Path searching problems in de Bruijn graphs

Workshop on Graph Assembly Algorithms for omics data

Leena Salmela[1]

University of Helsinki

November 18th, 2016

[1] Joint work with Eric Rivals, Kristoffer Sahlin, Veli Mäkinen and Alexandru I. Tomescu

Outline

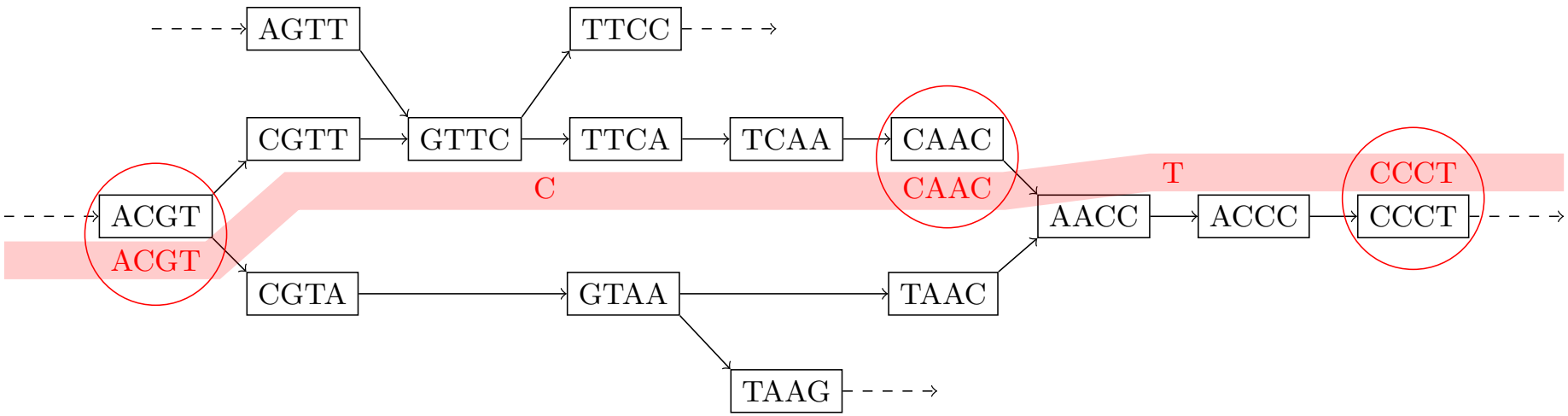
Outline

Read threading

Given

- a de Bruijn graph $G = (V, E)$
- and a read r

find a path $P = v_1, ..., v_k$ that minimizes the edit distance between the read r and the sequence corresponding to the path.



Applications of read threading

- Contig assembly
  - Disentangling the assembly graph
- Correction of sequencing errors
  - Assume a mostly correct de Bruijn graph
  - Thread a read to be corrected through the de Bruijn graph
  - Replace the read by the path

Error level

(Nearly) error free reads:

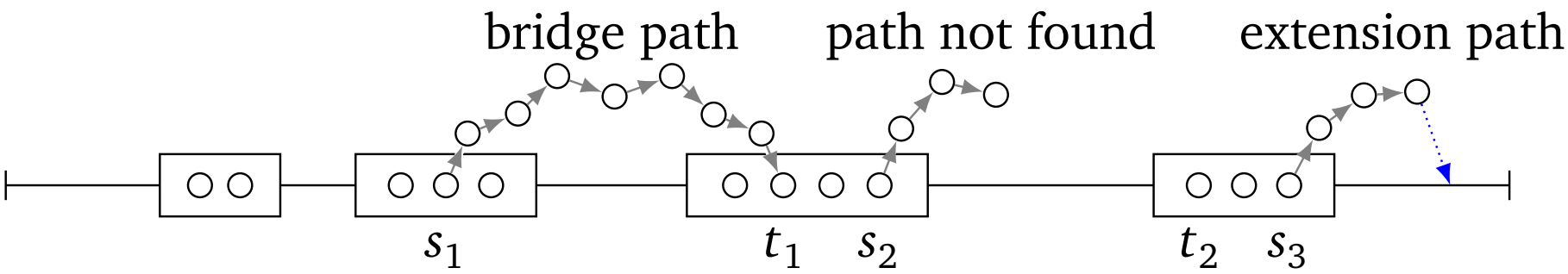
- Easy: k-mers of the read correspond one to one to the DBG nodes
- Contig assemblers targeting NGS reads

Highly erroneous reads (e.g. 15% error rate):

- Contig assemblers utilizing e.g. PacBio reads
- Correcting sequencing errors in e.g. PacBio reads

Seed-and-extend approaches

- Classify k-mers in the read: solid (= in the DBG) and weak
- Find paths in the DBG between the solid k-mers
  - Given a DBG $G = (V, E)$, a sequence R', a source s, and a sink t
  - Find a path from s to t with minimal edit distance between R' and the sequence spelled by the path.
  - Antipov et al. (2016) give a $O(|V||R'|\log(|V||R'|))$ time algorithm



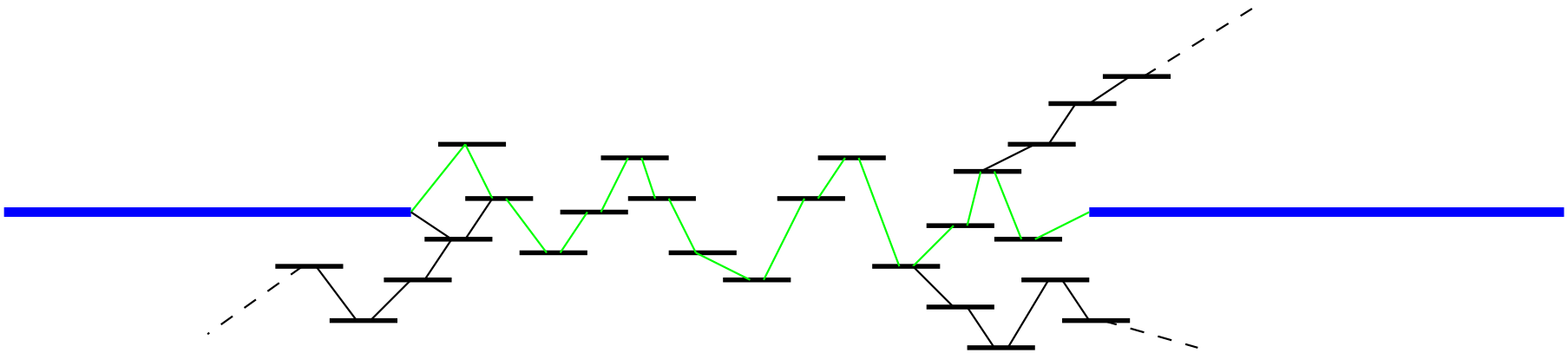
Can we have a large k?

- To find solid k-mers in the read, a small k must be used
- If k is small, the DBG is complex
- Miclotte et al. (2016):
  Index the k-mers in DBG for maximal exact match (MEM) queries
  - We can find solid substrings of the read even when k is large!

Outline

Gap filling

- Gap filling is the last phase in genome assembly
- Input: Scaffolds (=linearly ordered contigs) and reads
- Output: Scaffolds where gaps between contigs have been filled



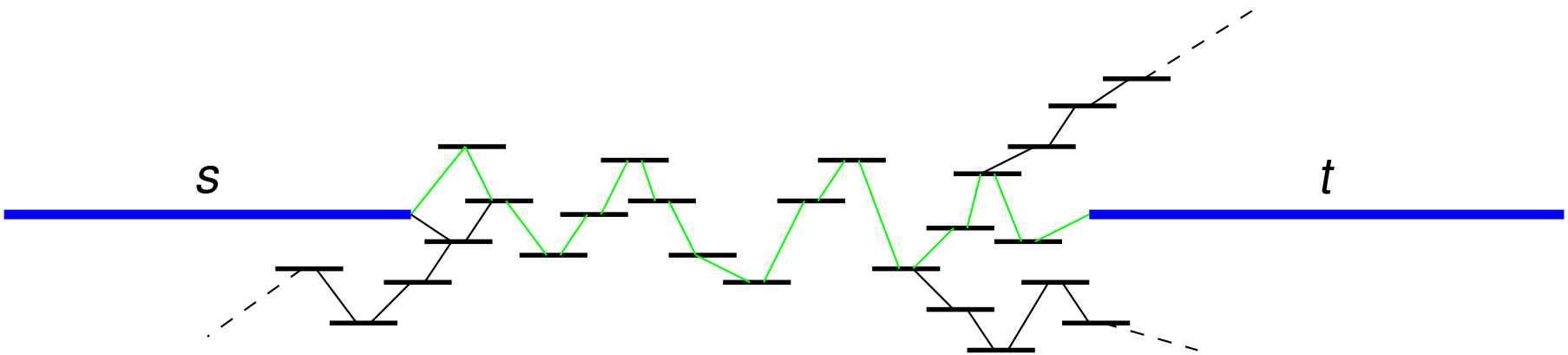
Gap filling: Problem definition

Given

- an (overlap or de Bruijn) graph $G = (V, E)$ of the whole read set
- a cost function $c : E \mapsto \mathbb{Z}_+$
- two vertices s and t representing the flanks of the contigs
- estimate of the gap length $[d', d]$

find for all $x \in [d', d]$ the number of paths $P = v_1, v_2, \ldots, v_k$ such that

$$cost(P) = \sum_{i=1}^{k-1} c(v_i, v_{i+1}) = x.$$



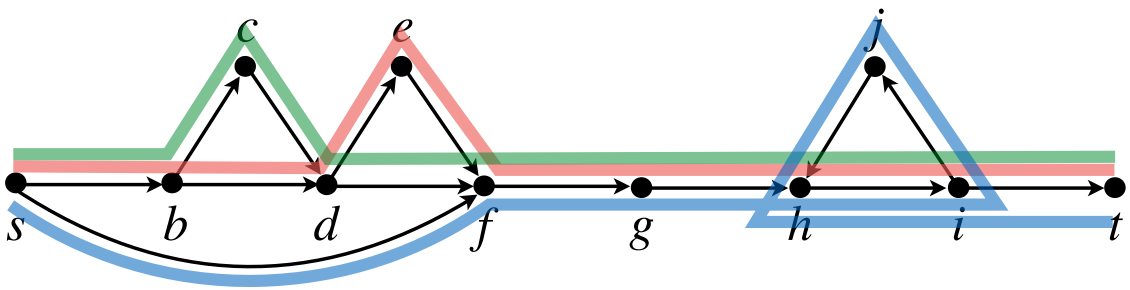
Complexity of gap filling

- Gap filling is NP-complete
- Pseudopolynomial algorithm running in $O(d|E|)$ time

Safe and complete solutions

An algorithm is

- safe: it returns only partial solutions that are common to all solutions
- complete: it returns all safe solutions

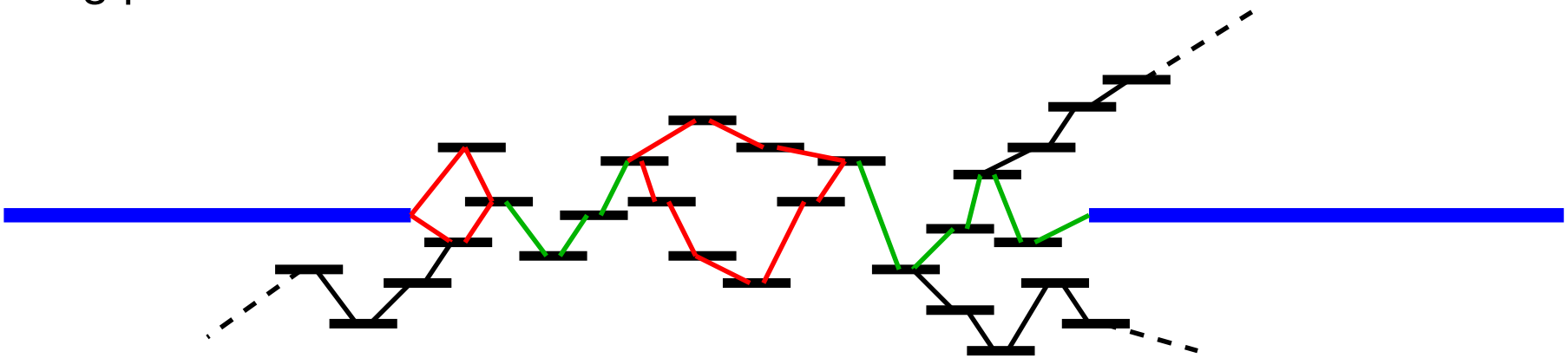


Safe and complete gap filling problem

Given

- an (overlap or de Bruijn) graph $G = (V, E)$ of the whole read set
- a cost function $c : E \mapsto \mathbb{Z}_+$
- two vertices s and t representing the flanks of the contigs
- estimate of the gap length d

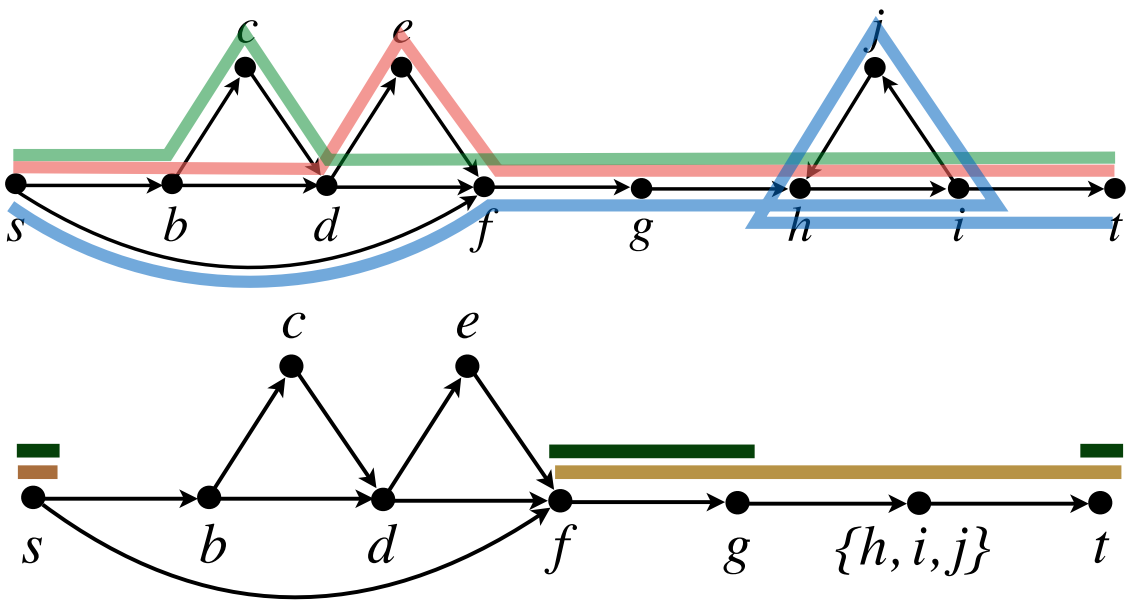
find all maximal paths in G that are sub-paths of all solutions to the gap filling problem.



...GTTTACGtggGATCgacgggGAGCTACTAGACGGTA...

Safe (but not complete) algorithm for gap filling

- Solve the safe and complete problem in a DAG
  $\Longrightarrow$ Safe (but not complete) algorithm for gap filling
- Time complexity: $O(d|E|)$



Outline

Conclusions

- Read threading in de Bruijn graphs
  - Seed-and-extend approaches
  - $O(|V||R'|\log(|V||R'|)$ time: Can we do better?
  - Can techniques to speed up edit distance computation be used?
- Gap filling
  - Safe (but not complete) algorithm
  - $O(d|E|)$ time
  - Open problem: Safe and complete algorithm for gap filling