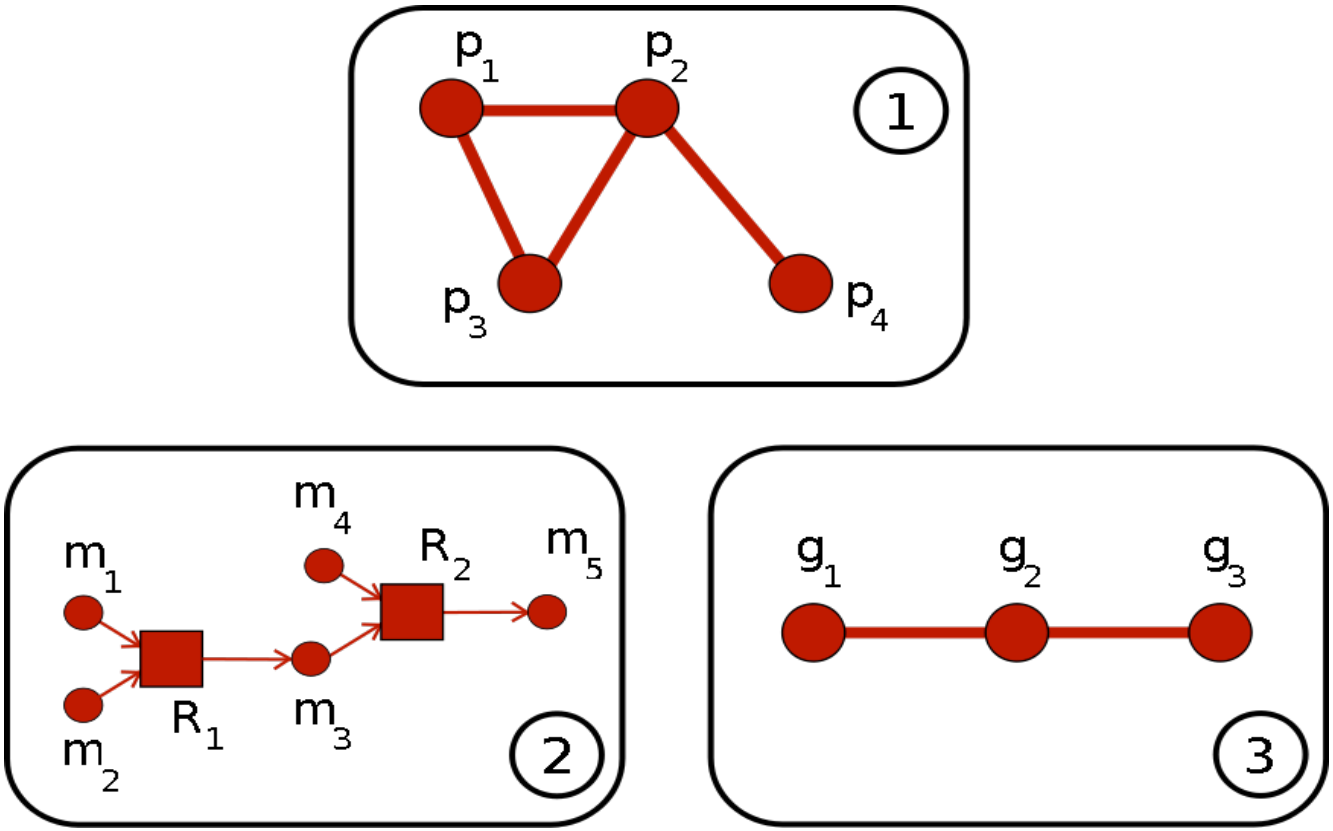
CPM' 09
20th Annual Symposium on Combinatorial Pattern Matching
June 22-24 2009, Lille, France
Special Anniversary Edition

Multiple alignment of biological networks: a flexible approach

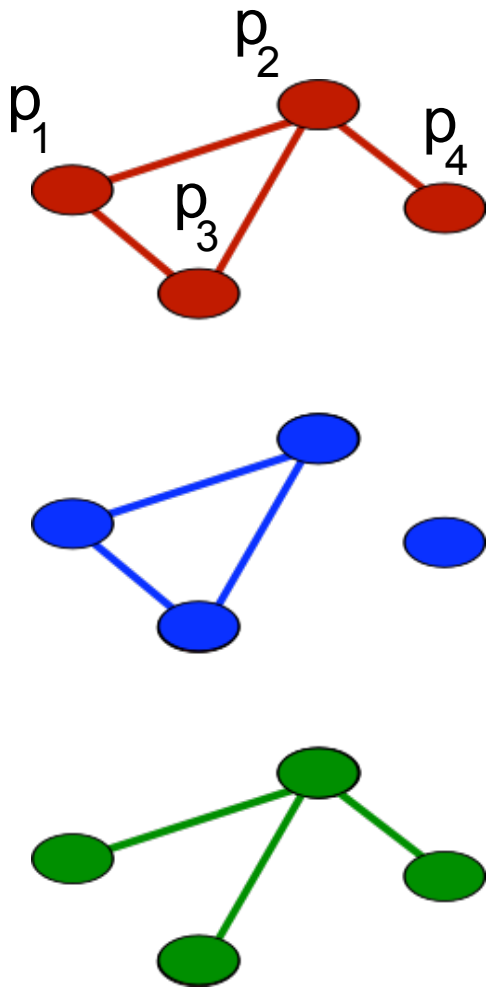
Yves-Pol Denielou, Fréderic Boyer,
Marie-France Sagot, Alain Viari
Bamboo team, Inria Rhône-Alpes

Motivation



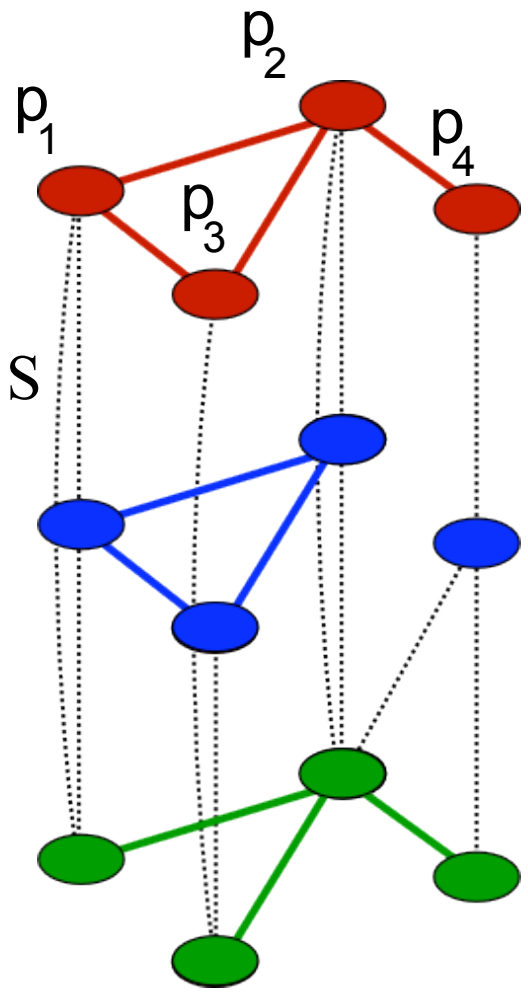
Aligning Networks

- **Local or Global**
- **Pairwise or Multiple**



Aligning Networks

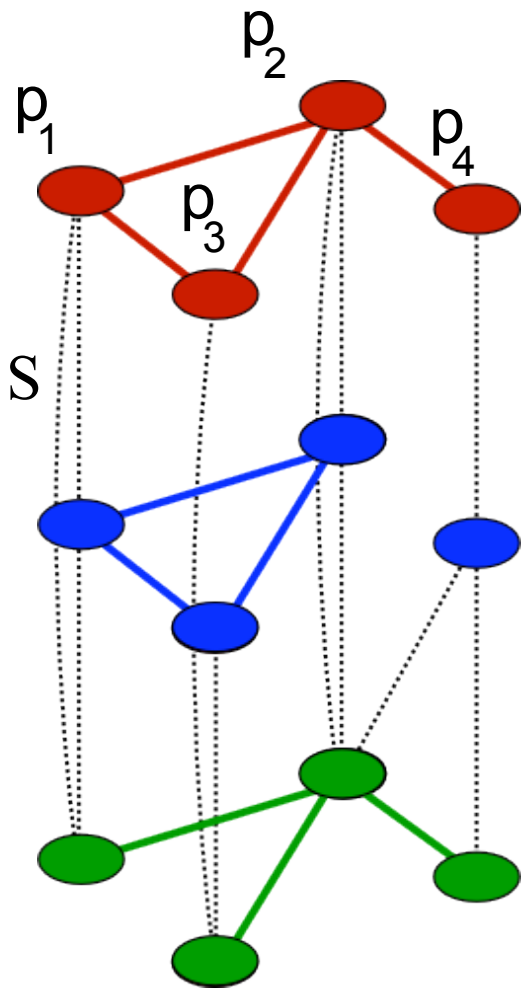
- **Local or Global**
- **Pairwise or Multiple**
- **Definition of a correspondence relation S between vertices of the different networks**



Aligning Networks

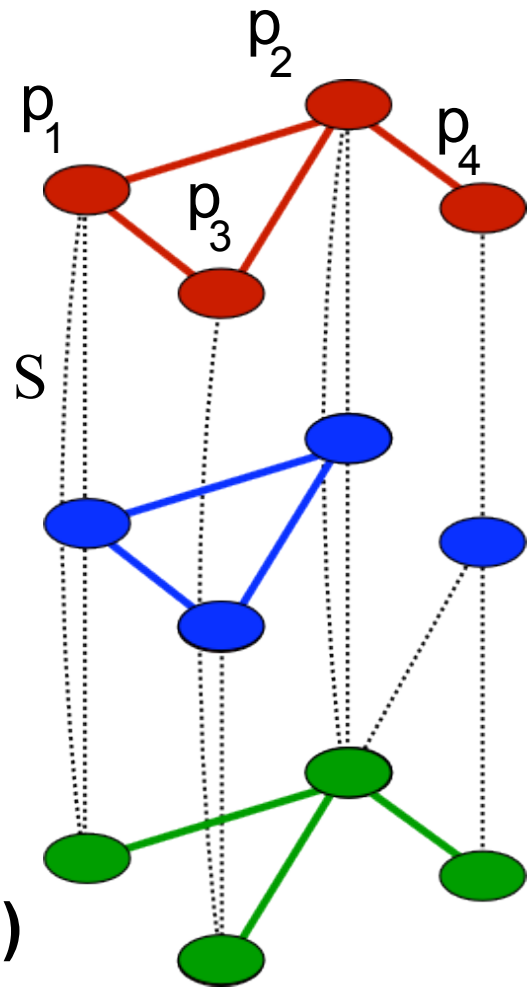
- **Local or Global**
- **Pairwise or Multiple**
- **Definition of a correspondence relation S between vertices of the different networks**

➢ **Extract conserved subnetworks**



Aligning Networks

- **Local or Global**
- **Pairwise or Multiple**
- **Definition of a correspondence relation S between vertices of the different networks**



➢ **Extract conserved subnetworks**

➢ **A merged representation :**

the Network Alignment Graph (NAG)

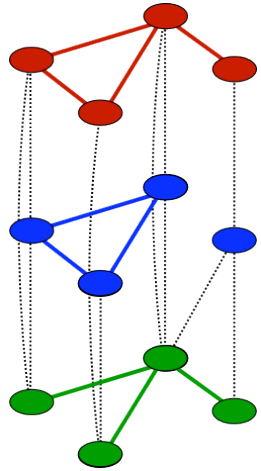
Sharan et al 2006

Network Alignment Graph

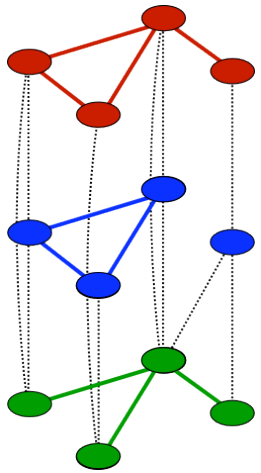
Given :

- n networks Gi = (Vi, Ei) i = 1..n

- a relation S between the vertices of V = $\cup_i$ Vi



Network Alignment Graph

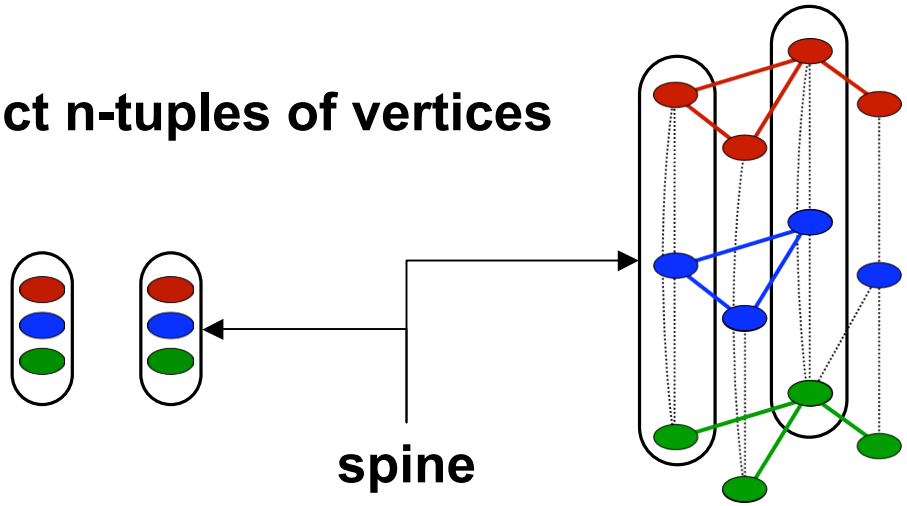


Given :

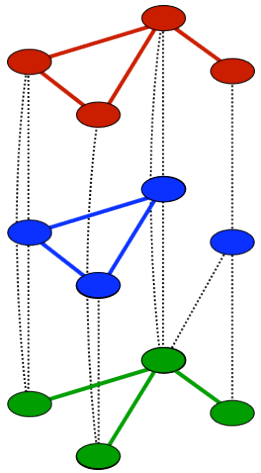
- n networks Gi = (Vi, Ei) i = 1..n

- a relation S between the vertices of V = $\cup_i V_i$

- **Select n-tuples of vertices**

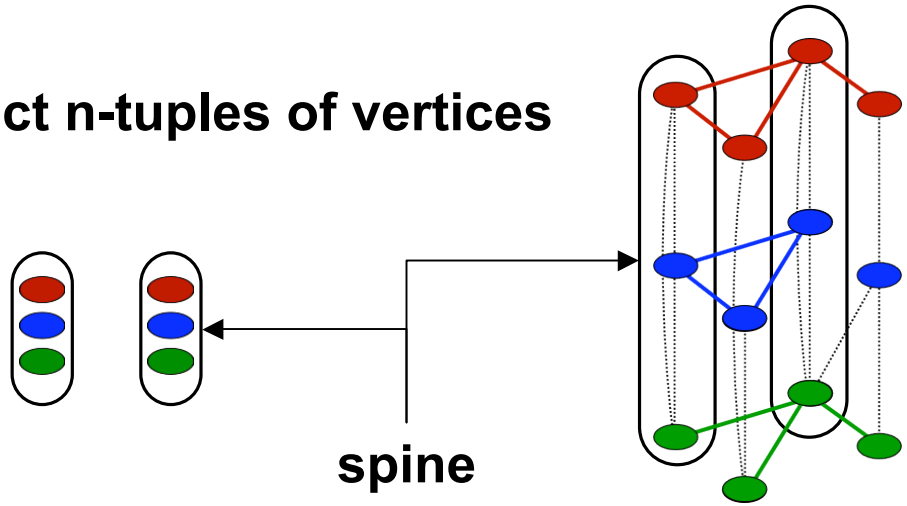


Network Alignment Graph

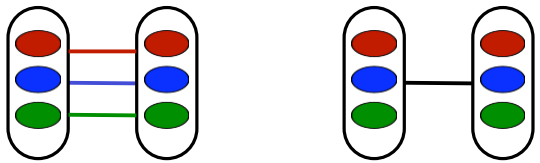


Given :

- n networks Gi = (Vi, Ei) i = 1..n
- a relation S between the vertices of V = $\bigcup_i V_i$

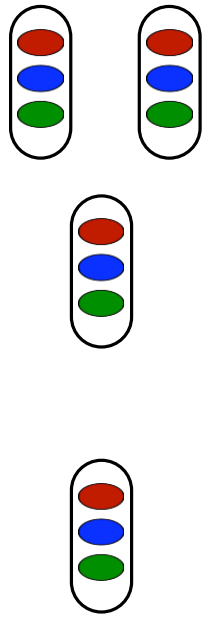


- Select n-tuples of vertices
- Connect these spines with edges from the sets Ei



3 main choices

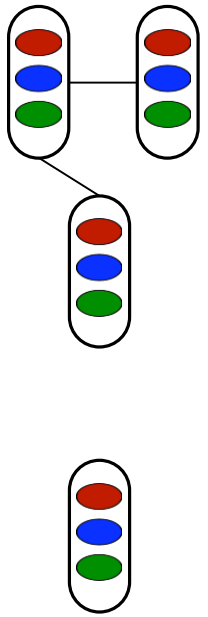
1- How to select n-tuples using S ?



3 main choices

1- How to select n-tuples using S ?

2- Which primary edges to keep ?

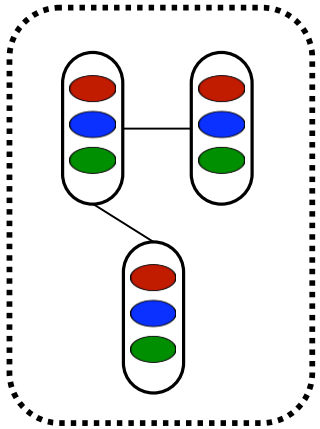


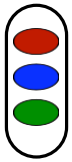
3 main choices

1- How to select n-tuples using S ?

2- Which primary edges to keep ?

3- Which topological condition on the subnetworks ?

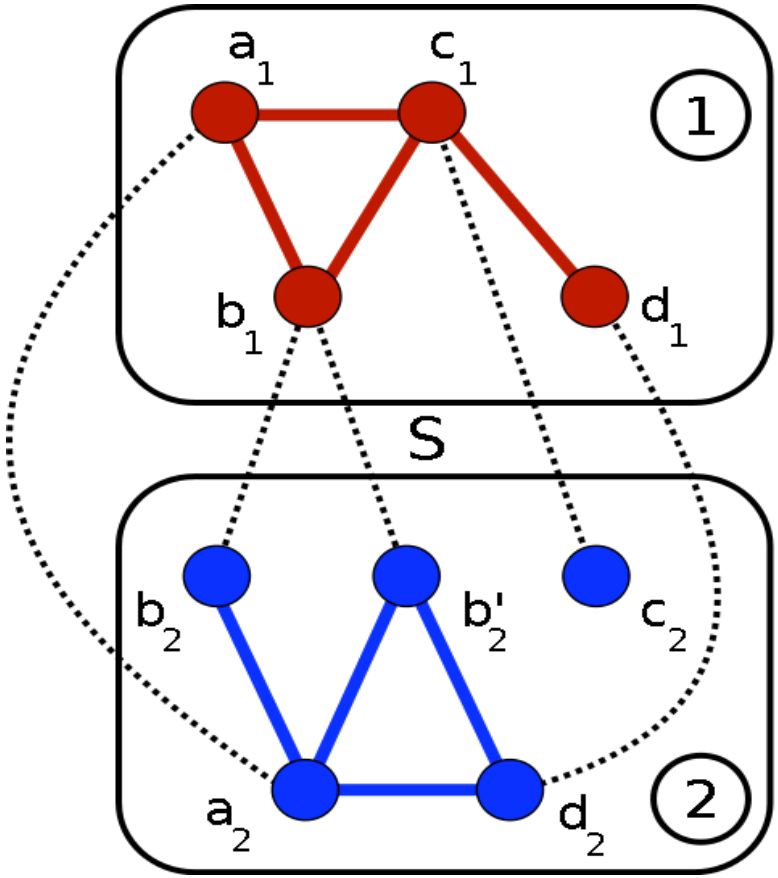




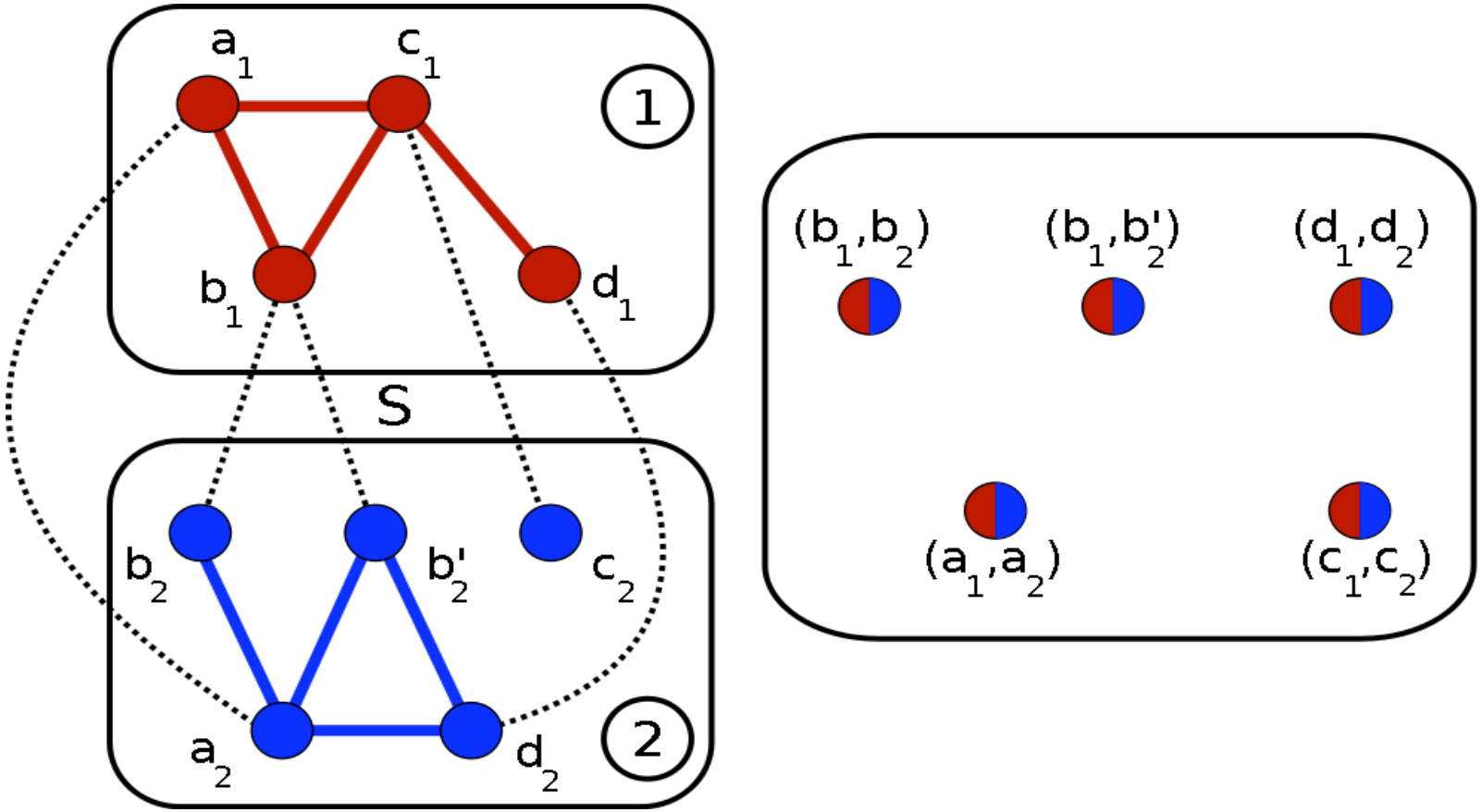
State of the art

Method	n-tuples construction	edges construction	topology
PathBLAST	pairs (n=2)	conserved	paths
NetworkBLAST	CC ($n \leq 3$)	conserved	clusters / paths
NetworkBLAST-M	paths	all edges	dense clusters on each network
GRAEMLIN 1.0	non-overlap--ping CC	conserved	user-defined
CAPPI	non-overlap--ping CC	all edges > threshold	CC
HopeMap	pairs	conserved	CC
MaWISH	pairs	conserved	max-weight subgraph
PHUNKEE	pairs	all edges	max shared-edges ratio
C3PART	cliques or stars	all edges	common CC

NAG: Pairwise case

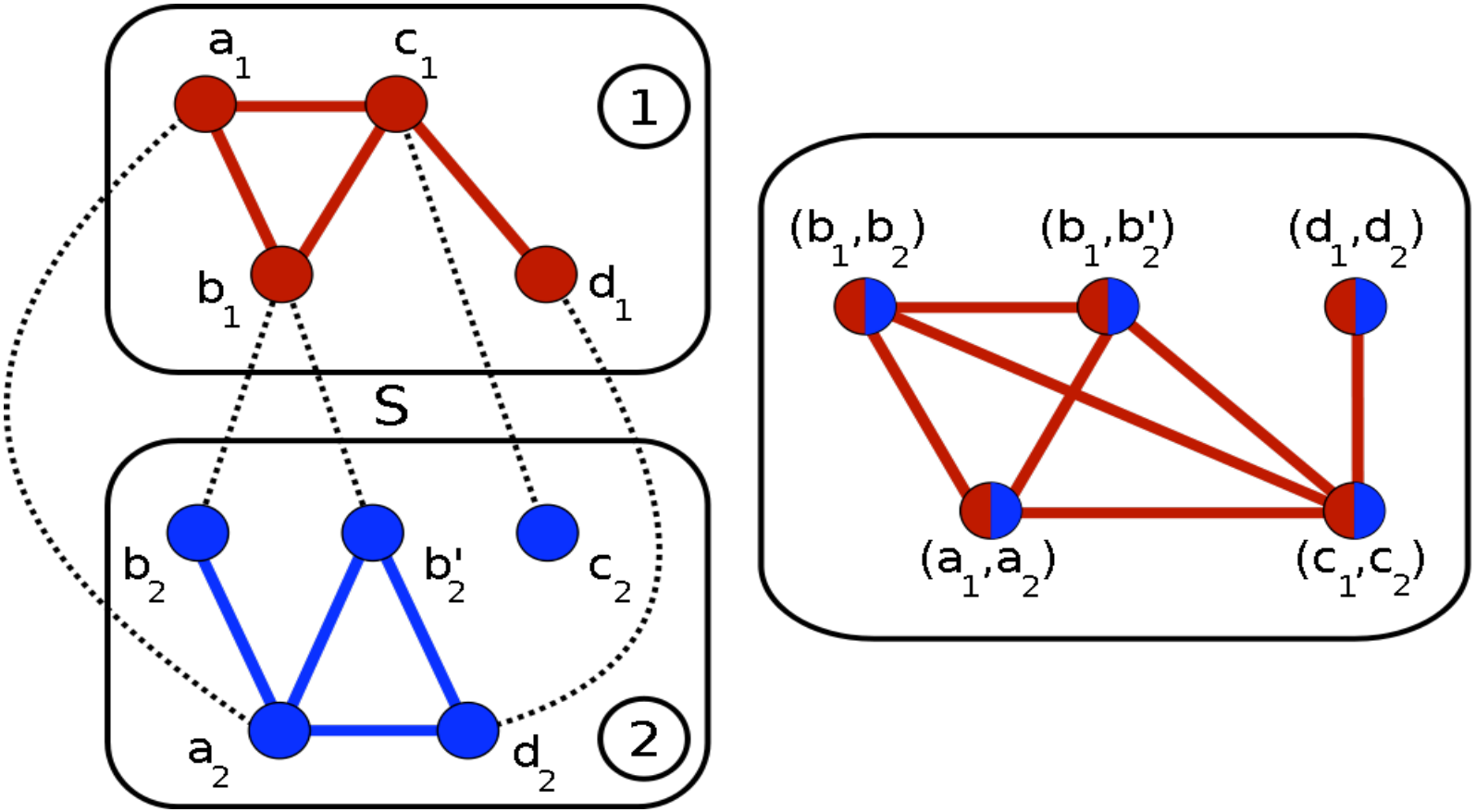


NAG: Pairwise case



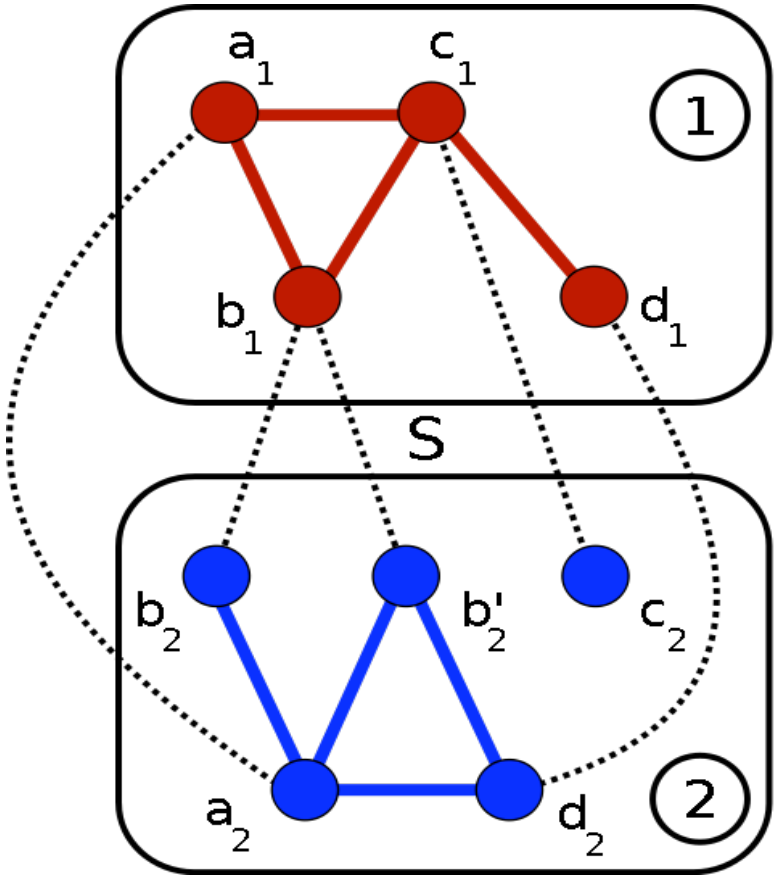
1: all S-related pairs

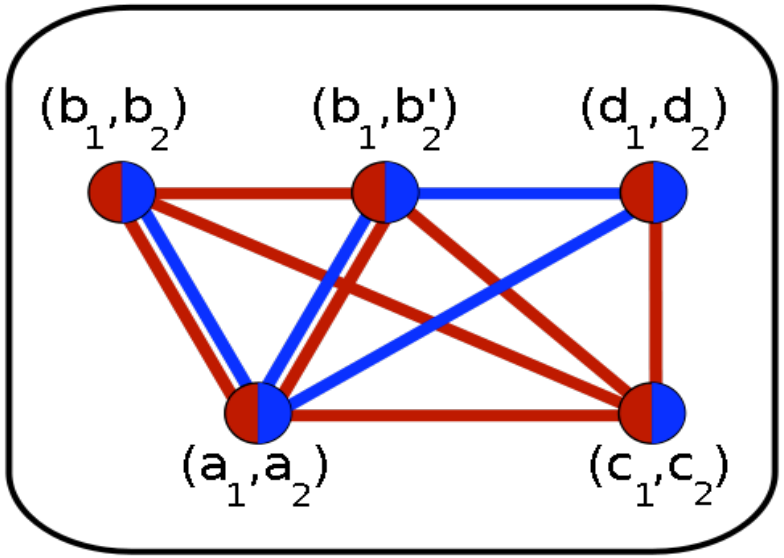
NAG: Pairwise case



1: all S-related pairs
2: keep all primary edges (multigraph)

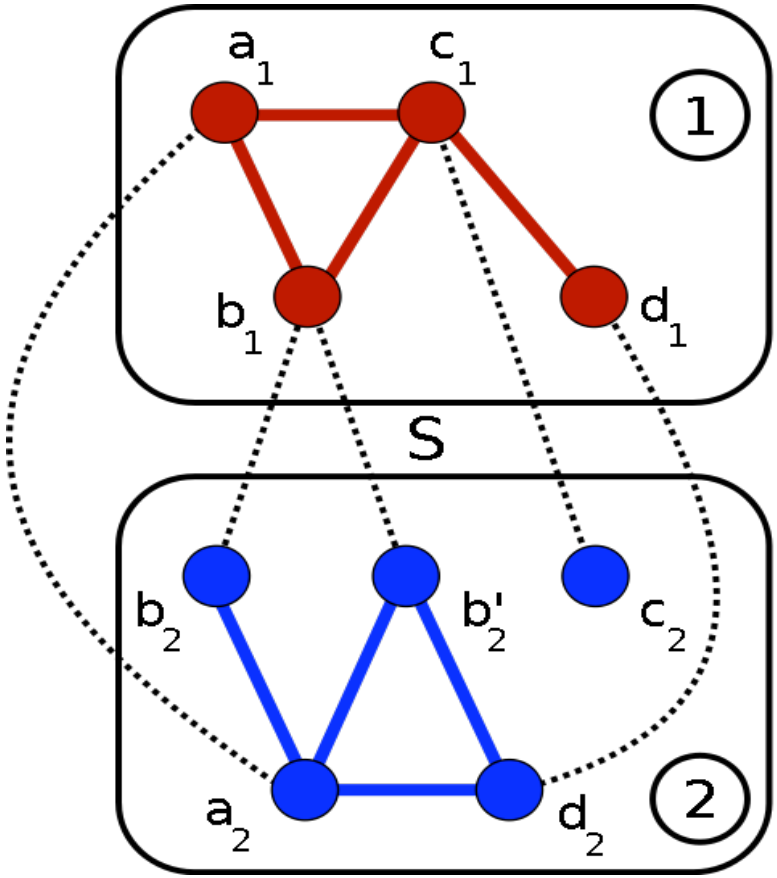
NAG: Pairwise case

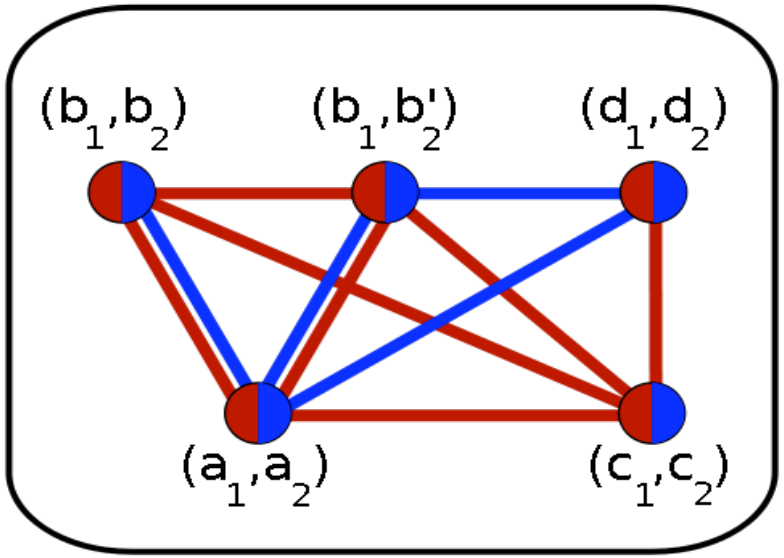




1: all S-related pairs
2: keep all primary edges (multigraph)

NAG: Pairwise case

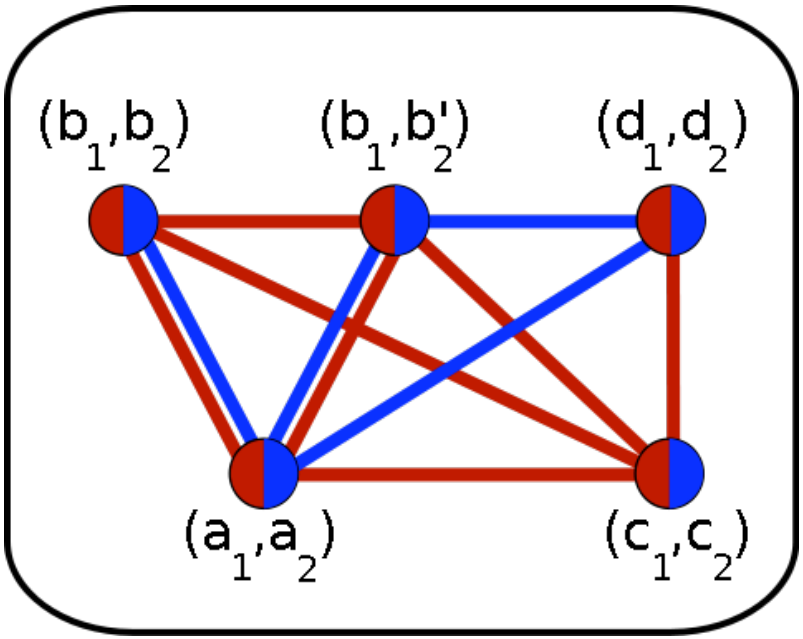




1: all S-related pairs
2: keep all primary edges (multigraph)
3: connected in each network

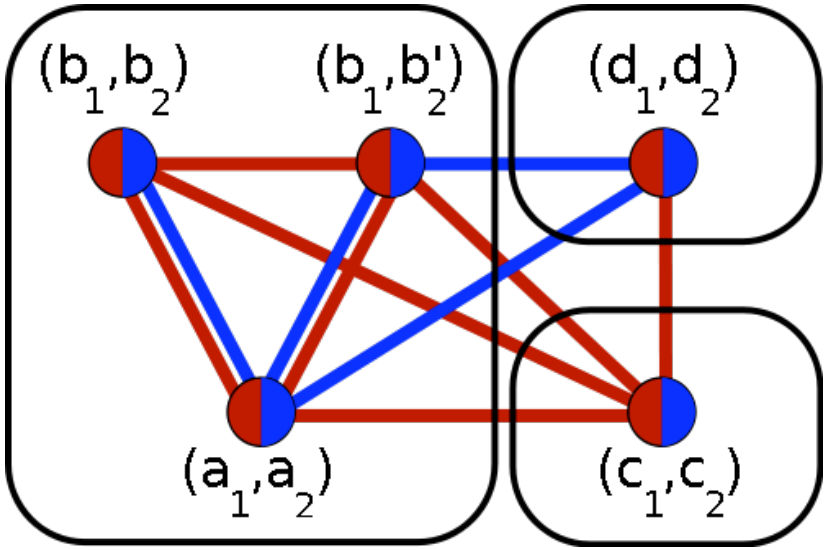
Defining Connectons

Connectons: maximal sets of vertices in the NAG connected on each network/color



Defining Connectons

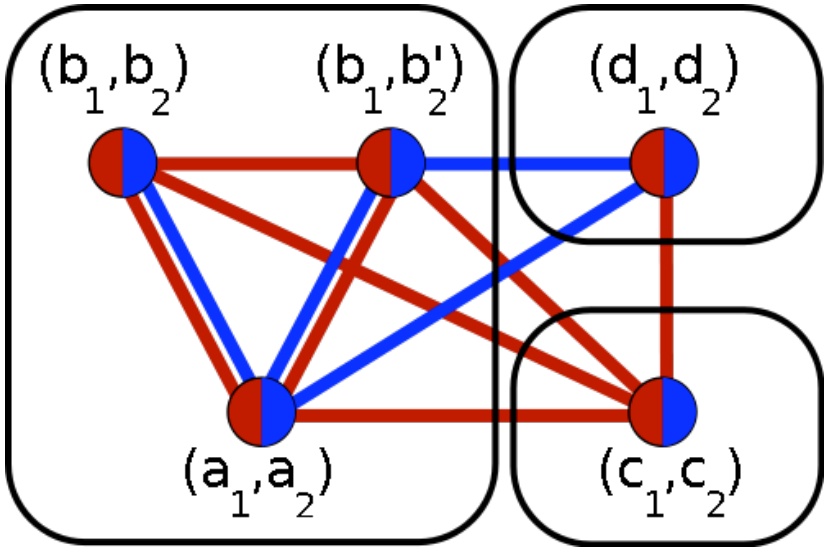
Connectons: maximal sets of vertices in the NAG connected on each network/color



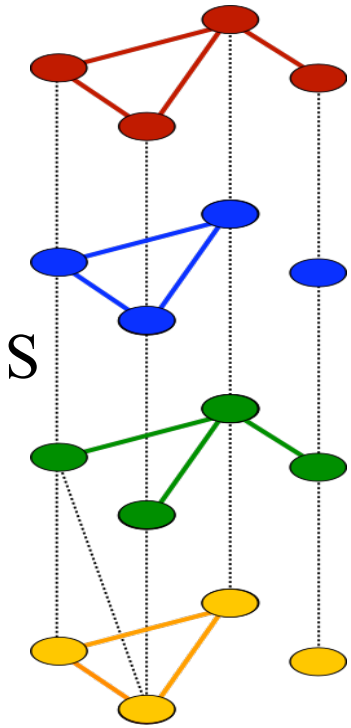
Defining Connectons

Connectons: maximal sets of vertices in the NAG connected on each network/color

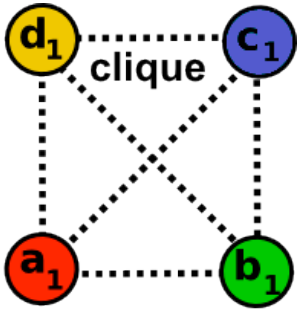
Connectons = Partition of the NAG vertices



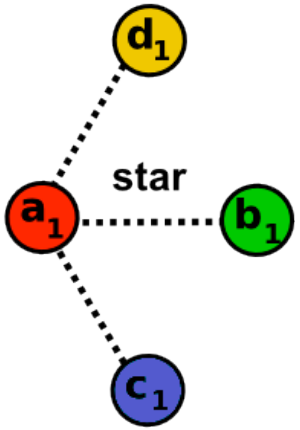
Extension to n networks

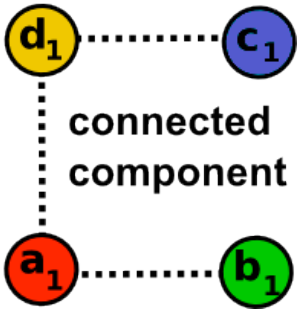


Selecting n-tuples :



(a_1, b_1, c_1, d_1)





in general :
S not one-to-one
S not transitive

Algorithm

Computing the connectons is doable, several partioning algorithms :

- **Boyer et al 2005 : O((N+M)*N) recursive intersection of CCs**
- **Gai et al 2003 : O((N+M*logN)*logN) Hopcroft-like partitioning approach**

Algorithm

Computing the connectons is doable, several partioning algorithms :

- **Boyer et al 2005 : O((N+M)*N) recursive intersection of CCs**
- **Gai et al 2003 : O((N+M*logN)*logN) Hopcroft-like partitioning approach**

But : N ~ |network|^{n} in general (because of S)

Algorithm

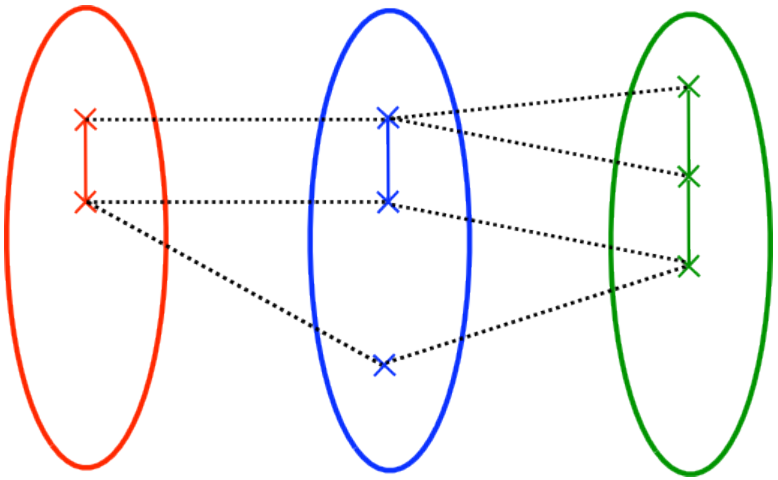
Computing the connectons is doable, several partioning algorithms :

- **Boyer et al 2005 : O((N+M)*N) recursive intersection of CCs**
- **Gai et al 2003 : O((N+M*logN)*logN) Hopcroft-like partitioning approach**

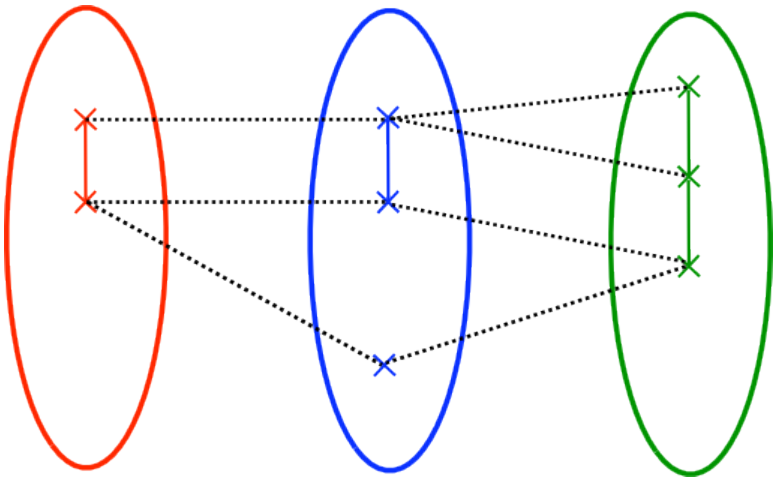
But : N ~ |network|^{n} in general (because of S)

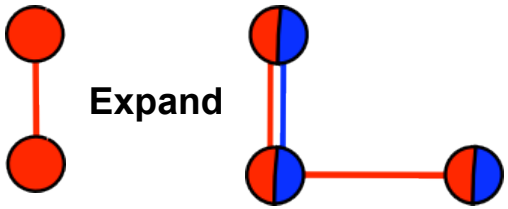
- **Avoid the explicit construction of the NAG**
- **On the fly partitionning**

On the fly expansion of NAG

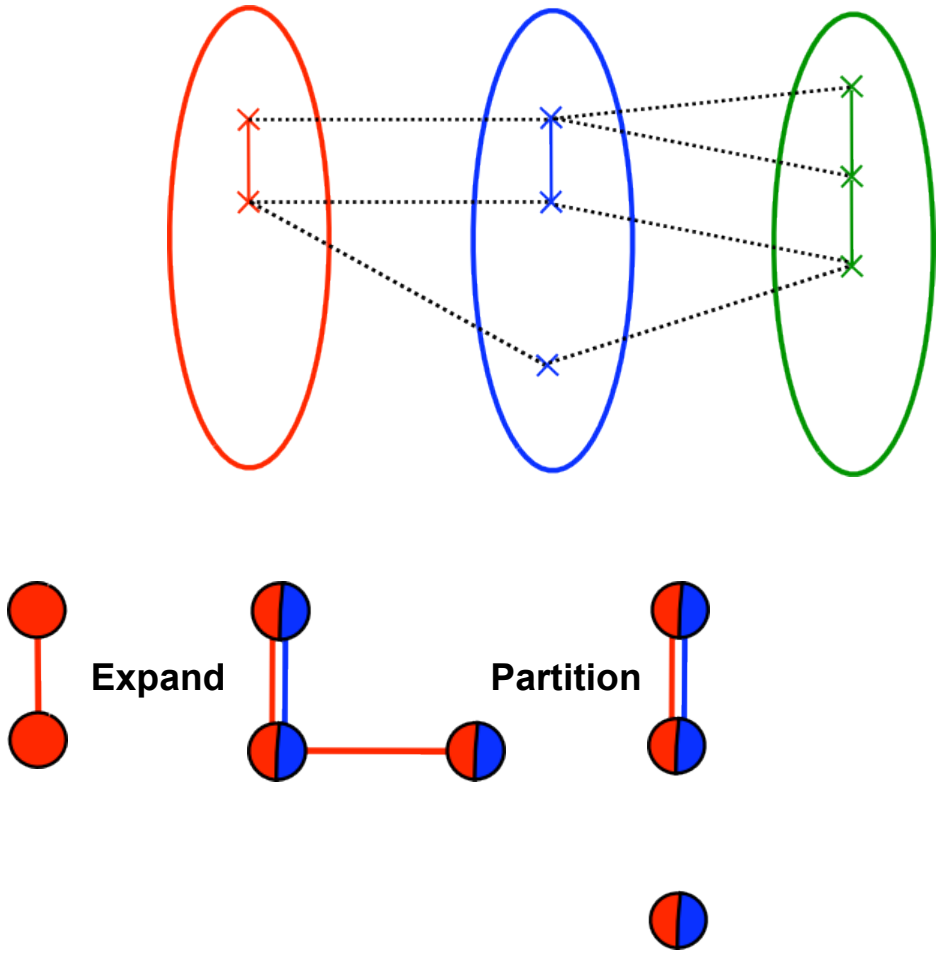


On the fly expansion of NAG

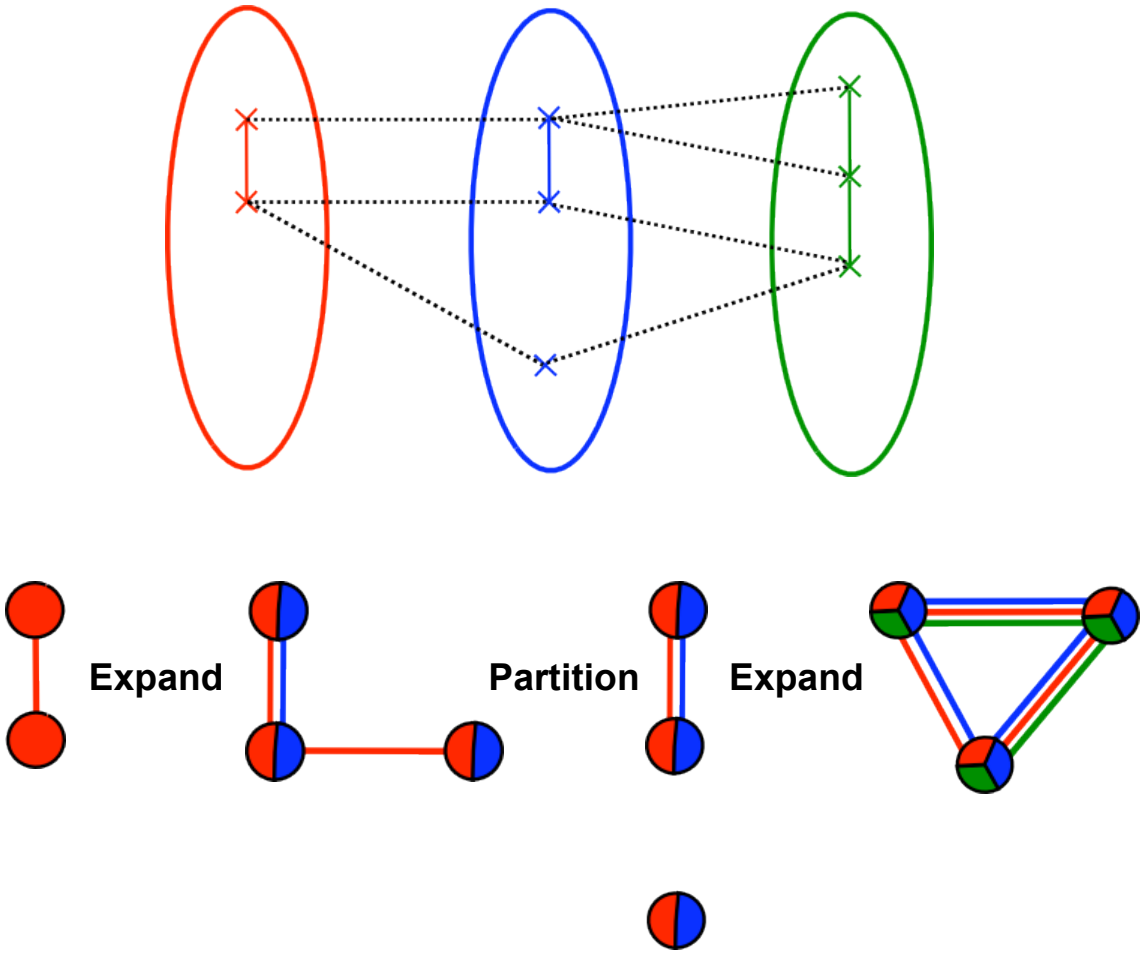




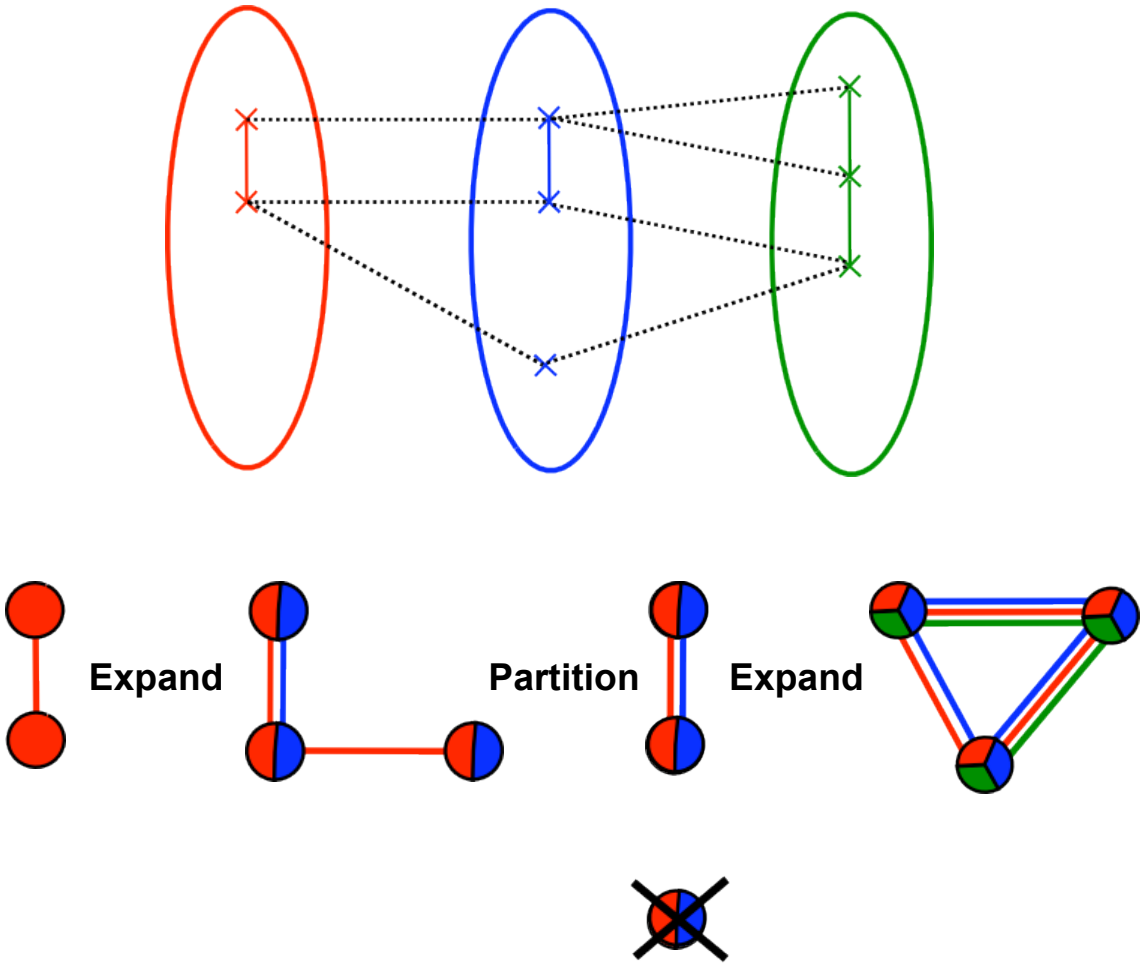
On the fly expansion of NAG



On the fly expansion of NAG



On the fly expansion of NAG



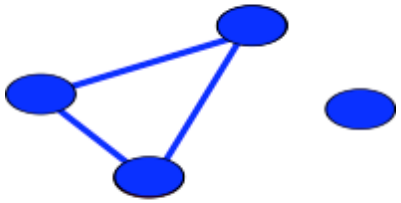
Results

Benchmark of 10 microbial PPI networks
from Srinivasan et al. 2006

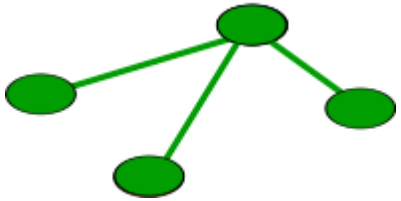
For a 0.5 threshold on the 10 networks :
N ~ 23000 proteins
E ~ 175000 interactions

Correspondence relation :
Sequence similarity with BlastP
Threshold : 10^{-10} Nb Hits /p : 5
~ 50000 edges

species A



species B



…

Selecting n-tuples : cliques / cc
Selecting edges : all edges
Topology : connectons

Comparison to Network Blast-M

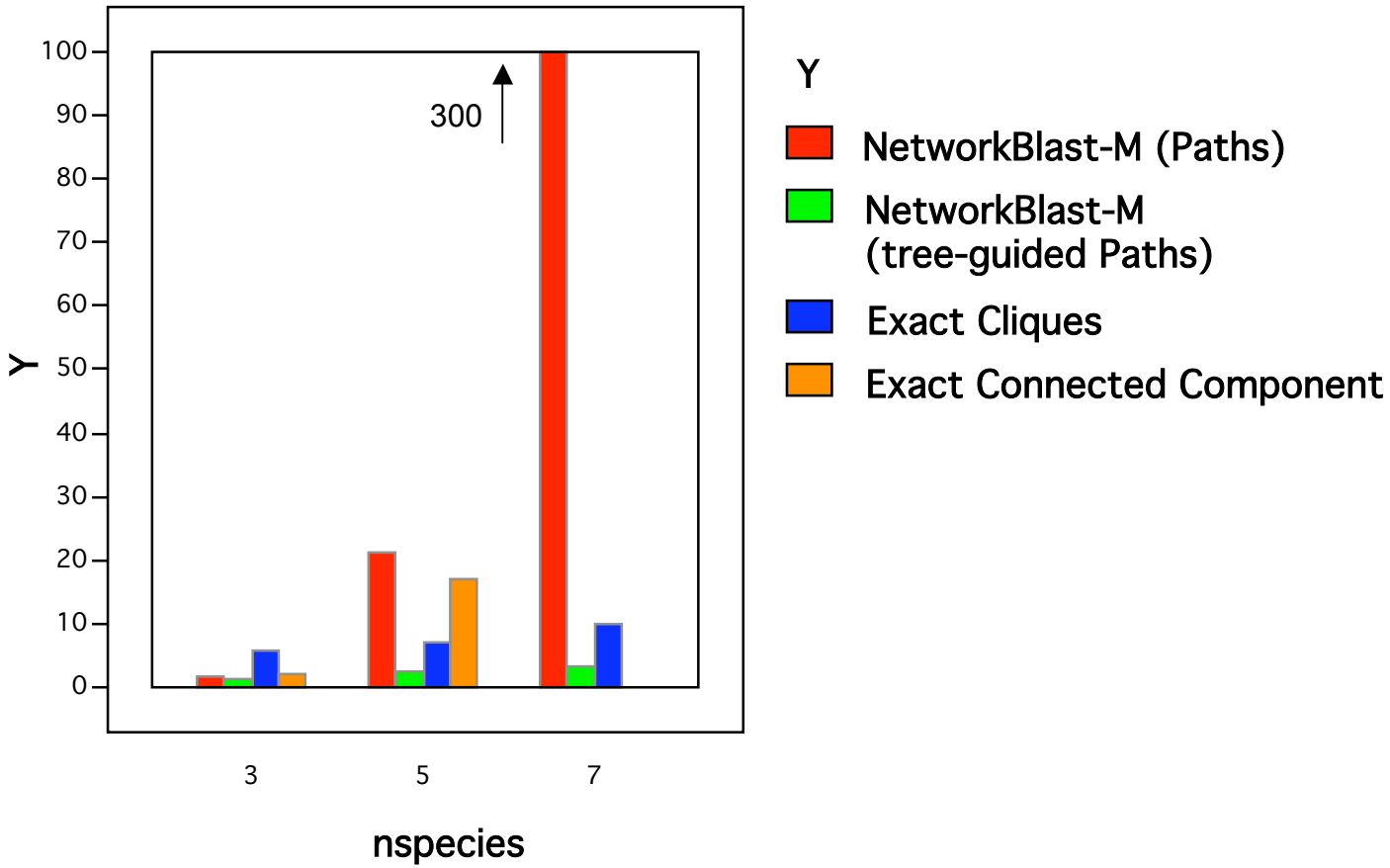
1 n-tuples : paths or tree-guided-paths
2 edges : all primary edges are kept
3 topology condition : dense in each network (*Sharan et al 2004*)

Heuristic algorithm

General idea:
- build a set of n-tuple seeds
- extend them greedily in order to maximise a sum of density scores, one per PPI network

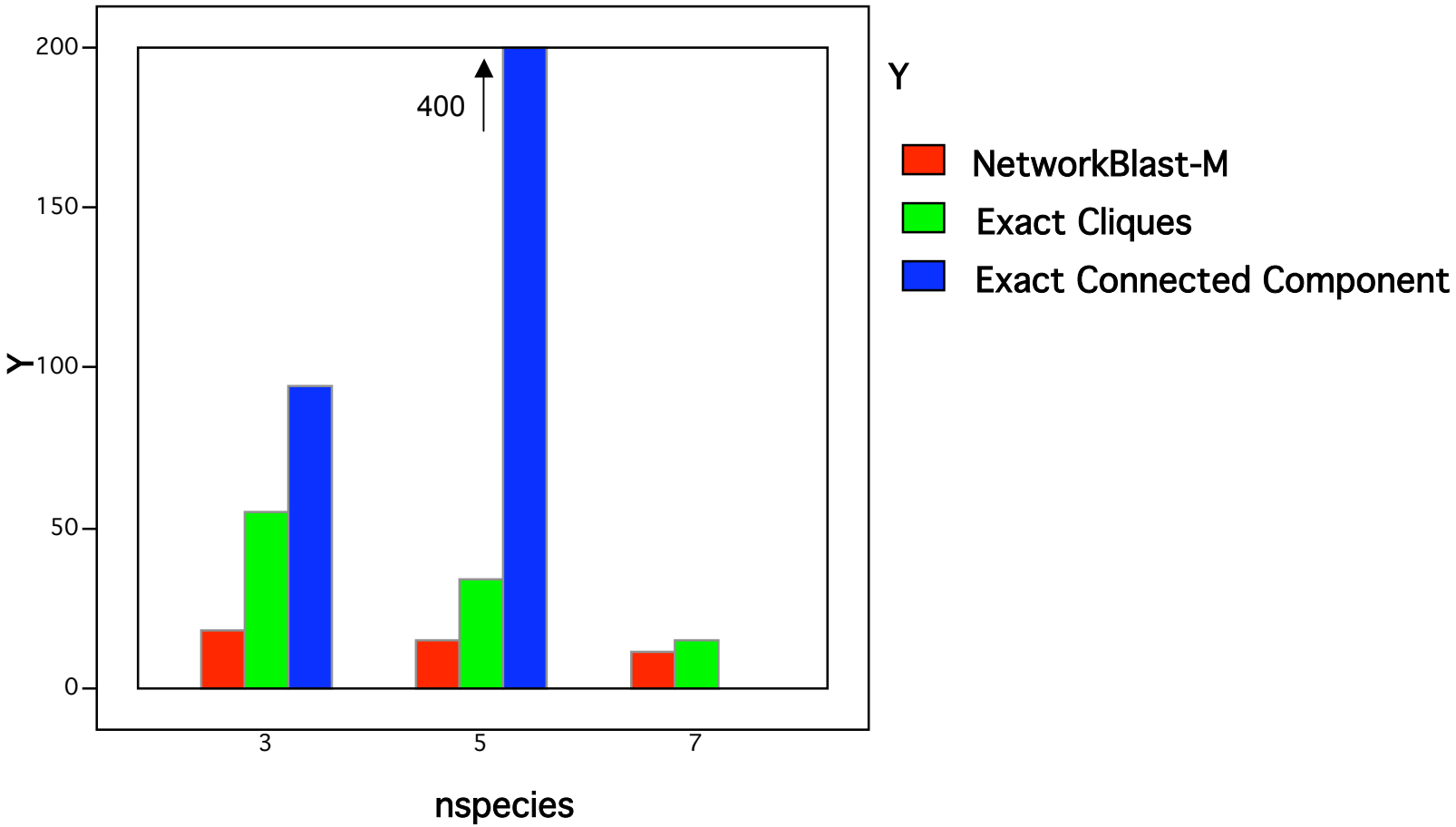
Comparison to NetworkBlast-M

-1- execution time



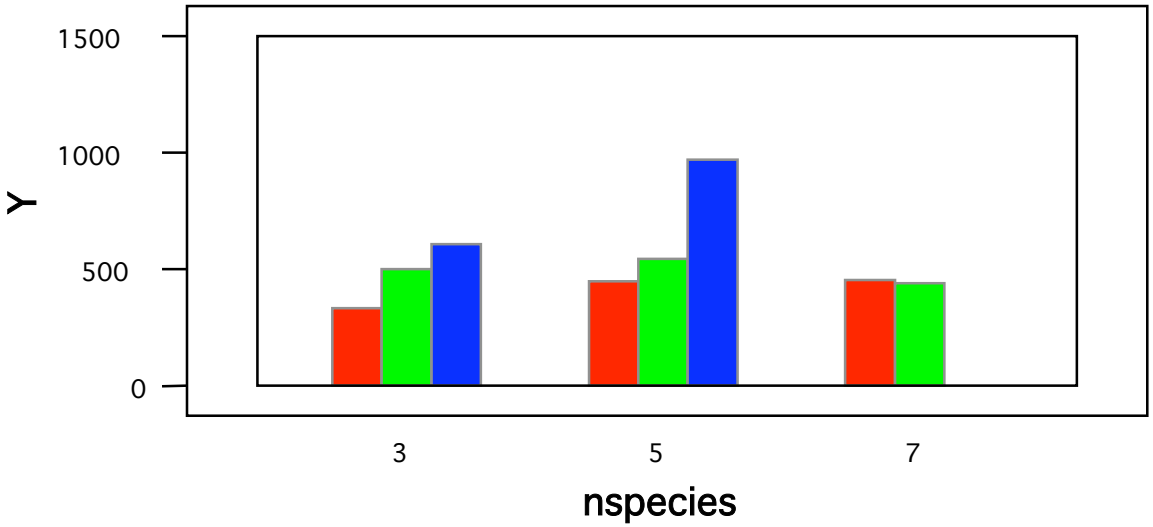
Comparison to NetworkBlast-M

-2- conserved subnetworks



Comparison to NetworkBlast-M

-3- proteins involved

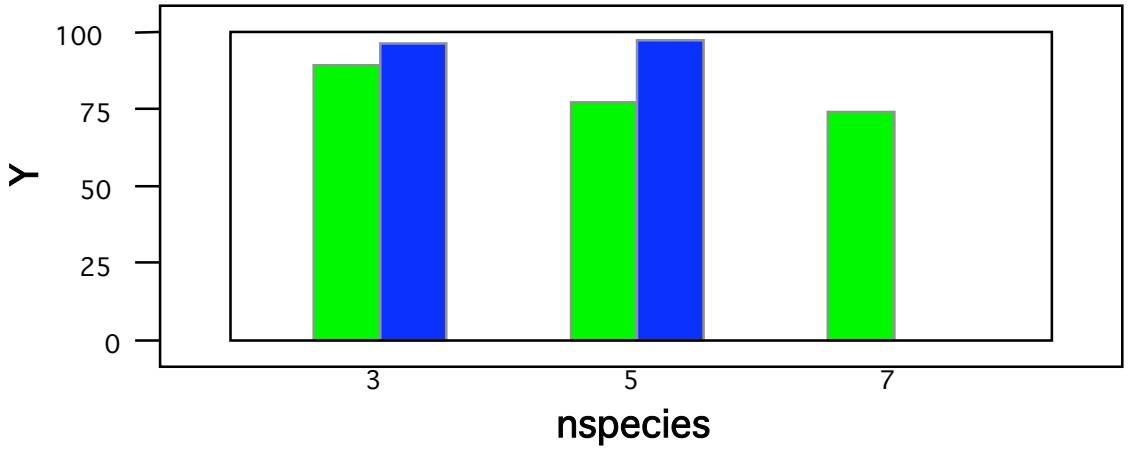


Y

NetworkBlast-M

Exact Cliques

Exact Connected Component



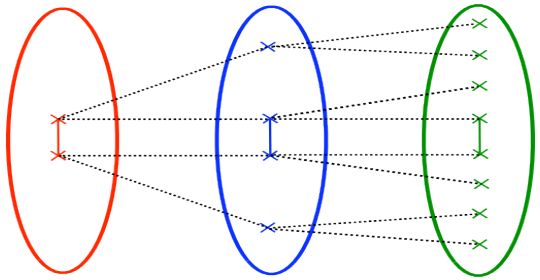
Y : %age of coverage of NetworkBlast-M results by

Exact Cliques

Exact Connected Component

Perspectives

- **Computing S correspondence on the fly**



- **Recovering connectons with missing vertices**

introducing k-tuples and quorum (q < k < n)