Computational assembly for prokaryotic sequencing projects

Lee Katz, Ph.D.
Bioinformatician, Enteric Diseases Laboratory Branch
January 21, 2015

Disclaimers
The findings and conclusions in this presentation have not been formally disseminated by the Centers for Disease Control and Prevention and should not be construed to represent any agency determination or policy. The findings and conclusions in this [report/presentation] are those of the author(s) and do not necessarily represent the official position of CDC.
Partners in Public Health
Graduated Oct 2010
Public Health Impact Award 2011: Helping to rid Africa of meningitis A
CDC 2010 - present
2011 to present

ENTERIC DISEASES LABORATORY BRANCH
Thanks to John Besser for letting me borrow this slide
Thanks to John Besser for letting me borrow this slide.
Lee Katz, Present

- Currently in the National Enteric Reference Laboratory
- *Vibrio, Campylobacter, Escherichia, Shigella, Yersinia, Salmonella*
- Current major projects
  - *Listeria* WGS for outbreak analysis
  - Spread of *Vibrio cholerae* into Haiti and into surrounding areas
  - High quality SNP analysis software (Lyve-SET)
  - Gulf coast *V. cholerae*
  - etc.
One of my projects is #2 on CDC’s list of accomplishments for 2013!

http://www.cdc.gov/features/endofyear/ (2014 accomplishments have not been published yet)
Outline

• Illumina sequencing

• Reads
  – Quality control (Q/C)
    • Read metrics
  – Read-cleaning

• Assembly
  – Algorithms
  – Assembly metrics
Prokaryotic Sequencing Projects

Stages
- Sequencing
- Assembly
- Feature prediction
- Functional annotation
- ...analysis...
- Display (Genome Browser)

Examples
- *Haemophilus influenzae*
- *Neisseria meningitidis*
- *Bordetella bronchiseptica*
- *Vibrio cholerae*
- *Listeria monocytogenes*


Kislyuk et al. (2010) “A computational genomics pipeline for prokaryotic sequencing projects” *Bioinformatics* 26:15
Illumina sequencing (2\textsuperscript{nd} Gen)

• I think I’ve heard you have sequences from the following machines (?)
  – GAIIx
  – HiSeq
Illumina sequencing video

- http://www.youtube.com/watch?v=womKfikWlxM
Q/C + cleaning

READS
Q/C

• You need to know if your data are good!
• Example software that give you read metrics
  – FastQC
  – Computational Genomics Pipeline (CG-Pipeline)
    • run_assembly_readMetrics.pl
  – AMOS
    • FastqQC (I promise it’s a different program than FastQC)
Quality Control

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
Quality Control bioinformatics

Sequence content across all bases

- %T
- %C
- %A
- %G

Position in read (bp)
The CG-Pipeline way

run_assembly_readMetrics.pl

<table>
<thead>
<tr>
<th>File</th>
<th>avgReadLength</th>
<th>totalBases</th>
<th>minReadLength</th>
<th>maxReadLength</th>
<th>avgQuality</th>
</tr>
</thead>
<tbody>
<tr>
<td>tmp.fastq</td>
<td>80.00</td>
<td>177777760</td>
<td>80</td>
<td>80</td>
<td>35.39</td>
</tr>
</tbody>
</table>
READ CLEANING
Read cleaning with CG-Pipeline
(not validated; please use with caution)

http://sourceforge.net/projects/cg-pipeline/
https://github.com/lskatz/CG-Pipeline

Graphs made with FastqQC (AMOS)
1. Trimming low-qual ends

run_assembly_trimLowQualEnds.pl

1A. %ACGT

1B. Phred

http://sourceforge.net/projects/cg-pipeline/
https://github.com/lskatz/CG-Pipeline

Graphs made with FastqQC (AMOS)
2a. Removing duplicate reads
2b. Sometimes: downsampling

run_assembly_removeDuplicateReads.pl

Trimmed reads

http://sourceforge.net/projects/cg-pipeline/
https://github.com/lskatz/CG-Pipeline
3. Trimming and filtering

run_assembly_trimClean.pl

3A. trimming

Min length

Min avg. quality

3B. filtering

Min length

Min avg. quality

http://sourceforge.net/projects/cg-pipeline/
https://github.com/lskatz/CG-Pipeline
More read cleaning and correction

- **Software**
  - Fastx toolkit [http://hannonlab.cshl.edu/fastx_toolkit](http://hannonlab.cshl.edu/fastx_toolkit)
  - AMOS amos.sourceforge.net
  - Sickle [https://github.com/najoshi/sickle](https://github.com/najoshi/sickle)
  - Quake [http://www.cbcb.umd.edu/software/quake](http://www.cbcb.umd.edu/software/quake)
  - BayesHammer [http://bioinf.spbau.ru/spades/bayeshammer](http://bioinf.spbau.ru/spades/bayeshammer)
  - … and more is out there!

- **Evaluation**
  - Fabbro et al 2013, “An extensive evaluation of read trimming effects on Illumina NGS data analysis”
ASSEMBLY CONCEPTS
Assembly

• Overlaps between reads or between nodes in a graph
• Generate **contigs** (contiguous sequences)
• Generate **scaffolds**
Further reading: Zhang et al 2011 *Plos One*; Miller et al 2011 *Genomics*
Greedy extension

Given a set of sequence fragments the object is to find the **shortest common supersequence**.

- Calculate pairwise alignments of all fragments.
- Choose two fragments with the largest overlap.
- Merge chosen fragments.
- Repeat step 2 and 3 until only one fragment is left.

The result is a suboptimal solution to the problem. [citation needed]
Overlap-layout-consensus

- Great at longer, fewer, imperfect reads, e.g., Ion Torrent, 454, PacBio
Derive consensus sequence

```
TAGATTACACAGATTACTGAA-TTGATGGC GTAA-CTA
TAGATTACACAGATTACTGACCTTGATGGC GTAAA-CTA
TAG- TTACACAGATTATTTGACTTTCATGGC GTAA-CTA
TAGATTACACAGATTACTGACCTTGATGGC GTAA-CTA
TAGATTACACAGATTACTGACCTTGATGGG GTAA-CTA
TAGATTACACAGATTACTGACCTTGATGGC GTAA-CTA
```

Derive each consensus base by weighted voting

De bruijn (kmer)

- Great at shorter, numerous, less error-prone reads, e.g., Illumina

http://www.homolog.us/blogs/blog/2012/06/17/an-intuitive-explanation-for-running-de-bruijn-assembler-with-varying-k-mer-sizes/
Another way to look at de Bruijn assembly

Unipath graph of the 1.8-Mb genome of *C. jejuni*

Possible paths:
- ABCDBCEFCEG
- ABCEFCDBCEG

Another way to look at de bruijn assembly

Unipath graph of the 1.8-Mb genome of *C. jejuni*

Possible paths:
- ABCDBCEFCEG
- ABCEFDDBCEG

Recap of assembly

reads

Paired end reads

contigs

Scaffold
Further reading

- Ekblom and Wolf 2013, A field guide to whole-genome sequencing, assembly, and annotation. *Evolutionary applications*
- I really like the glossary and introductions in this paper
ASSEMBLY

Algorithms + metrics
Some assemblers to try

• Open source
  – SPAdes
  – Velvet
  – Mira
  – Edena
  – CG-Pipeline (uses SPAdes + Velvet)

• Proprietary
  – Geneious
  – CLC Genomics Workbench
  – Lasergene
  – Bionumerics

• These are some ideas but it is up to you, the class, to do a more thorough review!
Comparisons of Illumina assemblers

• Zhang et al 2011 “A practical comparison of de novo genome assembly software tools for next-generation sequencing technologies”
• Genome Assembly Gold-standard Evaluations (GAGE) - http://gage.cbcb.umd.edu/
• Lin et al 2011, “Comparative studies of de novo assembly tools for next-generation sequencing technologies”
• Don’t forget to look at many newer assemblers including SPAdes (currently my favorite)
CG-Pipeline way for Illumina

run_assembly reads.fastq.gz -o assembly.fasta -e 2100000 --numcpus 8
# for a 2.1 MB genome
Assembler comparison (further reading)

- GAGE-B: http://ccb.jhu.edu/gage_b/
- Assemblathon 2: http://www.gigasciencejournal.com/content/2/1/10
- GABenchToB: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0107014#pone-0107014-g003
ASSEMBLY DIFFICULTIES
One problem: randomly low coverage (Lander-Waterman)

• Assuming random distribution of reads and ignoring repeat resolution issues,
  • \( G \) = genome length
  • \( L \) = length of a single read
  • \( N \) = number of reads sequenced
  • \( T \) = minimum overlap to align the reads together
• Then overall coverage is \( C = \frac{LN}{G} \)

• Coverage for any given base obeys the Poisson distribution:
  \[
  P(C_x = y) = \frac{C^y e^{-C}}{C_y!}
  \]
• The number of gaps (bases with 0 coverage) is:
  \[
  G = Ne^{\left[1 - \frac{T}{L}\right]}
  \]

http://www.cmb.usc.edu/papers/msw_papers/msw-081.pdf
Major Problem: repeat elements

Unipath graph of the 1.8-Mb genome of *C. jejuni*

Possible paths:
- ABCDBCEFCEG
- ABCEFCDBCEG
Another way to look at de Bruijn assembly

Unipath graph of the 1.8-Mb genome of *C. jejuni*

Possible paths:
ABCDBCEFCEG
ABCEFCDBCEG
A quick note on reference assembly

• It is possible to map reads to a reference genome using short-read mappers, and then deriving a consensus sequence

• What makes a good reference genome?
  – Must be a closely related genome (same species or same serogroup or same sequence type…)
  – Relatively few SNPs (>90 or 95% identity to your genome)
  – Relatively few rearrangements
Software to try

• Mapping software
  – Smalt
  – BWA
  – Bowtie/Bowtie2

• SNP-calling software
  – Samtools mpileup or bcftools
  – FreeBayes
  – GATK
  – Varscan
  – SolSNP

• Must know how to use samtools for this route
Reference assembly notes

• To my knowledge no paper exists that compares reference assemblers (could be wrong)

• Assembly could be biased
  – Miss genomic islands
  – Ends of contigs might be tapered

• Good practice for reference assembly: perform *de novo* assembly on unused reads just in case you missed something
ASSEMBLY EVALUATION
# Assembly Metrics

How do you tell if your assembly is good?

<table>
<thead>
<tr>
<th>Metric</th>
<th>description</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assembly Length</td>
<td>The size of the concatenated assembly</td>
<td></td>
</tr>
<tr>
<td>Number of Contigs</td>
<td>The count of contigs</td>
<td></td>
</tr>
<tr>
<td>N50</td>
<td>The size of the contig at where half the genome is located in size &gt;N50 and half is located in size &lt;N50</td>
<td></td>
</tr>
<tr>
<td>Longest Contig</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average contig length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kmer21</td>
<td>Frequency of kmers with k=21</td>
<td><a href="http://www.homolog.us/blogs/blog/2012/06/26/what-is-wrong-with-n50-how-can-we-make-it-better-part-ii/">http://www.homolog.us/blogs/blog/2012/06/26/what-is-wrong-with-n50-how-can-we-make-it-better-part-ii/</a></td>
</tr>
<tr>
<td>GC-content</td>
<td>Percentage of the genome that is either G or C</td>
<td></td>
</tr>
<tr>
<td>Assembly score</td>
<td>Log(N50/numberOfContigs)</td>
<td>CG-Pipeline/Lee Katz</td>
</tr>
</tbody>
</table>
QUAST: [http://bioinf.spbau.ru/quast](http://bioinf.spbau.ru/quast)

**QUAST**

**QUality ASsesment Tool for Genome Assemblies** by Algorithmic Biology Lab

QUAST evaluates genome assemblies by computing various metrics, including:

- **N50**, length of the shortest contig from all that cover 50% of all assembly,
- **NG50**, where the reference genome is being covered,
- **NA50** and **NGA50**, where aligned blocks instead of contigs are taken, misassemblies, misassembled and unaligned contigs or contigs bases, genes and operons covered.

Builds convenient plots for different metrics:

- cumulative contigs length,
- all kinds of N-metrics,
- genes and operons covered,
- GC content.

**Report example**

More details are on the project page and in Gurevich et al (2013), Bioinformatics. Supplementary material for the paper is available [here](http://bioinf.spbau.ru/quast).

**Download console tool**

For installation details and usage instructions, please read the manual.

We will be thankful if you help us make QUAST better by sending your comments, bug reports, and suggestions to quast.support@bioinf.spbau.ru.

**Quality Assessment**

- **Contigs**
  - Add files

  *drop files here*

- **Skip contigs shorter than** 500 bp

- **Scaffolds**
  - Adds split assemblies (continuous fragments of N's longer than 10 bp.)

- **Find genes**

- **Resolve** (find genes with GenemarkS, process circular chromosomes)
QUAST results

Contigs are ordered from largest (contig #1) to smallest.

You can also see a comparison of more single-cell assemblers on this data set.

There an assessment of these assemblers working with an isolate data set as well.
The CG-Pipeline way

$ run_assembly_metrics.pl assembly.fasta|column -t

<table>
<thead>
<tr>
<th>File</th>
<th>genomeLength</th>
<th>N50</th>
<th>numContigs</th>
<th>assemblyScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>assembly.fasta</td>
<td>2992976</td>
<td>2992976</td>
<td>1</td>
<td>14.9117787680924</td>
</tr>
</tbody>
</table>
AMOS

• amosvalidate
• http://amos.sourceforge.net
  – (build this from source control to avoid bugs)
Reminder: your goal is to make the very best assembly, and here are ways to go above and beyond.

POST ASSEMBLY METHODS
Merging assemblies

• You can take the output of two assemblers and merge them!
• However, this could lead to misassemblies and so you might not want to risk combining two assemblies.
• Some software:
  – NGS-GAM
  – Minimus
  – Reconciliator
Verifying the assembly

• View the pileups and see if you agree with base calls and synteny
  – Hawkeye (AMOS)
  – IGV viewer - very smooth GUI
  – Artemis viewer - many bells and whistles
  – Tablet viewer - no bells and whistles (lightweight)
  – Samtools tview (command line interface)

• Look for bad base calls; look for indications of bad structure/synteny
Sort contigs

• Compare to other genomes; sort contigs
  – Must have a closely related reference genome (See: reference assembly slides)
  – MAUVE Contig Mover
  – Mummer (mummerplot)
  – Abacas
  – CONTIGuator

• A sorted genome assembly will be easier to analyze and there will probably be fewer mistakes in an analysis

• **Result:** fasta file with sorted and unsorted contigs
Making scaffolds

- Babmus2 (AMOS)
- IMAGE
- SOPRA
- SSPACE

- Use paired ends to give an idea if two contigs belong together

- Result: sorted scaffolds with Ns (unknown bases)
Close gaps in scaffolds

- IMAGE
- GapFiller
- FinIS
- SOAPdenovo2 GapCloser

- Basic idea: map reads back onto your assembly and make base calls where gaps are

- **Result**: scaffold with fewer or no Ns
Acknowledgements

• Every single compgenomics class since 2008
• Many slides were borrowed from my 2014 talk
• For letting me off work, my supervisor Cheryl Tarr
• Many, many others who I work with on a daily basis
“Geny”

Credit to Gladys Gonzalez-Aviles in PulseNet, CDC
Based on a figure in Katz et al 2013 MBio
Questions?

Lee Katz
lkatz@cdc.gov

The findings and conclusions in this presentation have not been formally disseminated by the Centers for Disease Control and Prevention and should not be construed to represent any agency determination or policy.
The findings and conclusions in this presentation are those of the author and do not necessarily represent the official position of CDC.