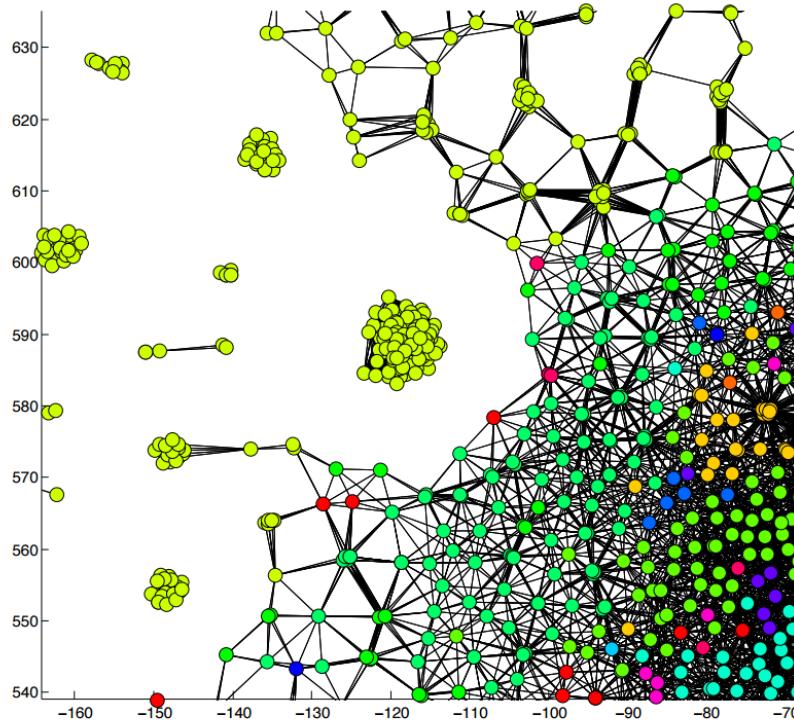


TreeScaper: Methods to Explore, Quantify, and Summarize Phylogenetic Tree Space



**Jeremy Ash, Wen Huang, Guifang Zhou, Melissa Marchand, Paul Van
Dooren, James Wilgenbusch, Kyle Gallivan, Jeremy M. Brown**

The Team



Jeremy Brown



David Morris



Kyle Gallivan



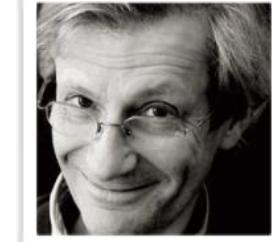
Guifang Zhou



Melissa Marchand



Wen Huang



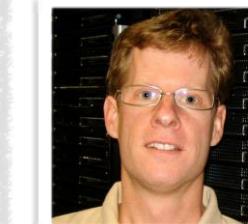
Paul Van Dooren



BELGIUM



Jeremy Ash



Jim Wilgenbusch



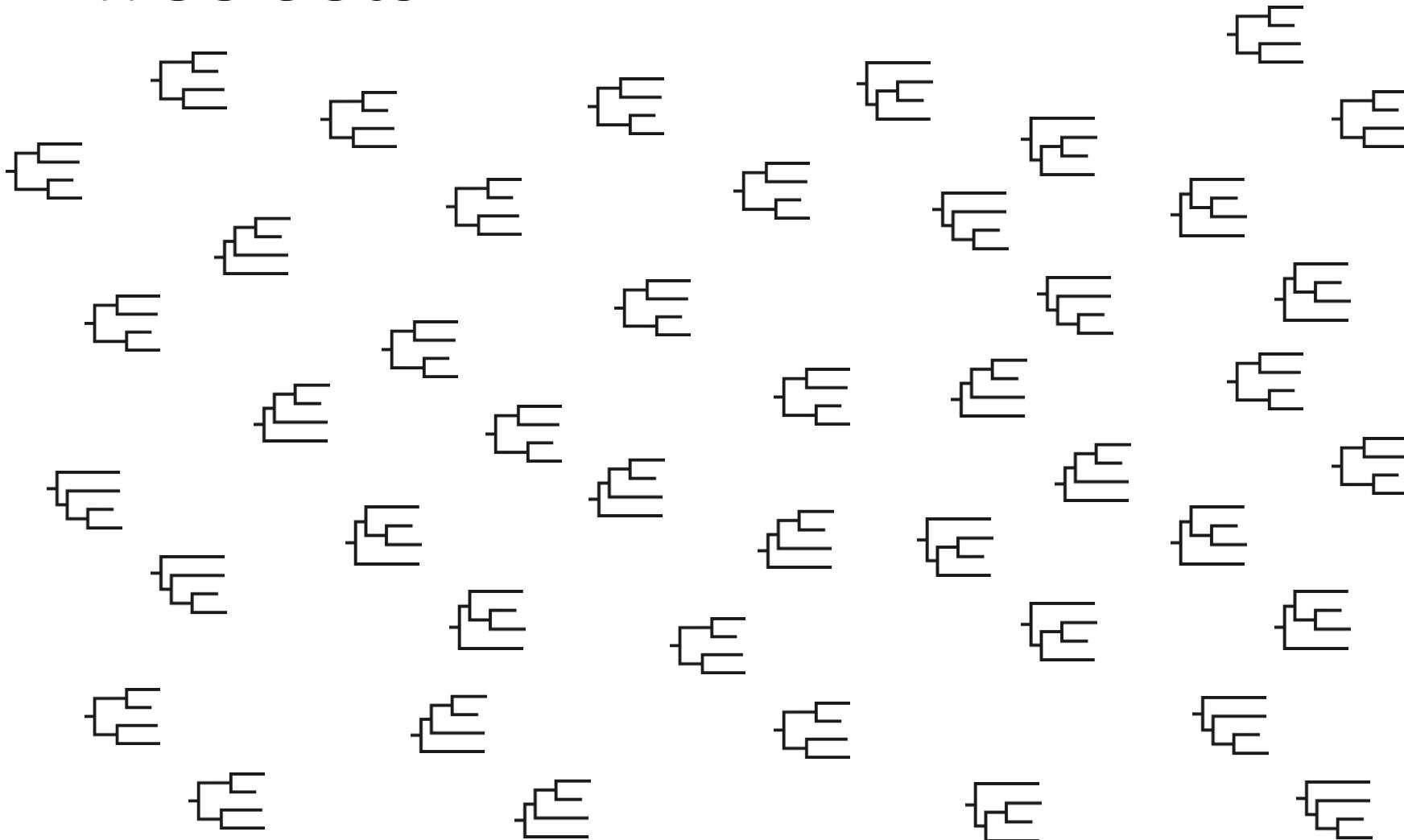
Part 1

Non-Linear Dimensionality Reduction in TreeScaper

Objectives

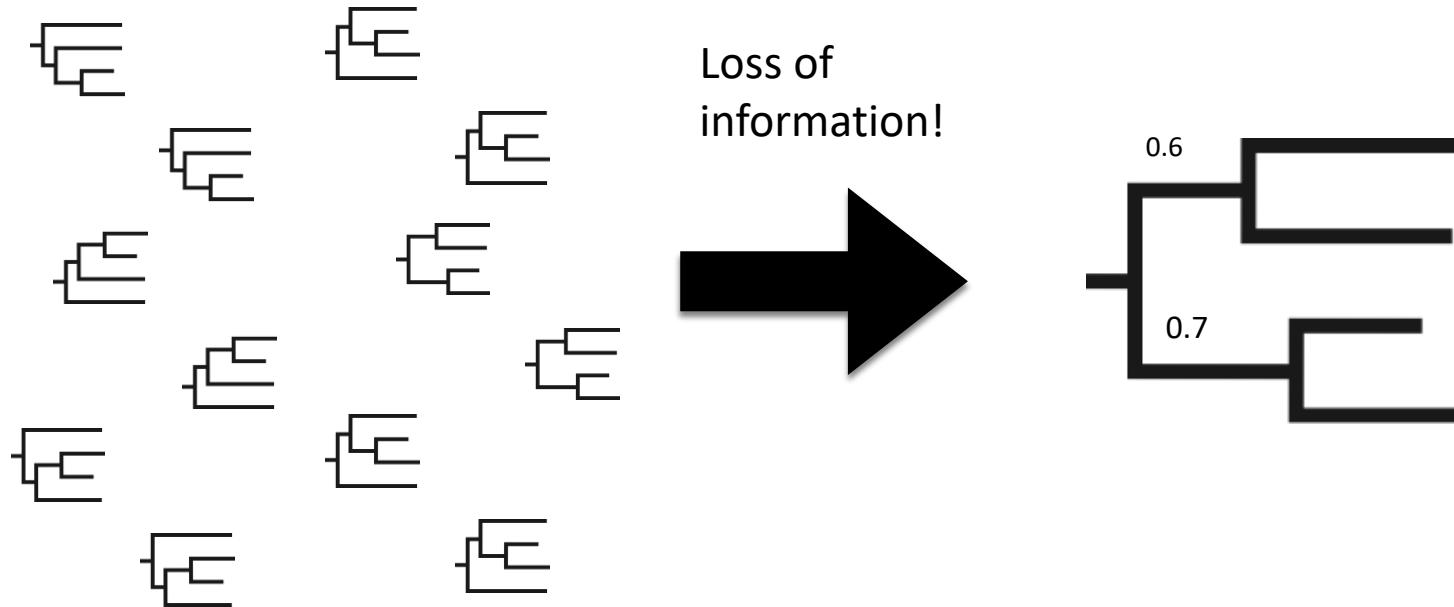
- Give you some intuition about how NLDR methods work
- Introduce the NLDR methods implemented in TreeScaper
 - Show why it is important to try multiple methods
- Introduce concepts related to NLDR visualization quality
- Demonstrate visualization capabilities of TreeScaper

Tree Sets



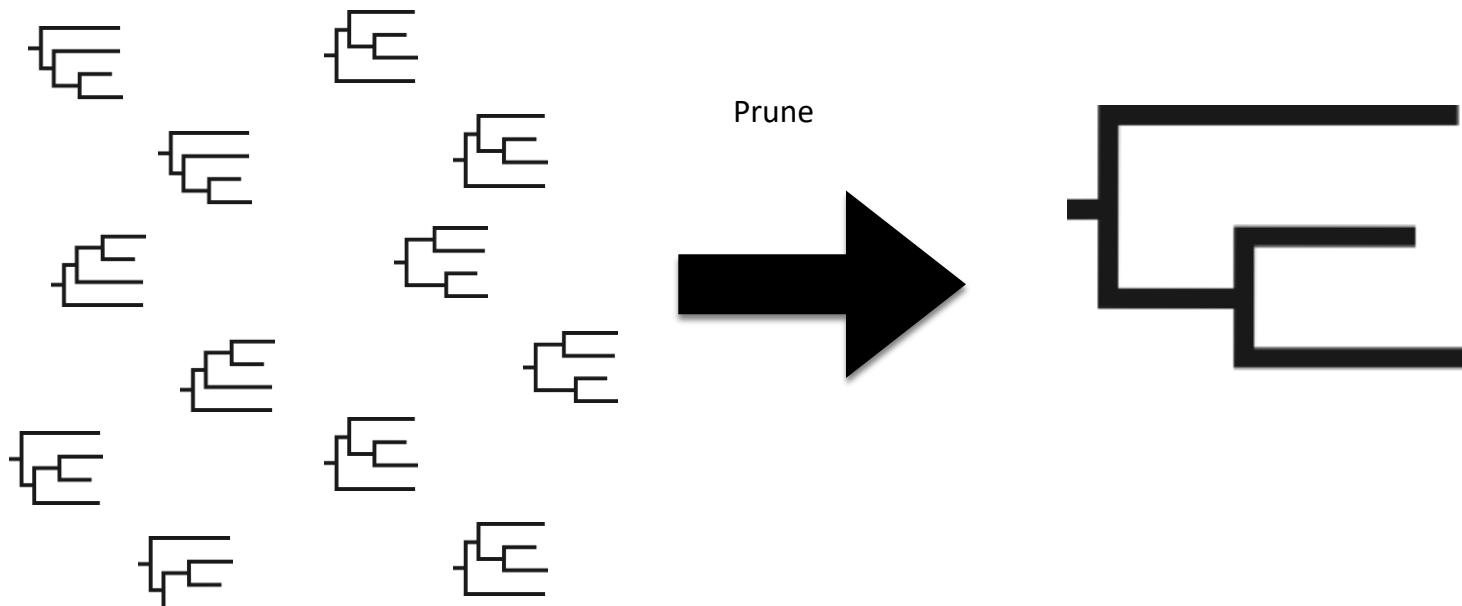
Summarizing Tree Sets

- Consensus trees



Summarizing Tree Sets

- Consensus trees
- Agreement subtrees



Summarizing Tree Sets

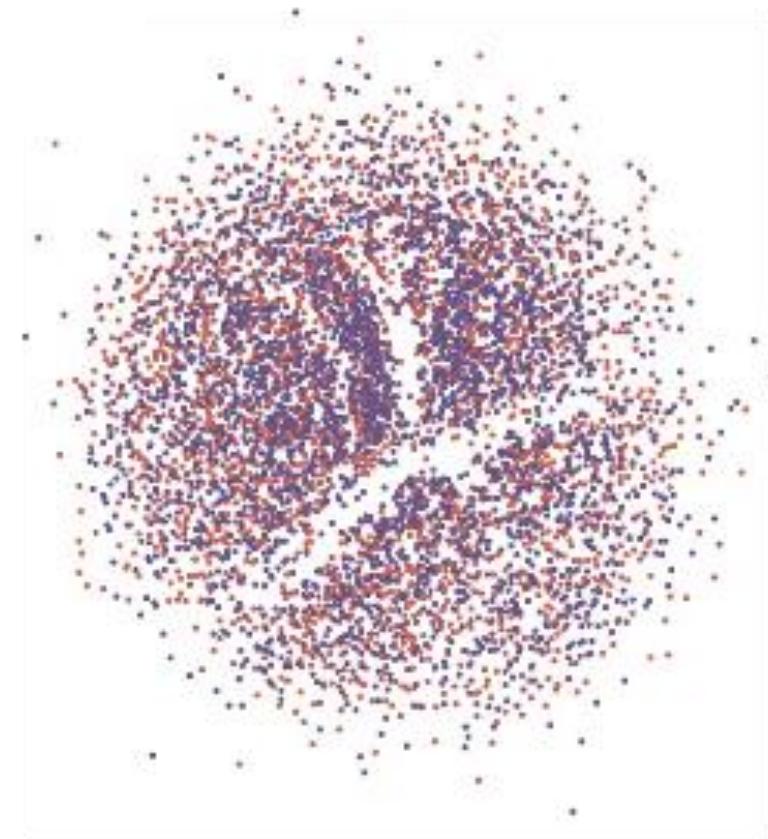
- Consensus trees
- Agreement subtrees
- Clustering

**Clustering genes of common
evolutionary history**

Kevin Gori¹, Tomasz Suchan², Nadir Alvarez², Nick Goldman^{1,*} and Christophe Dessimoz^{1,2,3,4,5,*}

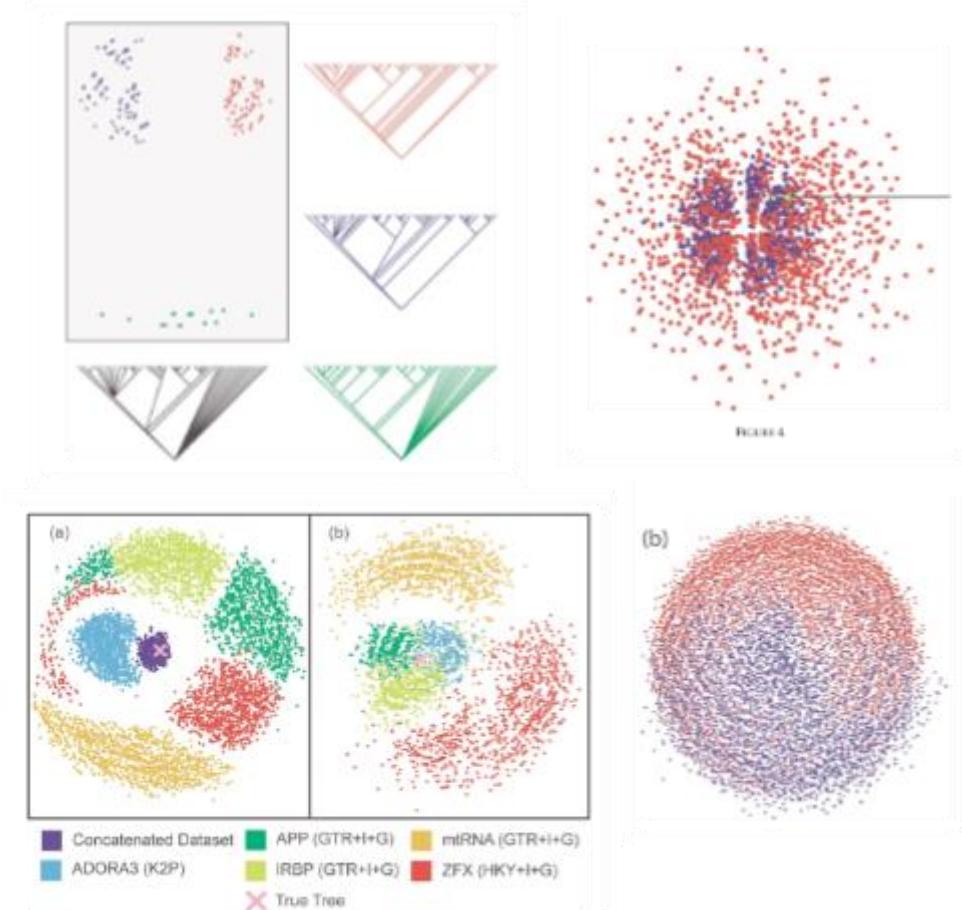
Summarizing Tree Sets

- Consensus trees
- Agreement subtrees
- Clustering
- Dimensionality Reduction



Dimensionality Reduction in Phylogenetics

- Explore tree islands
- Compare bootstrap and Bayesian results
- Compare trees obtained from different genes
- Compare independent runs in a Bayesian analysis



Hillis, Heath, and St. John, 2005. Analysis and visualization of tree space. Systematic Biology 54(3): 471-482.

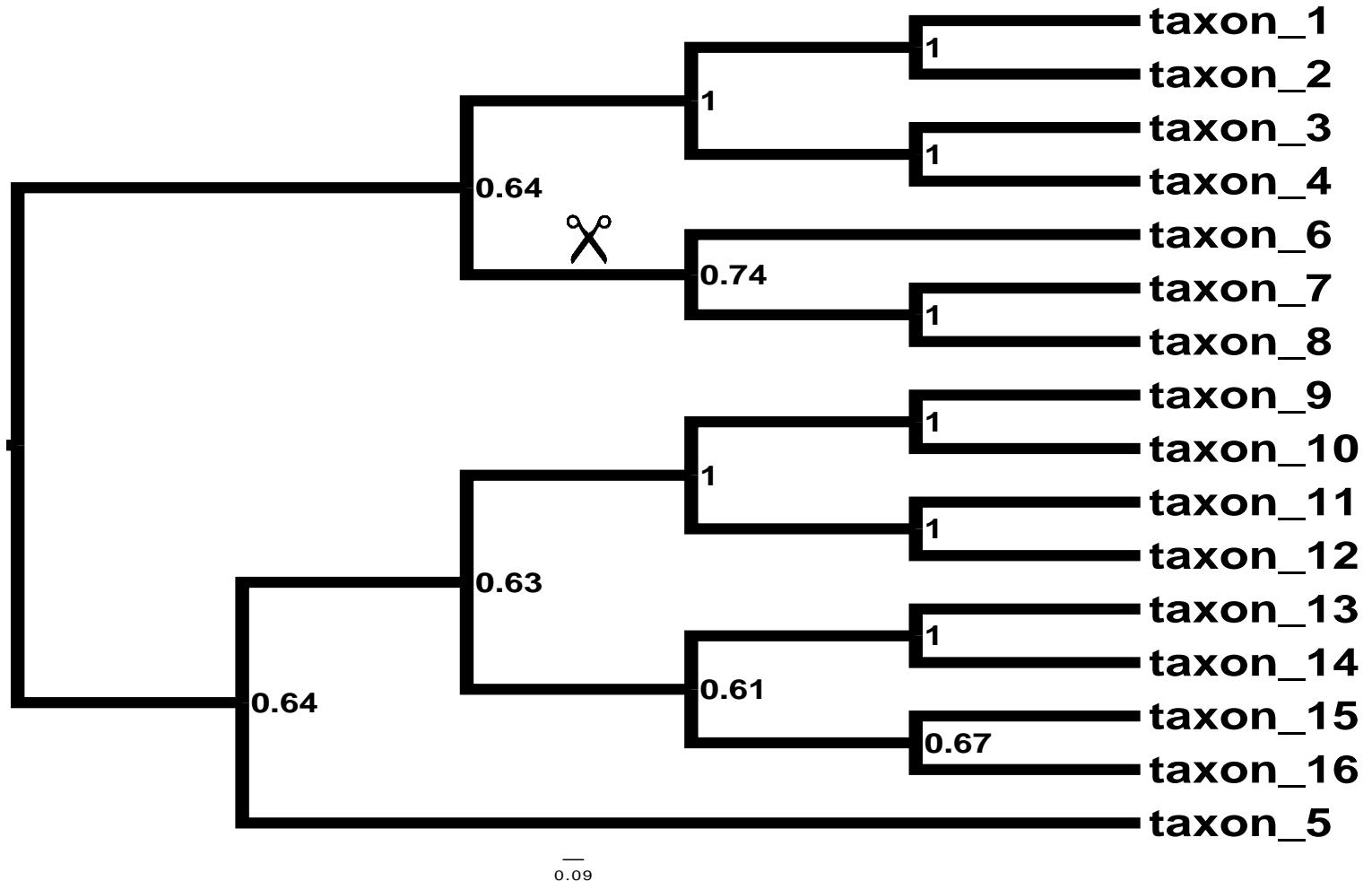
Our Contribution

- Does the Nonlinear Dimensionality Reduction (NLDR) method matter?
- Is 3D needed to fit tree-to-tree distances?
- Develop software for exploring large sets of trees.

Phylogenetic Bipartitions

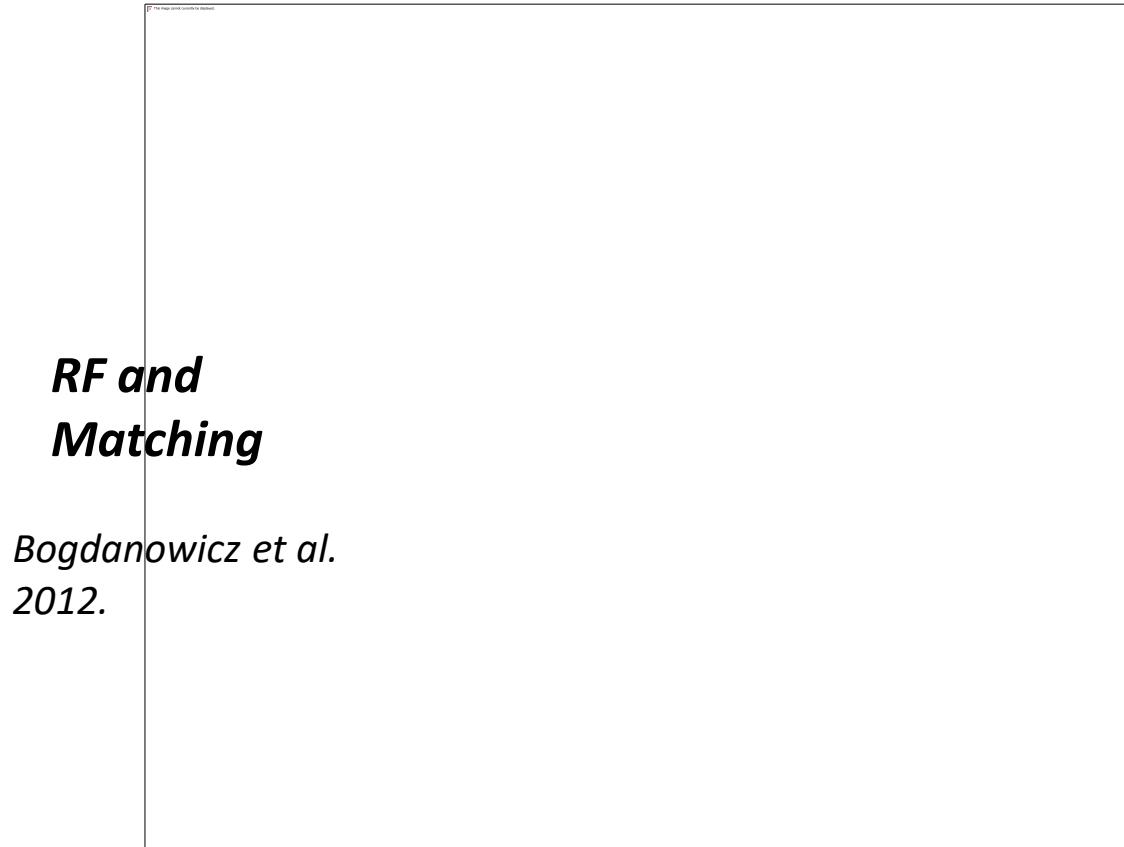
Bipartition:

- A division of the elements of the set into two groups
- In this case:
 - $(6,7,8) | (\text{the rest})$



Plot Trees as a “Phylogenetic Landscape”

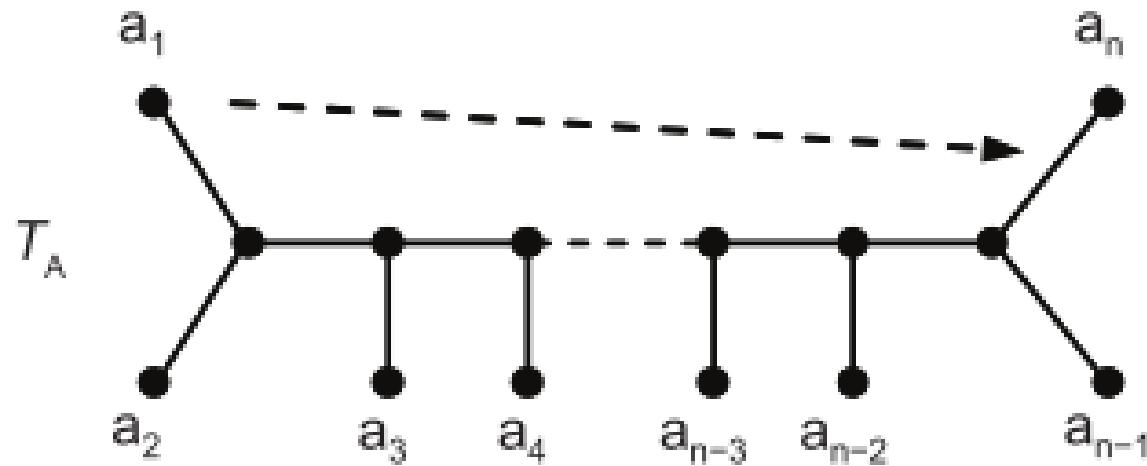
- Generate a matrix of tree-to-tree distances
 - Robinson Foulds (RF)
 - weighted or unweighted



*RF and
Matching*

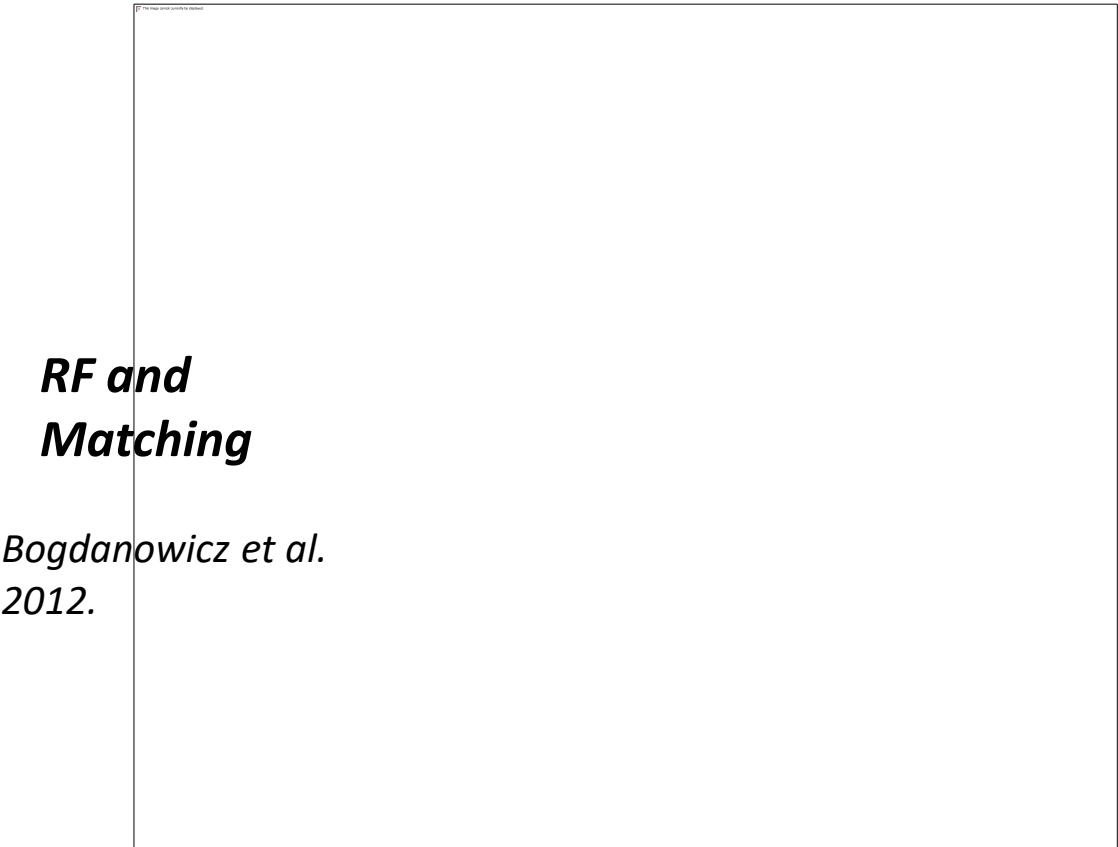
*Bogdanowicz et al.
2012.*

Problem with Robinson-Foulds



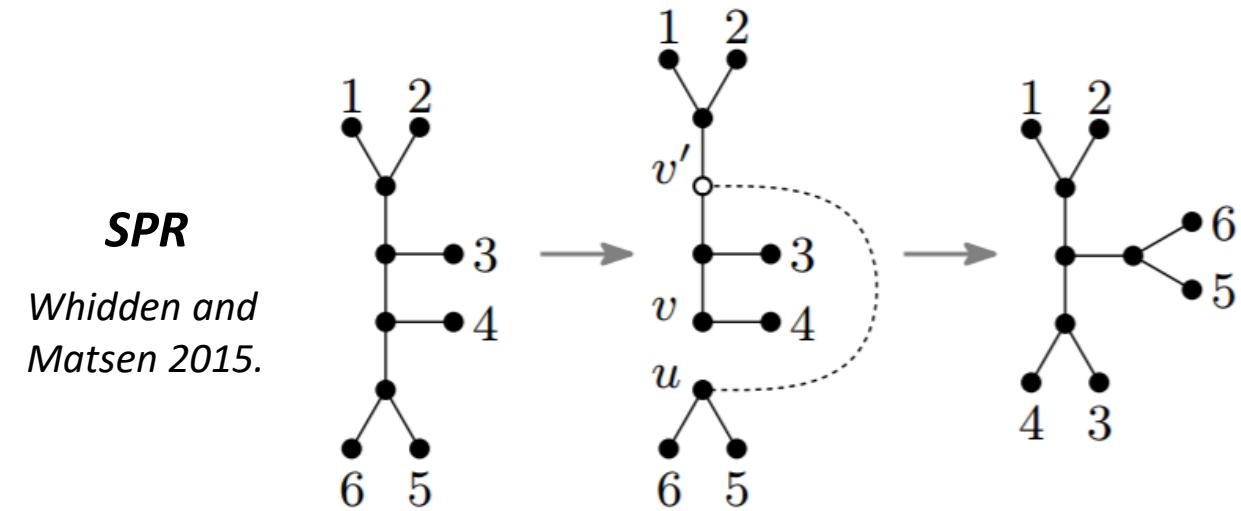
Plot Trees as a “Phylogenetic Landscape”

- Generate a matrix of tree-to-tree distances
 - Robinson Foulds (RF)
 - weighted or unweighted
 - Matching



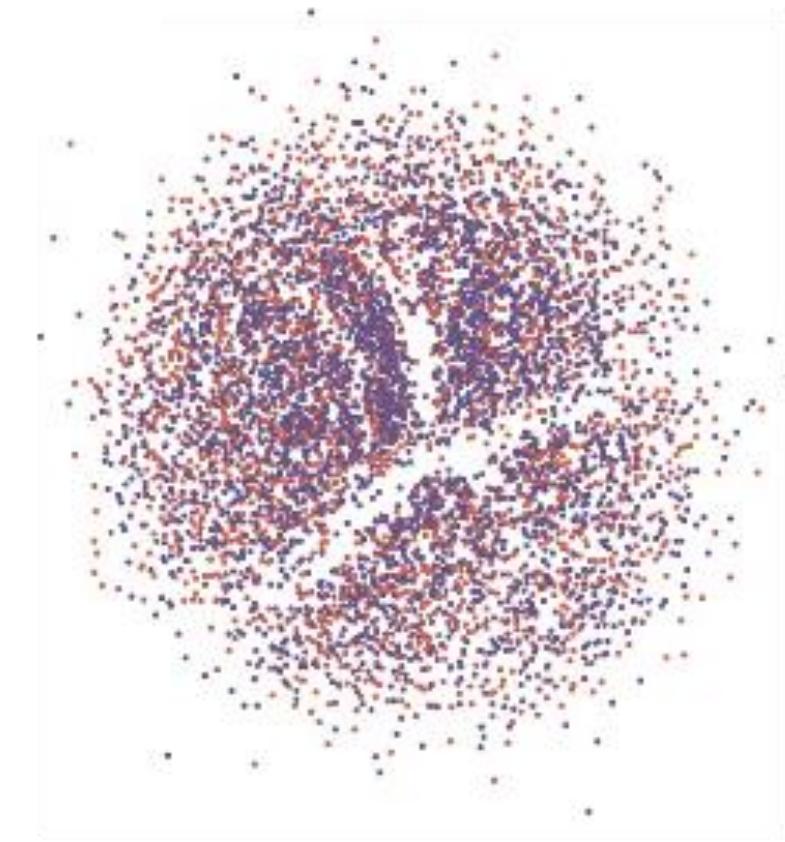
Plot Trees as a “Phylogenetic Landscape”

- Generate a matrix of tree-to-tree distances
 - Robinson Foulds (RF)
 - weighted or unweighted
- Matching
- Subtree Prune Refgraft (SPR)



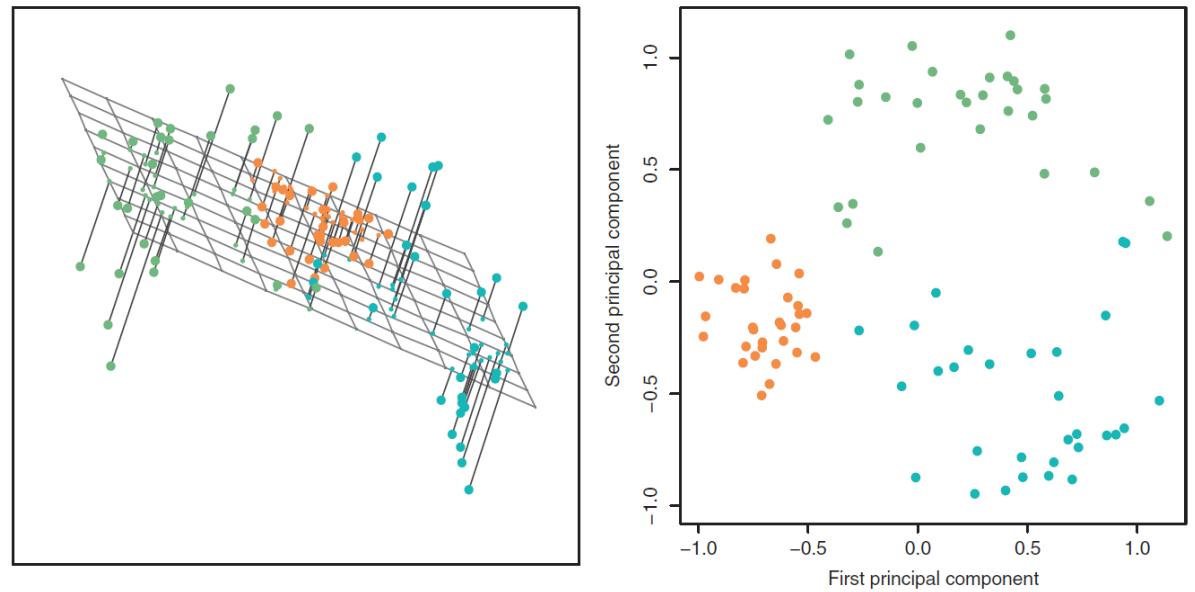
Plot Trees as a “Phylogenetic Landscape”

- Use a dimensionality reduction method to project tree-to-tree distances into a 2D space



Principle Components Analysis

- Finds best representation of data on a lower dimensional linear surface
 - Best in terms of sum of squared residuals
 - Percent of variability of the data explained on the PC axes provide a measure of quality of low dimensional representation

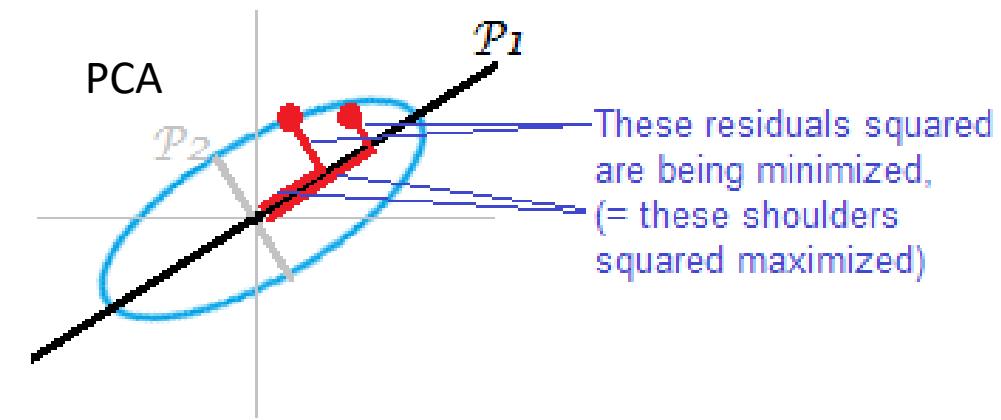
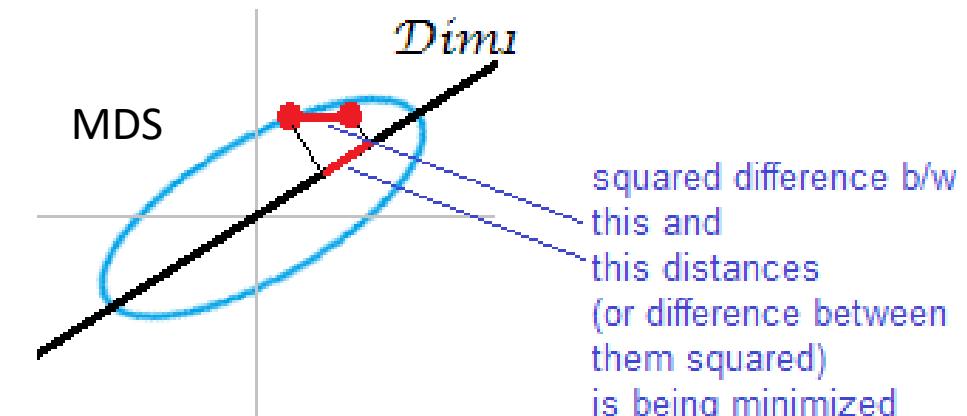


From “An Introduction to Statistical Learning” by Hastie et al.

Multidimensional Scaling

- Given a $n \times n$ matrix D of pairwise distances between n points, find a low-dimensional embedding of data points in \mathbb{R}^k such that distances between them approximate the given distances.
- Stress Majorization Cost Function:

$$S_M(z_1, z_2, \dots, z_N) = \sum_{i \neq i'} (d_{ii'} - \|z_i - z_{i'}\|)^2.$$



Non-Linear Dimensionality Reduction

- Methods assume the data lie close to an intrinsically low-dimensional non-linear surface embedded in a high-dimensional space.
- NLDR methods can be thought of as “flattening” the surface
 - Reduces the data to a set of low-dimensional coordinates that represent their relative positions in the surface.

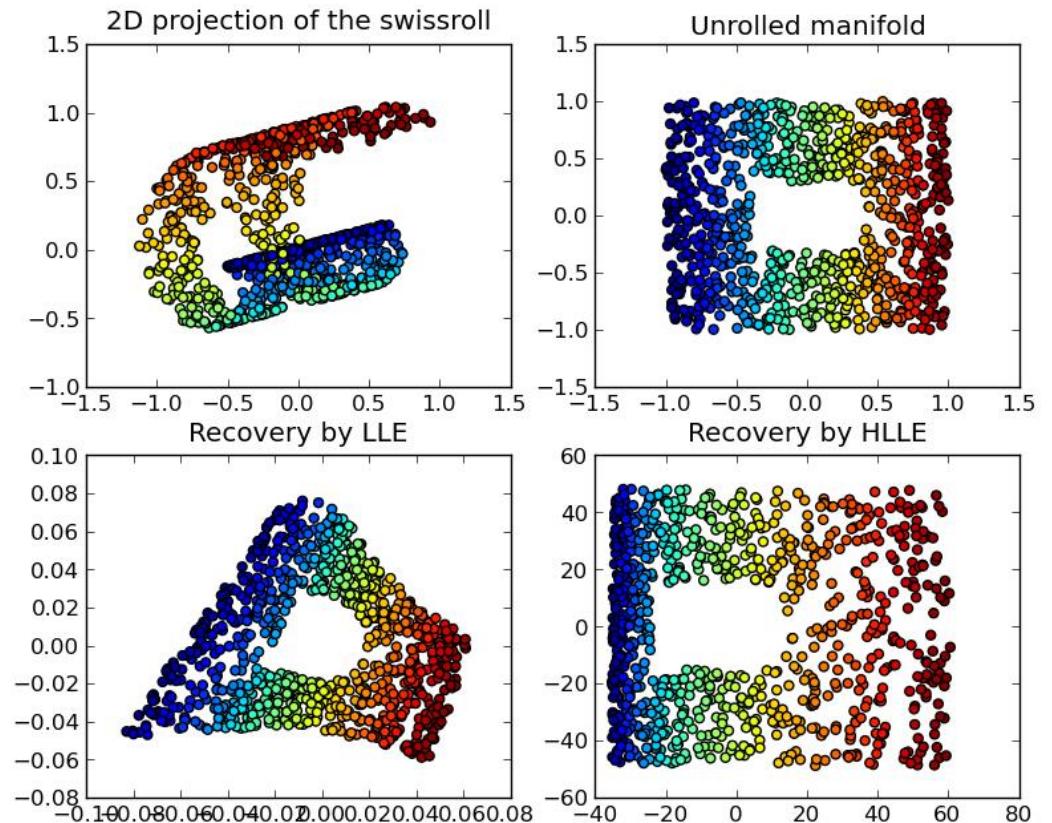


Figure By Olivier Grisel

TreeScaper NLDR

Cost Functions

- Sammon's Nonlinear Mapping

$$S_{Sm}(z_1, z_2, \dots, z_N) = \sum_{i \neq i'} \frac{(d_{ii'} - \|z_i - z_{i'}\|)^2}{d_{ii'}}.$$

More emphasis is put on preserving smaller pairwise distances.

TreeScaper NLDR

Cost Functions

- Sammon's Nonlinear Mapping
- Kruskal-1 Stress
- Normalized Stress
- Curvilinear Components Analysis (CCA)

Optimization Algorithms

- Linear Iteration
- Majorization
- Gauss-Seidel
- Stochastic Gradient Descent

Example Analysis

- Generate a large set of ML bootstrap trees
 - For 15 genes a salamander mtDNA genome alignments
- Evaluate trees with different NLDR Methods

Projections in 2D

Bootstrap Trees from 15 mtDNA Genes

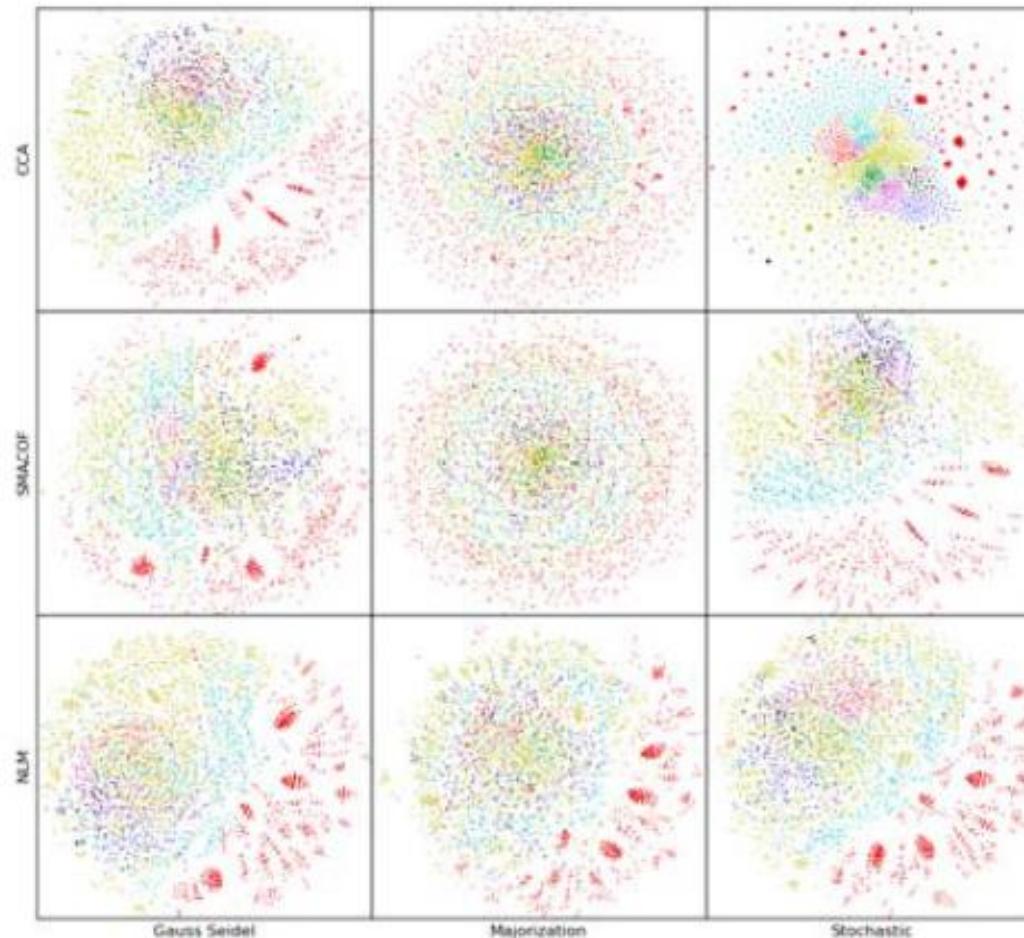
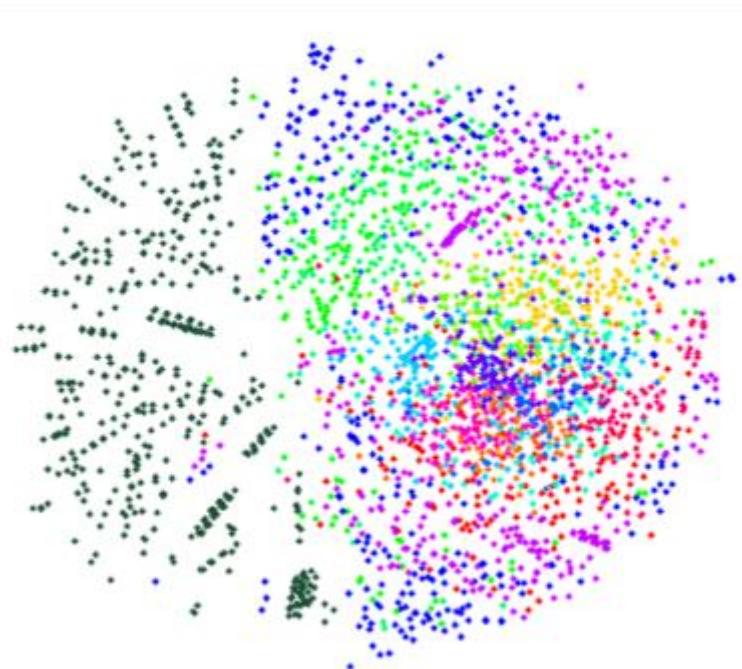


Figure by Wen
Huang

Projections in 2D

Bootstrap Trees from 15 mtDNA Genes

**Kruskal-1 & Linear
Iteration**



**Colors =
trees from
different
genes**

**CCA & Stochastic
Gradient**

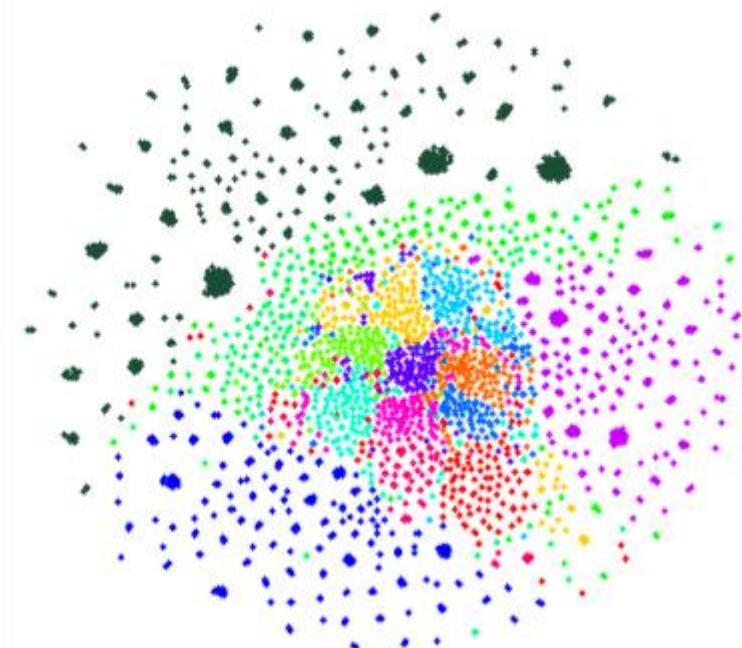
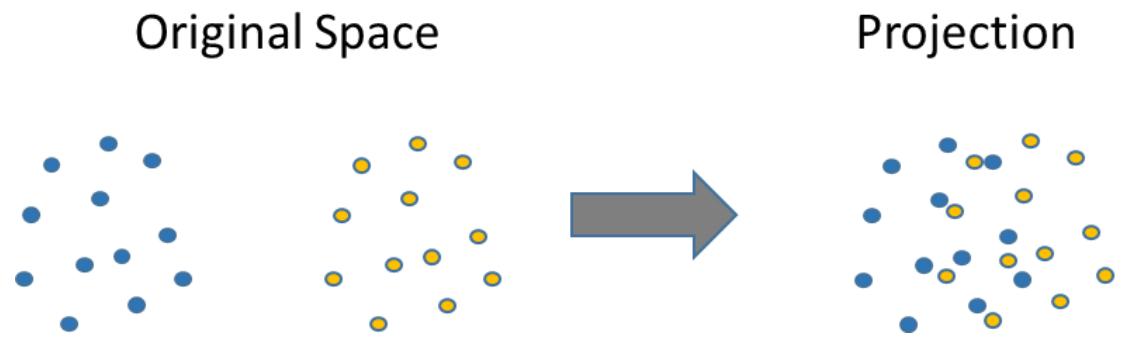


Figure by Wen
Huang

Measures of NLDR Quality

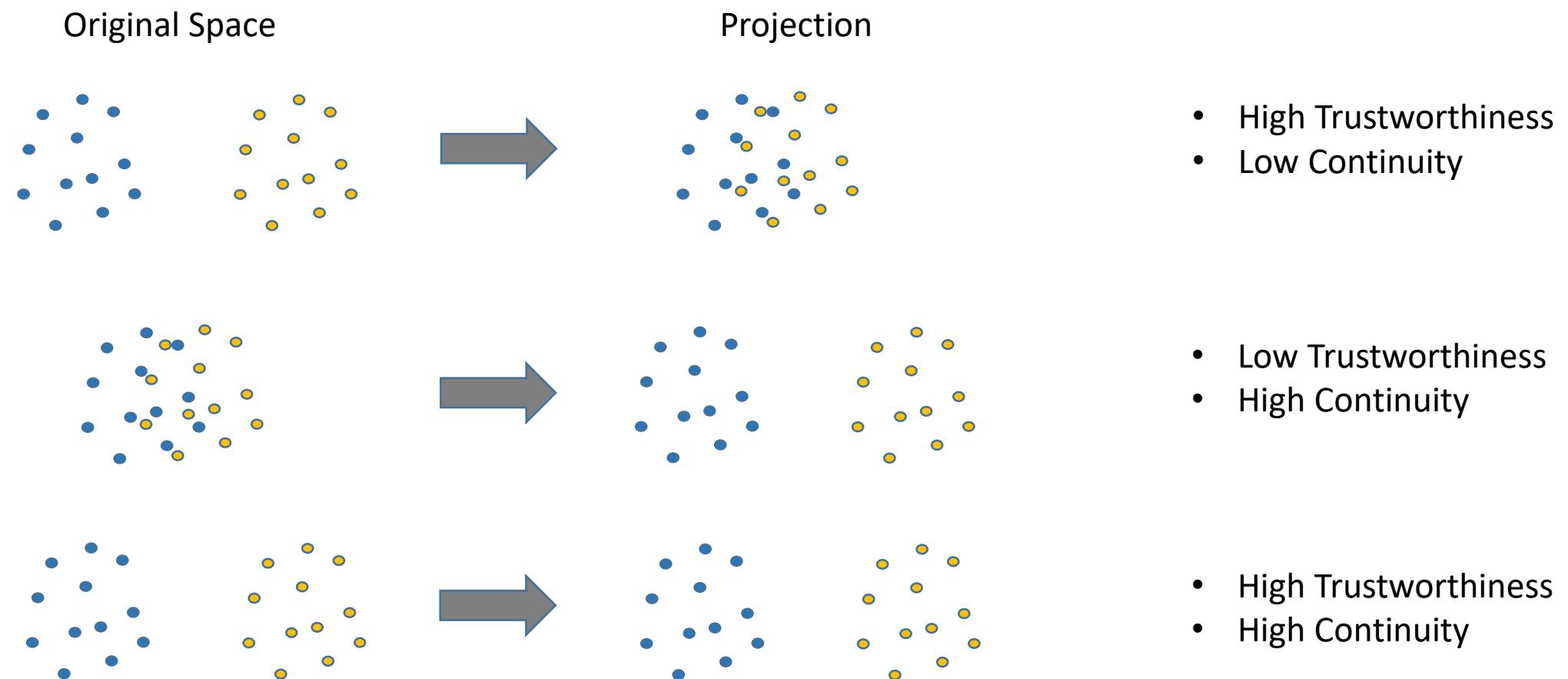


$$Tru = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^n \sum_{j \in U_k(i)} (r(i, j) - k).$$

$$Con = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^n \sum_{j \in V_k(i)} (s(i, j) - k).$$

- *Trustworthiness* measures to what extent the k closest neighbors of each point in the original space are close by in the display.
- *Continuity* reverses roles of display and original space. Measures to what extent the k closest neighbors of each point in the original space are close by in the display.
- Both measures are between 0 and 1

Both Measures Are Useful



Measures of NLDR Quality

CCA & Stochastic Gradient

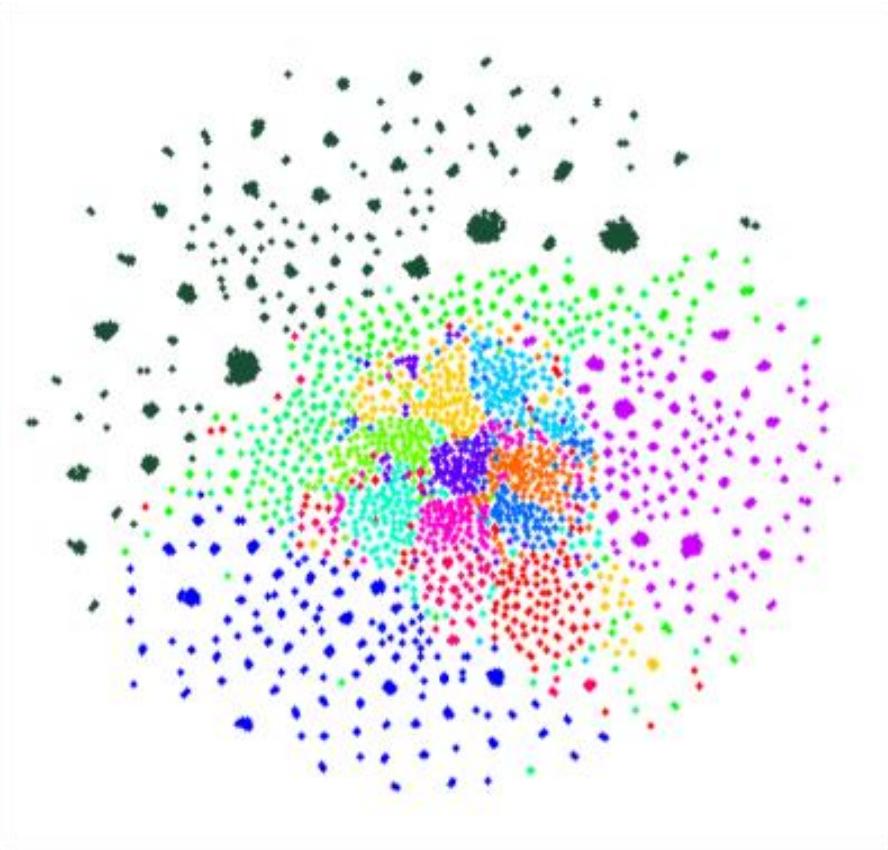
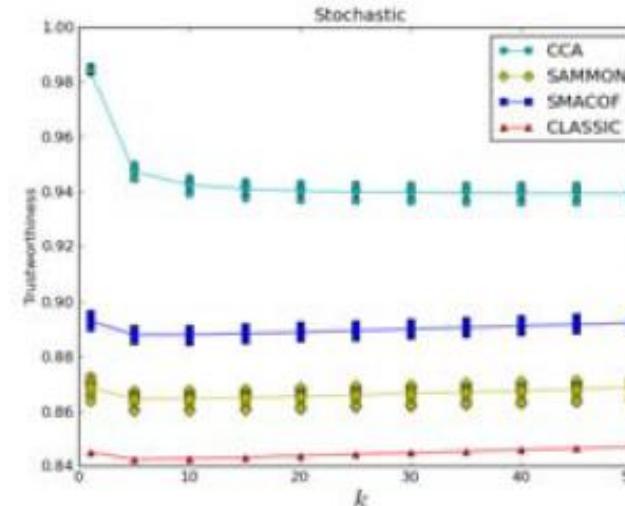
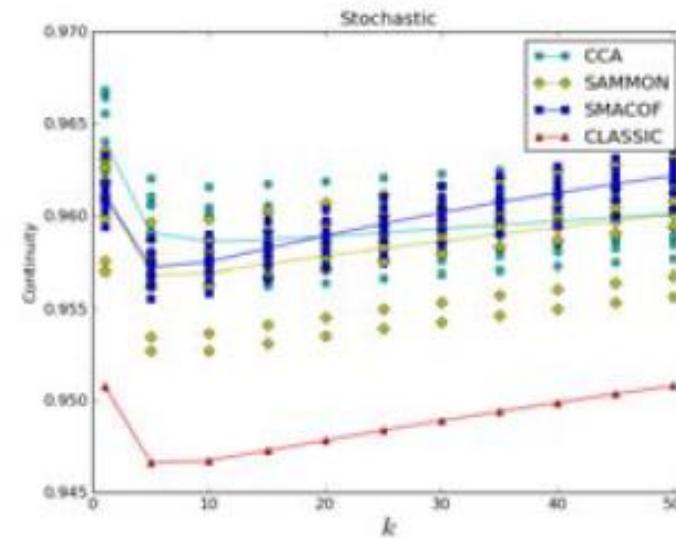
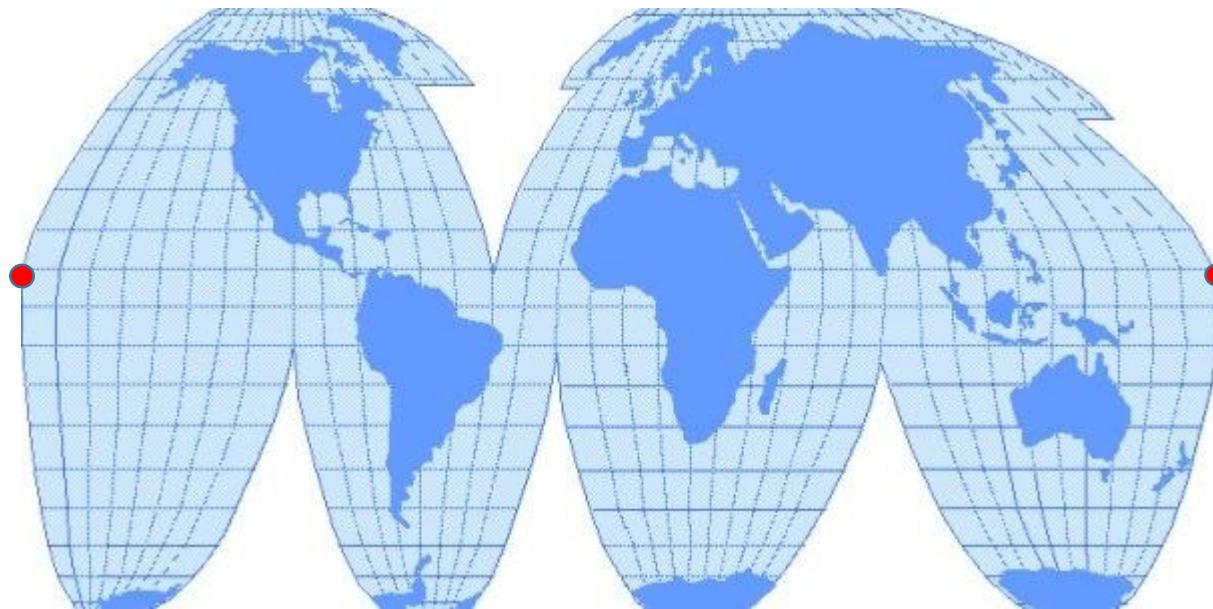


Figure by Wen
Huang



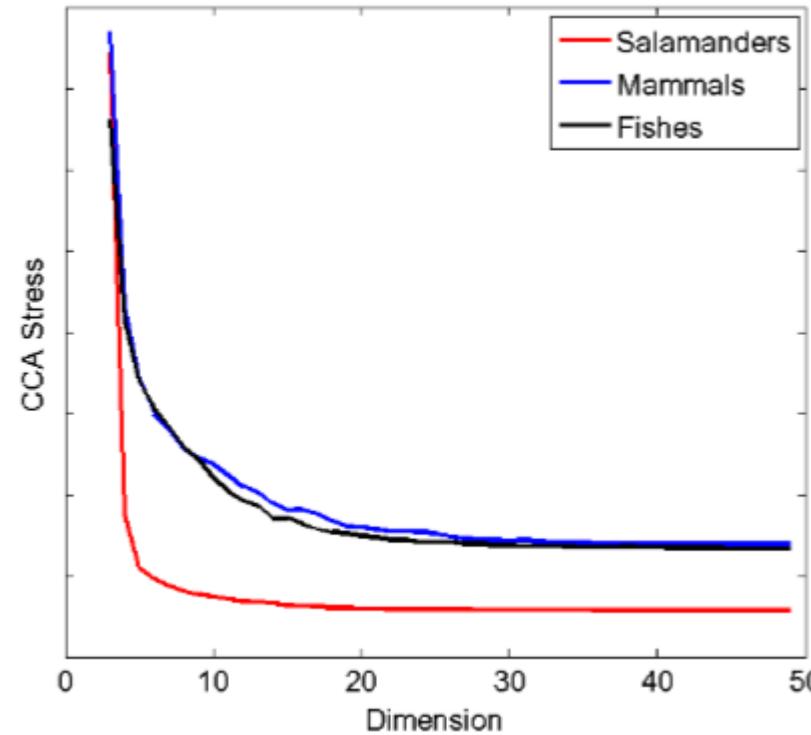
Discontinuity in NLDR: Flattening of a Globe



Is 3D Needed?

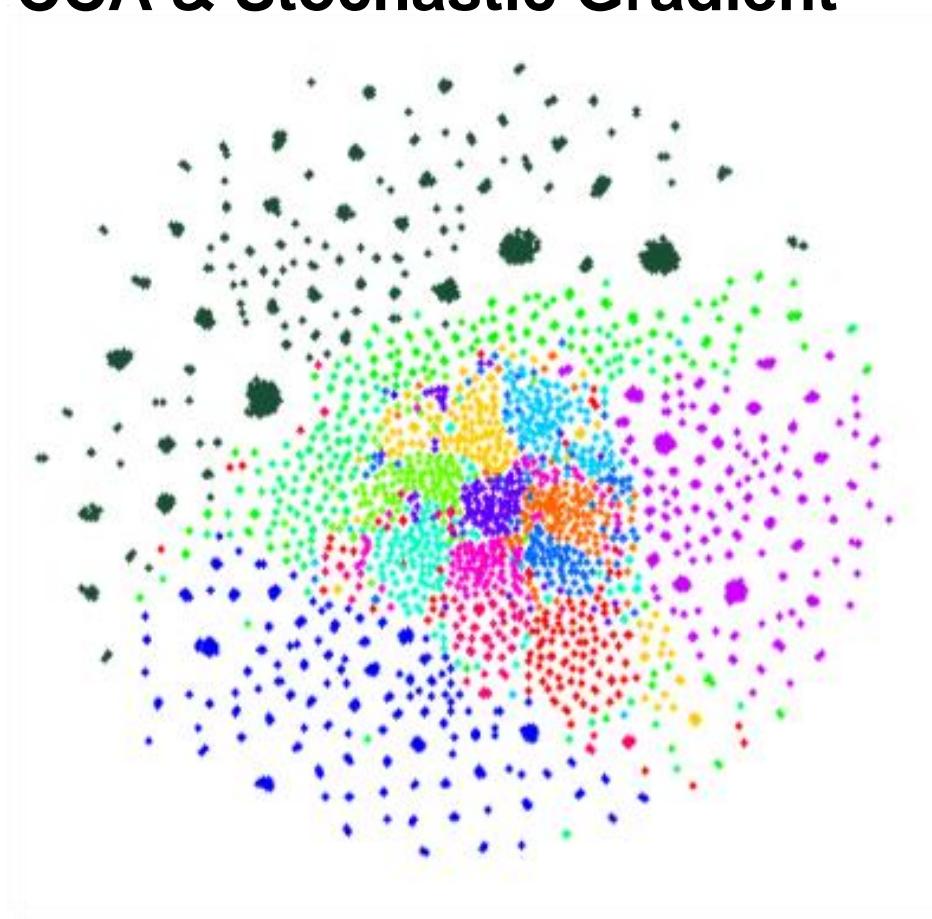
YES

- Intrinsic dimensionality estimators
 - Nearest Neighbor estimator
 - Correlation Dimension
 - Maximum Likelihood estimator
 - Visual inspection of stress versus dimension
- 3 to 15 dimensions are needed to fit tree-to-tree distances



Intrinsic Dimensionality

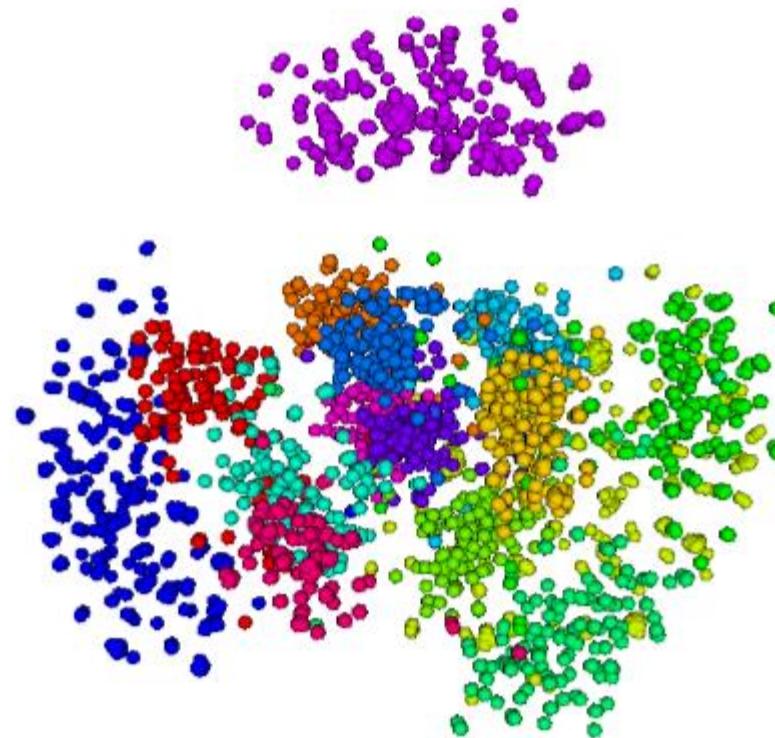
CCA & Stochastic Gradient



Dimensionality Test	Unfiltered
NN	6.91808
COR	4.53759
ML	17.2628

Projection in 3D

Bootstrap Trees from 15 mtDNA Genes

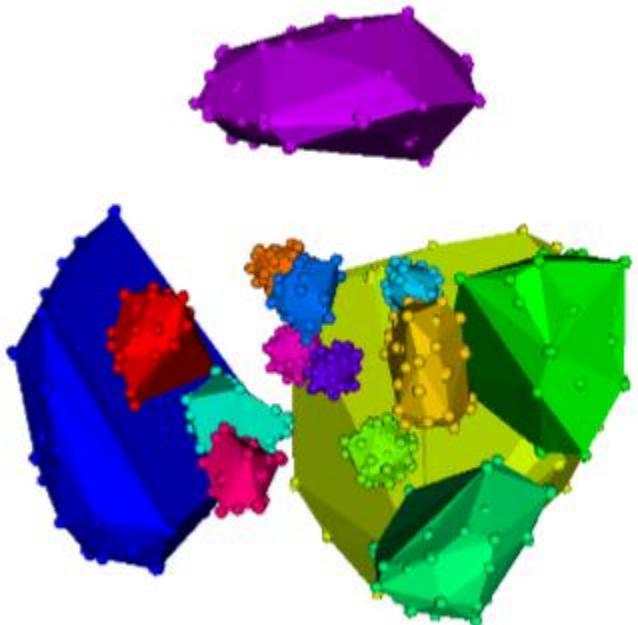


Colors =
trees from
different
genes

Projections in 3D

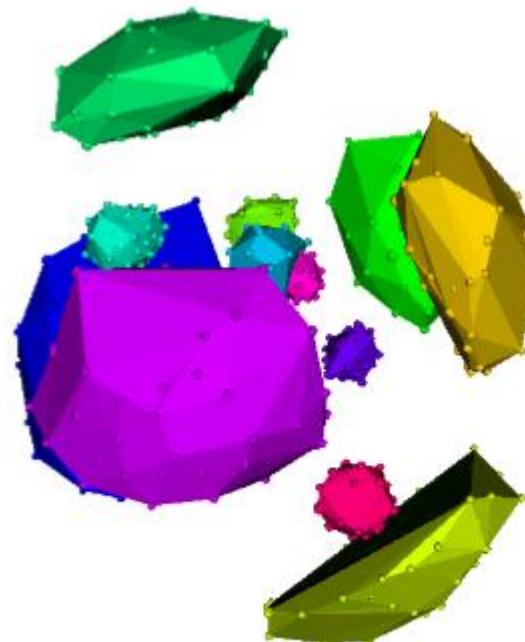
Bootstrap Trees from 15 mtDNA Genes

Salamanders



Colors =
trees from
different
genes

Mammals

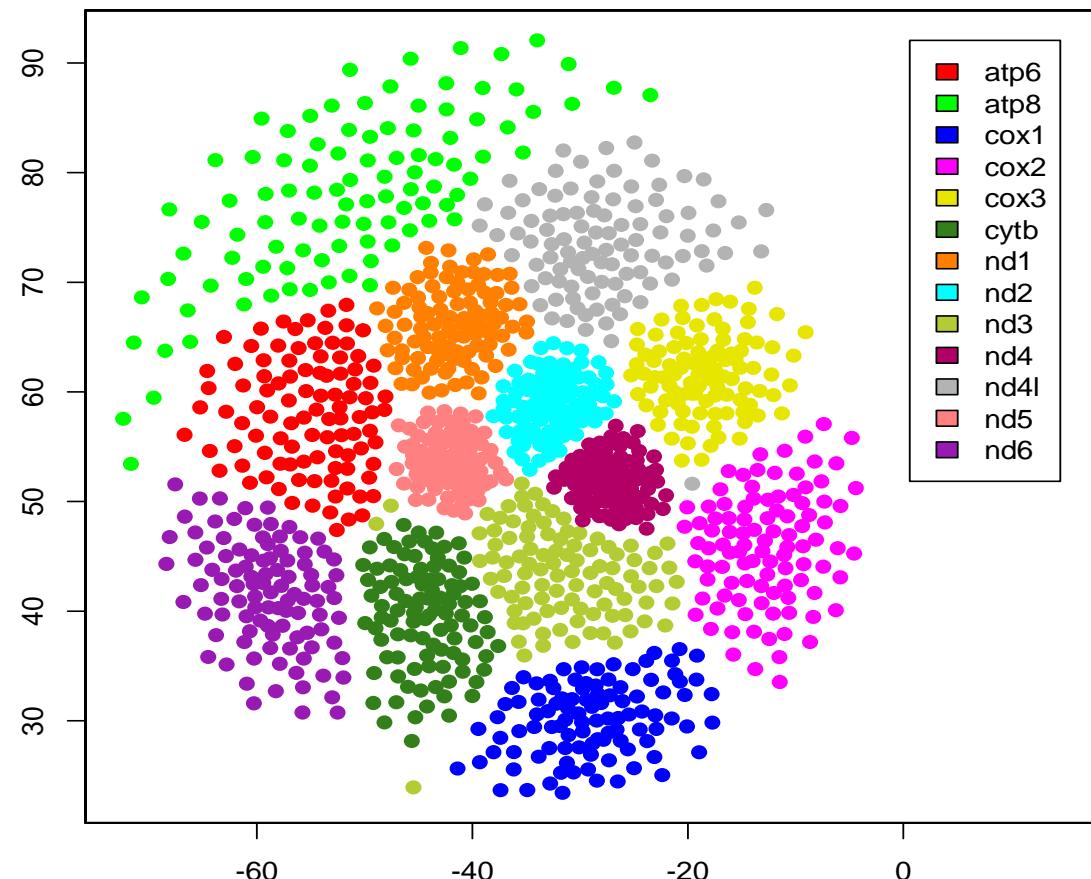


Part 2

Network Community Detection Methods in TreeScaper

Summarizing Tree Sets

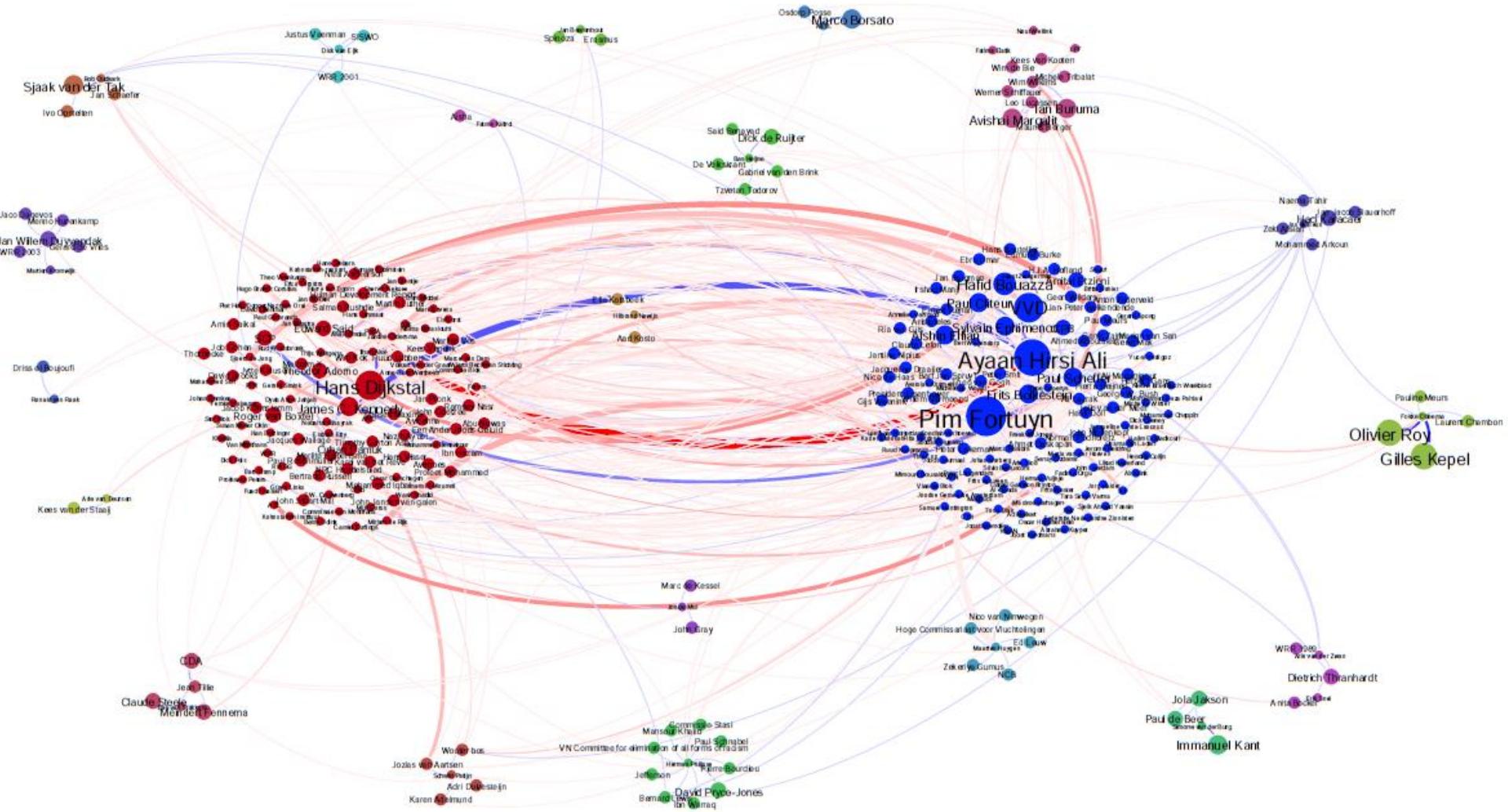
- Dimensionality Reduction
 - No objective way to discern distinct groups of trees



A New Approach: Phylogenetic Networks

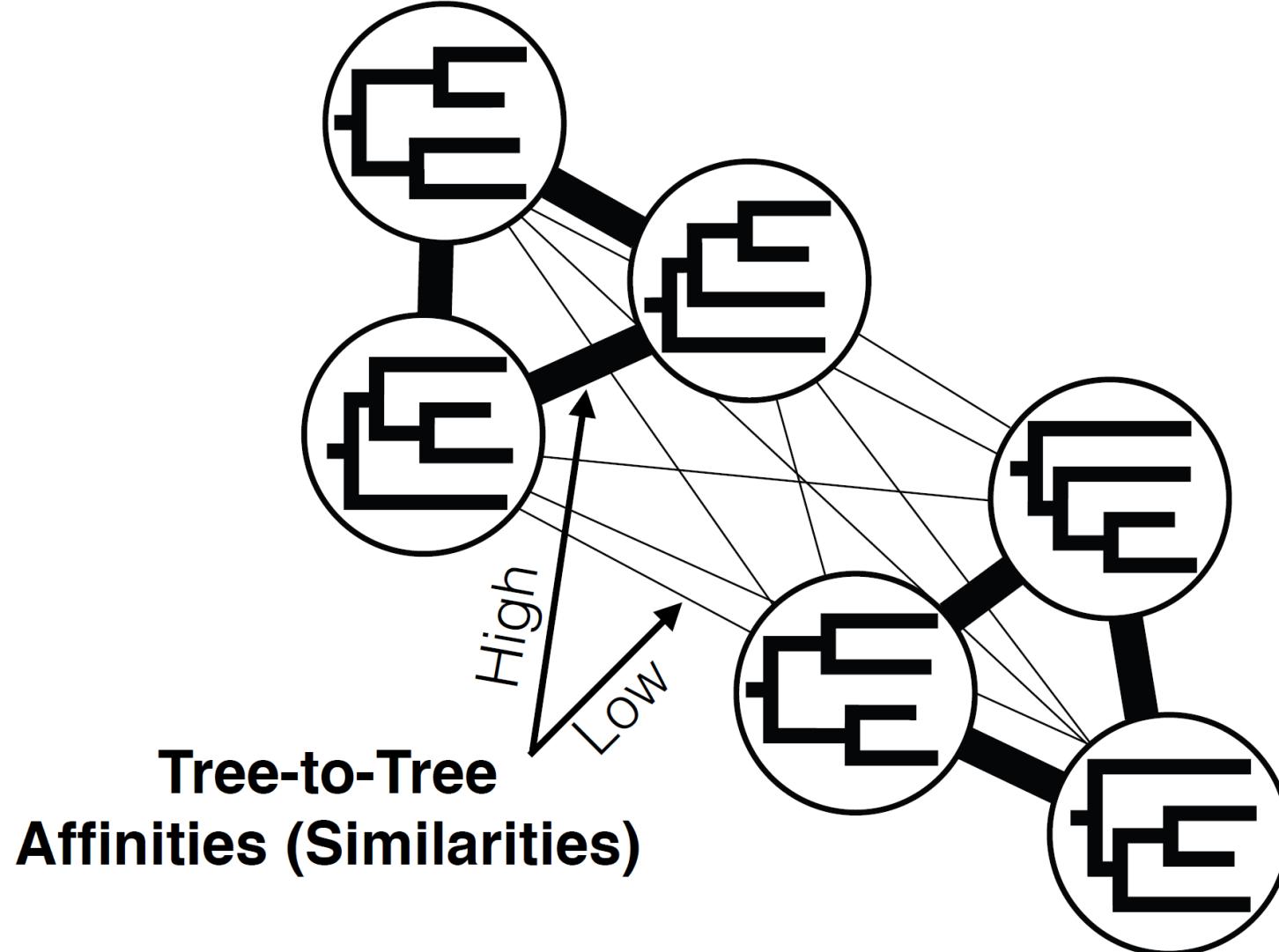
- Represent the tree set as a network
- Use community detection methods to identify distinct groups of topologies
- Indicative of interesting evolutionary history:
 - Recombination
 - Horizontal Gene Transfer
 - Hybridization
 - Incomplete Lineage Sorting
- May also indicate systematic error

Community Detection: Social Networks

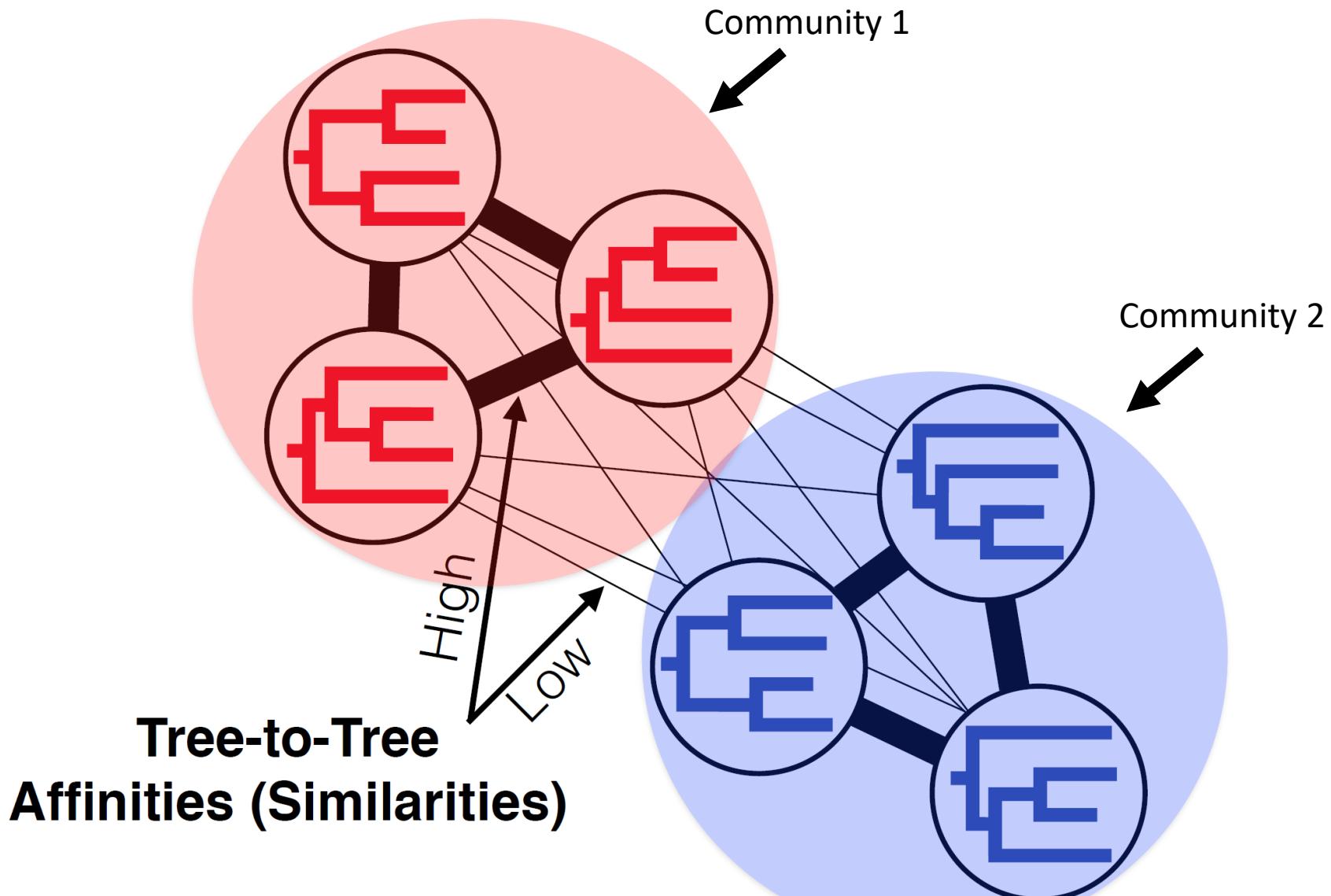


Author
Networks
(by Vincent
Traag)

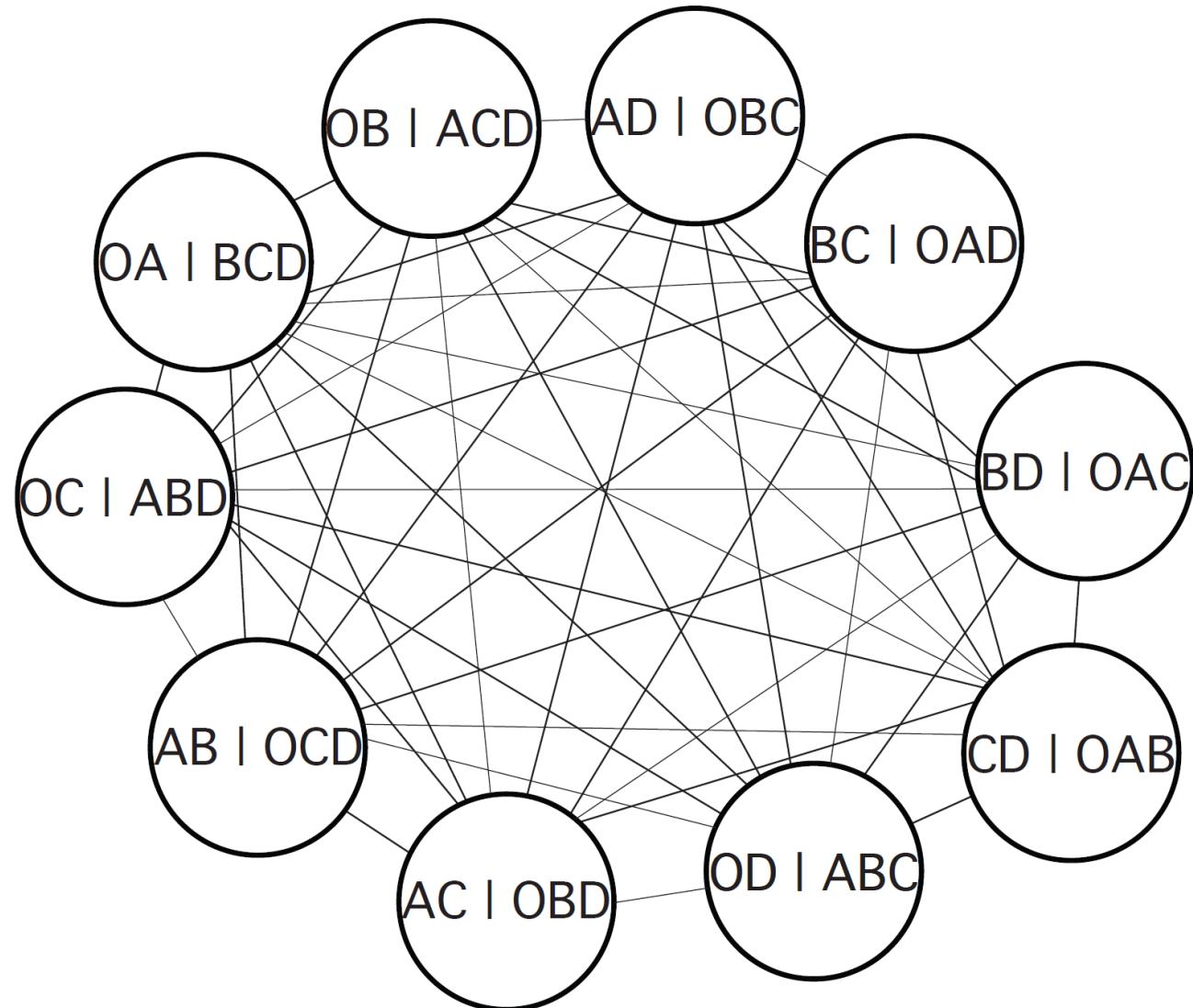
Networks of Trees



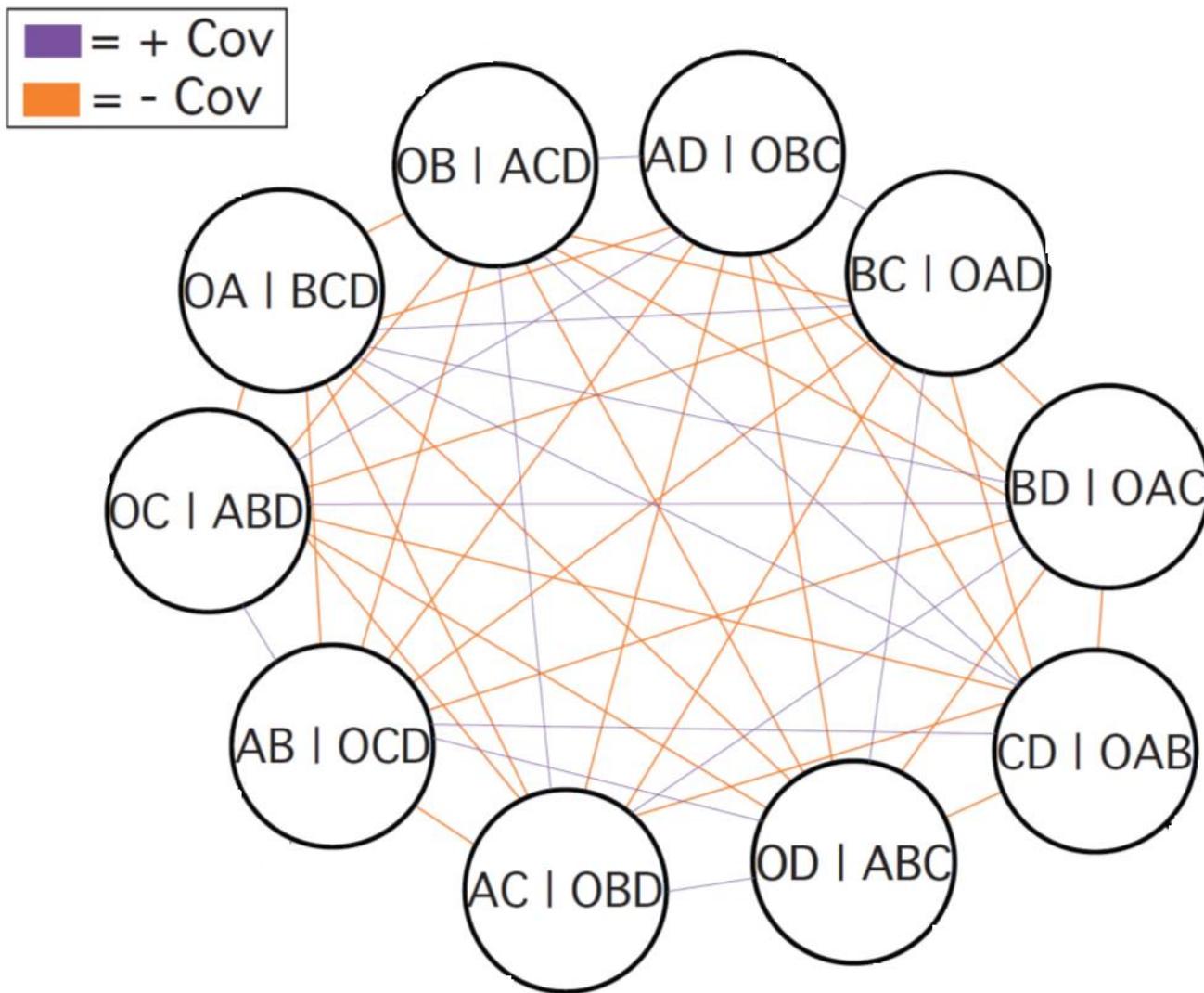
Networks of Trees: Community Detection



Networks of Bipartitions



Networks of Bipartitions: Covariance Edge Weights



Networks of Bipartitions: Covariances

For a tree topology \mathcal{T} the set of bipartitions corresponding to edges in \mathcal{T} is denoted $B(\mathcal{T})$.

Let X be a variable that encodes the presence/absence of bipartition a in the tree set. That is, for each tree topology \mathcal{T}_i in the tree set:

$$X_i = \begin{cases} 1 & a \in B(\mathcal{T}_i) \\ 0 & a \notin B(\mathcal{T}_i) \end{cases}$$

Let Y be a variable that encodes the presence/absence of bipartition b in the tree set.

Let p_1 be the fraction of the trees that contain a .

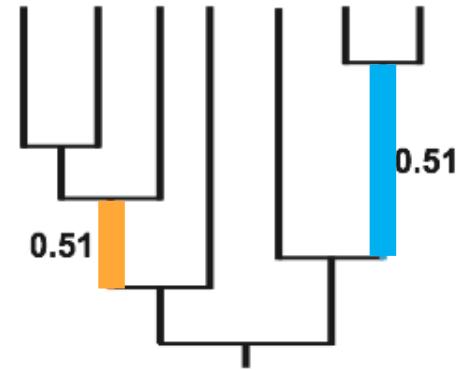
Let p_2 be the fraction of the trees that contain b .

Let p_3 be the fraction of the trees that contain both a and b .

It is straightforward to show that:

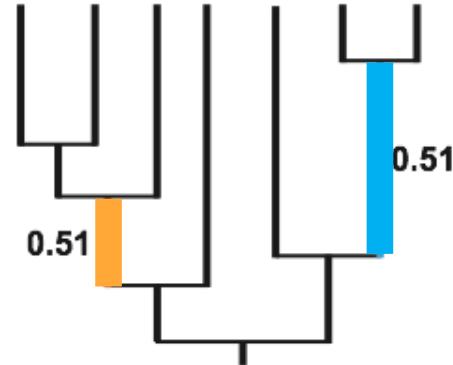
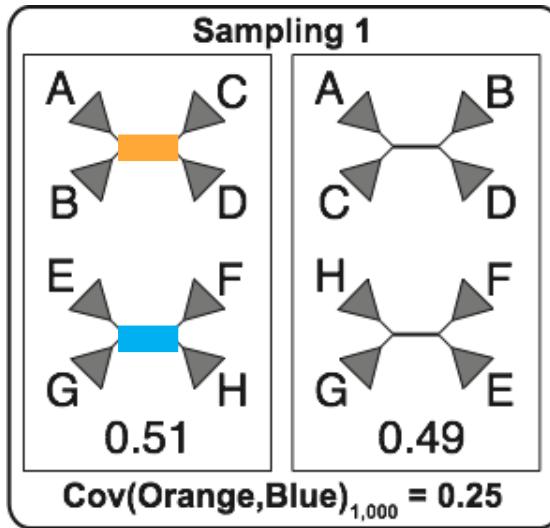
$$\begin{aligned} cov(X, Y) &= E[XY] - E[X]E[Y] \\ &= p_3 - (p_1)(p_2) \end{aligned}$$

Bipartition Covariances

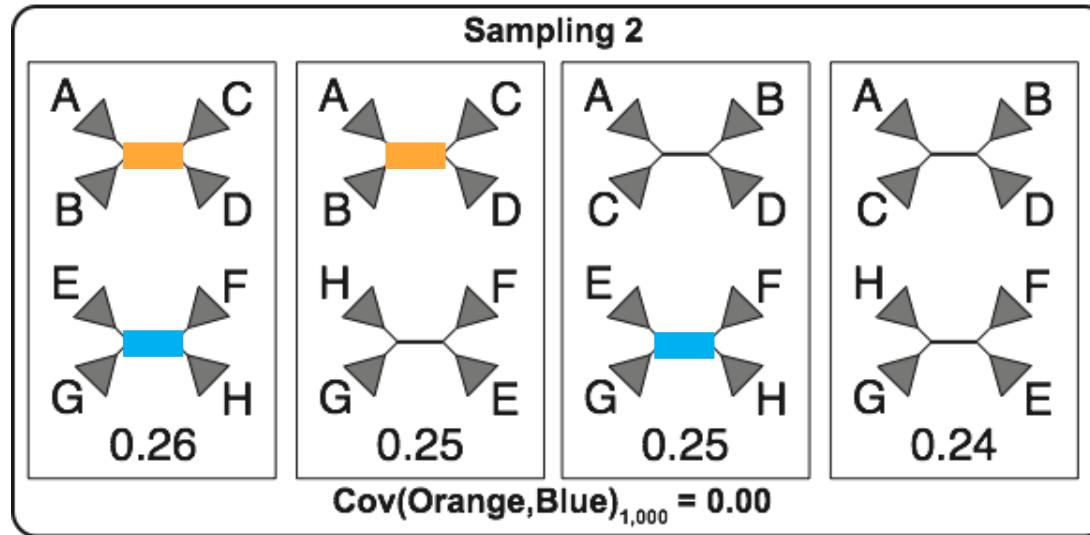
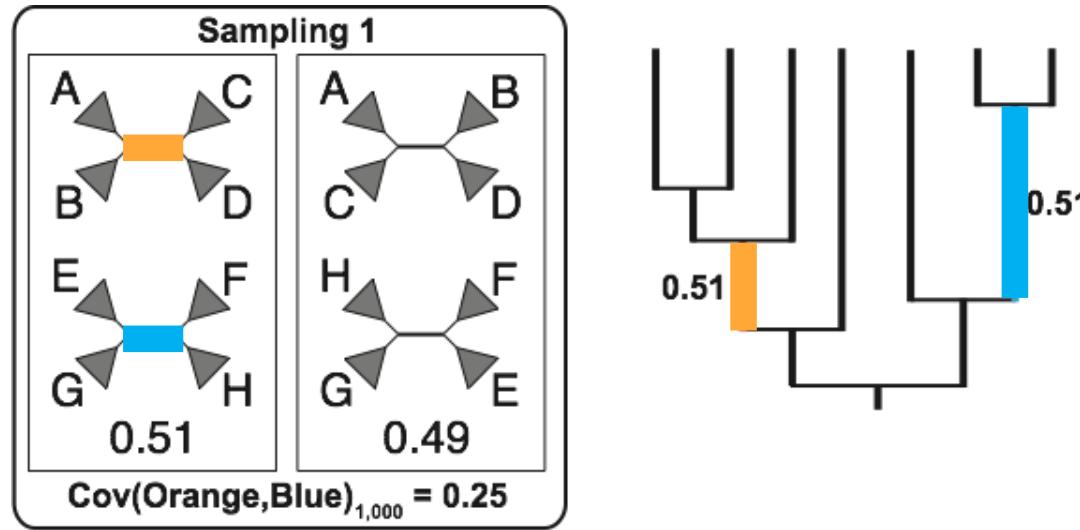


Bipartition Covariances

$$\begin{aligned}\text{Cov(orange, blue)} &= (.51) - (.51 * .51) \\ &= .25\end{aligned}$$

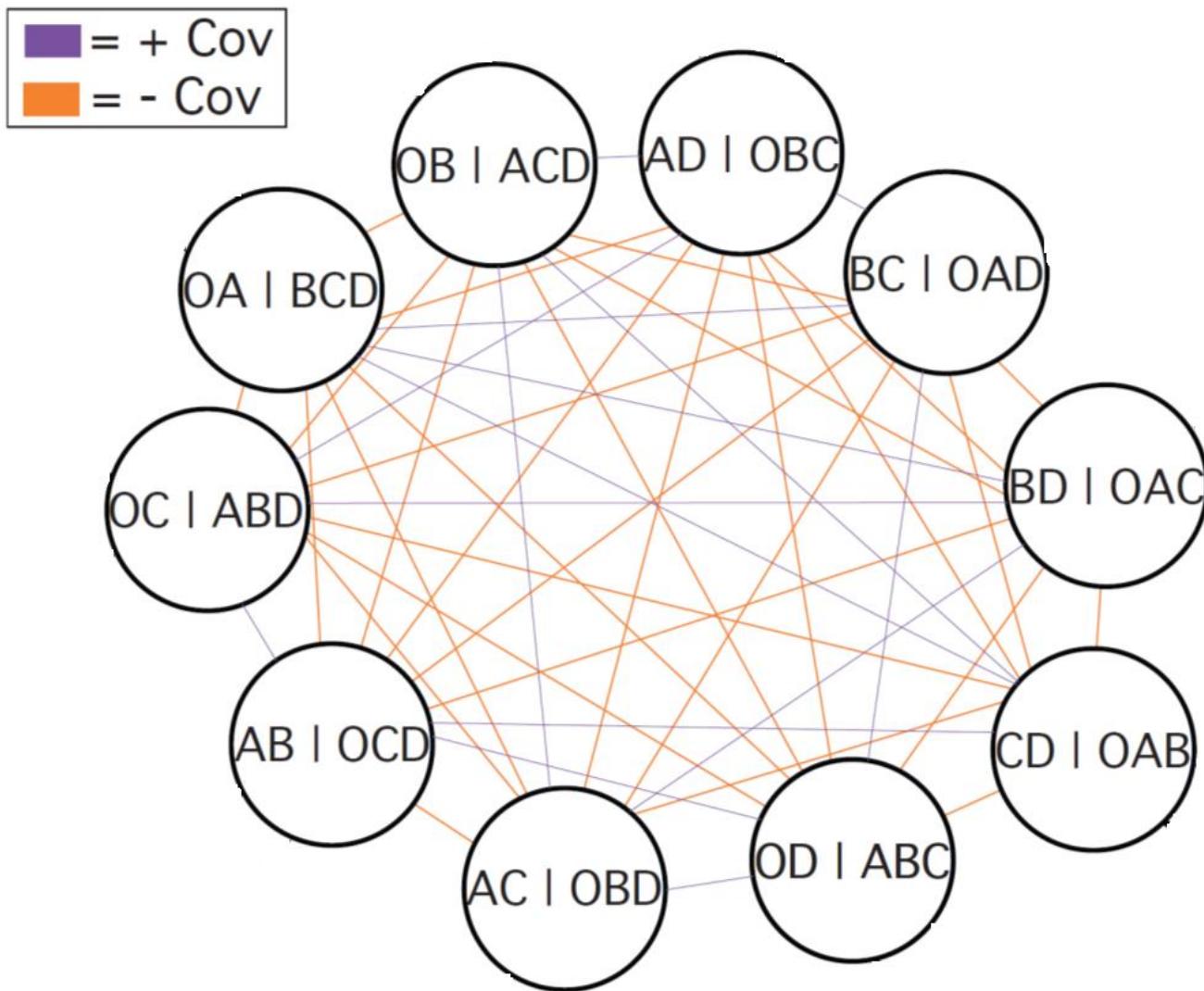


Bipartition Covariances



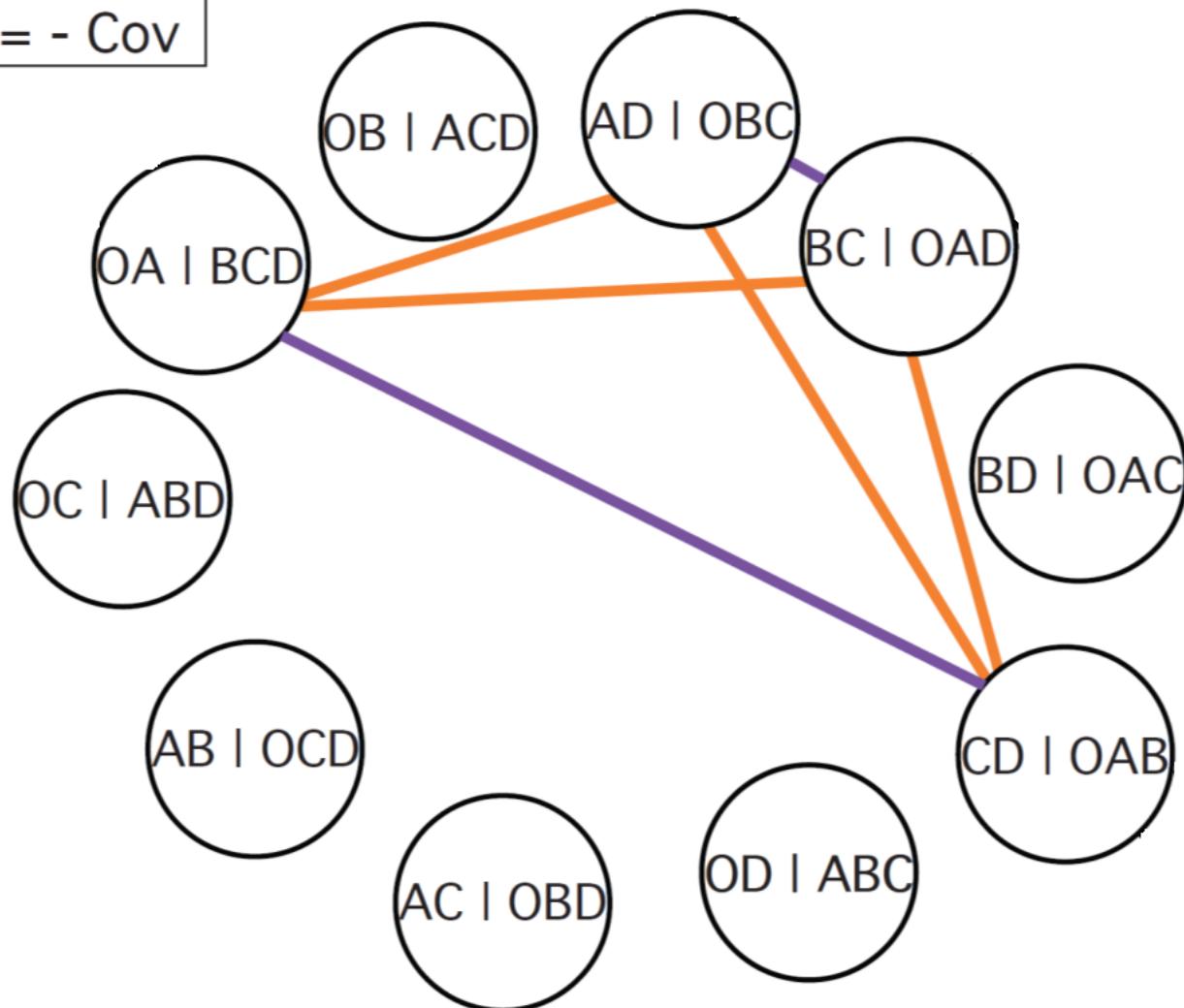
$$\begin{aligned}\text{Cov}(\text{orange, blue}) &= (.26) - (.51 * .51) \\ &= 0\end{aligned}$$

Networks of Bipartitions: Covariance Edge Weights



Networks of Bipartitions: Covariance Edge Weights

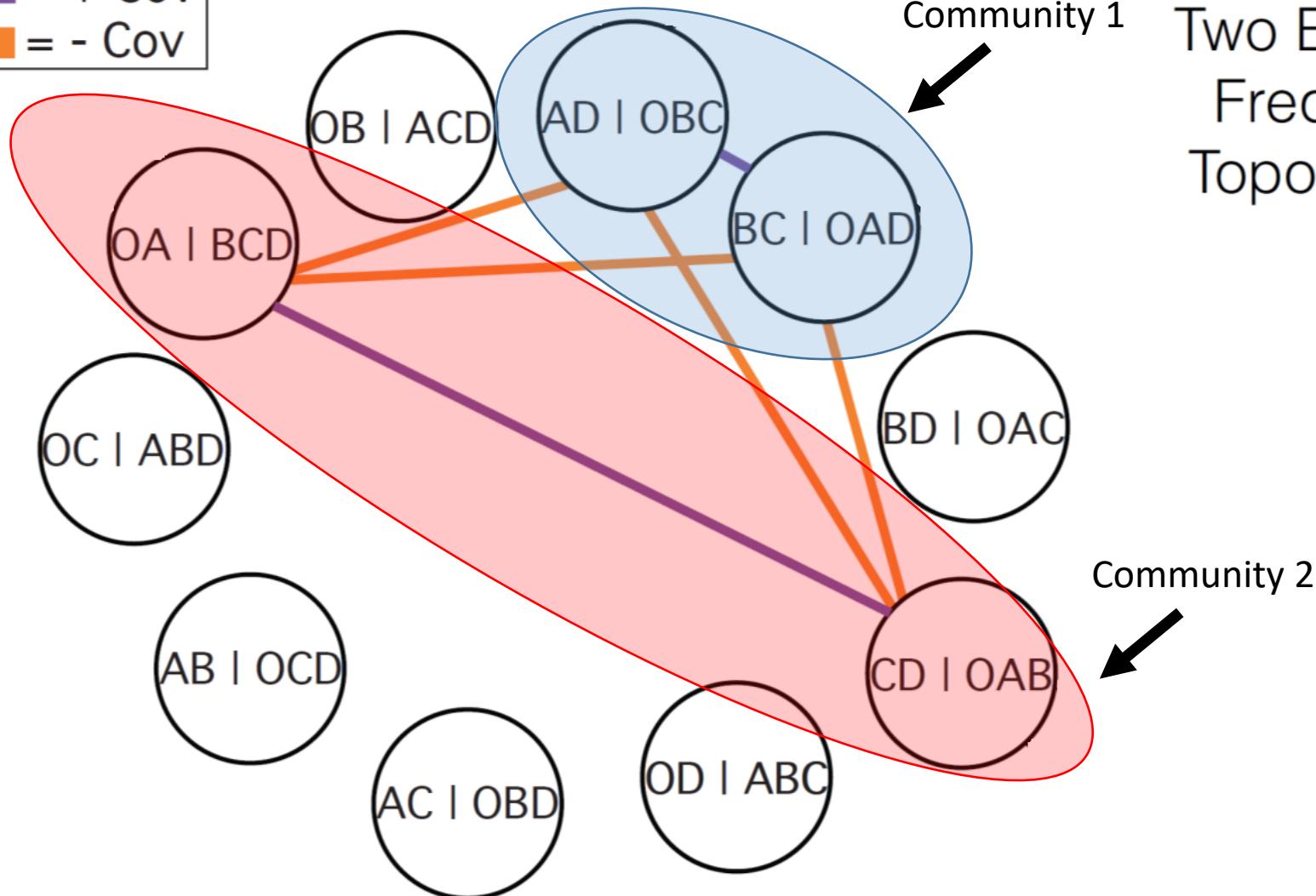
█ = + Cov
█ = - Cov



Two Equally
Frequent
Topologies

Networks of Bipartitions: Community Detection

■ = + Cov
■ = - Cov



Two Equally
Frequent
Topologies

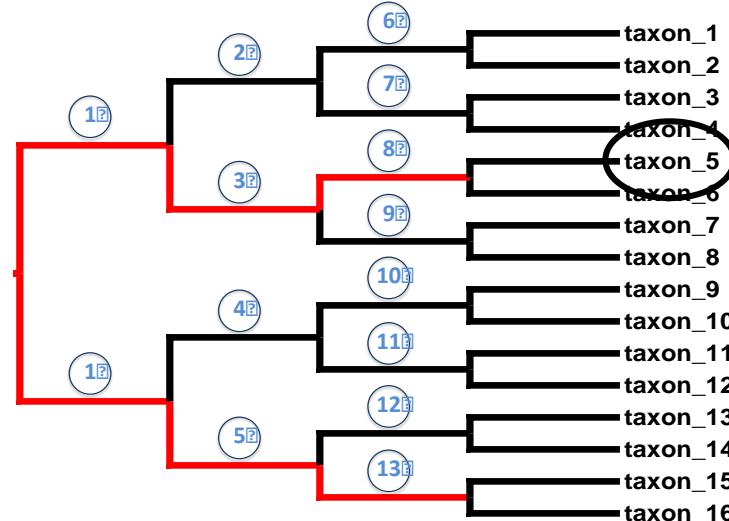
The “Perfect Case”: 50/50 Trees

$$\text{Cov}(\text{red}, \text{red}) = (.50) - (.50 * .50) \\ = .25$$

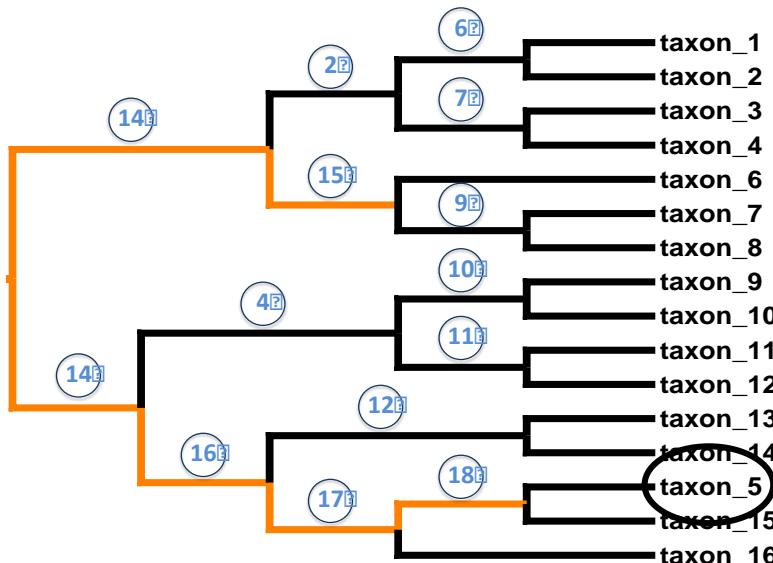
$$\text{Cov}(\text{black}, \text{black}) = (1) - (1 * 1) \\ = 0$$

$$\text{Cov}(\text{red}, \text{black}) = (.5) - (.5 * 1) \\ = 0$$

$$\text{Cov}(\text{red}, \text{orange}) = (0) - (.5 * .5) \\ = -.25$$

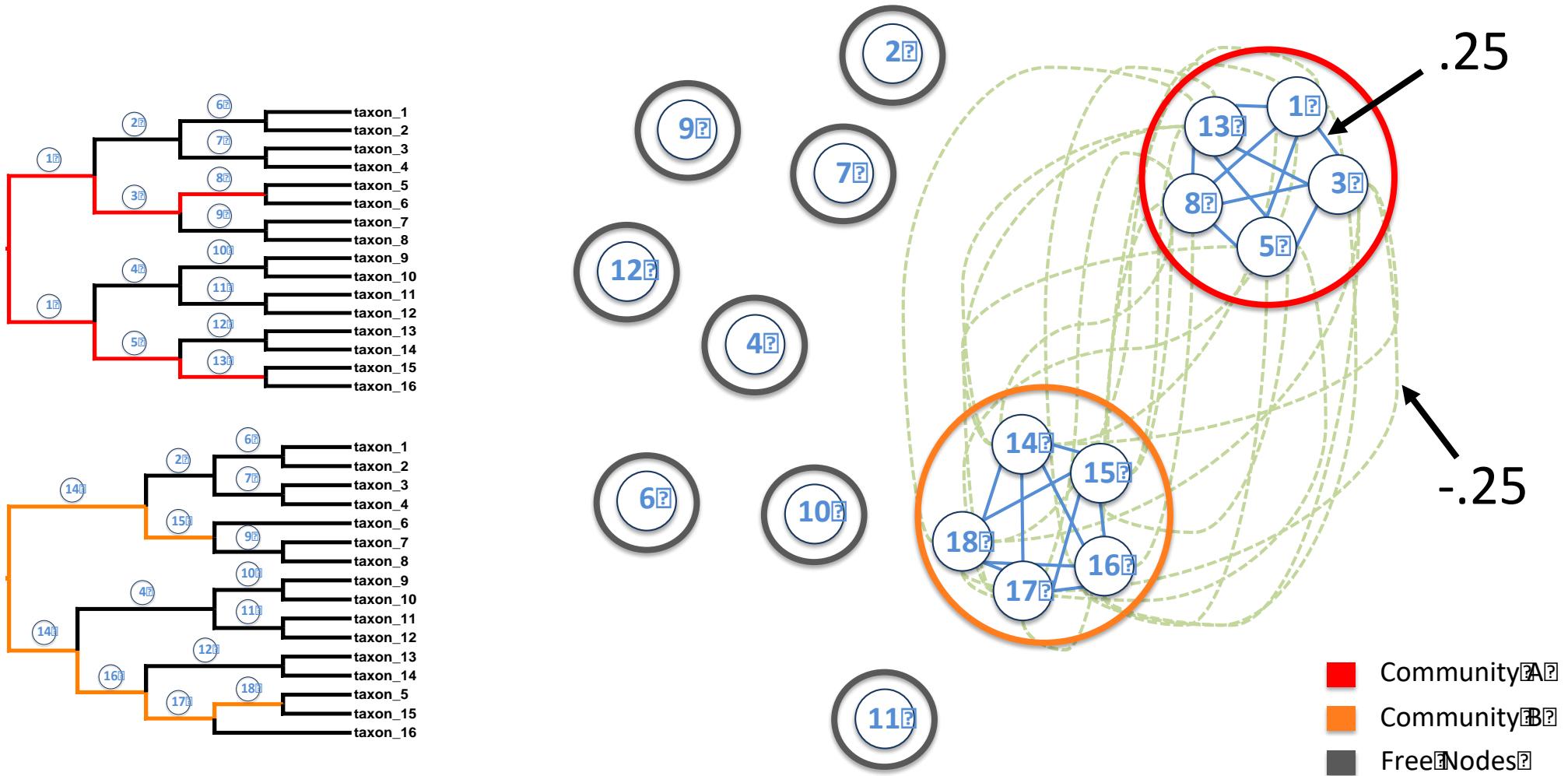


50% of Trees



50% of Trees

Hamiltonian Minimum



Method for Community Detection

- Constant-Potts Model(CPM):

$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [\mathcal{A}_{ij} - c^2(\lambda^+ - \lambda^-)]\delta(\sigma_i, \sigma_j)$$

Method for Community Detection

- Constant-Potts Model(CPM):

$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [A_{ij} - c^2(\lambda^+ - \lambda^-)]\delta(\sigma_i, \sigma_j)$$

- A_{ij} is the edge weight between node i and node j

Method for Community Detection

- Constant-Potts Model(CPM):

$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [A_{ij} - c^2(\lambda^+ - \lambda^-)] \delta(\sigma_i, \sigma_j)$$

- A_{ij} is the edge weight between node i and node j
- σ_i is the community node i belongs to

Method for Community Detection

- Constant-Potts Model(CPM):

$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [A_{ij} - c^2(\lambda^+ - \lambda^-) \delta(\sigma_i, \sigma_j)]$$

- A_{ij} is the edge weight between node i and node j
- σ_i is the community node i belongs to
- $\delta(\sigma_i, \sigma_j)$ is zero unless $\sigma_i = \sigma_j$

Method for Community Detection

- Constant-Potts Model(CPM):

$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [A_{ij} - c^2(\lambda^+ - \lambda^-)]\delta(\sigma_i, \sigma_j)$$

- A_{ij} is the edge weight between node i and node j
- σ_i is the community bipartition i belongs to
- $\delta(\sigma_i, \sigma_j)$ is zero unless $\sigma_i = \sigma_j$
- c^2 is community size squared

Method for Community Detection

- Constant-Potts Model(CPM):

$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [A_{ij} - c^2(\lambda^+ - \lambda^-)]\delta(\sigma_i, \sigma_j)$$

- A_{ij} is the edge weight between node i and node j
- σ_i is the community bipartition i belongs to
- $\delta(\sigma_i, \sigma_j)$ is zero unless $\sigma_i = \sigma_j$
- c^2 is community size squared
- Lambda values are tuning parameters
 - Only the difference matters in this case

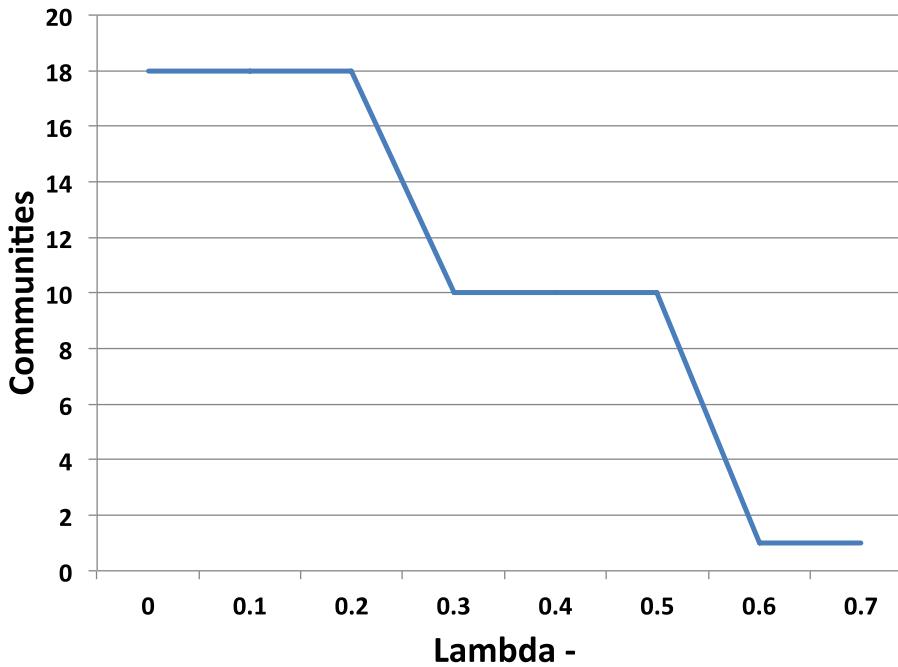
Method for Community Detection

- Constant-Potts Model(CPM):

$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [A_{ij} - c^2(\lambda^+ - \lambda^-)]\delta(\sigma_i, \sigma_j)$$

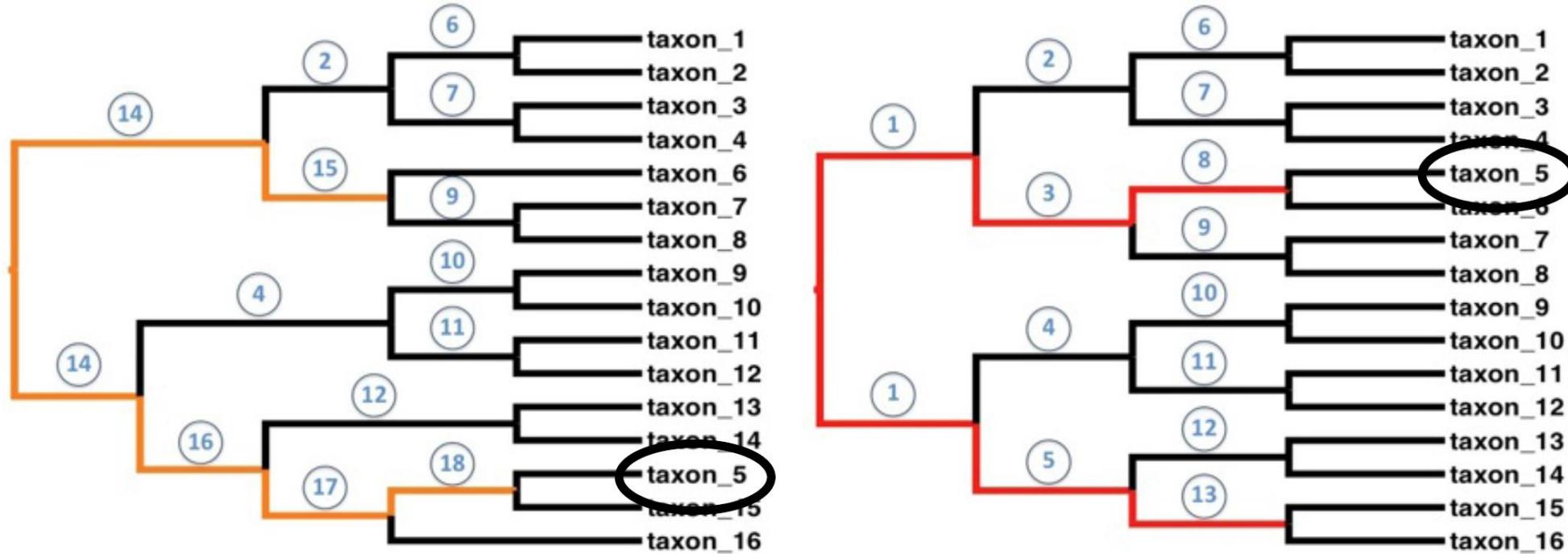
- A_{ij} is the edge weight between node i and node j
- σ_i is the community bipartition i belongs to
- $\delta(\sigma_i, \sigma_j)$ is zero unless $\sigma_i = \sigma_j$
- c^2 is community size squared
- Lambda values are tuning parameters
 - Only the difference matters in this case
- Hamiltonian, $\mathcal{H}(\{\sigma\})$, minimized by an optimization algorithm

Method for Community Detection



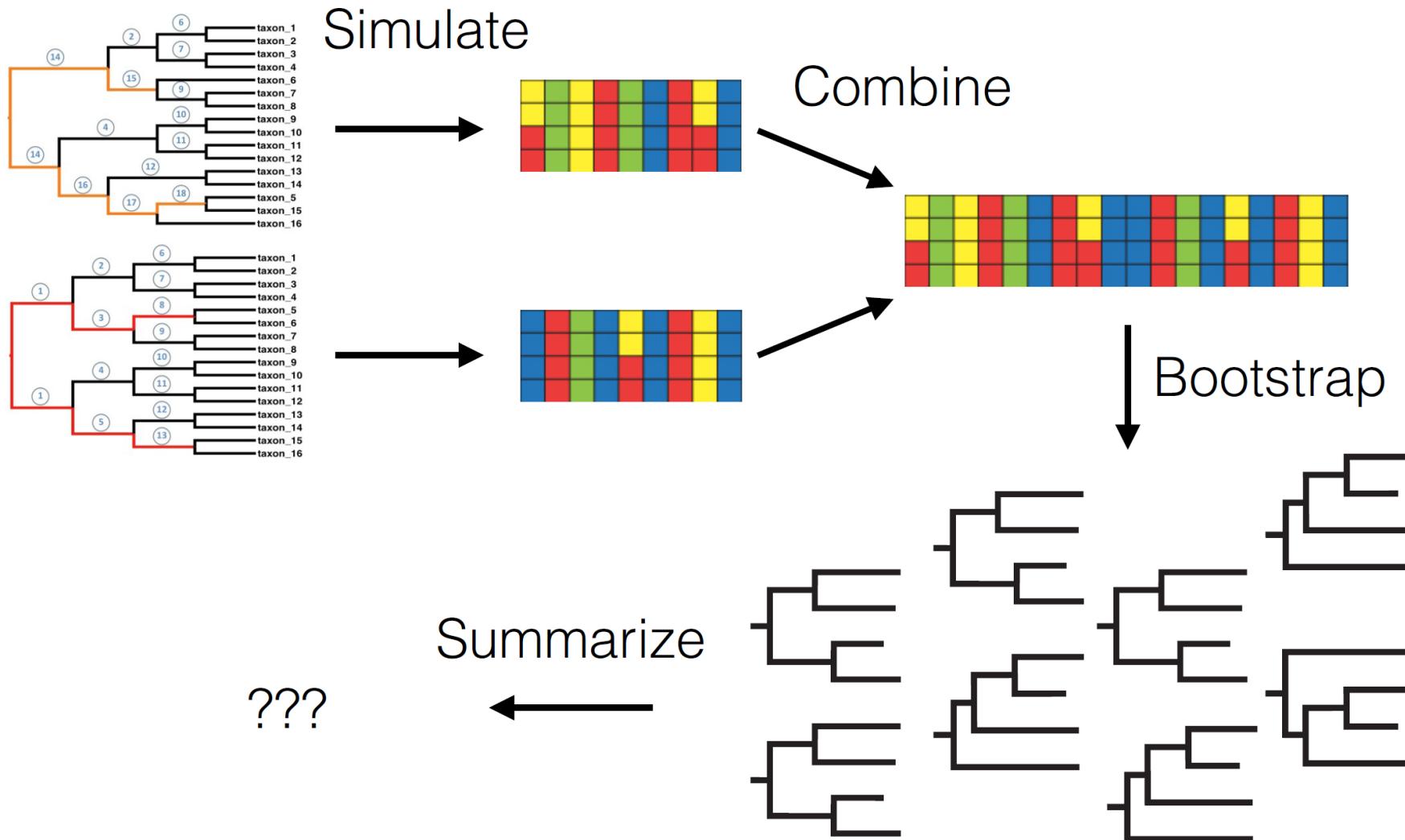
- λ^+ is fixed at some value and λ^- is varied
- The largest range of lambda values that produces a single community structure (and that is between the two extremes) is the most stable community structure in our dataset

Example: Simulation Experiment

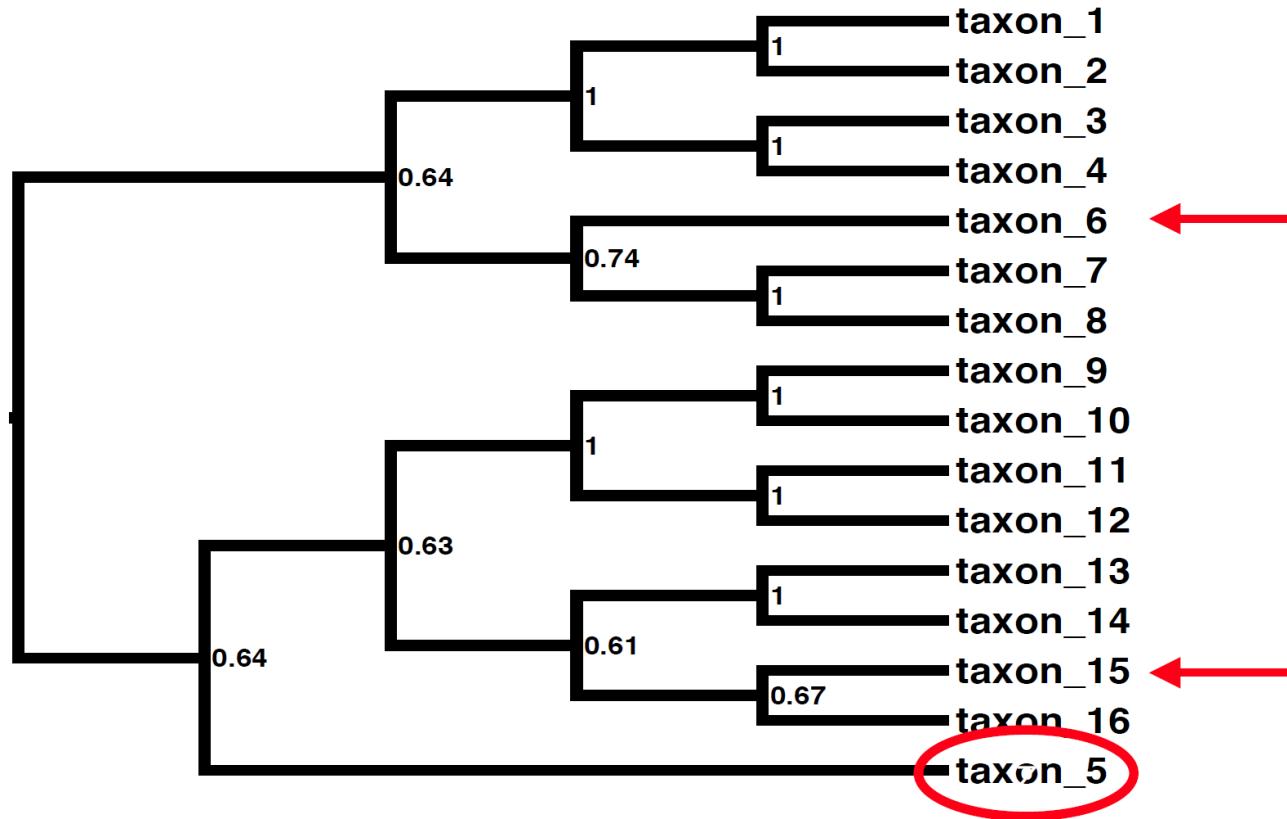


Topologies used for simulating two halves of an alignment.

Example: Simulation Experiment

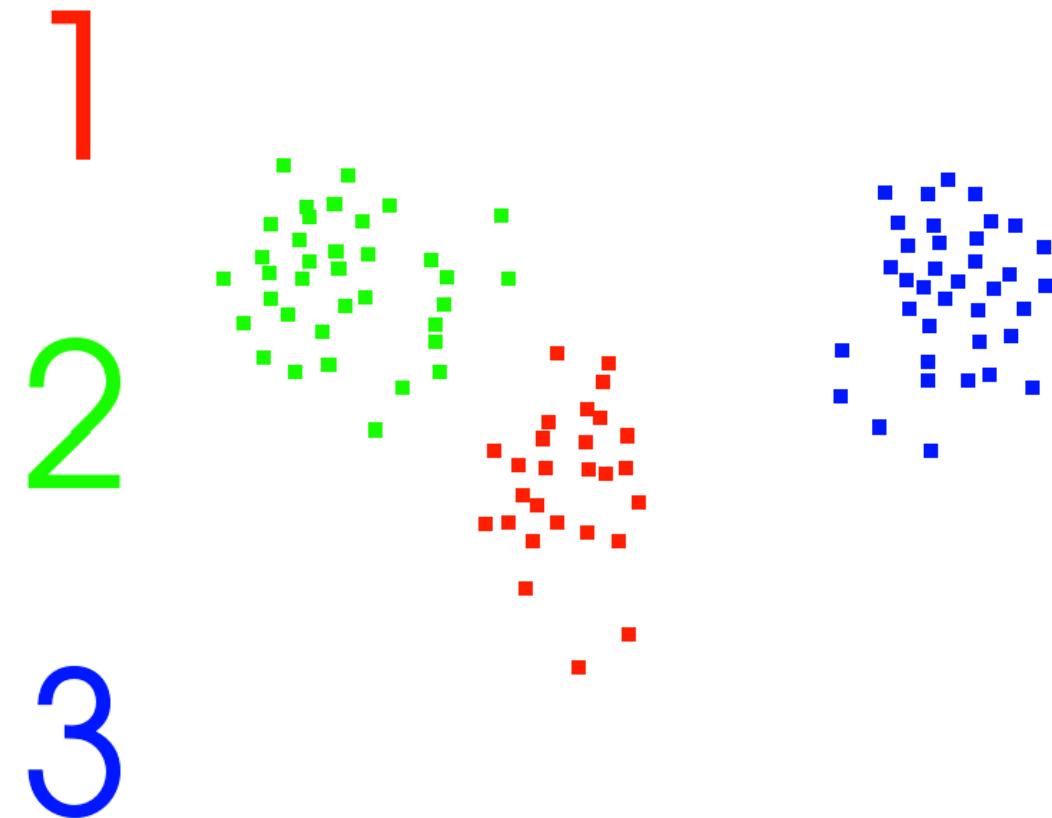


Consensus Tree Contains Misleading Information



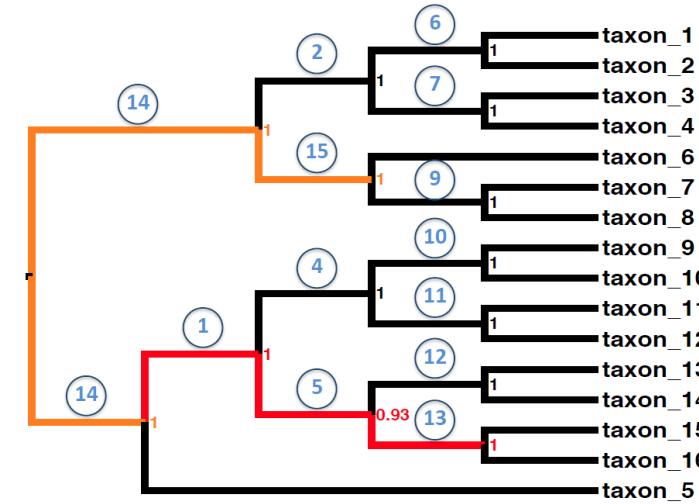
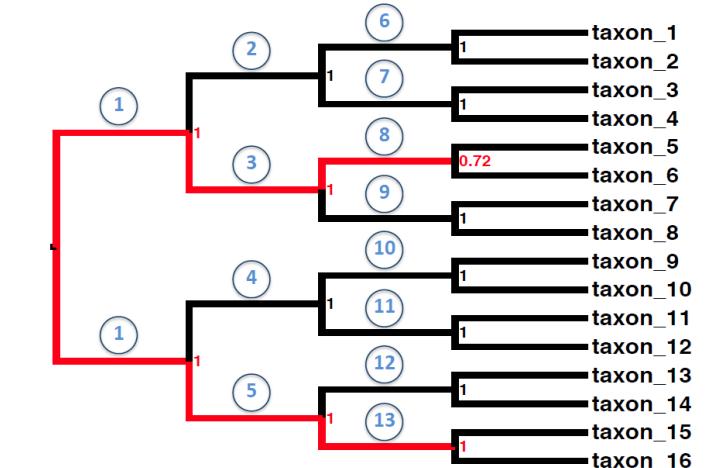
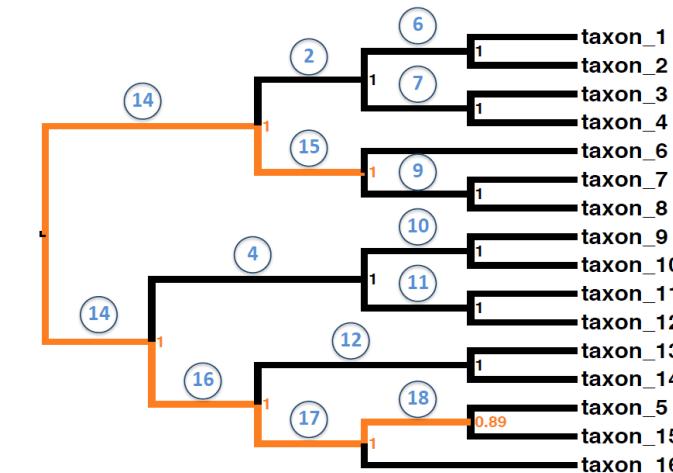
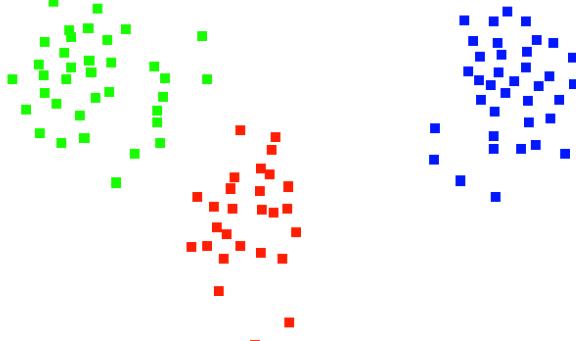
Majority-Rule Consensus Tree

Topology Based Community Detection

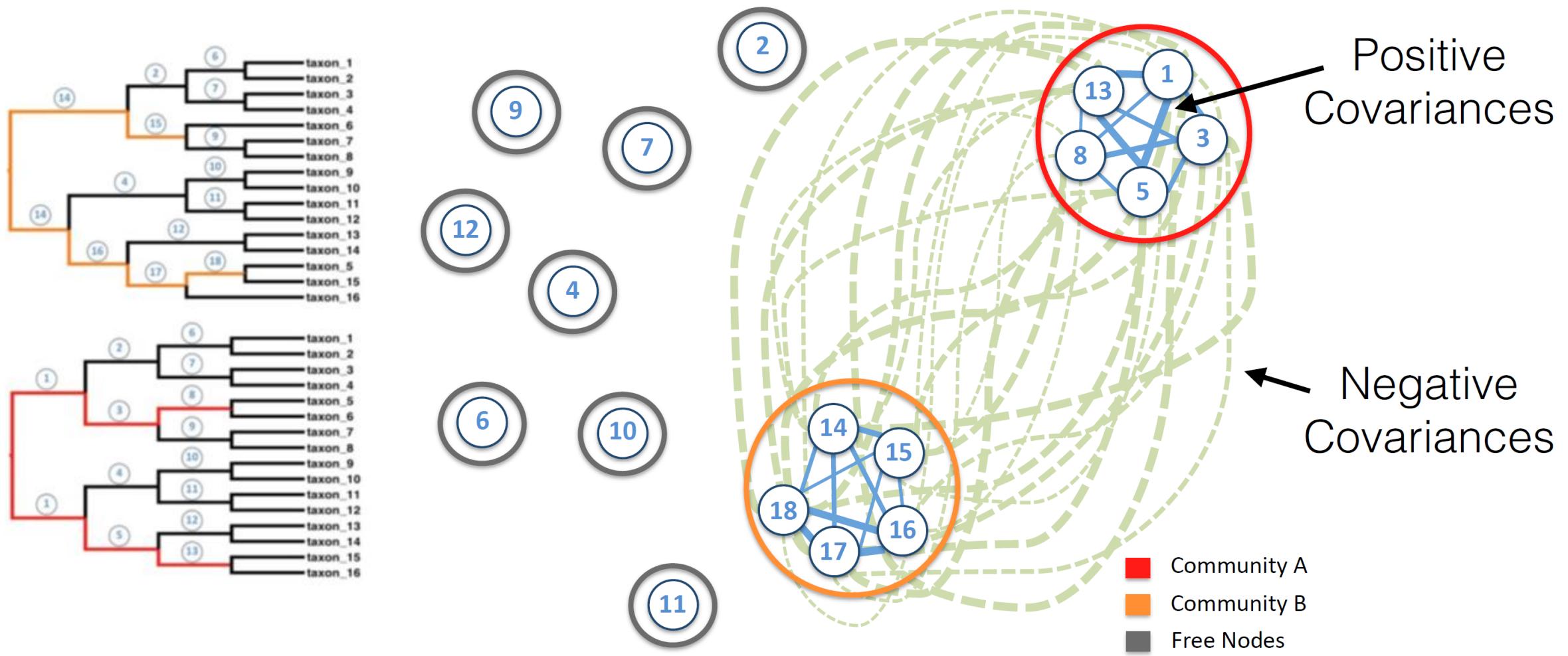


Networks Detect Strong Conflict

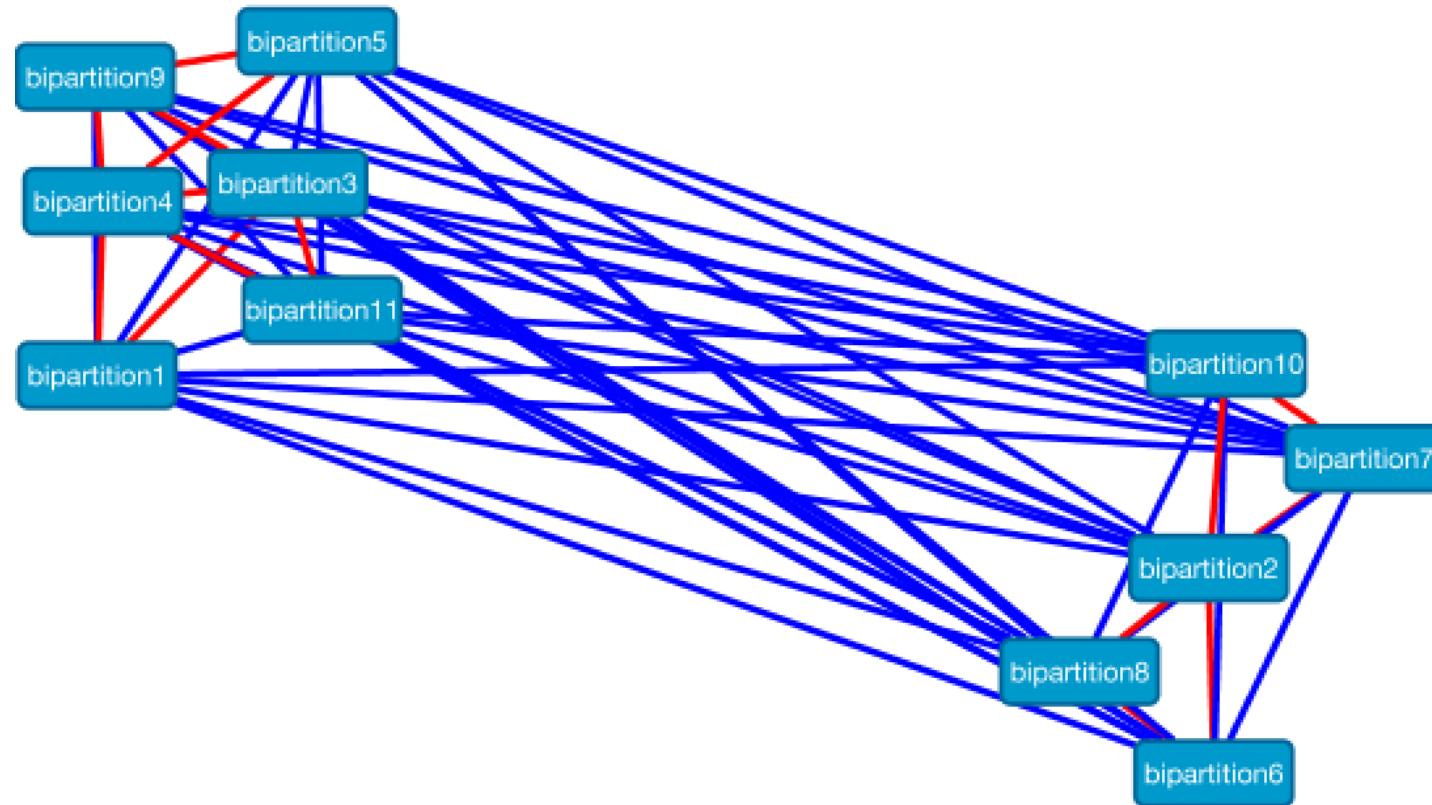
1
2
3



Network of Bipartitions

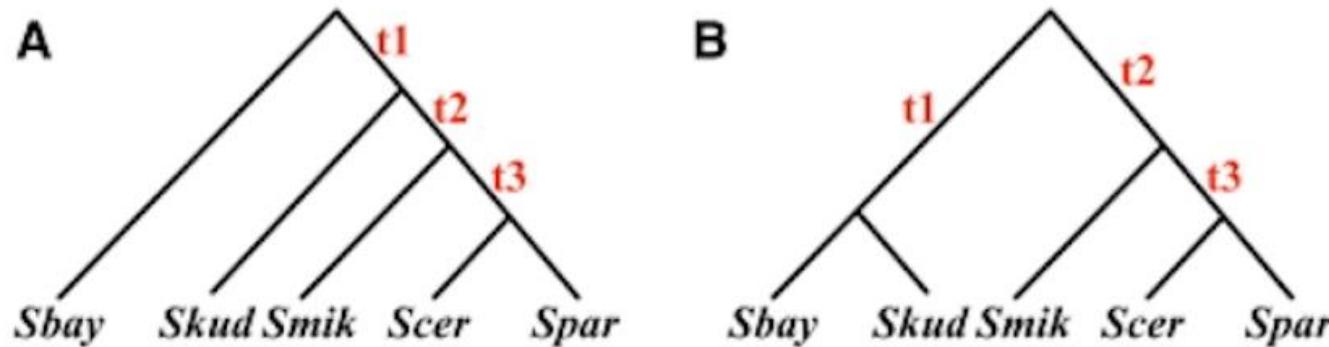


Networks of Bipartitions (in Cytoscape!)



Example: Empirical Data

- Yeast dataset with 5 species, 106 loci
- 106 gene trees were reconstructed using maximum parsimony



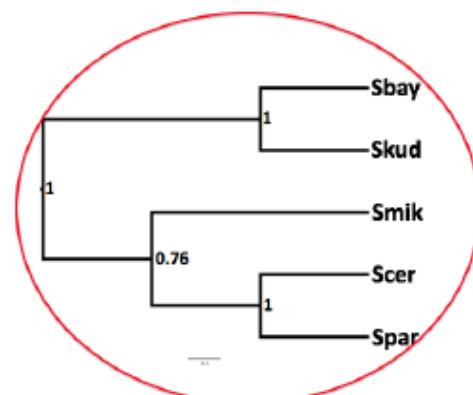
The Probability of a Gene Tree Topology within a Phylogenetic Network with Applications to Hybridization Detection

Yun Yu, James H. Degnan, Luay Nakhleh

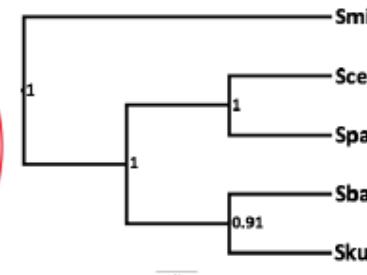
Published: April 19, 2012 • <http://dx.doi.org/10.1371/journal.pgen.1002660>

Topology-based Network Analysis

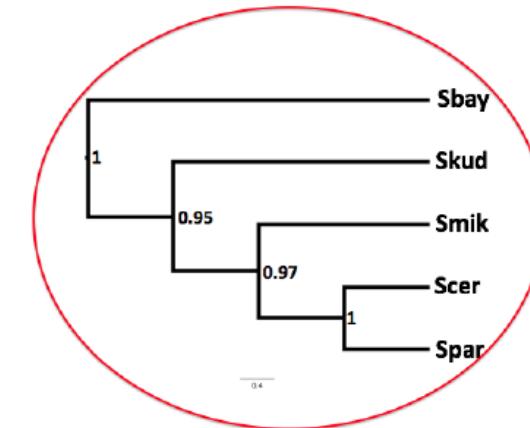
- Affinity matrix
 - Reciprocal of Tree-to-Tree distances
- Detect communities
 - Discovered 11 communities
- Consensus trees for each community
 - Consensus trees of two largest communities recovers 2 candidate species trees.



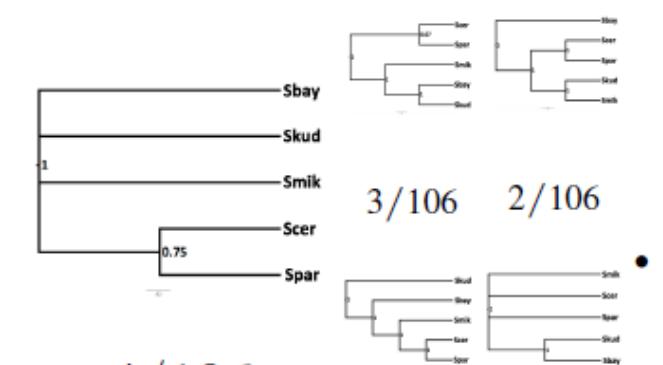
17/106



11/106



62/106



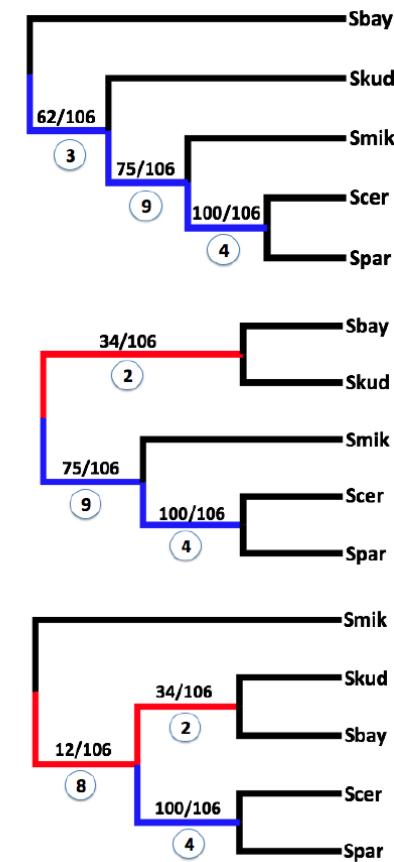
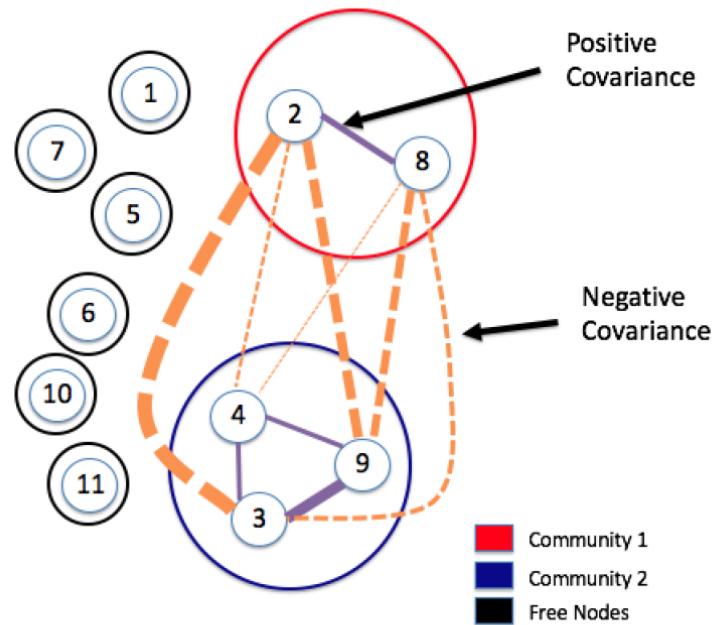
4/106

2/106 2/106

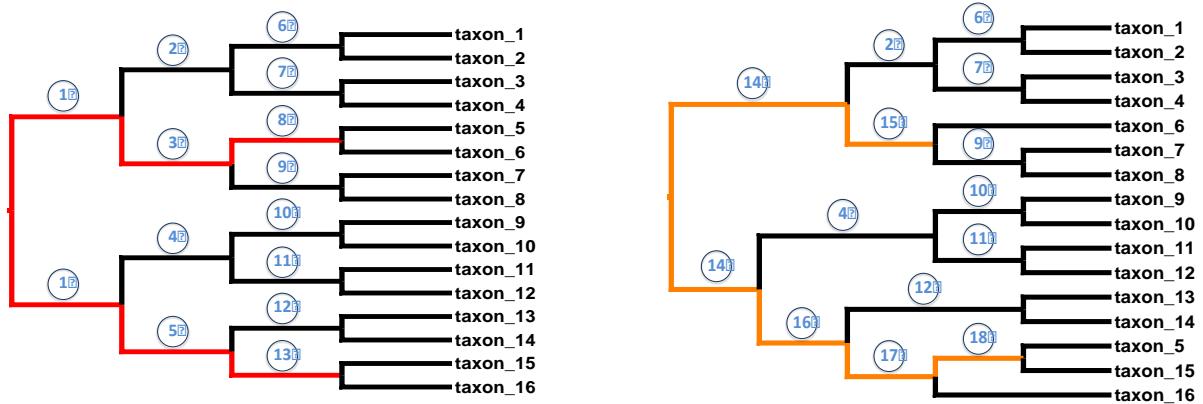
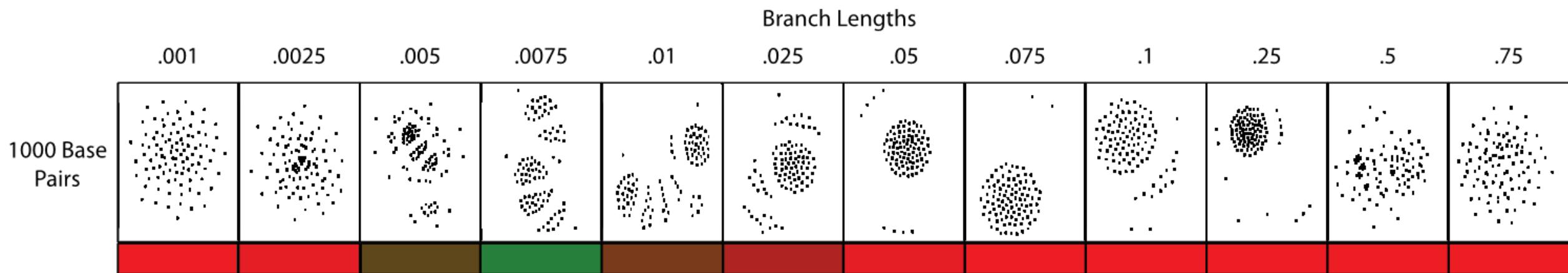
• • •

Bipartition-based Network Analysis

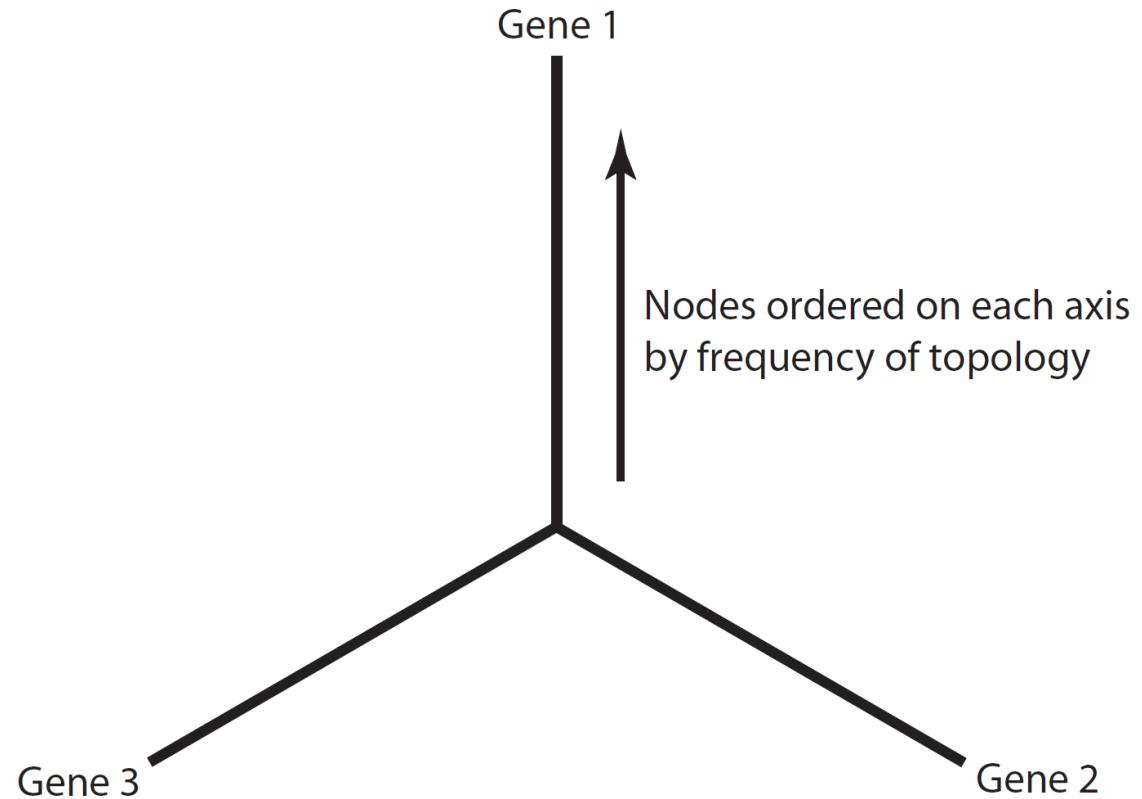
- Covariance matrix based on presence or absence of bipartitions in the gene trees



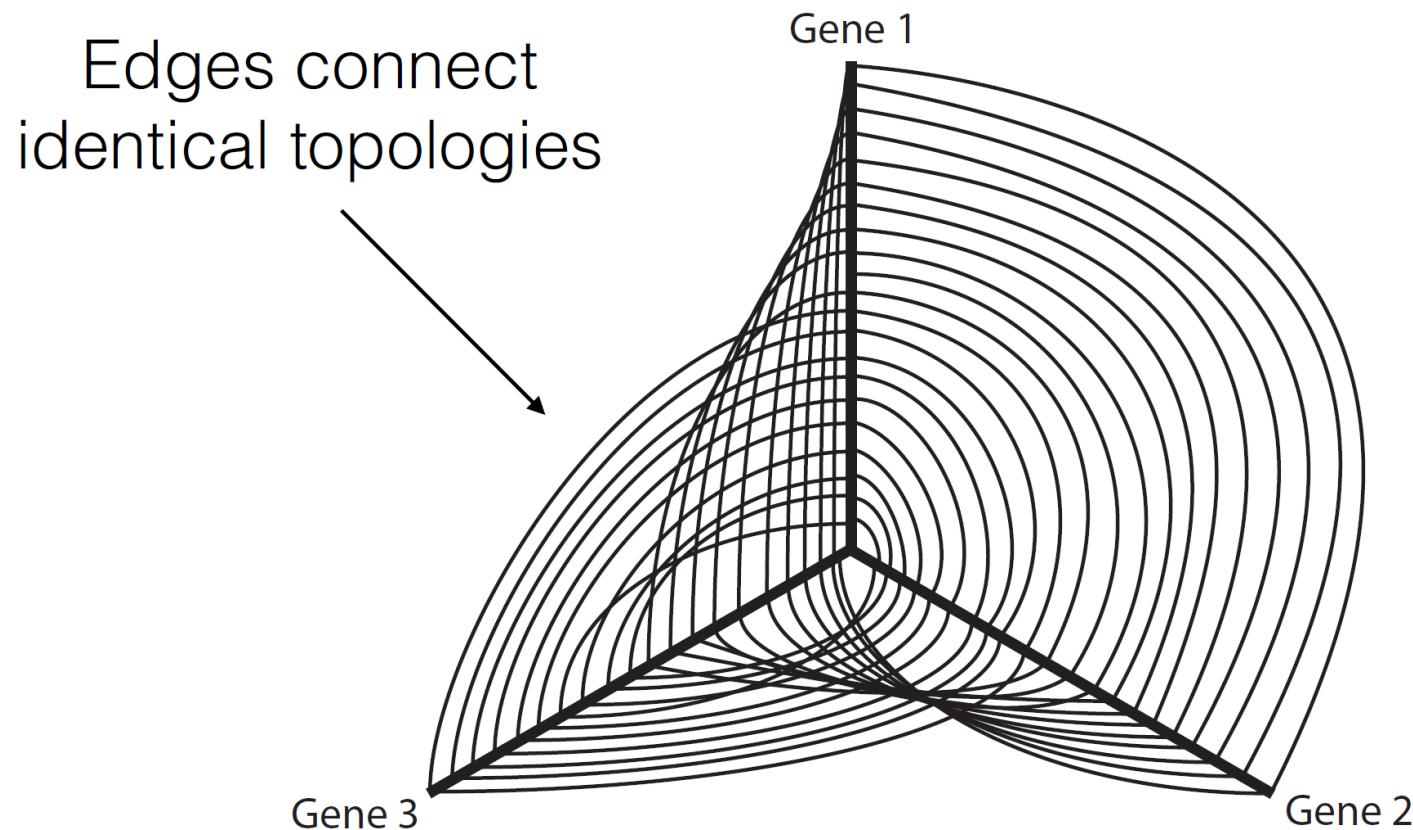
Variation of Branch Length



Other Network Visualizations

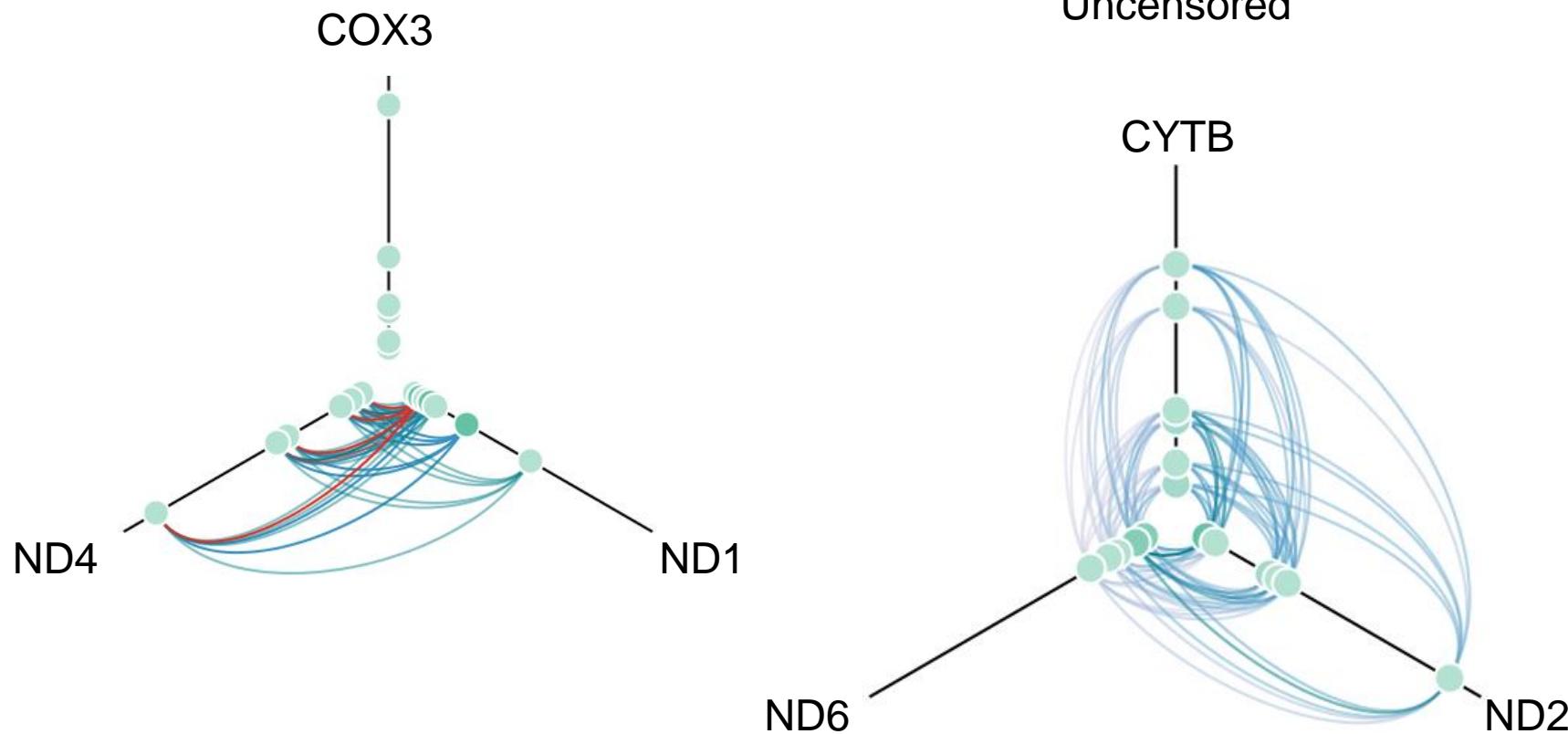


Other Network Visualizations



Network Visualizations

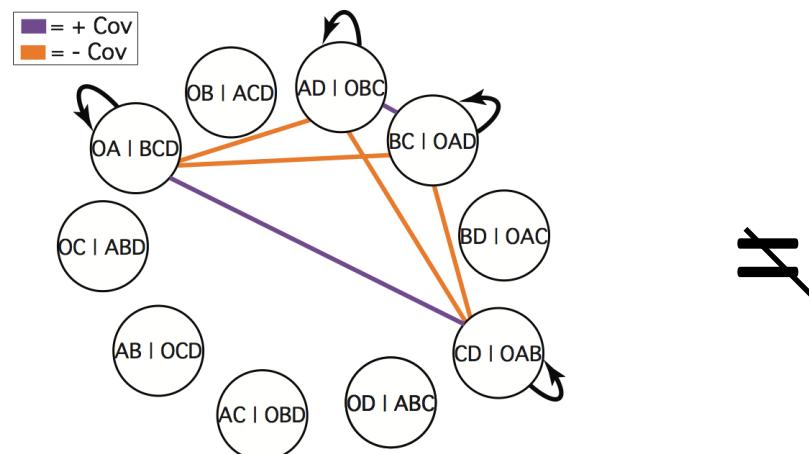
Censored
RF distances below 4 removed



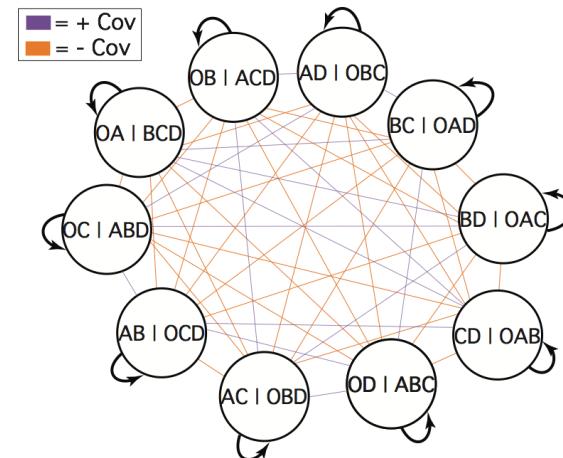
Assessing Model Fit

Using parametric bootstrapping or posterior prediction, we can compare network structures between observed and simulated datasets.

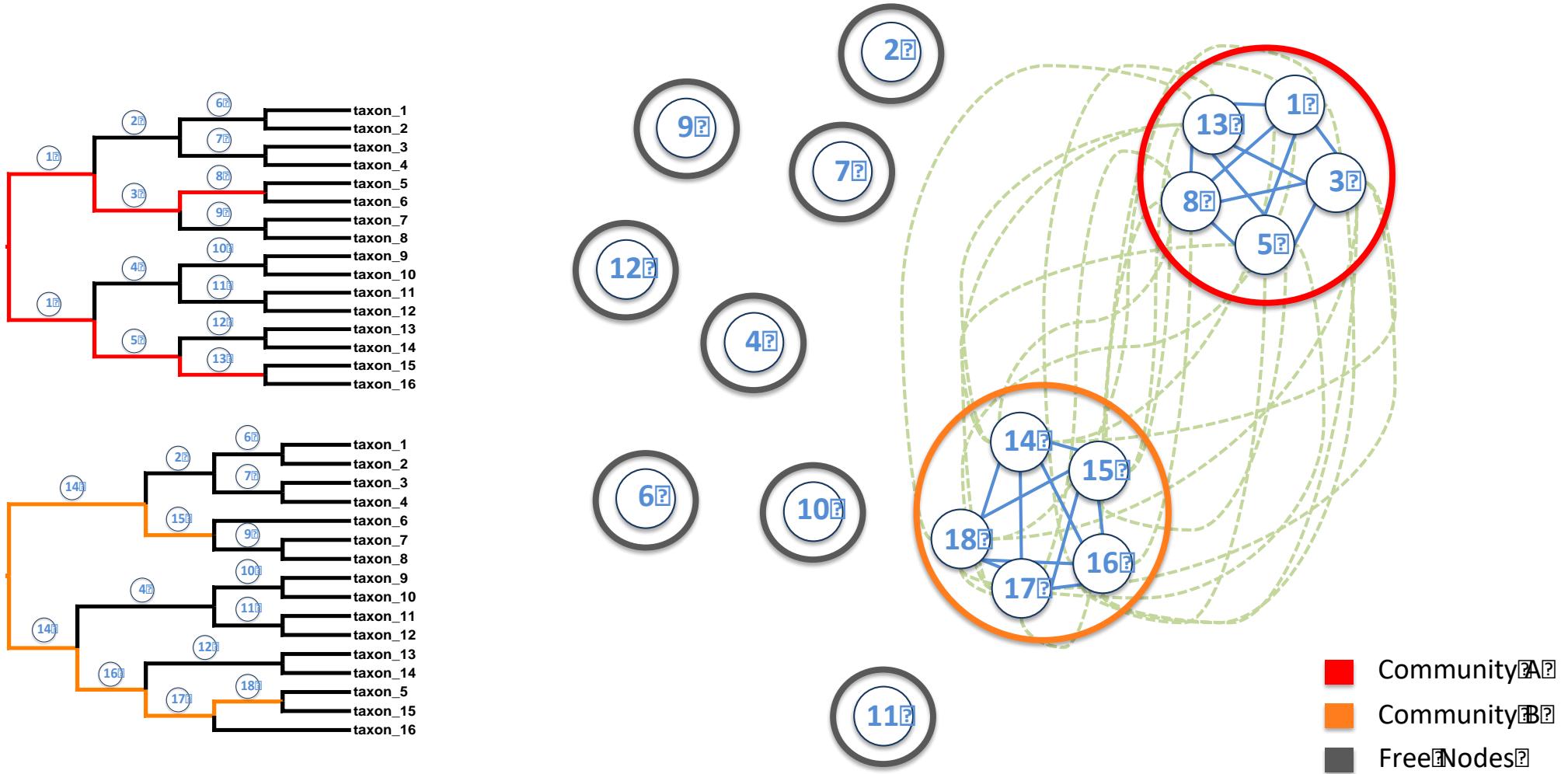
Empirical



Simulated



The Network



Calculating the Hamiltonian

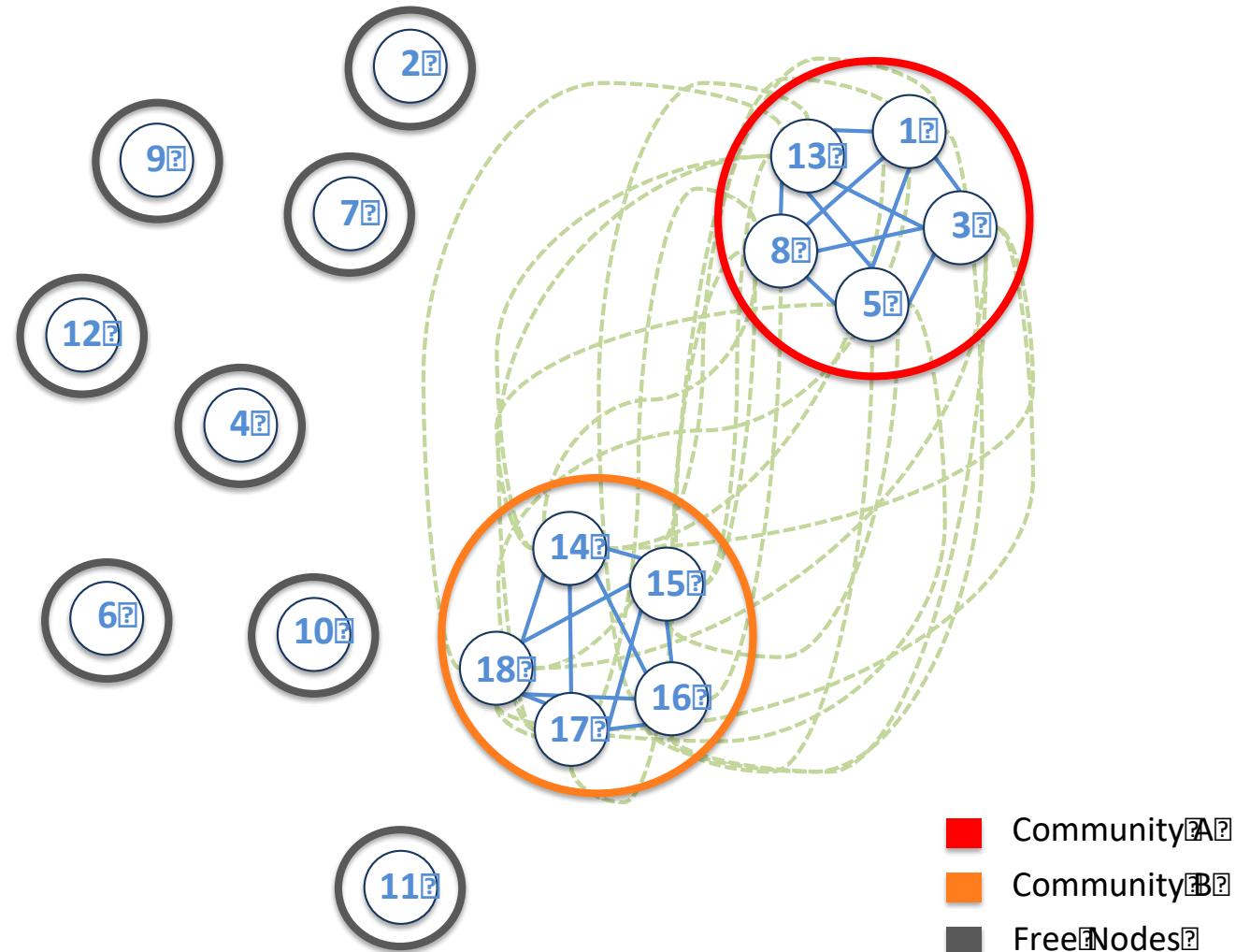
$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [\mathcal{A}_{ij} - c^2(\lambda^+ - \lambda^-)]\delta(\sigma_i, \sigma_j)$$

$$H_a = -(2(10*.25)+(5*.25)) = -6.25$$

$$H_b = -(2*(10*.25) + (5*.25)) = -6.25$$

$$H_a + H_b = H = -12.5$$

Hamiltonian minimum



Calculating the Hamiltonian

$$\mathcal{H}(\{\sigma\}) = - \sum_{ij} [\mathcal{A}_{ij} - c^2(\lambda^+ - \lambda^-)] \delta(\sigma_i, \sigma_j)$$

$$H_a = -(2*((6*.25) + 4*-.25) + (5*.25)) = -2.25$$

$$H_b = -(2*((6*.25) + 4*-.25) + (5*.25)) = -2.25$$

$$H_a + H_b = H = -4.5$$

Hamiltonian not minimized

No closed form solution!

**Need optimization algorithm
(simulated annealing)**

