

de novo assembly & *k*-mers

Rayan Chikhi

CNRS

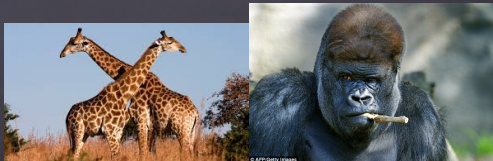
Workshop on Genomics - Cesky Krumlov
January 2018

YOUR INSTRUCTOR IS..

- Junior CNRS bioinformatics researcher in Lille, France
- Postdoc: Penn State, USA; PhD: ENS Rennes, France
- Computer Science background

Research:

- Software and methods for *de novo* assembly:
 - ▶ Minia
 - ▶ KmerGenie
 - ▶ BCALM
- Applications



@RayanChikhi on Twitter
<http://rayan.chikhi.name>

COURSE STRUCTURE

- Short intro
- Basic definitions
- Fundamentals: **why** assemblies are as they are
- Software
- Metrics: methods for **evaluation**
- Visualization: see pretty assembly **graphs**
- RNA-Seq & metagenome: major differences
- Reference-free variant detection
- In practice: best practices ; multi-k ; scaffolding

QUESTIONS TO THE AUDIENCE

- Already have data to assemble?
- Plans to sequence *de novo*?
- RNA-Seq?
- Metagenome?
- PacBio/Nanopore reference-free?

WHAT'S ASSEMBLY?

genome
not known

reads
*overlapping
substrings
that cover
the genome
redundantly*



assembly
*what we think
the genome is*



“A set of sequences which best approximate the original sequenced material.”

Example uses of genome assembly

- Generate a reference genome
- Alternative method of SNP discovery (even if you have a reference)
 - Mostly for small, haploid genomes
 - Provides better diversity calling for small indels and particularly difficult-to-align regions
- Discover structural variants
 - *De novo* assembly is the only way to get the sequence of a novel insertion
 - Complex structural variants can be more easily discovered through *de novo* assembly than read alignment to a pre-existing reference

k-mers:

- * reference-free discovery of variations
- * comparisons between datasets
- * QC
- * data search



PLAN

Fundamentals

Basics

Short Exercise

Some useful assembly theory

Graphs

Contigs construction

Exercise

RNA-seq and metagenomes

Visualizing and evaluating assemblies

Bandage

Reference-free metrics

Exercise

One small other thing you can do with k -mers

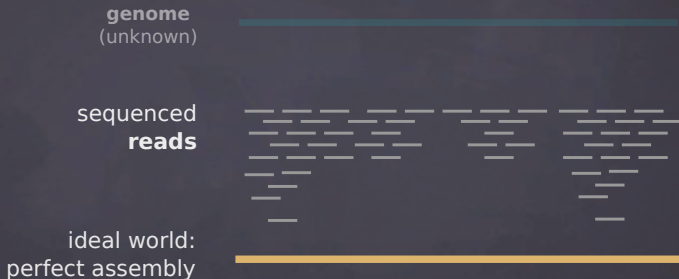
Assembly in practice

Exercise

BASIC EXPECTATIONS

An assembly generally is:

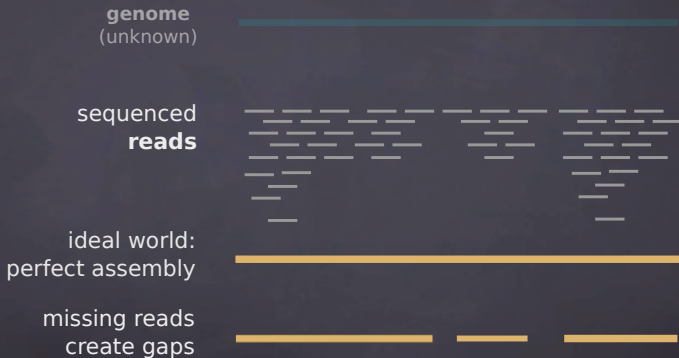
- smaller than the reference,
- fragmented



BASIC EXPECTATIONS

An assembly generally is:

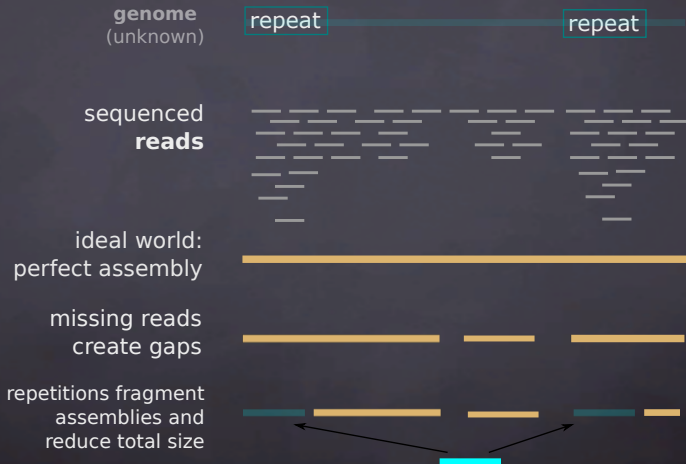
- smaller than the reference,
- fragmented



BASIC EXPECTATIONS

An assembly generally is:

- smaller than the reference,
- fragmented



Some vocabulary:

String What computer scientists mean for a sequence or a part of it

Read Any sequence that comes out of the sequencer

Paired read forward $read_1$, gap ≤ 500 bp, reverse $read_2$

Mate-pair reverse $read_1$, gap ≥ 1 kbp, forward $read_2$

Single read Unpaired read

Contig gap-less assembled sequence

Scaffold sequence which may contain gaps (N)

OLDSCHOOL ASSEMBLY



Greedy algorithms: won't be covered here

OVERLAPS

What does it mean for two strings to overlap?

→ a suffix of the first string equals (or is very similar to) a prefix of the other string.

Exact overlaps:

- | | |
|-----------|---|
| 1: ACTGCT | read 1 overlaps with read 2 and also with read 3. |
| 2: CTGCT | read 2 overlaps with read 3. |
| 3: TGCTAA | |

Inexact overlaps (here, allowing for ≤ 1 mismatch):

- | | |
|-----------|---|
| 1: ACTGCT | read 1 overlaps with read 2 with 1 mismatch.
(read 1 would overlap with read 3 but with 2 mismatches.) |
| 2: CTACT | |
| 3: ACGAA | read 2 overlaps with read 3 with 1 mismatch. |

THE FINE PRINT

TACGTG
GCGTCA

This isn't an exact overlap, even if CGT is common.

TACT
ACTG

AC is not considered an overlap, only ACT is.



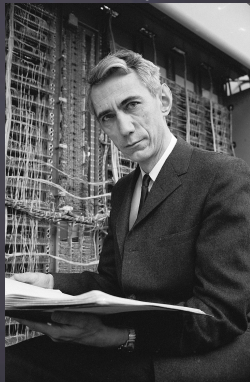
k -MERS

k -mer Any sequence of length k

N.G. de Bruijn (1946),
de Bruijn sequences ¹



C. Shannon (1948),
information theory ²



¹construct a short text that contains all k -mers exactly once

²improve communication by predicting what character is likely to follow a given k -mer

k -MER HISTOGRAMS

- x axis: *abundance*
- y axis: number of k -mers having abundance x (i.e. seen x times)

Example reads dataset:

ACTCA

GTCA

3-mers:

ACT

CTC

TCA

GTC

TCA

Abundance of distinct 3-mers:

ACT: 1

CTC: 1

TCA: 2

GTC: 1

3-mer histogram:

x	y
-----	-----

1	3
---	---

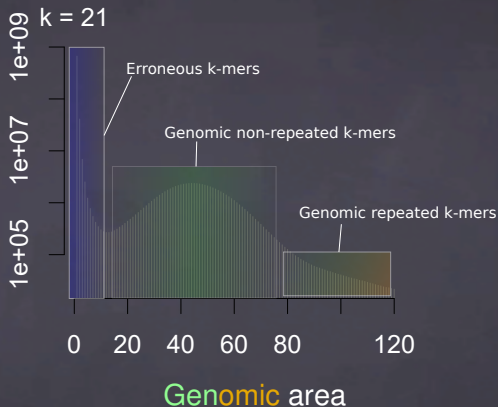
2	1
---	---

3	0
---	---

4	0
---	---

Tools: DSK, Jellyfish, KMC, KmerGenie

Chr 14 (≈ 88 Mbp) GAGE dataset; histogram $k = 21$



\approx

number of distinct k -mers covering the genome

\approx

size of the assembly

EXERCISE

Here is a set of reads:

```
TACAGT
  CAGTC
    AGTCA
      CAGA
```

1. How many k -mers are in these reads (including duplicates), for $k = 3$?
2. How many *distinct* k -mers are in these reads?
 - ▶ (i) for $k = 2$
 - ▶ (ii) for $k = 3$
 - ▶ (iii) for $k = 5$
3. How many distinct pair-wise overlaps of length ≥ 3 are there between the reads?
4. Pretend these reads come from the genome TACAGTCAGA. What is the largest k such that the set of distinct k -mers in the genome is exactly the set of distinct k -mers in the reads above?

EXERCISE (SOLUTION)

Here is a set of reads:

```
TACAGT
  CAGTC
    AGTCA
      CAGA
```

1. How many k -mers are in these reads (including duplicates), for $k = 3$? **12**
2. How many *distinct* k -mers are in these reads?
 - ▶ (i) for $k = 2$: **7**
 - ▶ (ii) for $k = 3$: **7**
 - ▶ (iii) for $k = 5$: **4**
3. How many distinct pair-wise overlaps of length ≥ 3 are there between the reads? : **3**
4. Pretend these reads come from the genome TACAGTCAGA. What is the largest k such that the set of distinct k -mers in the genome is exactly the set of distinct k -mers in the reads above? **3; for $k=4$, TCAG does not appear in the reads**

PLAN

Fundamentals

Basics

Short Exercise

Some useful assembly theory

Graphs

Contigs construction

Exercise

RNA-seq and metagenomes

Visualizing and evaluating assemblies

Bandage

Reference-free metrics

Exercise

One small other thing you can do with k -mers

Assembly in practice

Exercise

GRAPHS

A **graph** is a set of nodes and a set of edges (directed or not).



GRAPHS FOR SEQUENCING DATA

Overlaps between reads is the fundamental information used to assemble.

Graphs represent these overlaps.

Two different types of graphs for sequencing data:

- de Bruijn (DB) graphs for Illumina, 10X data
- string graphs for PacBio/Nanopore data

Knowledge of these graphs, useful for:

- assembly **parameters**
- type of regions **well or badly** assembled
- repetition **over-collapsing**
- selective **heterozygosity** collapsing

OVERLAP GRAPHS

*This is going to be fundamental for **PacBio/Nanopore** data.*

1. **Nodes** = reads
2. **Edges** = overlaps between two reads

OVERLAP GRAPHS

*This is going to be fundamental for **PacBio/Nanopore** data.*

1. **Nodes** = reads
2. **Edges** = overlaps between two reads

In this example, let's say that an overlap needs to be:

- exact
- over at least 3 characters,

Reads:

ACTGCT

CTGCT (overlap of length 5)

GCTAA (overlap of length 3)

Graph:



STRING GRAPHS

A **string graph** is obtained from an overlap graph by removing redundancy:

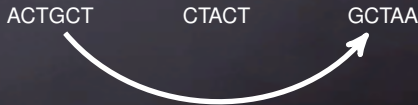
- redundant reads (those fully contained in another read)
- transitively redundant edges (if $a \rightarrow c$ and $a \rightarrow b \rightarrow c$, then remove $a \rightarrow c$)

Two examples:

ACTGCT
CTGCT (overlap length 5)
GCTAA (overlap length 3)

ACTGCT \longrightarrow GCTAA

ACTGCT
CTACT
GCTAA



STRING GRAPHS

A **string graph** is obtained from an overlap graph by removing redundancy:

- redundant reads (those fully contained in another read)
- transitively redundant edges (if $a \rightarrow c$ and $a \rightarrow b \rightarrow c$, then remove $a \rightarrow c$)

Two examples:

ACTGCT
CTGCT (overlap length 5)
GCTAA (overlap length 3)

ACTGCT \longrightarrow GCTAA

Let's have inexact overlaps now

ACTGCT
CTACT
GCTAA

ACTGCT \longrightarrow CTACT \longrightarrow GCTAA

FROM OVERLAP GRAPHS TO STRING GRAPHS

Overlap graph with exact overlaps ≥ 3 ,



String graph with exact overlaps ≥ 3 ,



The read CTGCT is contained in ACTGCT, so it is redundant

DE BRUIJN (DB) GRAPHS

*This is going to be fundamental for **Illumina** & 10X data.*

A **DB** graph for a fixed integer k :

1. **Nodes** = all k -mers in the reads.
2. **Edges** = all exact overlaps of length exactly $(k - 1)$ between k -mers

Example for $k = 3$ and a single read:

ACTG

ACT → CTG

Are DB graphs even a relevant topic anymore? There has been no research on assembly using DB graphs lately

J. Catchen

: -)

DB GRAPHS

Example for many reads and still $k = 3$.

ACTG

CTGC

TGCC

ACT → CTG → TGC → GCC

DB GRAPHS: REDUNDANCY

What happens if we add redundancy?

ACTG

ACTG

CTGC

CTGC

CTGC

TGCC

TGCC

dBG, $k = 3$:

ACT → CTG → TGC → GCC

DB GRAPHS: ERRORS

How is a sequencing error (at the end of a read) impacting the DB graph?

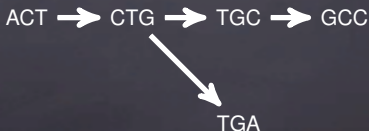
ACTG

CTGC

CTGA

TGCC

dBG, $k = 3$:



DB GRAPHS: REPEATS

What is the effect of a small repeat on the graph?

ACTG

CTGC

TGCT

GCTG

CTGA

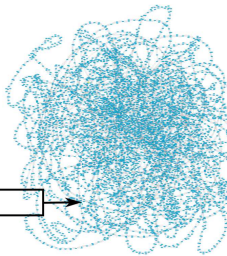
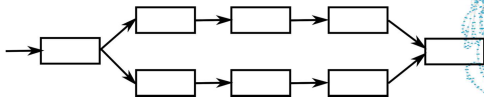
TGAT

dBG, $k = 3$:

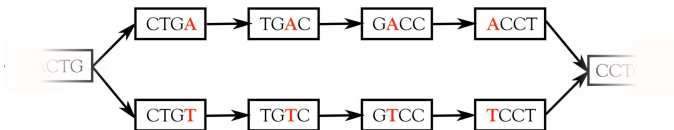


Variant in the *de Bruijn* graph: **Bubble**

Variant in the *de Bruijn* graph: **Bubble**

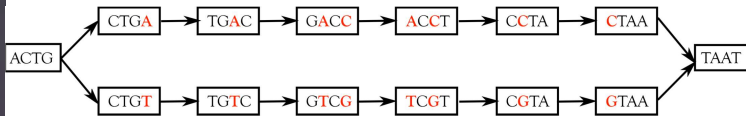


Topological model: SNP



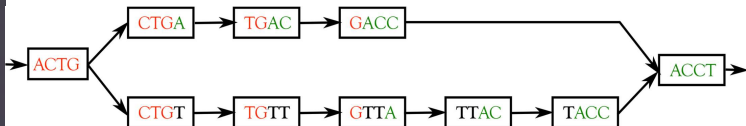
- Two paths of length k nodes:
- Provides two sequences of length $2k-1$:
 - CTG**A**CCT
 - CTG**T**CCT

Topological model: Close SNPs




- Two paths of same length $> k$ nodes
- Two sequences of length $> 2k-1$:
 - CTGACCTAA
 - CTGTCGTAA

Topological model: Indels

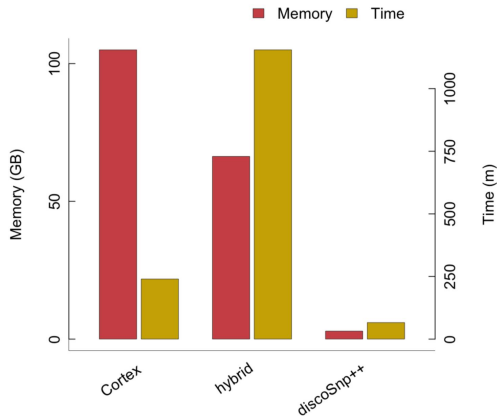


- Two paths
 - One of length $\leq k-1$ nodes ($\leq 2k-2$ nucleotides)
 - One of length $\leq k$ nodes ($\leq 2k-2 + |\text{indel}|$ nucleotides)
- Sequences:
 - CTGACC
 - CTGTTACC

DiscoSnp overview

- 1/ Extract kmers – construct the graph
- 2/ Detect bubbles 
- 3/ For each variant, finds read coverage
 - per allele and per read set
- 4/ Generate .fa and .VCF files
 - .vcf file may be generated using a reference genome

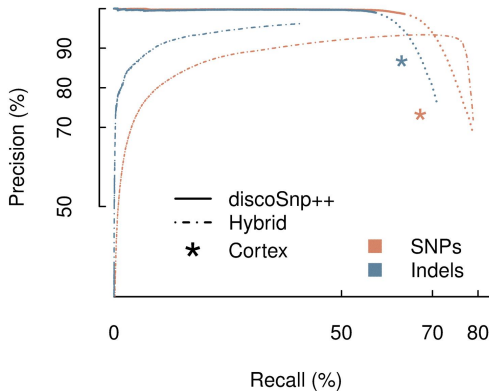
Performances



Human chr1
~100 million reads
100 bp reads


hybrid = SOAPdenovo2 + Bowtie 2 + GATK

Results quality



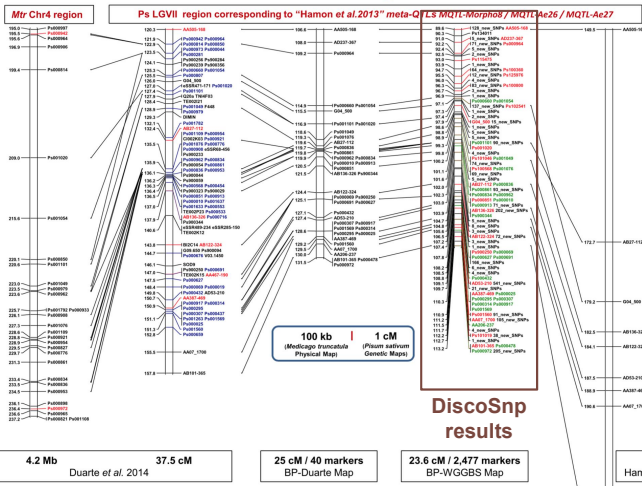
Human chr1
~100 million reads
100 bp reads

SNP predictions on pea genome

- Project PEAPOL 
- Context
 - Pea genome: 4.3 GB + complex, no reference
- Data
 - 4 illumina HiSeq2000 (7x each)
- discoSnp analyses
 - 1CPU, 4GB RAM, 48h computation
- Results
 - > 400,000 SNPs
 - Various filters (coverages, contexts, specific polymorphism)
 - > 88,000 SNPs *"of interest"*



Marker densification in QTL Regions



BACK TO ASSEMBLY THEORY



COMPARISON STRING GRAPH / DB GRAPH

On the same example, compare the DB graph with the string graph:

AGTGCT
GTGCTA
GCTAA

String graph with exact overlaps ≥ 3 :

AGTGCT \longrightarrow GTGCTA \longrightarrow GCTAA

DB graph, $k = 3$:

AGT \longrightarrow GTG \longrightarrow TGC \longrightarrow GCT \longrightarrow CTA \longrightarrow TAA

STRING GRAPH / DE BRUIJN GRAPH (2)

Let's add an error:

AGTGCT

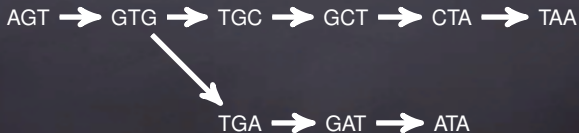
GTGATA

GCTAA

String graph where overlaps ≥ 3 may ignore up to 1 error:

AGTGCT \longrightarrow GTGATA \longrightarrow GCTAA

DB graph, $k = 3$:



STRING GRAPH / DB GRAPH (4)

So, which is better?

- String graphs capture whole read information
- DB graphs are conceptually simpler:
 - ▶ single node length
 - ▶ single overlap definition

Historically, **string graphs** were used for long reads and **DB graphs** for short reads.

String graphs are also known as the **Overlap Layout Consensus** (OLC) method.

HOW DOES ONE ASSEMBLE USING A GRAPH?

Assembly in theory

[Nagarajan 09]

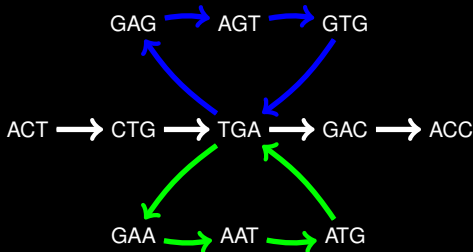
Return a path of *minimal length* that traverses **each node at least once**.

Illustration



The only solution is GATTACATTACAA.

An ambiguous assembly graph



Because of ambiguities and low-coverage regions, a single path is almost never found in theory, and is really never found in practice.

Assembly in practice

Return a **set of paths** covering the graph, such that *all possible assemblies* contain these paths.

Assembly of the above graph

An assembly is the following set of paths:

$\{\text{ACTGA}, \text{GACC}, \text{GAGTG}, \text{GAATG}\}$

CONTIGS CONSTRUCTION

Contigs are *node-disjoint simple paths*.

simple path: all internal nodes have a single in/out edge.

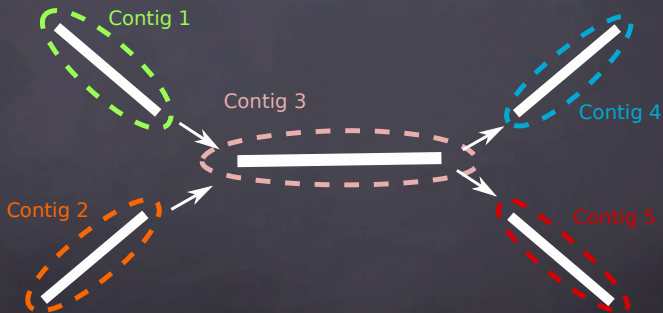
node-disjoint: two different paths cannot share a node.



CONTIGS GRAPH

The result of an assembly is a **contig graph**:

- nodes = contigs
- edges = overlaps between contigs



COFFEE BREAK?



COFFEE BREAK?

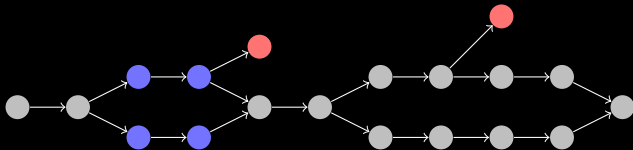


HOW AN ASSEMBLER WORKS

[SPAdes, Velvet, ABySS, SOAPdenovo, SGA, Megahit, Minia, FALCON, Canu, ..]

- 1) Maybe correct the reads. (SPAdes, HGAP, SGA, FALCON, Canu)
- 2) Construct a graph from the reads.

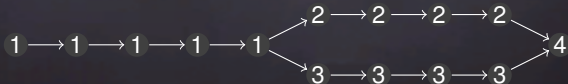
Assembly graph with variants & errors



- 3) Likely **sequencing errors** are removed. (not in Canu, FALCON)



- 3) Known biological events are removed. (not in Canu, FALCON)
- 4) Finally, **simple paths** (i.e. contigs) are returned.



SHORT NOTE ON REVERSE COMPLEMENTS

Due to strand ambiguity in sequencing:

In assembly, we always consider reads (and k-mers) are equal to their reverse complements.

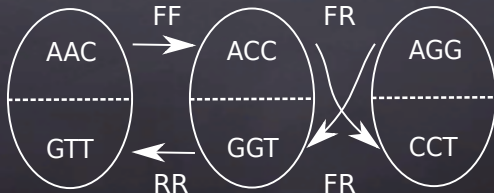
E.g:

AAA = TTT

ATG = CAT

In de Bruijn graphs, nodes implicitly represent both strands.

Lexicographically minimal k -mer is chosen as representative



EXERCISE

In this exercise, for simplicity, ignore reverse complements.

Reads:

TACAGT

CAGTC

AGTCAG

TCAGA

1. Construct the de Bruijn graph for $k = 3$.
(Reminder: nodes are k -mers and edges correspond to $(k - 1)$ -overlaps)
2. How many contigs can be created?
3. At which value of k is there a single contig?
4. (optional) Find a mathematical relationship between k_a , the smallest k value with which a genome can be assembled into a single contig (using a de Bruijn graph), and ℓ_r , the length of the longest exactly repeated region in that genome.

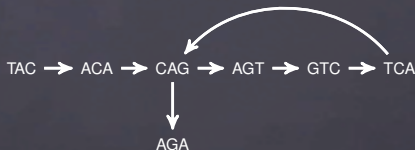
EXERCISE (SOLUTION)

In this exercise, for simplicity, ignore reverse complements.

Reads:

TACAGT
CAGTC
AGTCAG
TCAGA

1. Construct the de Bruijn graph for $k = 3$.
The 3-mers (nodes) are: TAC, ACA, CAG, AGT, GTC, TCA, AGA



2. How many contigs can be created? **3**
3. At which value of k is there a single contig? **5**
4. Find a mathematical relationship between k_a , the smallest k value with which a genome can be assembled into a single contig (using a de Bruijn graph), and ℓ_r , the length of the longest exactly repeated region in that genome. **$k_a = \ell_r + 2$**

PLAN

Fundamentals

- Basics

- Short Exercise

Some useful assembly theory

- Graphs

- Contigs construction

- Exercise

RNA-seq and metagenomes

Visualizing and evaluating assemblies

- Bandage

- Reference-free metrics

- Exercise

One small other thing you can do with k -mers

Assembly in practice

- Exercise

RNA-SEQ ANALYSIS



Haas & Zody, Nat Biotech 2010

RNA-SEQ AND METAGENOME ASSEMBLY

- Uneven coverage varying expression levels / abundance
- Contigs are re-used alternative splicing / different strains
- Short contigs average mRNA length: 2 kbp

RNA-SEQ AND METAGENOME ASSEMBLY

Despite these differences, DNA-seq assembly methods apply:

- Construct a de Bruijn graph (same as DNA)
- Remove errors and variants (somewhat similar)
- Output contigs (same as DNA)
- Allow to re-use the same contig in many different assembled transcripts or metagenome scaffolds (new part)

RNA-SEQ ASSEMBLY: TRINITY



Quick overview of Trinity steps:

- Inchworm
- Chrysalis
- Butterfly

RNA-SEQ ASSEMBLY: TRINITY



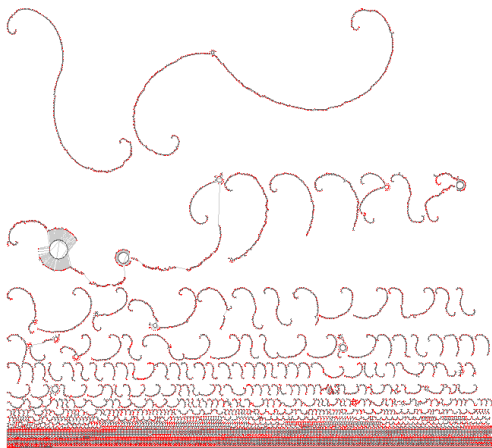
- Inchworm de Bruijn graph construction, part 1
- Chrysalis de Bruijn graph construction, part 2, then partitioning
- Butterfly Graph traversal using reads, isoforms enumeration

RNA-SEQ



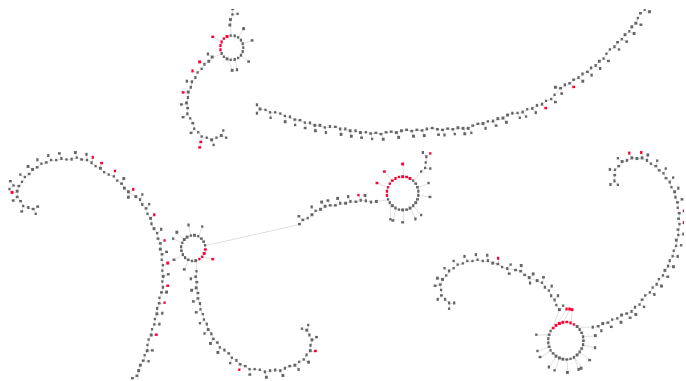
Haas & Zody, Nat Biotech 2010

Example of DBG built from RNA-seq data



Slides adapted from V. Lacroix / C. Benoit-Pilven

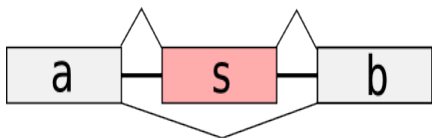
Variants in RNA-seq data



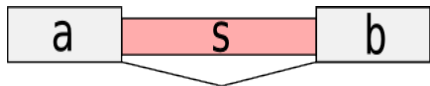
Variants in RNA-seq data

- If the purpose is to identify variants, then assemblers are not well suited
- The variable parts are precisely the ones that will be removed
- 3 types of variations are expected in RNA-seq :
 - SNP
 - indels
 - Alternative splicing

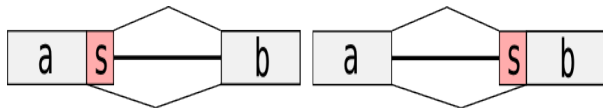
Alternative splicing events



Exon skipping

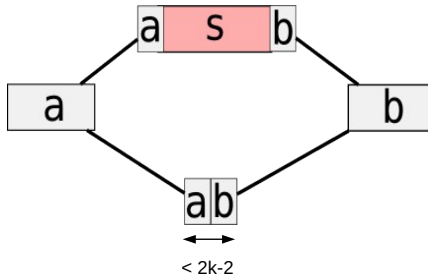


Intron retention



Alternative 5' or 3' splice site

Alternative splicing seen by a DB graph



(Alternative transcription start and end are not covered by this pattern)

Software

KisSplice

A local transcriptome assembler for SNPs and AS events

[HOME](#) [PUBLICATIONS](#) [CONTRIBUTORS](#) [DOWNLOAD](#) [DOCUMENTATION](#) [CONTACT](#)



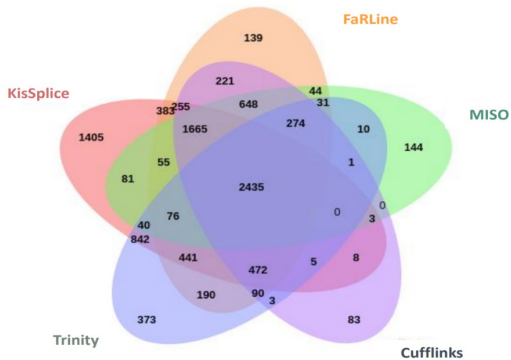
KisSplice

KisSplice is a software that enables to analyse RNA-seq data with or without a reference genome. It is an exact local transcriptome assembler, which enables to identify SNPs, indels and alternative splicing events. It can deal with an arbitrary number of biological conditions, and will quantify each variant in each condition. It has been tested on Illumina datasets of up to 1G reads. Its memory consumption is around 5Gb for 100M reads.

<http://kissplice.prabi.fr>

Sacomoto et al, Recomb-Seq 2012, Lopez-Maestre et al, NAR 2016

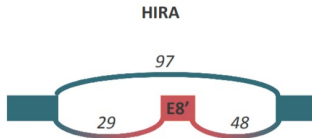
2 human cell lines RNA-seq with replicates, detected AS events :



More details : http://kissplice.prabi.fr/pipeline_ks_farline/

Events found only by **KisSplice**

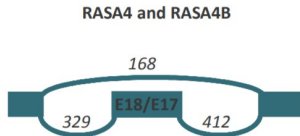
→ New event



SK-N-SH
RA
 $\psi = 0,28$



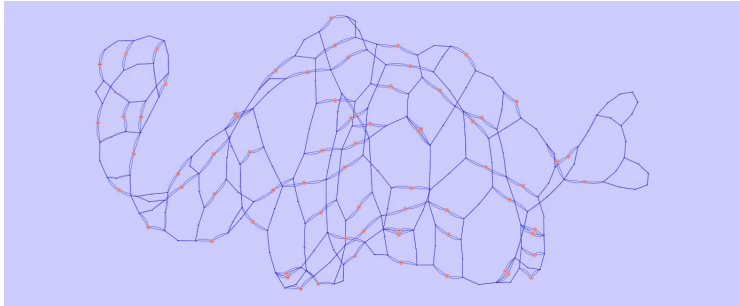
→ Recent paralogs



SK-N-SH
 $\psi = 0,69$

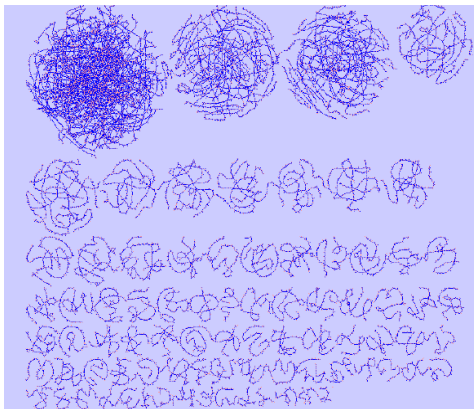


- new alternative exon
- new flanking exon
- new association of exons



This is not an elephant, this is a gene family:)

What about meta-transcritomics ?



PLAN

Fundamentals

Basics

Short Exercise

Some useful assembly theory

Graphs

Contigs construction

Exercise

RNA-seq and metagenomes

Visualizing and evaluating assemblies

Bandage

Reference-free metrics

Exercise

One small other thing you can do with k -mers

Assembly in practice

Exercise

ASSEMBLY GRAPH VISUALIZATION: BANDAGE

Bandage - /Users/Ryan/Desktop/E_coli_LastGraph

De Bruijn graph information

Nodes: 279
Edges: 332
Total length: 4,685,914

Graph drawing

Scope: Entire graph
Style: Single Double
Draw graph

Graph display

Zoom: 44.4%
Node width: 8.5
Random colours

Node labels

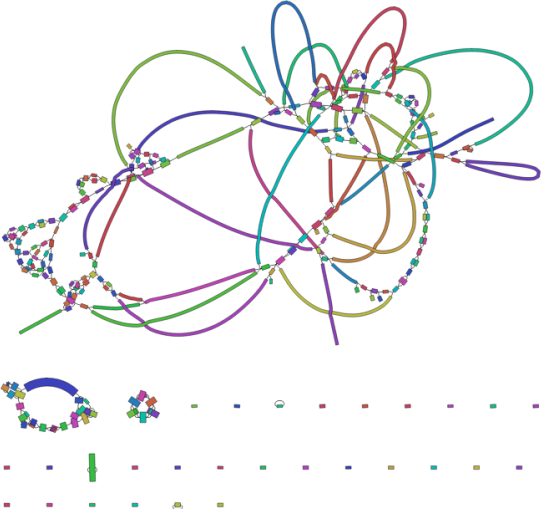
Custom Name
 Length Read depth
 BLAST hits
Font Text outline

BLAST

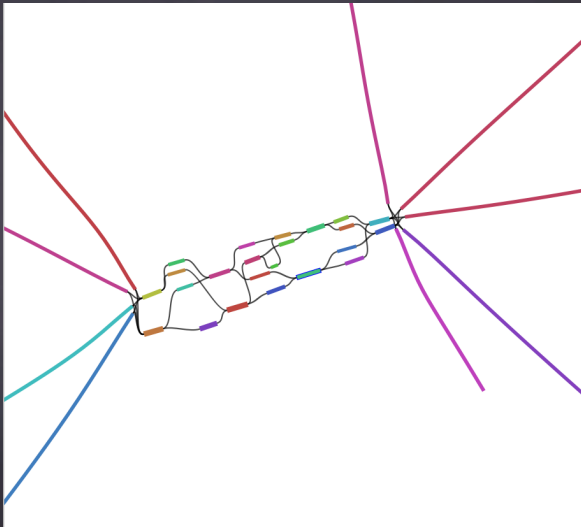
Create/view BLAST search
Query: none

Find nodes

Node(s):
Match:
Find

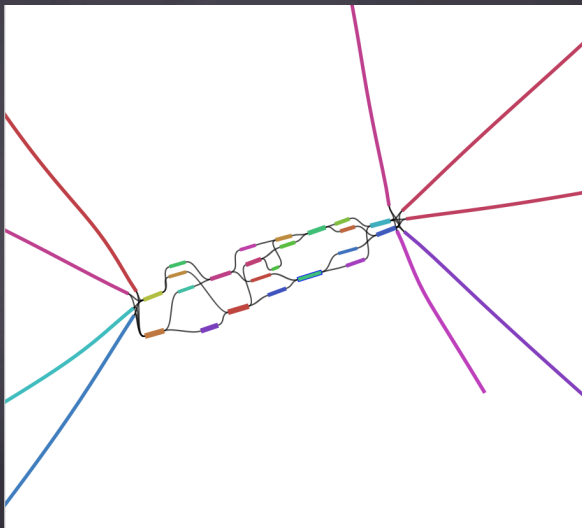


BANDAGE



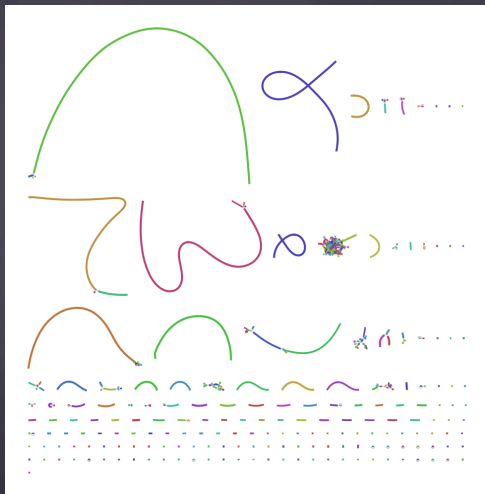
E. coli SPAdes assembly (excerpt). Fig from Lex Nederbragt. What is this knot?

BANDAGE



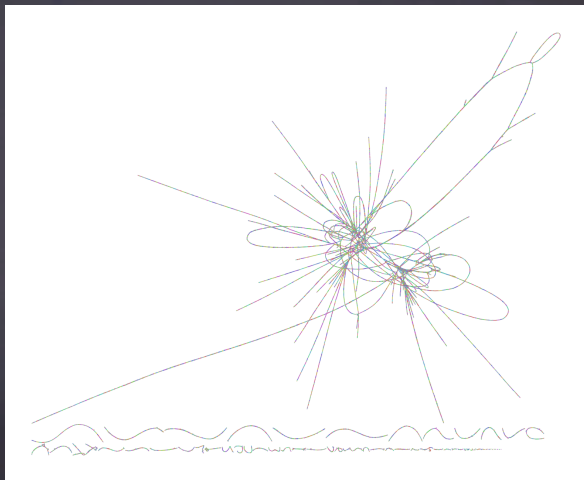
E. coli SPAdes assembly (excerpt). Fig from Lex Nederbragt. What is this knot?
collapsed ribosomal genes (16S, 2S, ..)

PACBIO ASSEMBLY VISUALIZATION



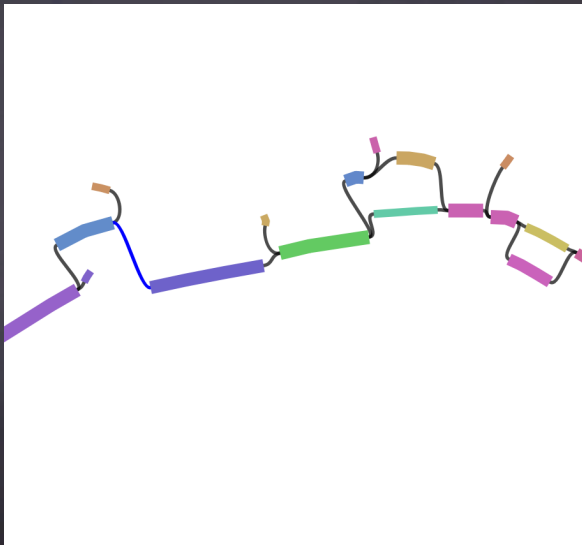
D. melanogaster FALCON assembly. Each node is a contig. (fig. @md5sam)

EFFECT OF SIMPLIFICATIONS ON THE GRAPH (ILLUMINA DATA)



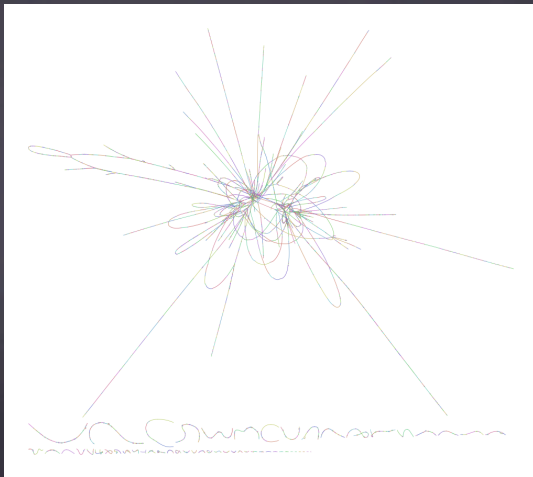
human chr14:20Mbp-20.5Mbp GAGE PE reads, Minia $k=31$, no graph simplifications at all, around 20k nodes

EFFECT OF SIMPLIFICATIONS ON THE GRAPH (ILLUMINA DATA)



same as previous slide, zoomed in to see tips and bubble

EFFECT OF SIMPLIFICATIONS ON THE GRAPH (ILLUMINA DATA)



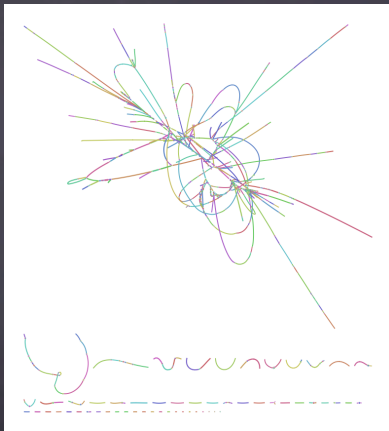
same data & assembler, with tips removed, around 6k nodes

EFFECT OF SIMPLIFICATIONS ON THE GRAPH (ILLUMINA DATA)



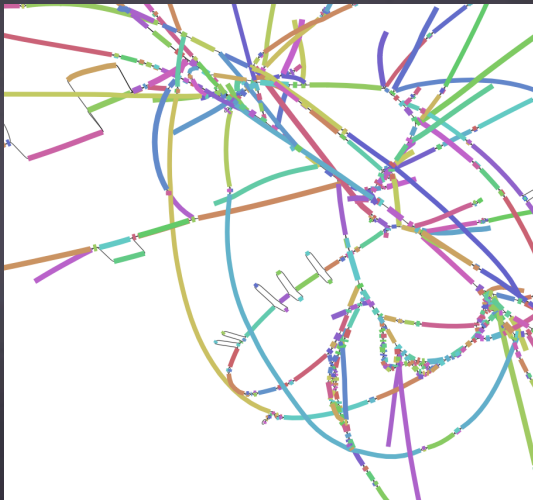
same as previous slide, detail

EFFECT OF SIMPLIFICATIONS ON THE GRAPH (ILLUMINA DATA)



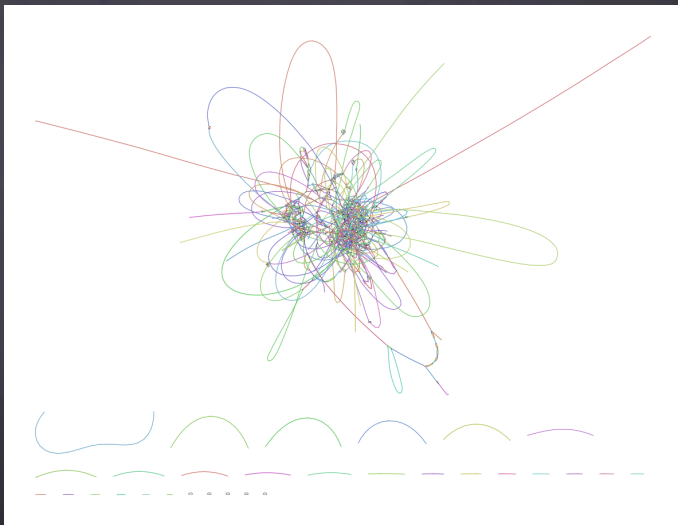
same data & assembler, all simplifications enabled, 1.3k nodes

EFFECT OF SIMPLIFICATIONS ON THE GRAPH (ILLUMINA DATA)



same data as previous slide, detail

EFFECT OF SIMPLIFICATIONS ON THE GRAPH (ILLUMINA DATA)



same data, SPAdes 3.8 $k=31$, 1k nodes

EFFECT OF SIMPLIFICATIONS ON THE GRAPH (ILLUMINA DATA)



same as previous slide, detail

METRICS

Preamble: There is no total order (i.e. ranking) between assemblies.

Why? > 2 independent criteria to optimize (e.g., total length, and average size of assembled sequences)

Example Would you rather have an assembly with **high** coverage and **short** contigs, or an assembly with **low** coverage and **long** contigs?

OVERVIEW OF REFERENCE-FREE METRICS

1. Individually evaluate a single assembly
2. Compare several assemblies made from different parameters or assemblers

Classical metrics:

[QUAST]

- Number of contigs/scaffolds
- Total length of the assembly
- Length of the largest contig/scaffold
- Percentage of gaps in scaffolds ('N')
- N50/NG50 of contigs/scaffolds
- Number of predicted genes
- Number of core single-copy genes

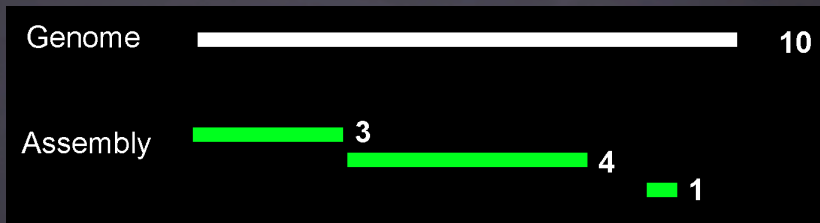
[BUSCO]

```
./quast.py assembly.fa
```


REFERENCE-FREE METRICS: N50

N50 = Largest contig length at which that contig and longer contigs cover 50% of the total **assembly** length

NG50 = Largest contig length at which that contig and longer contigs cover 50% of the total **genome** length



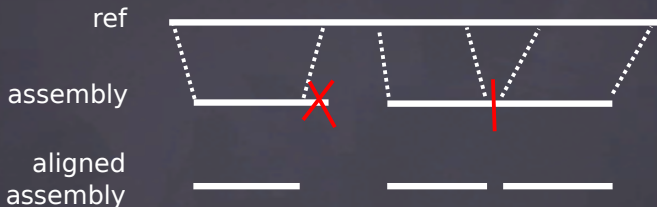
If you didn't know N50, write it down, there will be an exercise !

A practical way to compute N50:

- Sort contigs by decreasing lengths
- Take the first contig (the largest): does it cover 50% of the assembly?
- If yes, its length is the N50 value.
- Else, consider the two largest contigs, do they cover 50%?
- If yes, then the N50 is the length of the second largest contig.
- And so on..

REFERENCE-BASED: NA50

The best metric no-one has heard of.



- Align contigs to reference genome.
- Break contigs at misassemblies and remove unaligned bases.
- Compute N50 of the result. (for NGA50: NG50)

OTHER METRICS OF INTEREST

Internal consistency : Percentage of paired reads correctly aligned back to the assembly (*happy pairs*).

Can pinpoint certain misassemblies (mis-joins).

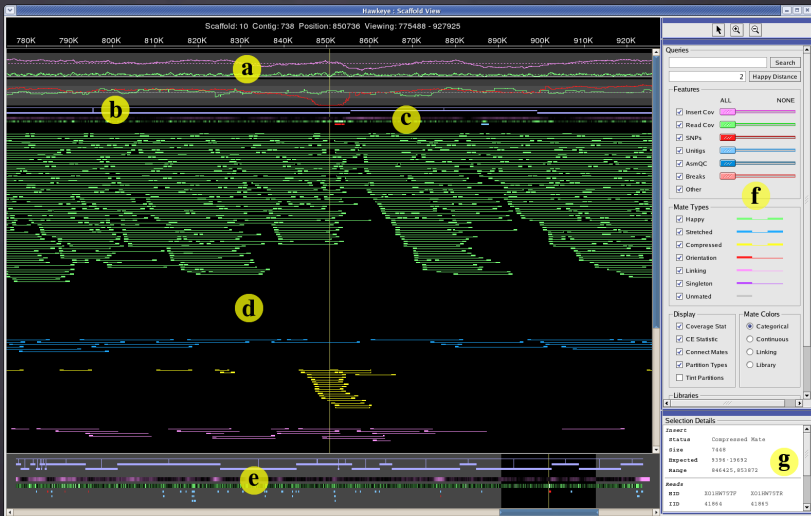
- REAPR [M Hunt, .. (Gen. Biol.) 2013]
- FRCurve [F. Vezzi, .. (Plos One) 2013]

Assembly Likelihood : $\prod_i p(r_i|A)$, where $p(r_i|A)$ is the probability that read r_i is sequenced if the genome was A

In practice, $p(r_i|A)$ is estimated by aligning r_i to the assembly.

- ALE [Clark, (Bioinf.) 2013]
- CGAL [Rahman, (Gen. Biol.) 2013]
- LAP [Ghodsi, (BMC Res. Notes) 2013]

INTERNAL CONSISTENCY: EXAMPLE



Hawkeye software

SUMMARY

Google 'assembly uncertainty' for a nice summary, blog post by Lex Nederbragt.

In summary:

- No total order for metrics
- Use QUAST
- Use BUSCO

EXERCISE

Because at some point in life, one may need to compare genome assemblies.

Here are two assemblies, aligned to the same reference, 1 dot = 1 base pair:

Reference ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ 10




Assembly 1 ■ ■ ■ ■ 4
 ■ ■ 2
 ■ ■ 2






Assembly 2 ■ ■ ■ 3
 ■ 1
 ■ 1
 ■ ■ ■ 3
 ■ ■ 2

- For each, compute the following metrics:
 - ▶ Total size of the assembly, N50, NG50 (bp)
 - ▶ Coverage (%)
- Which one is better than the other?

EXERCISE (SOLUTION)

Reference  10

Assembly 1  4
 2
 2

Assembly 2  3
 1
 1
 3
 2

- For each, compute the following metrics:
 - ▶ Total size of the assembly (8 bp, 10 bp), N50 (4 bp, 3 bp), NG50 (2 bp, 3 bp)
 - ▶ Coverage (%) (80, 100)
- Which one is better than the other? (I would say second one: higher NG50, higher coverage. But: more contigs. Mean contig lengths: 2.6 bp versus 2 bp)

PLAN

Fundamentals

- Basics

- Short Exercise

Some useful assembly theory

- Graphs

- Contigs construction

- Exercise

RNA-seq and metagenomes

Visualizing and evaluating assemblies

- Bandage

- Reference-free metrics

- Exercise

One small other thing you can do with k -mers

Assembly in practice

- Exercise

k -MER MATRIX

k -mer	Abundance per sample				
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
AAACTG	0	0	0	0	10
ACTGAA	10	12	11	10	9
GTACTG	10	12	11	0	0

Computed for all k -mers seen in at least one sample

FINDING DIFFERENCES ACROSS CONDITIONS

- differential gene expression
- —”— transcript expression
- —”— usage of exon
- —”— alternative splicing
- —”— allele-specific expression
- etc..

Tools: DESeq2, DEXSeq, edgeR (see RNAseq lecture)

Need GTF or transcriptome.

RNA-Seq reads



Align reads to genome

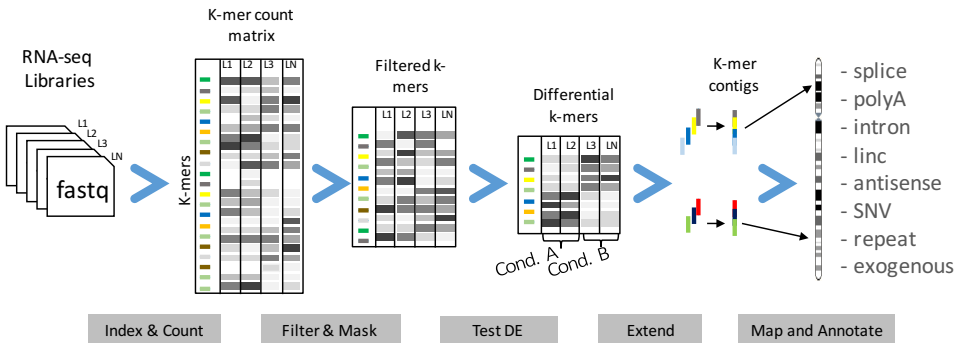
Assemble transcripts *de novo*







DE-KUPL



Audoux *et al*, Genome Biology (2018)

DE-KUPL RESULTS

condition A: samples 1 & 2

condition B: samples 3 & 4

Differential <i>k</i> -mer	Abundance per sample				DE-Kupl	
	Sample 1	Sample 2	Sample 3	Sample 4	logFC	p-value
ACTGAA	10	12	11	10	0.1	0.9
GTACTG	10	12	1	0	3	0.01

In the paper:

- 12 human RNA-Seq, 2 conditions
- 76k DE contigs, 6 hours analysis
- Differential splicing, polyadenylation, lincRNA, antisense RNA, allele-specific expression intron, expressed repeats retention

PLAN

Fundamentals

- Basics

- Short Exercise

Some useful assembly theory

- Graphs

- Contigs construction

- Exercise

RNA-seq and metagenomes

Visualizing and evaluating assemblies

- Bandage

- Reference-free metrics

- Exercise

One small other thing you can do with k -mers

Assembly in practice

- Exercise

RECOMMENDED PRACTICES (GENOMES)

- Illumina:

- ▶ 10XGenomics if possible, otherwise mate-pairs
- ▶ Long read lengths, beware of longest
- ▶ $\geq 50x$ coverage, \times ploidy number
- ▶ Bacteria: no point going $\geq 200x$

and

- PacBio:

- ▶ $\geq 30x$, for now

ASSEMBLERS, PERSONAL EXPERIENCE, 2018

Most genomes SPAdes

PacBio, Nanopore Canu

10X Supernova

RNA-Seq Trinity

Memory issues Minia pipeline

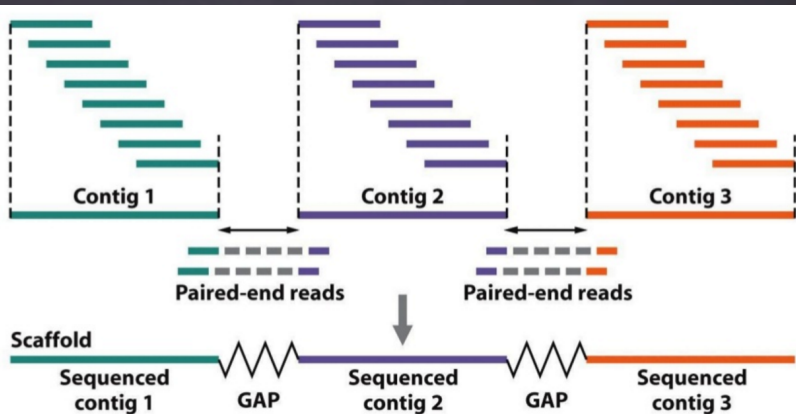
Large metagenomes Megahit

Disclaimer: there is a really long list of other amazing assemblers and I apologize for not having the time to present their benefits and use-cases.

META-PRACTICES

1. Read [Twitter](#) and [blogs](#) for PacBio, Nanopore, metagenomes, assembly news.
2. Pick two assemblers
3. Run each assembler at least two times (different parameters set)
4. Compare assemblies
5. If possible, visualize them using Bandage

SCAFFOLDING



Introduction to Genetic Analysis, Tenth Edition
© 2012 W. H. Freeman and Company

- Many scaffolders: SSPACE, BESST, Opera, SWALO
- Best strategy: mate-pairs libraries with many insert sizes
- Note: misjoins are mainly made during scaffolding

HYBRID ASSEMBLY

When you have multiple sources of data, e.g.

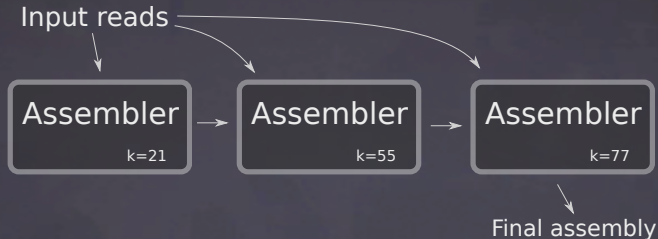
1. high-coverage Illumina paired-end / mate-pairs
2. low-coverage PacBio

Improve an Illumina assembly using:

- SSPACE-LR (scaffolding using PacBio reads)
- PBJelly (same but also gap-filling)

Not aware of any 10X hybrid assembler.

MULTI-K ASSEMBLY



In principle, **better** than single-k assembly.

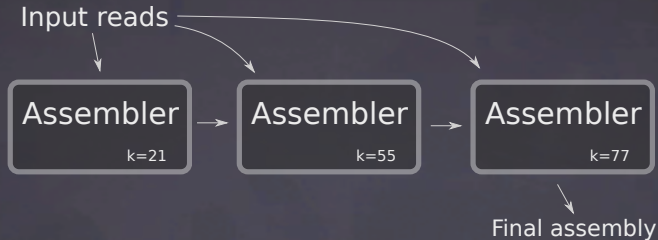
Notable assemblers that implement multi-k:

- IDBA, SPAdes, Megahit

Notable assemblers that don't:

- Velvet, SOAPdenovo, Trinity, ABySS

MULTI-K ASSEMBLY



In principle, **better** than single-k assembly.

Notable assemblers that implement multi-k:

- IDBA, SPAdes, Megahit

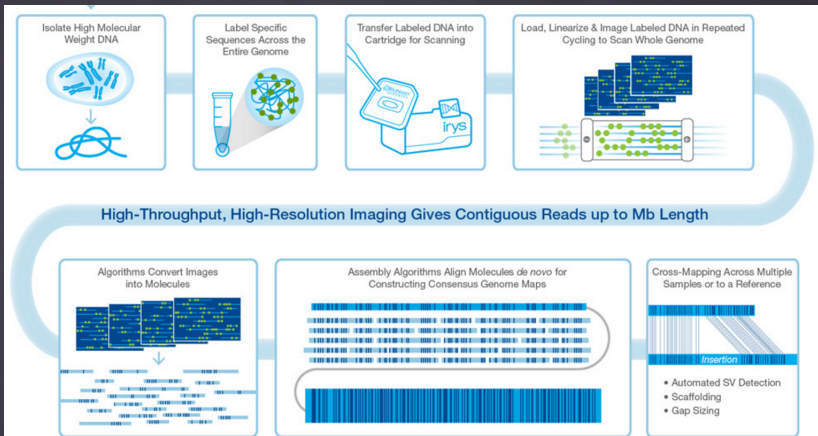
Notable assemblers that don't:

- Velvet, SOAPdenovo, Trinity, ABySS



it's 2017-2018, so really? single-k assembly? really?

GENOME MAPS



Bionano workflow

Other technologies: Dovetail, Nabsys, OpGen.. similar principles.

Hi-C (different)

Chromonomer: Create or correct scaffolds using RAD-seq, or other markers.

COMMON QUESTION: SHOULD I TRIM THE READS?

To check and remove adapters: yes absolutely

Quality-trim: I'd say no

WHAT'S INSIDE THE 10X ASSEMBLER

Supernova is..

- Discover, heavily modified
- de Bruijn graph
- barcode-informed extensions
- open-source

POLISHING

- Improve base-level accuracy of assemblies
- For PacBio/Nanopore
- Pilon
- Racon
- Nanopolish

Slide is work in progress, needs to be polished

ASSEMBLY: A SOLVED PROBLEM?

Still challenging, even in 2018.

- PacBio/Nanopore tools are slowly maturing
- No competition for 10X assemblers
- ~~Hard~~Hopeless to obtain high-contiguity assemblies from Illumina data
- High computational requirements overall

State of the research

1. More PacBio assemblers!
2. *k*-mer methods for RNA-seq, metagenomes, everything
3. de Bruijn graphs

LAST EXERCISE

Reads:

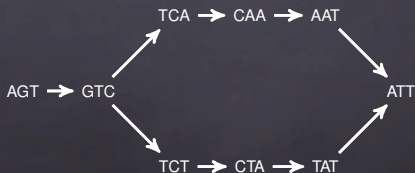
1. AGTC
2. TCAA
3. AATT
4. GTCT
5. TATT
6. TCTA
7. TCAA
8. TCTA

1. Assemble these reads
2. What was special about this genome?

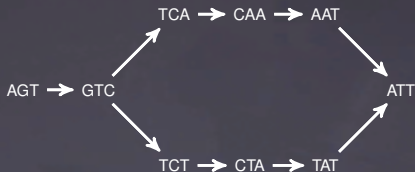
LAST EXERCISE (DETAILED SOLUTION)

Step by step:

- **Choose an assembly model:** de Bruijn graph or string graph
- *The reads are short, let's choose the de Bruijn model*
- **Choose a k-mer size:**
- *Tempting to use $k = 4$, as it is the highest value such that k-mers exist in the reads. However, to obtain a good assembly, all 4-mers from the (unknown) sequenced genome need to be seen in the reads. This is a risky bet. Hence, let's pick a smaller k , $k = 3$.*
- The **nodes** of the graph are all the distinct 3-mers in the reads: AGT, GTC, TCA, CAA, AAT, ATT, TCT, TAT, CTA
- With an appropriate layout, the graph is:



LAST EXERCISE (DETAILED SOLUTION)



- *To assemble this graph, using the contigs construction used before, there would be 4 contigs. Depending on how branching nodes are included in contigs, a possible solution is: AGTC, TCAAT, TCTAT, ATT.*
- But we can actually do better. There are two ways to traverse this graph, yielding an assembly of two "haplotypes":
AGTCAATT
AGTCTATT
- This could be a tiny diploid genome with an heterozygous SNP. The bubble is unlikely to be a sequencing error, as I have purposely added reads 7 and 8, which make the k -mer coverage of both paths equally high.
- An assembler would collapse this bubble and output only one of the two haplotypes.

CONCLUSION, WHAT WE HAVE SEEN

- Assembly evaluation
 - ▶ No total order
 - ▶ Use QUAST, BUSCO
- Techniques
 - ▶ de Bruijn graph (Illumina, 10X), string graph (PacBio, Nanopore)
 - ▶ Errors and small variations are removed
 - ▶ Contigs are just simple paths from the graph
 - ▶ Scaffolds are linked contigs, prone to misassemblies
- *k*-mers
 - ▶ Histograms, reference-free variant detection in DNA and RNA
 - ▶ Not covered: taxonomic assignment, digital normalization, error correction, pseudoalignment
- Takeaways
 - ▶ Try another assembler
 - ▶ Try different parameters
 - ▶ An assembly is not the **absolute truth**, it is a **mostly complete, generally fragmented and mostly accurate hypothesis**

SUPPLEMENTAL SLIDE: THE CHOICE OF k

Choice of k is critical in dBG applications:

- k -mers with sequencing errors are noise
- only *non-erroneous* k -mers matter
- $k < \log_4(|\text{genome}|)$: nearly complete graph, uninformative
- small k : **collapses** repeats, **more** non-erroneous k -mers
- large k : **less** repeat collapsing, **less** non-erroneous k -mers (due to error and shortness of reads)

Generally, $k \geq 20$.

(Compare 4^k to the genome size.)

Higher sequencing coverage means larger k values can be used.