de novo assembly

Simon Rasmussen
27626: Next Generation Sequencing analysis
CBS - DTU
Generalized NGS analysis

- Raw reads
- Pre-processing
- Assembly: Alignment / de novo
- Application specific: Variant calling, count matrix, ...
- Compare samples / methods
- Answer?
What is *de novo* assembly?

Merge small DNA fragments together so they form previously unknown sequences.

Merge millions of reads together so they form previously unknown sequences.
**de novo assembly**

- Assemble reads into longer fragments
- Find overlap between reads
- Many approaches
Let's try to assemble some reads!

- Rules:
  - a minimum of 7-bp overlap
  - overlap must not include any N bases
  - same orientation so that the sequence can be read left to right
  - there may be 1-bp differences
  - simplified - no double stranded DNA

..NNNGGACTATGATTTCG
   |||||
   TGATTCGAGGCTAANN..

..NNNNNNNNCGAGGACTATGATT
   |||||
   ATGATTCGAGGCTAANN..

..NNNNNNNNCGATTCTGATCCGA
   |||||
   GTCCTCGATTCTGNNNNNNNN..

..NNNNNNNNCGCTACTGATCCGA
   |||||
   GTCCTCGATTCTGNNNNNNNNN..
Which are valid?

..NNNNCGGACTATGATT

..NNNNNGGACTATGATTCG

..NNNNGGACTATGATTCG

..NNNNCGGACTATGATT

..NNNNNNNNCGATTCTGATCCGA

..NNNNNNNNCGAATTCTGATCCGA

..NNNNNNNNCGATTCTGATCCGA

..NNNNNNNNCGATTCTGATCCGA

..NNNNNNNNCGATTCTGATCCGA

..NNNNNNNNCGATTCTGATCCGA

..NNNNNNNNCGATTCTGATCCGA

..NNNNNNNNCGATTCTGATCCGA

..NNNNNNNNNNCGATTCTGATCCGA

..NNNNNNNNNNCGATTCTGATCCGA

..NNNNNNNNNNCGATTCTGATCCGA
Which are valid?

..NNNCGGACTATGATT

..NNNNGGACTATGATTTCG

ATGATTTCGAGGCTAANN..

..NNNNNNNNCGATTCTGATCCGA

GTCCTCGATTCTGNNNNNNNN..

..NNNNNNNNNCGCTACTGATCCGA

GTCCTCGATTCTGNNNNNNNN..

..NNNNNNNNNCGATTTCTGATCCGA

GTCCTCGATTCTNNNNNNNNN..
Which approaches?

• Greedy ("Simple" approach)
• Overlap-Layout-Consensus (OLC)
• de Bruijn graphs
Simple approach - Greedy

- Pseudo code:
  1. Pairwise alignment of all reads
  2. Identify fragments that have largest overlap
  3. Merge these
  4. Repeat until all overlaps are used

- Can only resolve repeats smaller than read length
- High computational cost with increasing no. reads
Reads > Contigs > Scaffolds

• Overlap Layout Consensus and de Bruijn use a similar general approach.

1. Try to correct sequence errors in reads with high coverage
2. Assemble reads to contiguous sequence fragments “contigs”
3. Identify repeat contigs
4. Combine and order contigs to “scaffolds”, with gaps representing regions of uncertainty
Overlap-Layout-Consensus

- Create overlap graph by all-vs-all alignment (Overlap)
- Build graph where each node is a read, edges are overlaps between reads (Layout)

R₁: GACCTACA
R₂: ACCTACAA
R₃: CCTACAAG
R₄: CTACAAGT
A: TACAAGTT
B: ACAAGTTA
C: CAAGTTAG
X: TACAAGTC
Y: ACAAGTCC
Z: CAAGTCCG
Overlap-Layout-Consensus

- Create consensus sequence
- We need to use graph theory to solve the graph
- *Walk the Hamiltonian path*
- Eg. visit each node *exactly once*

Imagine trying to solve this for a graph of hundred of thousands of nodes (=reads) - this is an **NP-complete problem**
Overlap-Layout-Consensus

- Relative high assembly error rate
- Very hard for high coverage and large genomes (all-vs-all)
- With short read lengths, hard to resolve repeats
- Good for small genomes and large read lengths (454, Ion Torrent, Pac Bio)
de Bruijn graph

- Directed graph of overlapping items (here DNA sequences)
- Instead of comparing reads, decompose reads into $k$-mers
  - Graph is created by mapping the $k$-mers to the graph
  - Each $k$-mer only exists once in the graph
  - Problem is reduced to walking Eulerian path (visiting each edge once) - this is a solve-able problem
Drawbacks ...

•Lots of RAM required (1-1000 GB !)

•Optimal $k$ can not be identified *a priori*, must be experimentally tested for each dataset

•small $k$: very complex graph, large $k$: limited overlap in low coverage areas

•Iterative approach to find best assembly
How is the graph constructed?

- Same 10 reads, extract $k$-mers from reads and map onto graph, $k = 3$:

\[ R_1: \text{GACCTACA} \]

No alignment is used!

Different assemblers use different modifications of the de Bruijn graphs.
**Complicated graphs**

Large genomes with many repeats/errors creates very large graphs
Create the *de Bruijn* graph of this genome using $k=3$

AAGACTCCGACTGGGACTTTT
A de Bruijn graph of a sequence
After building: Simplify

Clip tips
(seq err, end)

Pinch bubbles
(seq err, middle, SNP)

Remove low cov. links
Create contigs and scaffolds

Cut graph at repeat boundaries to create contigs

Use paired end information to resolve repeats and combine to scaffolds

Fill potential gaps using PE reads

The assembly is done
Iterate parameters

- Re-run with different k-sizes, find optimum
- Compare assembly statistics such as, assembly length, N50, no. contigs

- Assembly refinement
  - Break contigs not supported by PE/MP reads
  - Analyze assembly using REAPR or QUAST
Successful de novo assembly

- Success is a factor of:
  - Genome size, genomic repeats(!), ploidy
  - High coverage, long read lengths, PE/MP libraries

Repeats in E. coli
Improving *de novo* assemblies

- Paired end & Mate pair for long range continuity
- Hybrid approaches (combine Illumina with PacBio/Oxford Nanopore)
- Synthetic long reads: Illumina Synthetic Reads (Moleculo) or 10X Genomics
- Hi-C contact maps
Two bacterial genomes *de Bruijn* graphs

**Few repeats**

**“more” repeats**

Flicek & Birney, Nat.Methods 2009

Zerbino, 2009
N50: Assembly quality

N50: What is the smallest piece in the largest half of the assembly?

- Calculate sum of assembly
- Order contigs by size
- Sum contigs starting by largest
- When half the sum is reached, N50 is the length of the contig
N50 example

5 scaffolds, calculate N50:

200kb
180kb
150kb
125kb
35kb

Sum: 200 + 180 + 150 + 125 + 35 = 690kb
Half: 630 / 2 = 395kb

200kb + 180kb = 380kb

Start adding: 380kb + 150kb = 530kb

530kb > 395kb => N50: 150kb
Some assemblers

- OLC: Newbler, (the only one I really use)
- de Bruijn: Allpaths-LG, SPAdes, Velvet, SOAPdenovo, Megahit
- other: MIRA, SGA