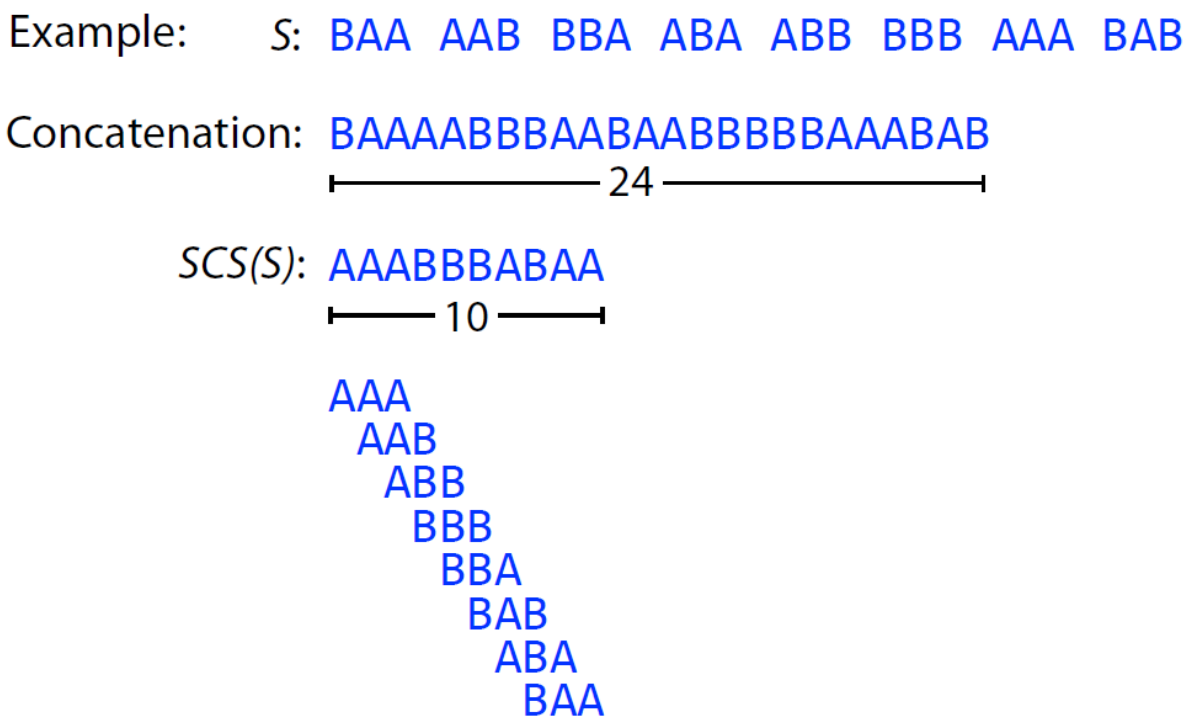
De Novo Genome Assembly

MPA-PRG: Programming Bioinformatics

Exercise 9

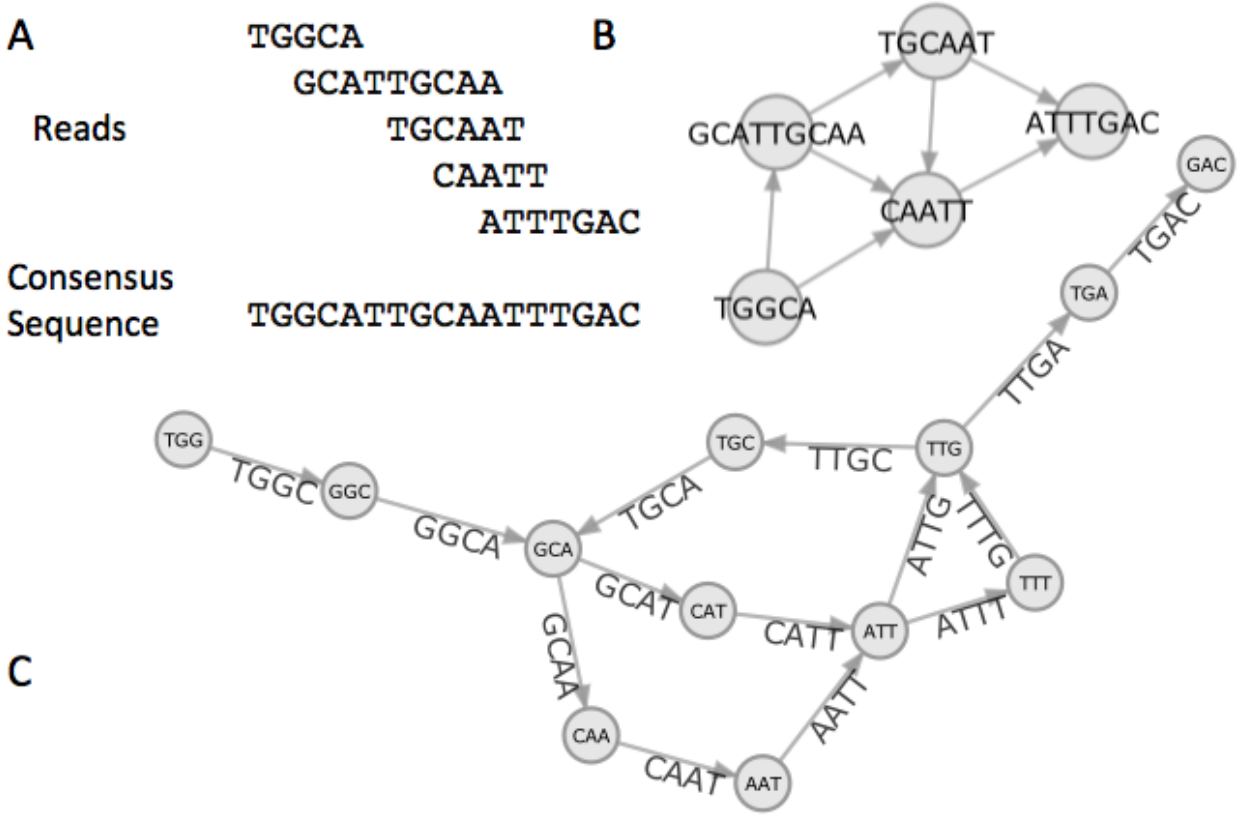
Shortest common superstring

- given a set of strings *S*, find SCS(*S*): the shortest string that contains all strings in *S* as substrings



De Novo Assembly

- string methods – greedy algorithms
- graph methods – OLC (overlap layout concensus) × de Bruijn graph

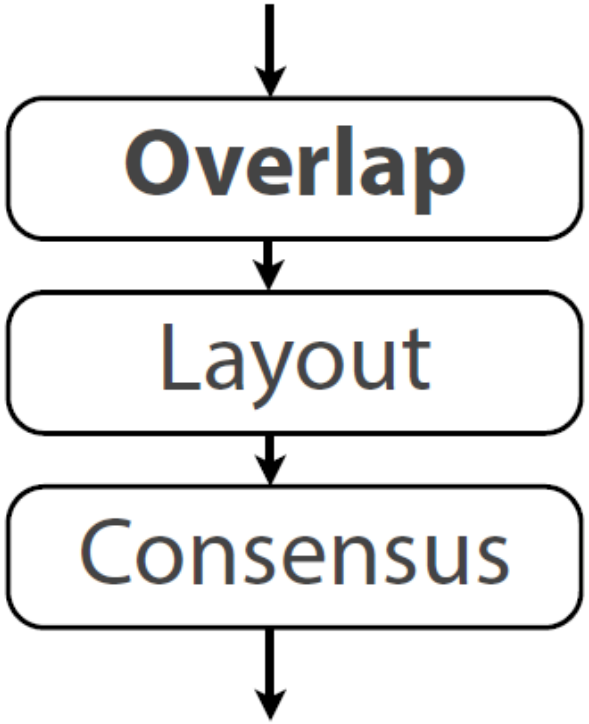


Graph Algorithms

- Overlap Layout Consensus (OLC)
  - reads represent the vertices of the graph and edges the overlaps between them
  - search for a Hamiltonian path (each vertex once)

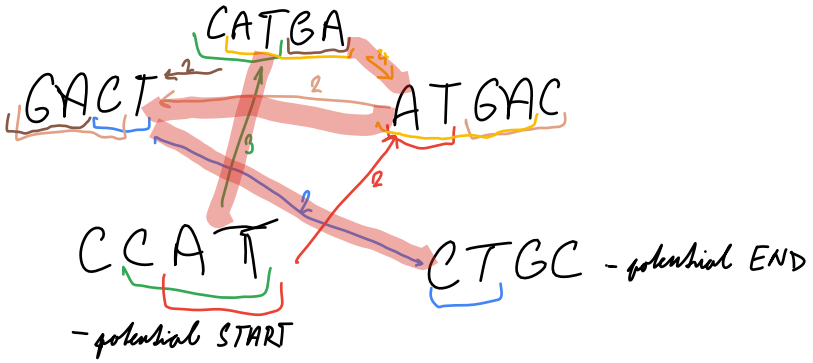
- de Bruijn graph (DBG)
  - reads represent edges of the graph, and vertices represent overlaps
  - search for a Eulerian walk (each edge once)

Overlap Layout Consensus



- create an overlap graph
- find an appropriate path through the graph
- select a putative nucleotide sequence

nejdřív se hledá overlap konce jednoho se vším ostatním.



OLC Examples

1. S = {GACT, ATGAC, CCAT, CTGC, CATGA}

from START → biggest path (weight)

layout: CCAT
CATGA
ATGAC
GACT
CTGC

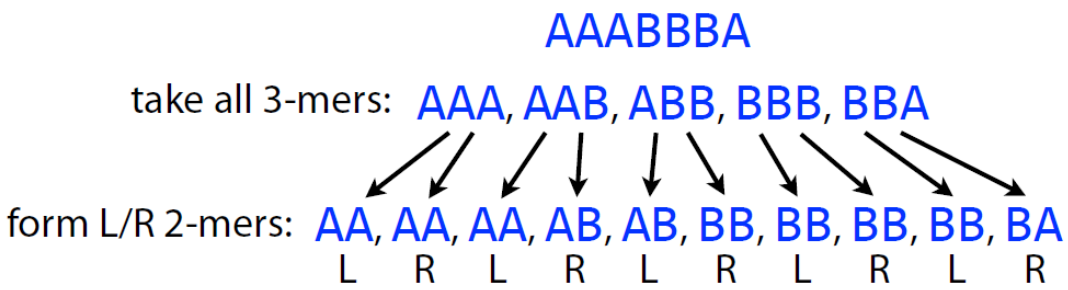
consensus: CCATGACTGC

2. S = {ATG, GACT, CTTA, CATG, ACTTA, TAGCA, GCAT}

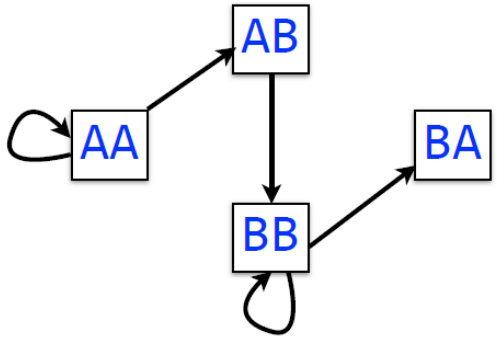
3. S = {TAG, AGAG, AGA, GAGT, AGTAGC, GCAG, AGCAG}

DBG

Take each length-3 input string and split it into two overlapping substrings of length 2. Call these the *left* and *right 2-mers*.

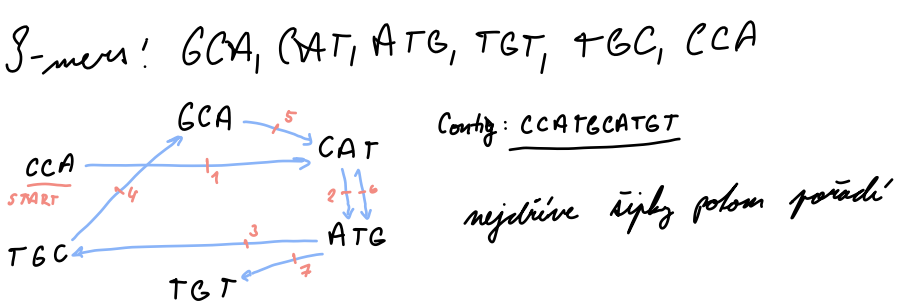


Let 2-mers be nodes in a new graph. Draw a directed edge from each left 2-mer to corresponding right 2-mer:



Each *edge* in this graph corresponds to a length-3 input string

DBG Examples



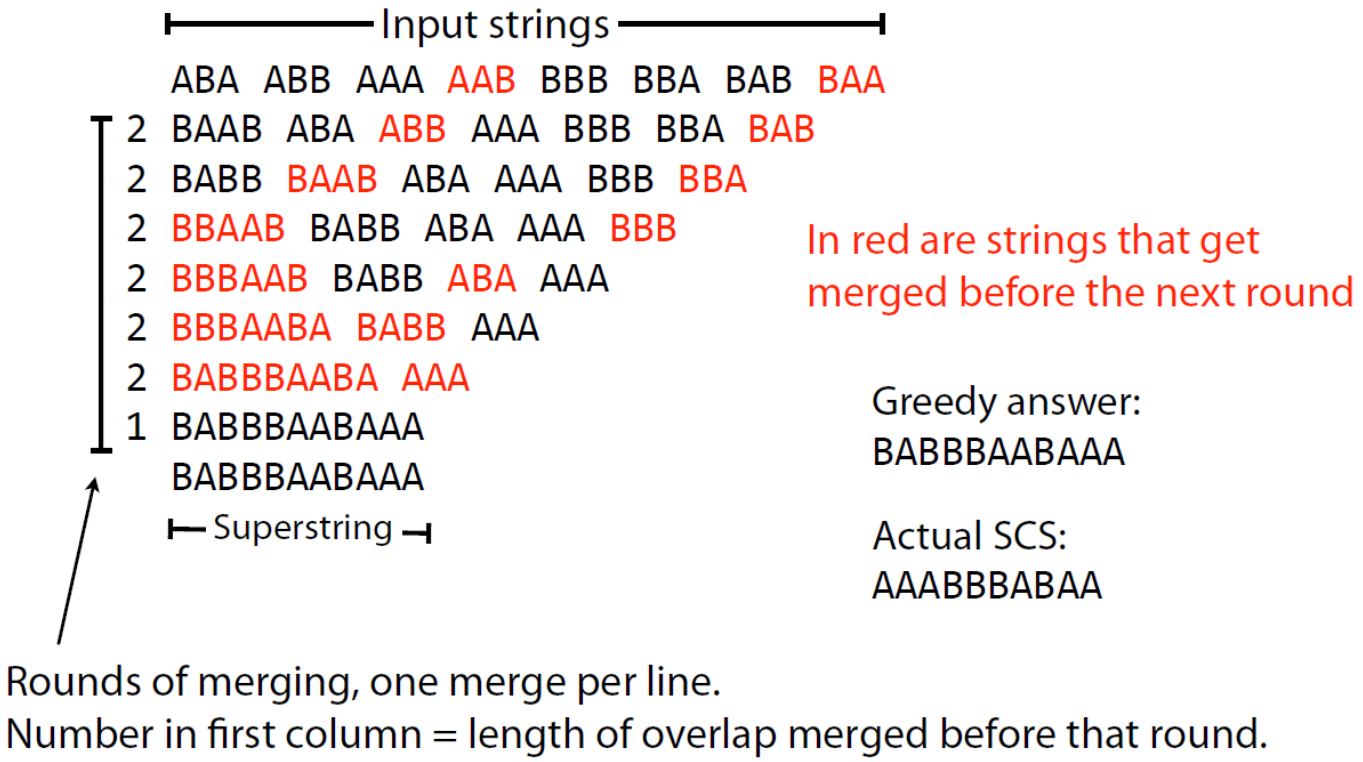
1. S = {GCAT, CATG, ATGT, CATG, ATGC, TGCA, CCAT}

2. S = {ACGT, AATC, ATCA, AGTA, ACGC, CGTA, CAGT, GTAA, GTAC, TAAT, TCAG, TACG}

3. S = {ACGTA, TAAT, ATCAGT, GTA, TACG, CGT}

String Methods

- greedy algorithms
  - in each step merge the two reads with the largest overlap



Greedy SCS Examples

1. S = {CATGC, CTAAGT, GCTA, TTCA, ATGCATC}

2. S = {ATC, TCAGAG, ATG, AGCCAT, TGCAT}

	CATGC	CTAAGT	GCTA	TTCA	ATGCATC
CATGC	—	1^{C}	2^{GC}	0	4^{ATGC}
CTAAGT	0	—	0	1^{T}	0
GCTA	2^{GC}	3^{CTA}	—	0	1^{A}
TTCA	2^{CA}	0	0	—	1^{A}
ATGCATC	1^{C}	1^{C}	0	0	—

Task

- In R, implement a function `GreedySuperstring()` according to the pseudocode.
- Input:
  - `S` a `DNAStringSet` of strings (reads)
- Output:
  - `S` a `DNAStringSet` of the shortest common superstring (contig)

```
GreedySuperstring(S)
1   while length of S > 1
2     overlapMat ← OverlapMatrix(S)
3     if max(overlapMat) = 0
4       return S
5     else
6       seq1, seq2 ← Two sequences from S with the longest overlap
7       Merge seq1 and seq2 and add the new sequence to S
8       Remove seq1 and seq2 from S
9   return S
```

Hint: Create also functions:

- `Overlap()` to calculate overlap between two sequences
- `OverlapMatrix()` to create a matrix of overlaps among all sequences in `S`