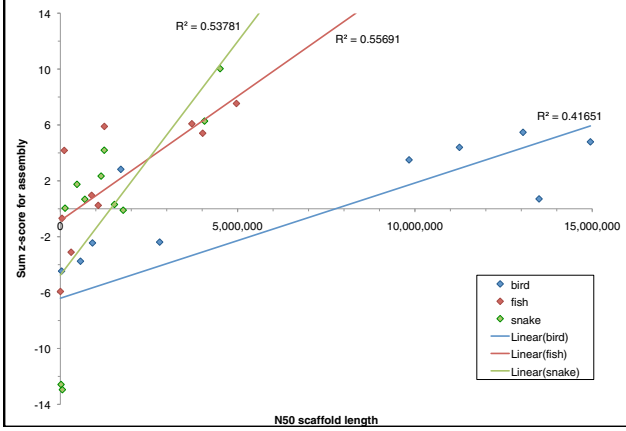
- Bad assembly could have a high N50:

> "The standard of judging assembly quality by size of contigs is questionable. Large contigs may simply reflect overly aggressive joining of contigs, thereby creating larger contigs with mis-assemblies. As a consequence, genome scientists who are not experts at assembly can be completely misled by statistics about contig sizes, and as a result might prefer the 'larger' but incorrect assembly when given a choice."
>
> Salzberg & Yorke, 2005

28

# Utility of N50?

From the "Assemblathon 2" genome assembler assessment (Bradnam *et al.*, Gigascience, 2013):




"[W]e find that N50 remains highly correlated with our overall rankings […]. However, it may be misleading to rely solely on this metric when assessing an assembly's quality."

29

# Quality: E-size

- Definition: the size of a randomly chosen contig
- First appeared 2012
- *My opinion*: reflects fragmentation better

## Example

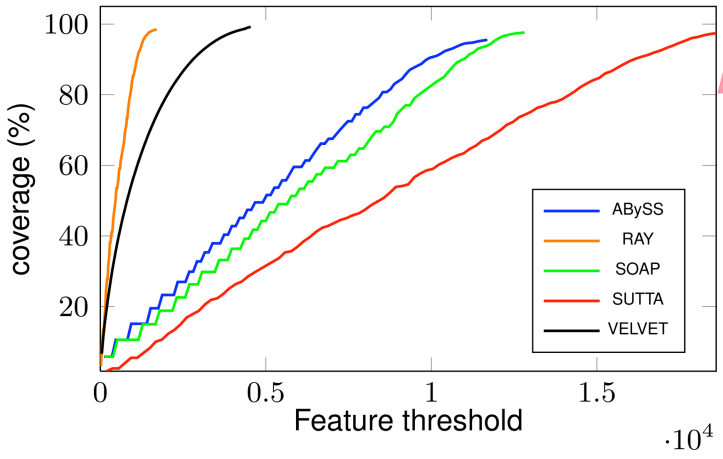
- Assembly *A*:
  Contigs 10 * 100 bp
- Assembly *B*:
  Contigs 499 + 5 * 100 bp

| | N50 | E-size |
|---|---|---|
| A | 100 | 100 |
| B | 100 | 166 |

30

# Feature Response Curve

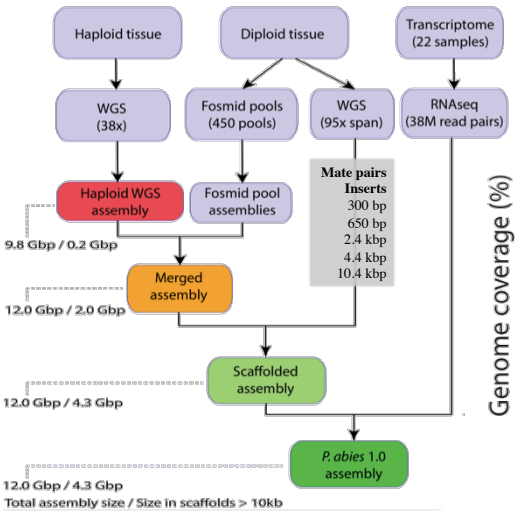
*by* Vezzi, Narzisi, and Mishra, PLoS One, 2012

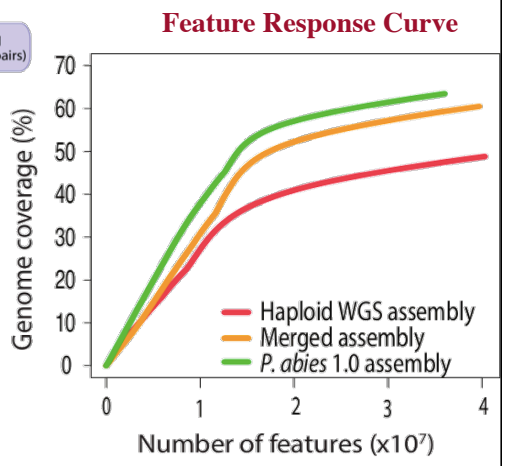



For comparing assemblies on the same data

31

# Norway spruce: assembly process




32

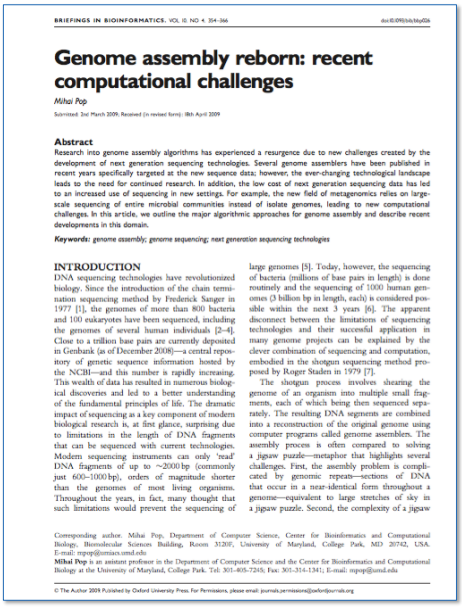
# Assembly resources

- NCBI's Trace Archive
- Lots of assemblers
  - Cap3
  - Phrap
  - Minimus
  - Velvet
  - AllPaths
  - ABySS
  - CABOG
  - MaSuRCA
  - Minia
  - SPAdes
  - … and many more

33

# Mandatory reading

- Mihai Pop's review paper
- Prepares for the quiz



BRIEFINGS IN BIOINFORMATICS. VOL 10. NO 4. 354–366

**Genome assembly reborn: recent computational challenges**

*Mihai Pop*

Submitted: 2nd March 2009; Received (in revised form): 18th April 2009

**Abstract**

Research into genome assembly algorithms has experienced a resurgence due to new challenges created by the development of next generation sequencing technologies. Several genome assemblers have been published in recent years specifically targeted at the new sequence data; however, the ever-changing technological landscape leads to the need for continued research. In addition, the low cost of next generation sequencing data has led to an increased use of sequencing in new settings. For example, the new field of metagenomics relies on large-scale sequencing of entire microbial communities instead of isolate genomes, leading to new computational challenges. In this article, we outline the major algorithmic approaches for genome assembly and describe recent developments in this domain.

*Keywords:* genome assembly; genome sequencing; next generation sequencing technologies

**INTRODUCTION**

DNA sequencing technologies have revolutionized biology. Since the introduction of the chain termination sequencing method by Frederick Sanger in 1977 [1], the genomes of more than 800 bacteria and 100 eukaryotes have been sequenced, including the genomes of several human individuals [2–4]. Close to a trillion base pairs are currently deposited in Genbank (as of December 2008)—a central repository of genetic sequence information hosted by the NCBI—and this number is rapidly increasing. This wealth of data has resulted in numerous biological discoveries and led to a better understanding of the fundamental principles of life. The dramatic impact of sequencing as a key component of modern biological research is, at first glance, surprising due to limitations in the length of DNA fragments that can be sequenced with current technologies. Modern sequencing instruments can only 'read' DNA fragments of up to ~2000 bp (commonly just 600–1000 bp), orders of magnitude shorter than the genomes of most living organisms. Throughout the years, in fact, many thought that such limitations would prevent the sequencing of large genomes [5]. Today, however, the sequencing of bacteria (millions of base pairs in length) is done routinely and the sequencing of 1000 human genomes (3 billion bp in length, each) is considered possible within the next 3 years [6]. The apparent disconnect between the limitations of sequencing technologies and their successful application in many genome projects can be explained by the clever combination of sequencing and computation, embodied in the shotgun sequencing method proposed by Roger Staden in 1979 [7].

The shotgun process involves shearing the genome of an organism into multiple small fragments, each of which being then sequenced separately. The resulting DNA segments are combined into a reconstruction of the original genome using computer programs called genome assemblers. The assembly process is often compared to solving a jigsaw puzzle—metaphor that highlights several challenges. First, the assembly problem is complicated by genomic repeats—sections of DNA that occur in a near-identical form throughout a genome—equivalent to large stretches of sky in a jigsaw puzzle. Second, the complexity of a jigsaw

Corresponding author. Mihai Pop, Department of Computer Science, Center for Bioinformatics and Computational Biology, Biomolecular Sciences Building, Room 3120F, University of Maryland, College Park, MD 20742, USA. E-mail: mpop@umiacs.umd.edu

**Mihai Pop** is an assistant professor in the Department of Computer Science and the Center for Bioinformatics and Computational Biology at the University of Maryland, College Park. Tel: 301-405-7265; Fax: 301-314-1341; E-mail: mpop@umd.edu



34

# Student presentations

Half of next lecture!
(Aim for 10 min presentations).

1. Assemblathon 2
   - Based on Bradnam *et al*, GigaScience 2013
   - What can we learn from Assemblathon 2?
2. Assembly comparison/evaluation
   - Based on Vezzi *et al* " Feature-by-Feature – Evaluating *De Novo* Sequence Assembly", PLoS ONE, 2012
   - What "features" are they using?
   - How do they compute the graphs?
   - Any limitations?

- **Mandatory:**
  - Browse paper!
  - Email me a question regarding the paper!
    - At the latest the evening before the presentations

35