De Bruijn Graphs for DNA Sequencing (Part 2)[1]

Course "Discrete Biological Models" (Modelli Biologici Discreti)

Zsuzsanna Lipták

Laurea Triennale in Bioinformatica
a.a. 2014/15, fall term

[1]These slides mainly based on Compeau, Pevzner, Tesler: *How to apply de Bruijn graphs to genome assembly*, Nature Biotechnology 29 (11).

Sanger sequencing vs. short read sequencing

NGS
Next generation sequencing technologies (Illumina, 454, SOLiD, ...) generate a much larger number of reads

- high-throughput: fast acquisition, low cost
- lower quality (more errors)
- short reads (Illumina: typically 60-100 bp)
- much higher number of reads

While overlap graph approach (with many additional details and modifications!) worked for Sanger type sequences, it no longer works for NGS data. Reason: Input too large, no efficient (= polynomial time in input size) algorithms known, since all problem variants NP-hard.

Solution: Use Euler cycle/path approach

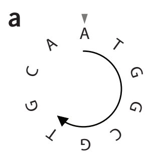
Solution:
Use Euler cycle/path in de Bruijn graph approach instead of finding heaviest Hamiltonian cycle/path in overlap graph.

Finding an Euler cycle (or Euler path) can be solved in polynomial time.

But:
We have to find a way of modelling our problem in the right way.

Modelling our problem with de Bruijn graphs

N.B.
For simplicity, for now our sequence to be reconstructed is assumed to be circular. E.g. bacterial genomes are circular.



String can be read as: `ATGGCGTGCA`, `TGGCGTGCAA`, `GGCGTGCAAT`, ...

Definition of de Bruijn graphs

Let Σ be our alphabet.
(E.g. $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$ or $\Sigma = \{0, 1\}$ or $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$)

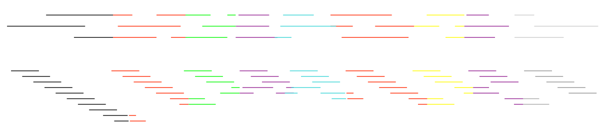
Definition
A digraph $G = (V, E)$ is called a de Bruijn graph of order k if $V \subseteq \Sigma^{k-1}$ and for all $u, v \in V$: if $(u, v) \in E$ then there exists a word $w \in \Sigma^k$ s.t. u is the $(k-1)$-length prefix of w and v is the $(k-1)$-length suffix of w.

Example
$u = \mathtt{GCA}$, $v = \mathtt{CAA}$, $w = \mathtt{GCAA}$.
Note that this graph can have loops, e.g. if $u = \mathtt{AAA}$, then $(u, u) \in E$ is possible.

N.B.
Named after Nicolaas de Bruijn, who introduced a related class of graphs in 1946, for a different problem.

Modelling our problem with de Bruijn graphs

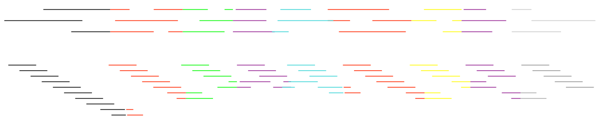
Input: A collection $\mathcal{F}$ of strings.
First step: Generate all k-length substrings of fragments in $\mathcal{F}$.



Example
$\mathcal{F} = \{\mathtt{ATGGCGT}, \mathtt{CAATGGC}, \mathtt{CGTGCAA}, \mathtt{GGCGTGC}, \mathtt{TGCAATG}\}$.
For $k = 3$, we get:

Modelling our problem with de Bruijn graphs

Input: A collection $\mathcal{F}$ of strings.
First step: Generate all k-length substrings of fragments in $\mathcal{F}$.



Example
$\mathcal{F} = \{\texttt{ATGGCGT}, \texttt{CAATGGC}, \texttt{CGTGCAA}, \texttt{GGCGTGC}, \texttt{TGCAATG}\}$.
For $k = 3$, we get:
$\texttt{AAT}, \texttt{ATG}, \texttt{CAA}, \texttt{CGT}, \texttt{GCA}, \texttt{GCG}, \texttt{GGC}, \texttt{GTG}, \texttt{TGC}, \texttt{TGG}$.

Modelling our problem with de Bruijn graphs

Now from the k-mers, we generate the $(k-1)$-length prefixes and suffixes: `AA, AT, CA, CG, GC, GG, GT, TG`. These are the vertices. The edges are the k-mers.

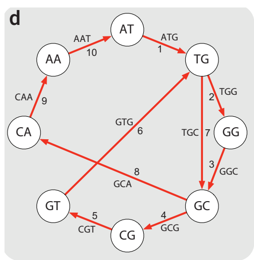
- $\mathcal{F} = \{\texttt{ATGGCGT}, \texttt{CAATGGC}, \texttt{CGTGCAA}, \texttt{GGCGTGC}, \texttt{TGCAATG}\}$, $k = 3$
- edges: $\texttt{AAT}, \texttt{ATG}, \texttt{CAA}, \texttt{CGT}, \texttt{GCA}, \texttt{GCG}, \texttt{GGC}, \texttt{GTG}, \texttt{TGC}, \texttt{TGG}$
- vertices: $\texttt{AA}, \texttt{AT}, \texttt{CA}, \texttt{CG}, \texttt{GC}, \texttt{GG}, \texttt{GT}, \texttt{TG}$

Modelling our problem with de Bruijn graphs

- edges: $\texttt{AAT}, \texttt{ATG}, \texttt{CAA}, \texttt{CGT}, \texttt{GCA}, \texttt{GCG}, \texttt{GGC}, \texttt{GTG}, \texttt{TGC}, \texttt{TGG}$
(remember to only put an edge is the k-mer is present!)
- vertices: $\texttt{AA}, \texttt{AT}, \texttt{CA}, \texttt{CG}, \texttt{GC}, \texttt{GG}, \texttt{GT}, \texttt{TG}$

Modelling our problem with de Bruijn graphs

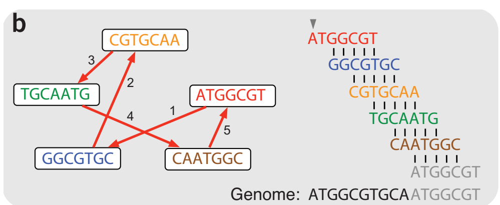
- edges: $\texttt{AAT}, \texttt{ATG}, \texttt{CAA}, \texttt{CGT}, \texttt{GCA}, \texttt{GCG}, \texttt{GGC}, \texttt{GTG}, \texttt{TGC}, \texttt{TGG}$
(remember to only put an edge is the k-mer is present!)
- vertices: $\texttt{AA}, \texttt{AT}, \texttt{CA}, \texttt{CG}, \texttt{GC}, \texttt{GG}, \texttt{GT}, \texttt{TG}$



The numbers on the edges give an Eulerian cycle in this graph: `ATGGCGTGCA`

Comparison to other models

Compare to modelling the same problem with overlap graphs:
$\mathcal{F} = \{\texttt{ATGGCGT}, \texttt{CAATGGC}, \texttt{CGTGCAA}, \texttt{GGCGTGC}, \texttt{TGCAATG}\}$

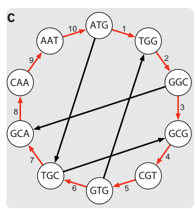


Note that not all non-zero weight edges are included in the figure. The numbers on the edges give a Hamiltonian cycle: `ATGGCGTGCA`.

Comparison to other models

Compare to modelling the same problem with overlap graphs using k-mers as nodes:

- $\mathcal{F} = \{\texttt{ATGGCGT}, \texttt{CAATGGC}, \texttt{CGTGCAA}, \texttt{GGCGTGC}, \texttt{TGCAATG}\}$, $k = 3$
- k-mers are nodes: $\texttt{AAT}, \texttt{ATG}, \texttt{CAA}, \texttt{CGT}, \texttt{GCA}, \texttt{GCG}, \texttt{GGC}, \texttt{GTG}, \texttt{TGC}, \texttt{TGG}$



Put an edge if the overlap equals $k-1$. The numbers on the edges give a Hamiltonian cycle: `ATGGCGTGCA`.

Practical strategies for applying de Bruijn graphs: all k-mers

Generating nearly all k-mers

In reality, only a small fraction of all 100-mers (e.g.) are really sampled. Solution: Take shorter k than readlength. E.g. if reads have length approx. 100, then taking $k = 55$ will yield nearly all k-mers of the genome.

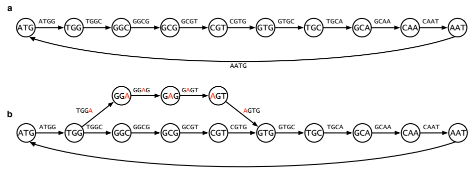
Ex.

In the example, not all 7-mers are present as reads, but all 3-mers are:

- genome: `ATGGCGTGCA`
- 7-mers: `ATGGCGT`, `CAATGGC`, `CGTGCAA`, `GGCGTGC`, `TGCAATG`
- 3-mers: `AAT`, `ATG`, `CAA`, `CGT`, `GCA`, `GCG`, `GGC`, `GTG`, `TGC`, `TGG`

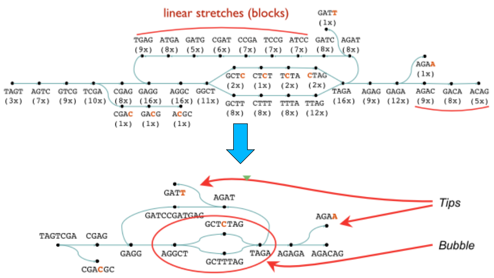
Practical strategies for applying de Bruijn graphs: errors

Errors is reads result in *bubbles* (= *bulges*) in the de Bruijn graph. This can be detected and handled, using multiplicity of k-mers (multigraphs!)



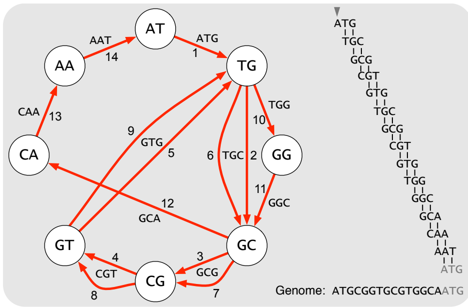
Practical strategies for applying de Bruijn graphs: errors

Errors is reads result in *bubbles* (= *bulges*) in the de Bruijn graph. This can be detected and handled, via multiplicity of k-mers (multigraphs!) or of $(k-1)$-mers



E.g. the software Velvet (Zerbino and Birney, 2008) uses detection and elimination of bubbles and tips.

Practical strategies for applying de Bruijn graphs: repeats



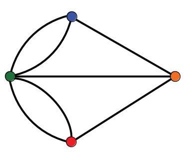
Repeats can be detected using multiplicity of k-mers (edges). Again, using multigraphs (edges have multiplicities).

Eulerian cycles in multigraphs

Theorem

A connected multigraph is Eulerian (has an Eulerian cycle) if and only if every vertex is balanced.

Now indegree = sum of multiplicities of incoming edges (= number of incoming edges counted with their multiplicities), outdegree defined similarly.



Recall the Bridges of Königsberg problem.

Homework

- On page 8, is this the only Euler tour? If not, find the other circular string(s) which might give a solution. Do they also yield a superstring for the input fragments of length 7?
- Repeat the algorithm from p. 7-8 with $k = 4$. How many Euler tours exist now?

Origins of de Bruijn graphs