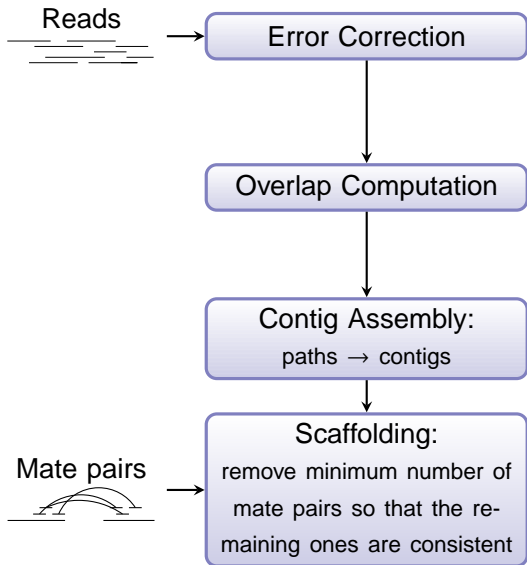


# Fast scaffolding with small independent mixed integer programs

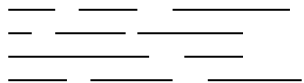
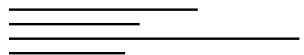
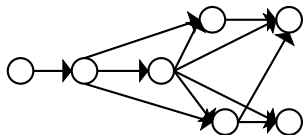
Leena Salmela, Veli Mäkinen, Niko Välimäki,  
Johannes Ylinen, and Esko Ukkonen

October 28th, 2011

# DNA fragment assembly workflow

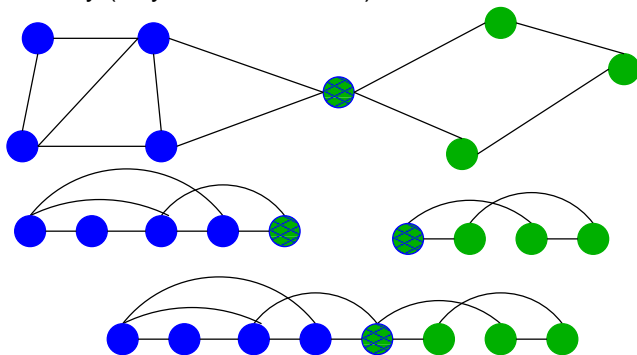


GTCAG - A  
C CGAAGT  
AGAAGTC  
C AAGTCA



## Previous work

- ▶ Even determining the orientation of contigs is NP-complete:  
⇒ All approaches use heuristics
- ▶ Biconnected components of the scaffolding graph can be solved independently (Dayarian et al. 2010)



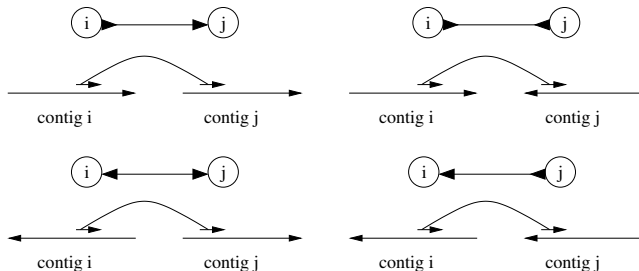
- ▶ Several tools developed: SOPRA, Bambus, SSPACE, OPERA...

# Overview of our work

- ▶ Cleaning input:
  - ▶ Keeping only more reliable mate pairs
  - ▶ Bundling mate pairs that connect the same contigs together
  - ▶ Estimating the distance between contigs based on the mate pairs
- ▶ Partitioning the problem into smaller subproblems of **restricted** size
- ▶ Solving each subproblem as a mixed integer program (MIP)

# Scaffolding graph

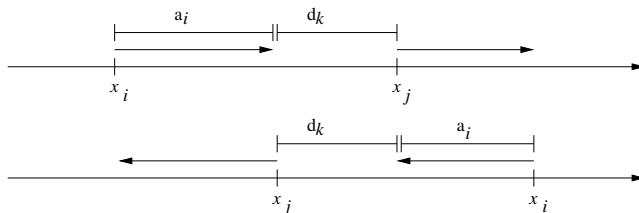
- ▶ Nodes: contigs
- ▶ Edges: mate pairs connecting contigs
  - ▶ **Support** is the number of mate pairs connecting the contigs
  - ▶ **Distance** is the estimated distance of the contigs based on the mate pairs linking the contigs directly
  - ▶ **Orientation** of the contigs



# Partitioning the problem

- ▶ Initially: Nodes=contigs, no edges
- ▶ Sort the edge candidates according to their support
- ▶ Add edges to the graph in descending order of their support but only if the edge does not create a too large biconnected component in the graph.
- ▶ Biconnected components of the graph can be maintained under updates efficiently using a data structure by Westbrook and Tarjan (1992)

# MIP formulation



- ▶  $x_i \in \{1 \dots N\}$ : location of contig  $i$
- ▶  $o_i \in \{0 = \text{reverse}, 1 = \text{forward}\}$ : orientation of contig  $i$
- ▶  $l_k \in [0, 1]$ : smoothed indicator of edge  $k$
- ▶  $a_i$ : length of contig  $i$
- ▶  $s_k$ : support of edge  $k$
- ▶  $d_k$ : distance of edge  $k$
- ▶  $C$ : large constant

maximize  $\sum_k s_k l_k$   
such that

$$o_i - o_j - (1 - l_k) \leq 0$$

$$o_i - o_j + (1 - l_k) \geq 0$$

$$x_i + a_i + d_k - C(1 - l_k) - C(1 - o_i) \leq x_j$$

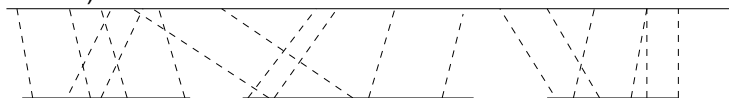
$$x_i + a_i + d_k + C(1 - l_k) + C(1 - o_i) \geq x_j$$

$$x_j + d_k + a_i - C(1 - l_k) - C o_i \leq x_j$$

$$x_j + d_k + a_i + C(1 - l_k) + C o_i \geq x_j$$

# Validation

- ▶ Align the scaffolds to the reference genome:
  - ▶ Find local maximal approximate matches (swift by Rasmussen et al. 2006)



- ▶ Produce maximal colinear chains of the above matches (colinear chaining algorithm by Abouelhoda 2007)
- ▶ N50: “length-weighted median”, sequences longer than the N50 value cover half of the combined length of a sequence set
- ▶ Normalized N50: we computed the N50 statistic for the aligned parts of the scaffolds



## Experimental results: Normalized N50 values

Scaffolder	<i>E.Coli</i>	<i>C.Elegans</i>	<i>P.Syringae</i>	Human
SOPRA	185,227	130,346	72,714	-
SSPACE	-	-	93,850	179,418
MIP Scaffolder	170,796	183,891	84,779	190,008

# Thanks!

- ▶ Acknowledgements:

Rainer Lehtonen, Virpi Ahola, Ilkka Hanski, Panu Somervuo, Lars Paulin, Petri Auvinen, Liisa Holm, Patrik Koskinen, and Pasi Rastas.

- ▶ More information:

L. Salmela, V. Mäkinen, E. Ukkonen, N. Välimäki, and J. Ylinen: Fast scaffolding with small independent mixed integer programs. To appear in *Bioinformatics*.