A greedy, graph-based algorithm for the alignment of multiple homologous gene lists

Jan Fostier, Sebastian Proost, Bart Dhoedt, Yvan Saeys,
Piet Demeester, Yves Van de Peer, and Klaas Vandepoele

Bioinformatics **27** (2011) 749–756.

Speaker: Joseph Chuang-Chieh Lin

Department of Computer Science and Information Engineering
National Chung Cheng University.
Taiwan

March 30, 2011

Self introduction

Basic information

- Name: Joseph, Chuang-Chieh Lin
- Birth: 5 Dec. 1979 at Tainan City
- Married since 2007 & one daughter
- Hobbies: running, reading, & eating

- ⋆ Homepage: http://www.cs.ccu.edu.tw/~lincc
- ⋆ Email addresses: lincc@cs.ccu.edu.tw josephcclin@gmail.com

Education background

- Bachelor of Science: Mathematics, **National Cheng Kung University**, 1998 – 2002
- Master of Science: CSIE, **National Chi Nan University** 2002 – 2004, Thesis supervisor: Prof. Richard Chia-Tung Lee
- Ph.D. (dissertation in progress): CSIE, **National Chung Cheng University** 2004 – 2011, Dissertation supervisors: Prof. Maw-Shang Chang & Prof. Peter Rossmanith

Self introduction (contd.)

Research interests

- Fixed-parameter algorithms
- Randomized algorithms (property testing)
- Graph theory & algorithms
- Computational biology (phylogenetic tree reconstruction)

Outline

Comparative genomics

- Comparative genomics:
  - The study of the relationship of genome structure and function across different species.
  - It exploits similarities & differences in the proteins, RNA, and regulatory regions (a DNA segment).
  - An attempt to understand the evolutionary processes acting on genomes.
- Identification of homologous genomic regions is heavily relied.
  - Using alignment tools.

Sequence alignment

S_1: A R G C R C
S_2: A G A G C

$\Downarrow$

S_1: A R G C R C
S_2: A – G A G C

- Using dynamic programming (table-filling method)

$$M(i,j) = \min \begin{cases} M(i-1,j-1) + \sigma(S_1(i), S_2(j)), \\ M(i-1,j) + \sigma(S_1(i), -), \\ M(i,j-1) + \sigma(-, S_2(j)). \end{cases}$$

$\sigma(\cdot,\cdot)$: score function
(e.g., PAM 250, BLOSUM 62)

- $O(\ell^2)$ time.
  - ℓ: the maximum sequence length.

Multiple sequence alignment

S_1:	A	R	G	C	R	C
S_2:	A	G	A	G	C	
S_3:	R	R	C	R	G	

$\Downarrow$

S_1:	A	R	G	C	R	C
S_2:	A	–	G	A	G	C
S_3:	R	R	–	C	R	G

- $M(i,j,k) =$

$$\min \begin{cases} M(i-1,j-1,k-1) + \sigma(S_1(i), S_2(j), S_3(k)), \\ M(i-1,j-1,k) + \sigma(S_1(i), S_2(j), -), \\ M(i-1,j,k-1) + \sigma(S_1(i), -, S_3(k)), \\ M(i,j-1,k-1) + \sigma(-, S_2(j), S_3(k)), \\ M(i-1,j,k) + \sigma(S_1(i), -, -), \\ M(i,j-1,k) + \sigma(-, S_2(j), -), \\ M(i,j,k-1) + \sigma(-, -, S_3(k)). \end{cases}$$

- $O((2^3-1)\ell^3)$ time.

A heuristic: progressive MSA

Main steps of progressive MSA:

- Compute pairwise sequence similarities;
- Create a phylogenetic tree (or use a user-defined tree);
- Use the phylogenetic tree to carry out a MSA.

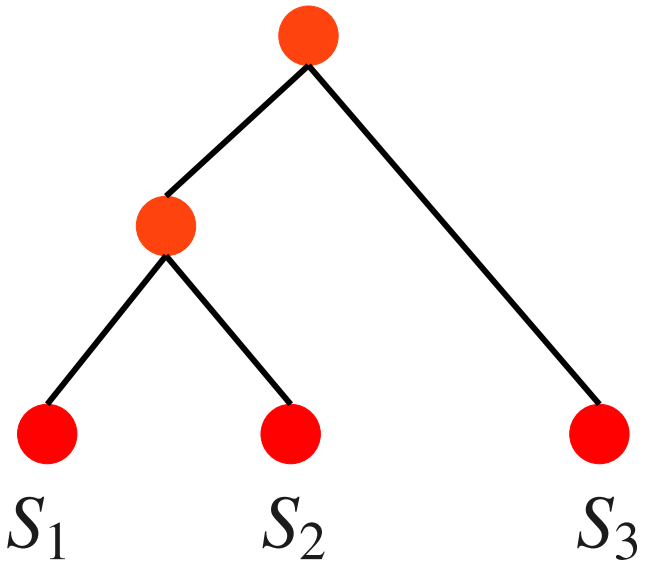
S_1:	A	T	G	C	T	C
S_2:	A	–	G	A	G	C

S_2:	A	G	A	G	C
S_3:	G	G	G	G	–

S_1:	A	T	G	C	T	C
S_3:	G	G	G	G	–	–

$\Downarrow$

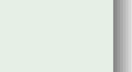
S_1:	A	T	G	C	T	C
S_2:	A	–	G	A	G	C



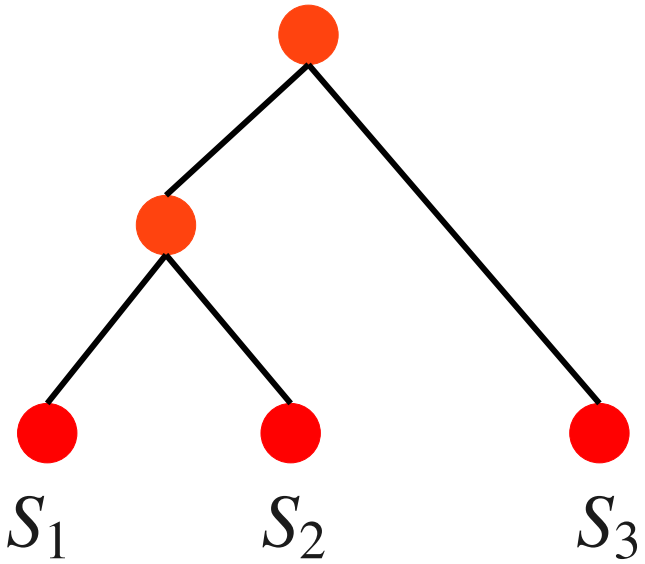
S_1:	A	T	G	C	T	C
S_2:	A	–	G	A	G	C

S_2:	A	G	A	G	C
S_3:	G	G	G	G	–

S_1:	A	T	G	C	T	C
S_3:	G	G	G	G	–	–



S_1:	A	T	G	C	T	C
S_2:	A	–	G	A	G	C
S_3:	G	–	G	G	G	–



Well known MSA tools

- CLUSTAL (W) (Higgins & Sharp 1988; Thompson *et al*, 1994)
- T-COFFEE (Notredame *et al*., 1998)
- MUSCLE (Edgar, 2004)
- MAFFT (Katoh *et al*., 2002)

Contribution of this paper

- Focus on alignment of multiple, mutually homologous genomic segments (gene lists).
  - Homologous: *derived from a common ancestor*.
  - The homologous relationships between individual genes
    $\Rightarrow$ established as a preprocessing step.
- A graph-based approach to create accurate alignments of homologous genomic segments.
  - Find a maximal number of homologous genes.
  - The detection of additional homologous genomic segments can be facilitated.

Some remarks

- The MSA of gene lists is much different from that at the nucleotide or amino acid level.

Nucleotides & amino acids

- alphabet size: nucleotides (4) amino acids (20).
- Through evolution:
  - mainly undergo *character substitutions*.

Genes (or chromosomes)

- alphabet size: combinations of nucleotides...
- Through evolution:
  - mainly undergo *gene loss/insertion*, *inversion*, etc.

Contribution of this paper (contd.)

- The alignment algorithm: a subroutine of the software `i-ADHoRe`.
  - `i-ADHoRe 3.0` is available at http://bioinformatics.psb.ugent.be/software

original `i-ADHoRe`

- Simillion *et al.* 2004.
- progressively applies the Needleman-Wunsch (pNW) aligner
- × 'once a gap, always a gap'.

`i-ADHoRe 2.0`

- Simillion *et al.* 2008.
- Using a greedy, graph-based aligner (GG-aligner).
- ✓ Avoiding 'once a gap, always a gap'.
- × Correctly aligned genes are not more.

`i-ADHoRe 3.0`

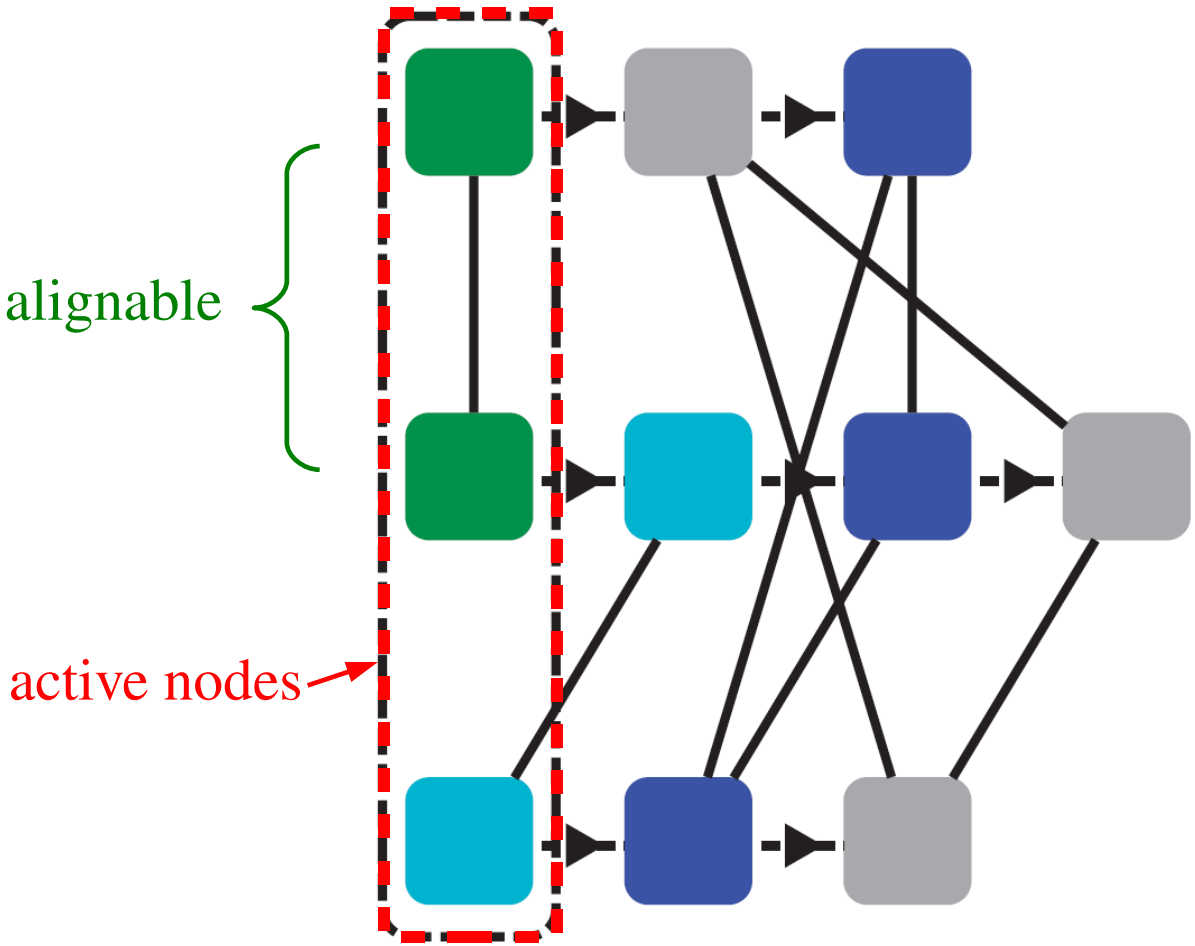
- Fostier *et al.* 2011.
- New greedy, graph-based aligner (GG2-aligner).
- Using heuristics to resolve *conflicts*.
- ✓ Outperform pNW and GG aligners.

- The main dataset of genome: *Arabidopsis thaliana*.

Arabidopsis thaliana

The algorithm

The graph structure (cf. Lenhof *et al.* 1999)

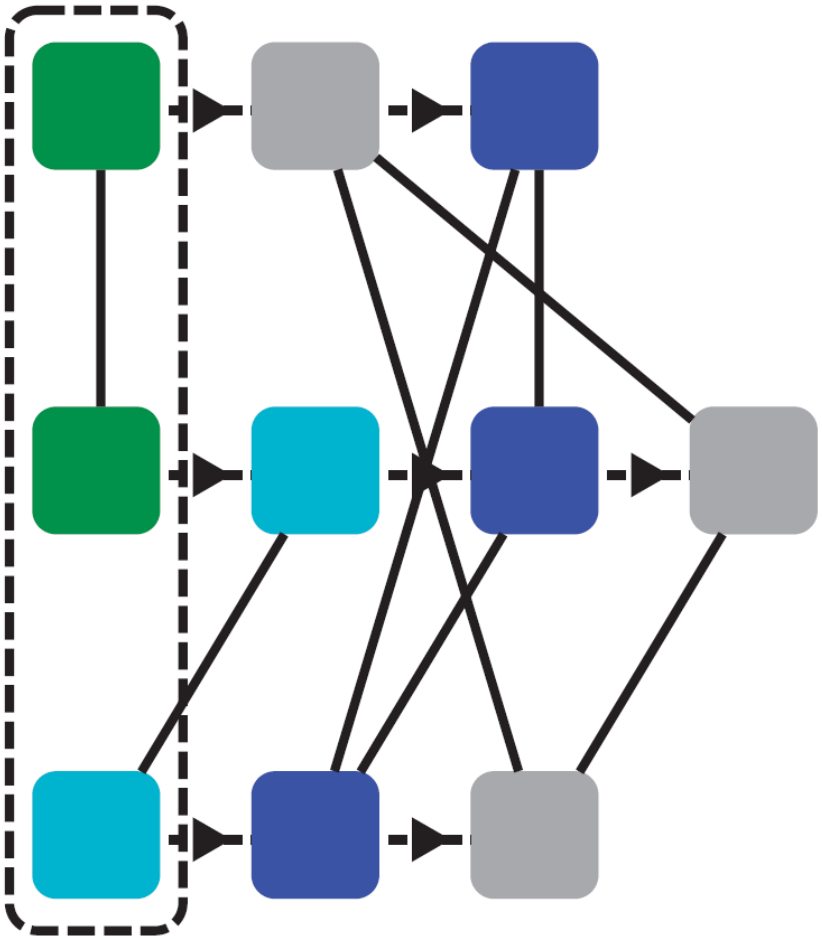


The graph structure (contd.)

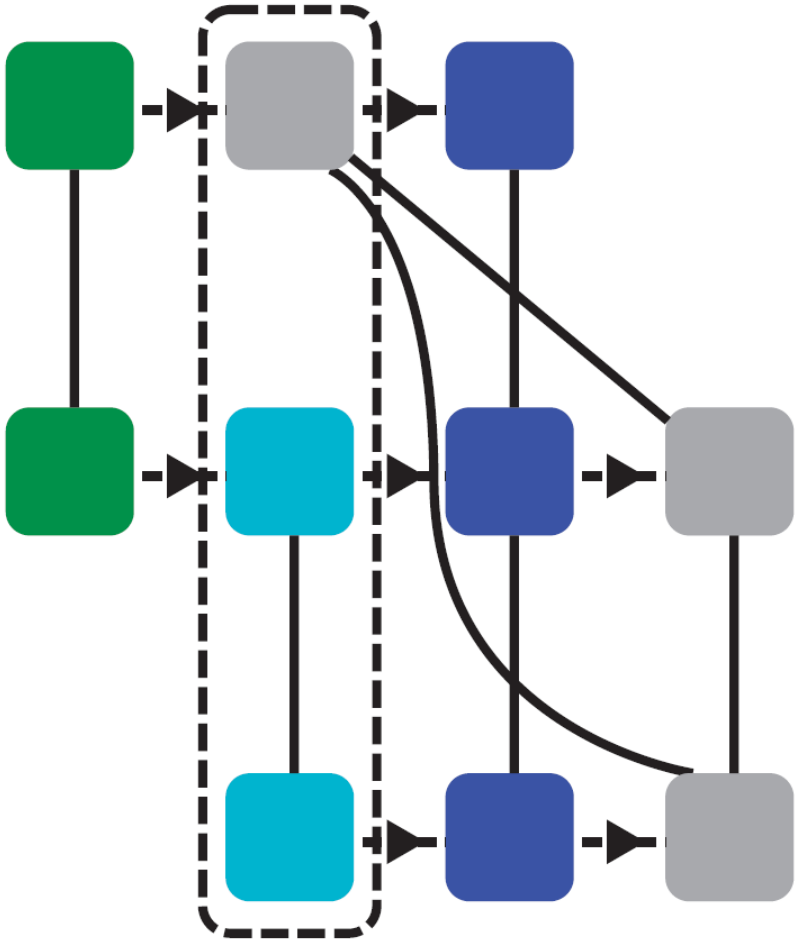
$G(V, E, w)$:

- N genomic segments.
- vertices in V: genes.
  - $n_{i,j}$: the j-th gene on the i-th gene-list.
  - n_{i,a_i}: the active node which is the a_i-th gene on the i-th gene-list.
- Consecutive genes on a segment are connected through a (directed) edge.
- Homologous genes located on different lists are connected through a link (undirected edge).
- Weight w: only attributed to links.

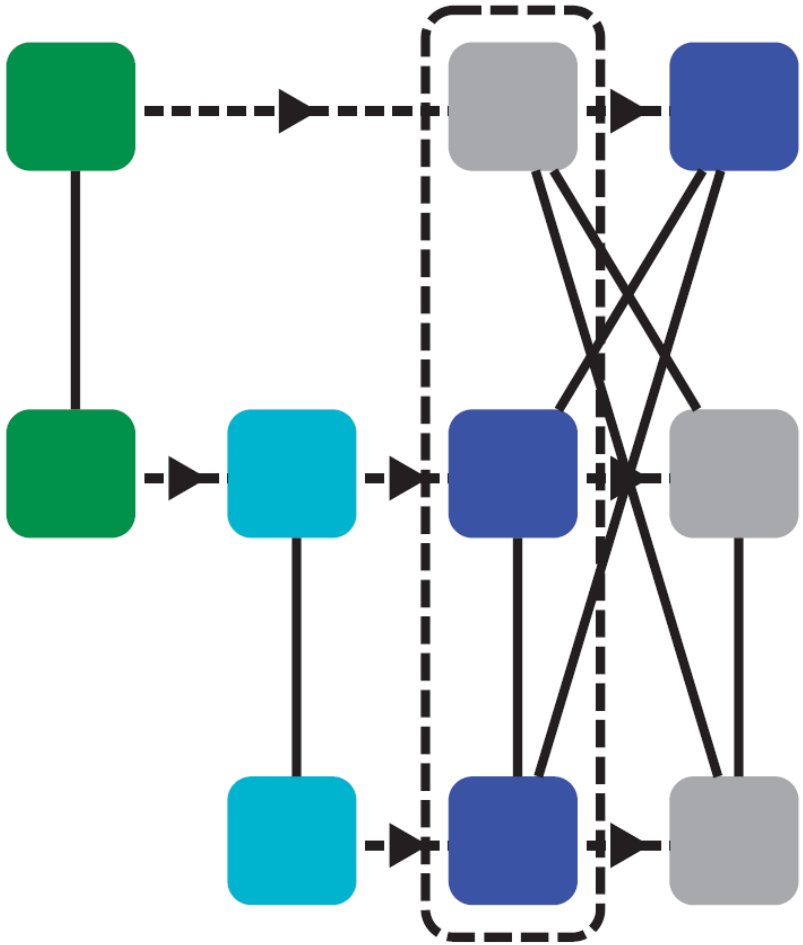
The basic procedure



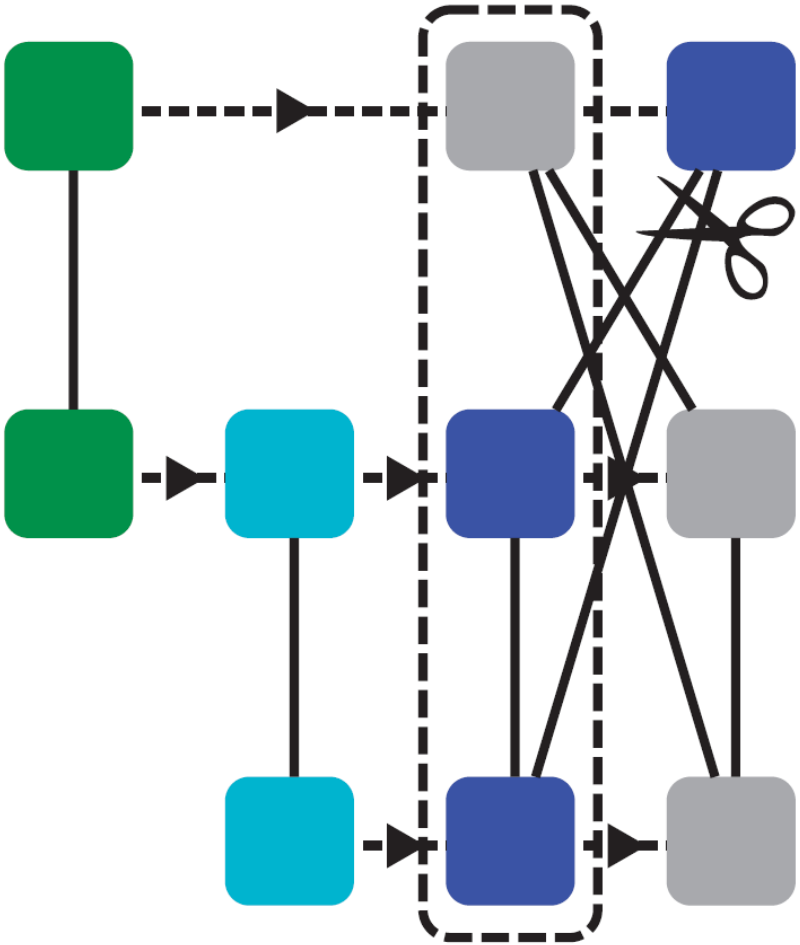
The basic procedure



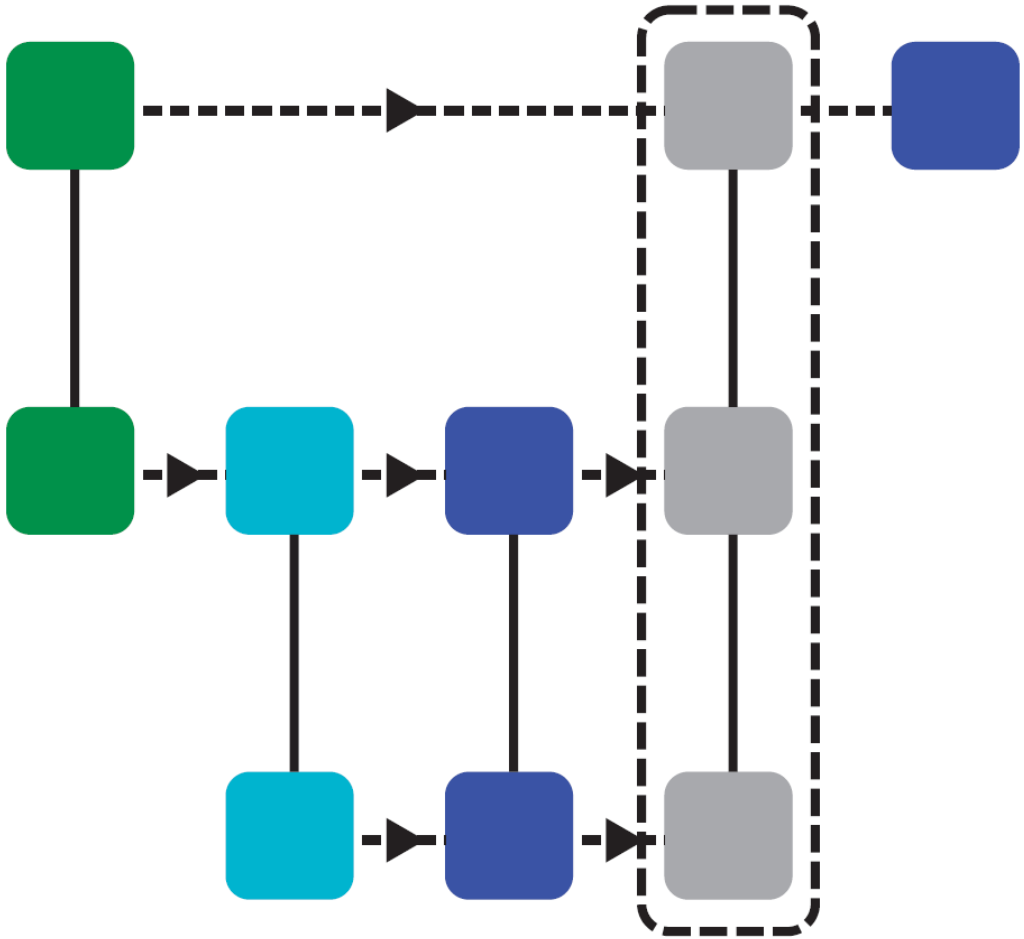
The basic procedure



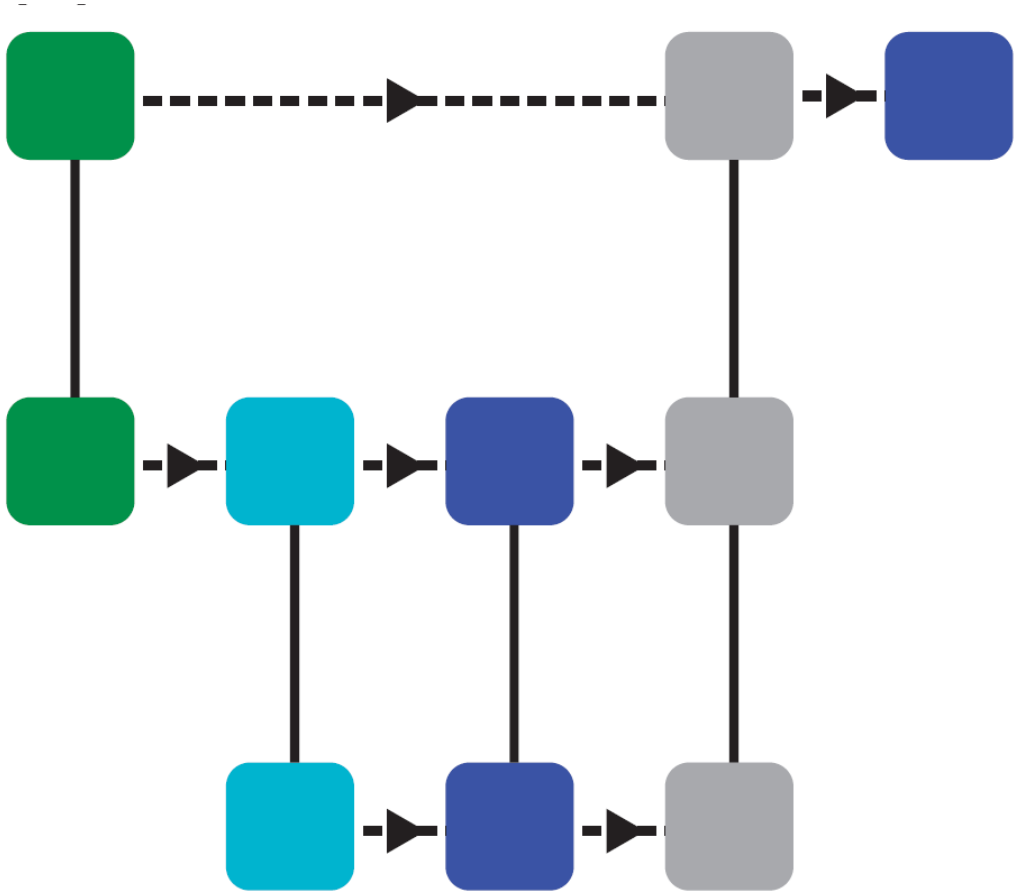
The basic procedure



The basic procedure



The basic procedure



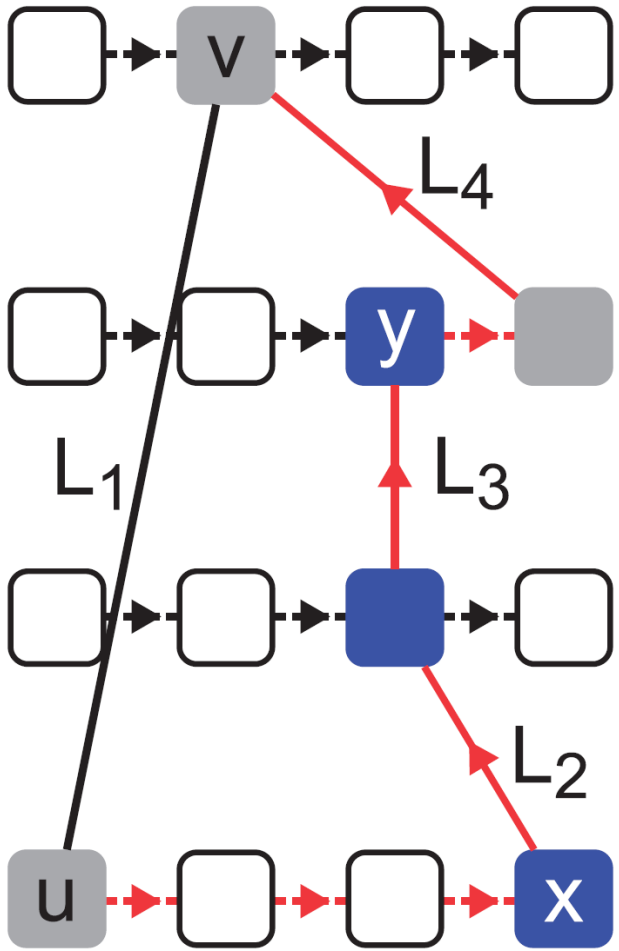
Conflicts

- alignment of a link: aligning two nodes connected by a link.
- a conflict: a set of links such that the alignment of some of them prohibits the alignment of others.
  - It can be resolved by removing one or more links.
  - ▷ Though certain homologous genes will not be placed in the same column finally.
- **Goal:** minimize the number of misaligned genes.
  - It's imperative to carefully select the link(s) to be removed.

Blocking paths/cycles & direct paths

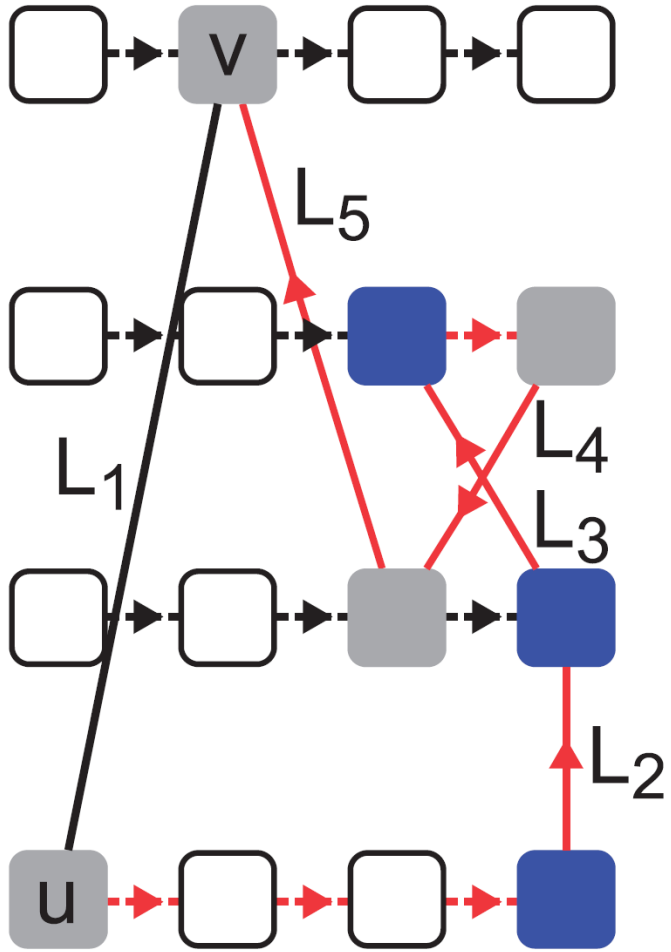
- $P_B = \{L_2, L_3, L_4\}$:
  an elementary blocking path from u to v.
- $P_D = \{L_2, L_3\}$:
  an elementary direct path between x and y.
- $C_C = \{L_1, P_B\}$:
  an elementary blocking cycle.

⋆ $\exists$ conflicts $\Leftrightarrow$ $\exists$ blocking cycle.



Blocking paths/cycles & direct paths (contd.)

- $P_B = \{L_2, L_3, L_4, L_5\}$:
  a blocking path from u to v (NOT elementary).
- $C_C = \{L_1, P_B\}$:
  a blocking cycle (NOT elementary).
- Yet $C'_C = \{L_3, L_4\}$ is an elementary blocking cycle.
- Removing either L_3 or L_4 resolves the conflict!



Minimal conflicts & elementary blocking cycles

- a conflict is minimal: the removal of **ANY** link from its corresponding blocking cycle resolves **ALL** conflicts among the remaining links.

Proposition

- *Minimal conflicts corresponds to elementary blocking cycles and vice versa.*
- *A non-elementary blocking cycle corresponds to one or more minimal conflicts.*

- In what follows, we only consider elementary blocking paths/cycles and minimal conflicts.

Active conflicts

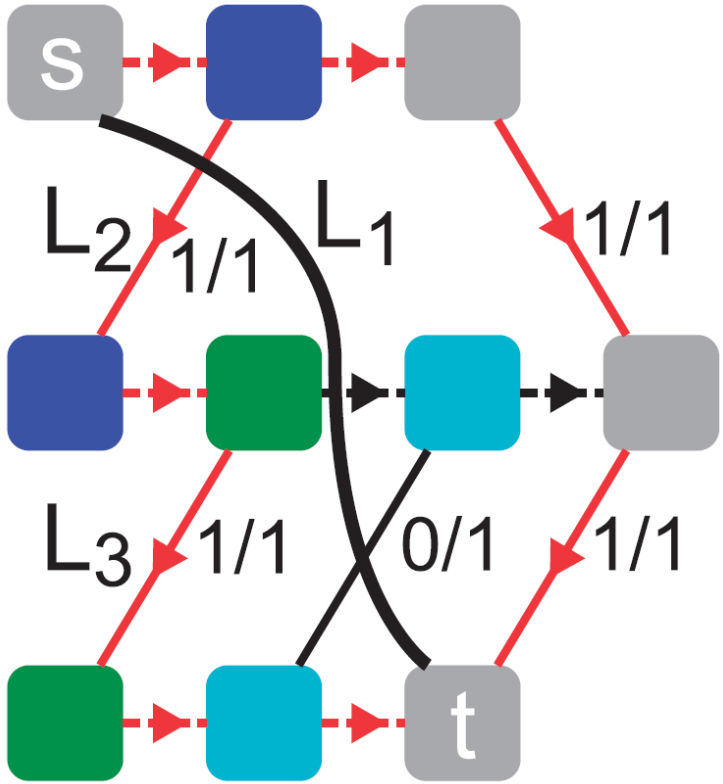
- Consider a situation that no alignable set S can be found among the N active nodes in G.
- Let $G' = (V, E')$ be the subgraph that only contains the active links.

Proposition

- *If no alignable set can be found among the N active nodes in G during the basic alignment procedure, then $\exists$ one conflict among the active links.*
- *Furthermore, no alignable set can be found among the same active nodes in G'.*

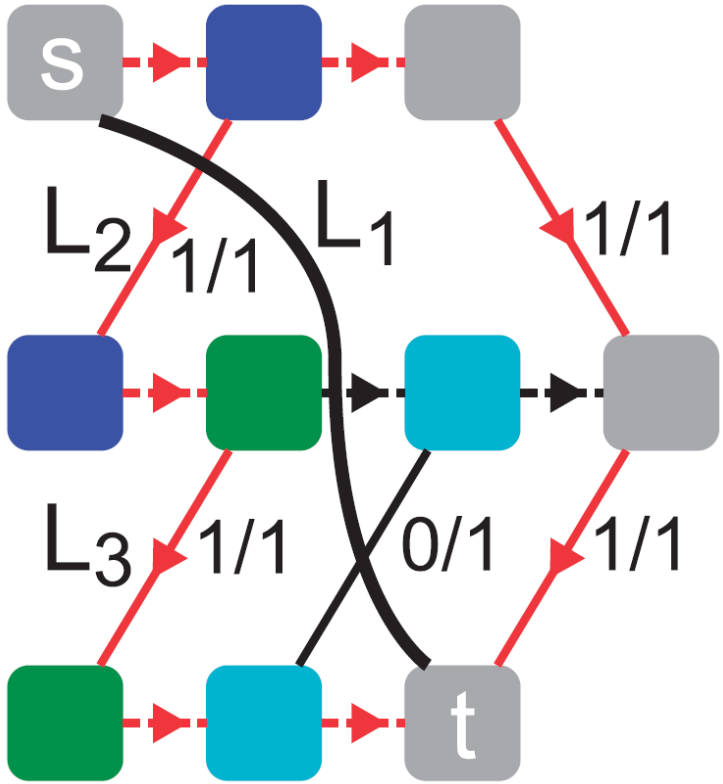
Heuristic by maximum flow

- For a link L_{st} (linking s and t), we want to assess *which degree they are connected*
  $\Rightarrow$ maximum flow f_{st}.
- Restrictions:
  - Only passing through element blocking/direct paths $s \longrightarrow t$.
  - Links: *no direction* & *capacity equal to their weight* w.
  - Edges: keep their directions & unlimited capacities.



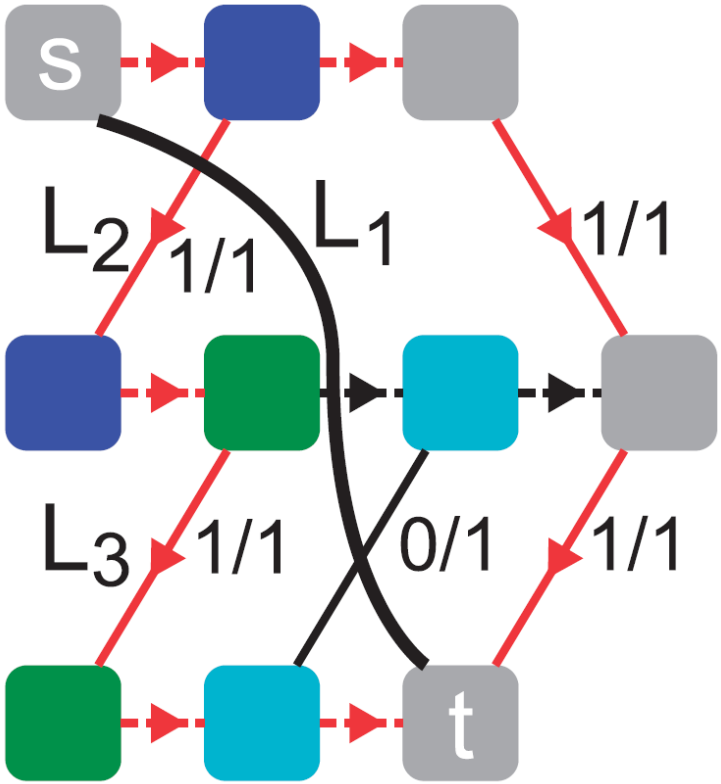
Heuristic by maximum flow

- f_{st}^D (resp., f_{st}^C): the max flow of $s \longrightarrow t$ through directed elementary paths (resp., elementary blocking paths).
- $f_{st}^C = f_{st} - f_{st}^D$.
- Let $S_{L_{st}} = f_{st}^D - |f_{st}^C - f_{ts}^C|$.
- The link L with the lowest score S_L will be removed.



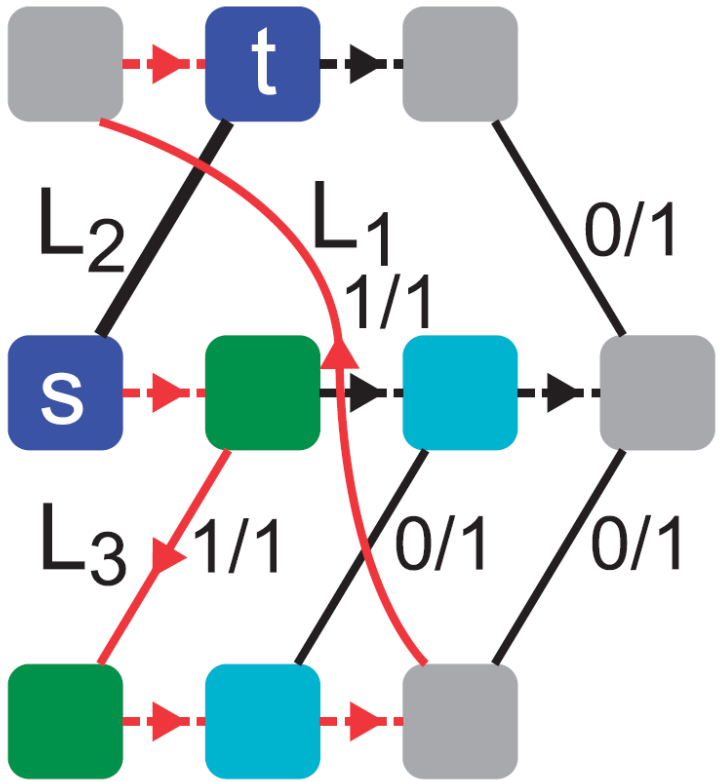
Heuristic by maximum flow

- $f_{st}^{C} = 2$ for link L_1.
- $f_{ts}^{C} = 0$.
- $f_{st}^{D} = 1$.
- $\therefore S_{L_1} = 1 - 2 = -1$.



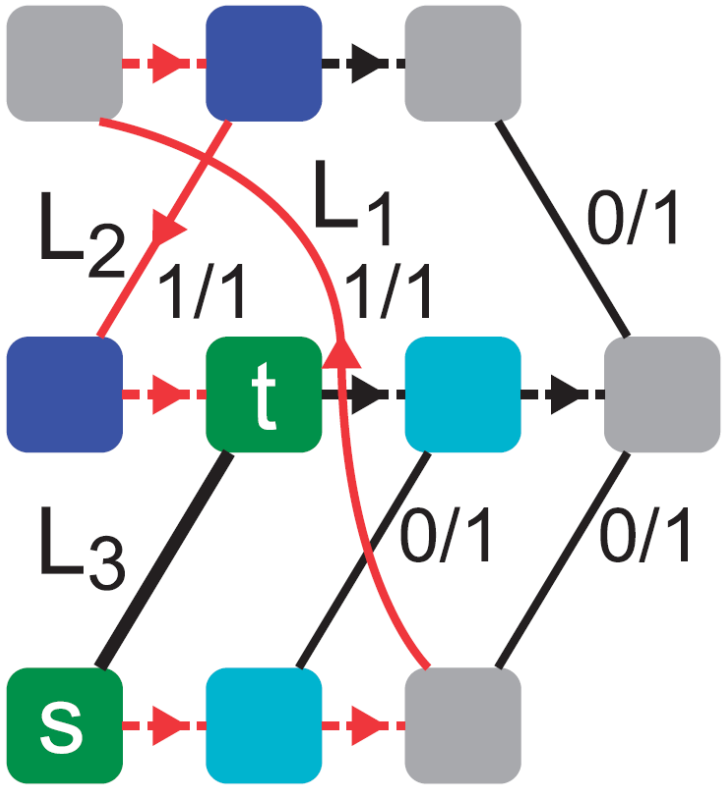
Heuristic by maximum flow

- $f_{st}^{C} = 1$ for link L_2.
- $f_{ts}^{C} = 0$.
- $f_{st}^{D} = 1$.
- $\therefore S_{L_2} = 1 - 1 = 0$.

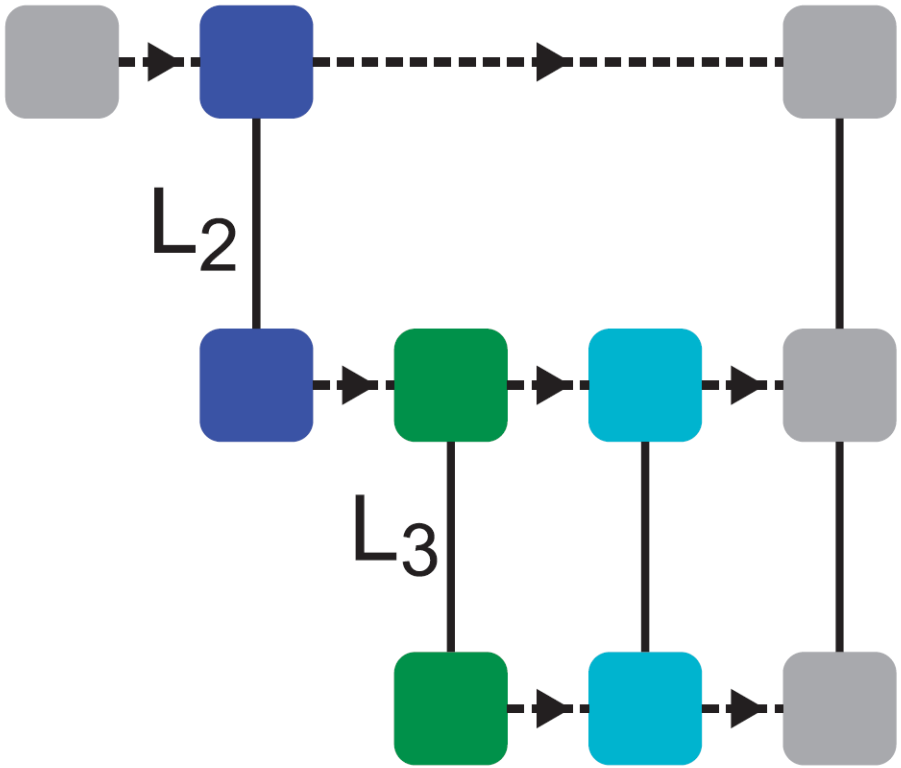


Heuristic by maximum flow

- $f_{st}^{C} = 1$ for link L_3.
- $f_{ts}^{C} = 0$.
- $f_{st}^{D} = 1$.
- $\therefore S_{L_3} = 1 - 1 = 0$.
- Thus we remove the link L_1.



Heuristic by maximum flow

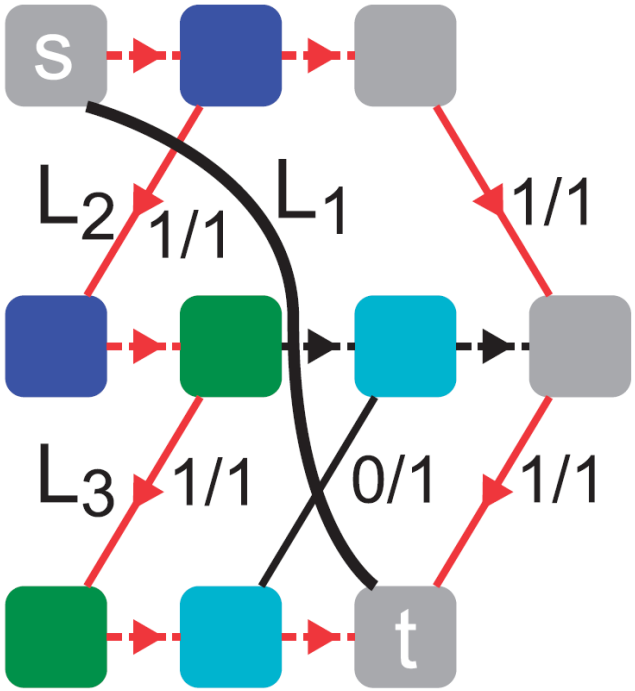


Faster heuristics

- The computation of the max flow (f): $O(Ef)$ time (Ford & Fulkerson 1962).
- Seeking for upper bounds on f^C_{st} is easier.
  - Find $f^C_{st,\mathtt{UB}}$ and $f^C_{ts,\mathtt{UB}}$.
- Then we derive a lower bound on $S_{L_{st}}$:

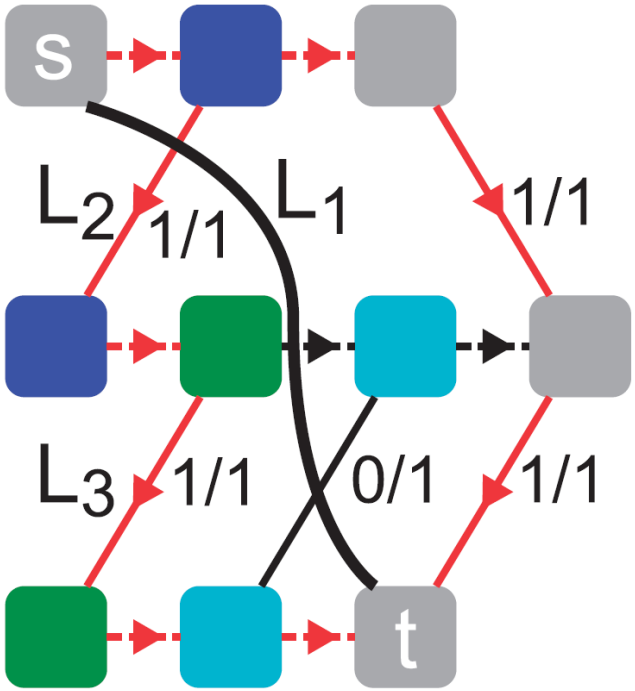
$$S_{L_{st},\mathtt{LB}} = f^D_{st,\mathtt{LB}} - \max\{f^C_{st,\mathtt{UB}}, f^C_{ts,\mathtt{UB}}\},$$

  where we set $f^D_{st,\mathtt{LB}} = w(L_{st})$.
- Select the active link L with the lowest lower bound $S_{L,\mathtt{LB}}$.



Faster heuristics (contd.)

- Select the "longest link".
  - Let L_{st} be an active link with $t = n_{j,k}$.
  - Select the active link incident to $n_{j,k}$ for which $k - a_j$ is maximal.



Summary of the heuristics

To resolve a conflict, select one of the following **active** link for removal:

RA (RAndom)	a random link
RC (Random Conflict)	a random link involved in at least one (active or non-active) conflict
RAC (Random Active Conflict)	a random link involved in at least one *active* conflict
LL (Longest Link)	the link L, involved in at least one active conflict, incident to node $n_{j,k}$ for which $(k - a_j)$ is maximum.
LLBS (Lowest Lower Bound Score)	the link L, involved in at least one active conflict, with the lowest lower bound score $S_{L,\mathrm{LB}}$.
LS (Lowest Score)	the link L, involved in at least one active conflict, with the lowest score S_L.

Datasets

Dataset I:

Arabidopsis thaliana genome (2000)

- ▷ It contains both strongly diverged and more closely related homologous chromosomal regions.
- Running i-ADHoRe on the genome separately.
- 921 sets of homologous genome segments are produced.
- N: varies from 2 to 11.

Dataset II:

Genomes of
Arabidopsis thaliana,
Populus trichocarpa (Tuskan *et al.*, 2006) &
Vitis vinifera (by Jaillon *et al.*, 2007).

- Running i-ADHoRe on the genomes.
- 7821 sets of homologous genome segments are produced.
- N: varies from 2 to 15.

Table 1. Comparison of the number of correctly aligned homologous genes in Dataset I.

N	Number of input sets	$\sum$ number of correctly aligned homologous genes							
				GG2: the proposed greedy, graph-based aligner					
		pNW	GG	RA	RC	RAC	LL	LLBS	LS
2	447	4877	4719	4108	4109	4109	4843	4871	4874
3	169	2823	2704	2477	2544	2574	2810	2817	2852
4	119	2924	2769	2502	2611	2684	2921	2971	3008
5	49	1634	1533	1375	1454	1514	1627	1655	1697
6	41	1715	1516	1375	1485	1577	1773	1747	1814
7	39	2114	1803	1572	1732	1884	2149	2152	2275
8	24	1229	1062	882	995	1094	1278	1263	1358
9	16	807	713	623	725	773	880	885	921
10	13	703	670	602	704	741	810	822	825
11	4	211	228	203	239	246	263	259	259
$\sum$	921	19 037	17 717	15 719	16 596	17 196	19 354	19 442	19 883

Table 2. Comparison of alignment scores and run times for Dataset II

Alignment method	Alignment score	Alignment time (s)
pNW	518 247	1.7
GG	497 771	5.9
RA	489 020	6.2
RC	501 241	5.2
RAC	505 447	3.3
LL	525 665	2.3
LLBS	525 652	2.2
LS	529 633	6742

- LLBS is used as the default heuristic for i-ADHoRe.
- Experiments on an extremely large dataset:
  - Dozens of eukaryotic speices (Hubbard *et al.*, 2009).
  - This method can handle $N = 50$ (enough for practical uses).

Thank you.