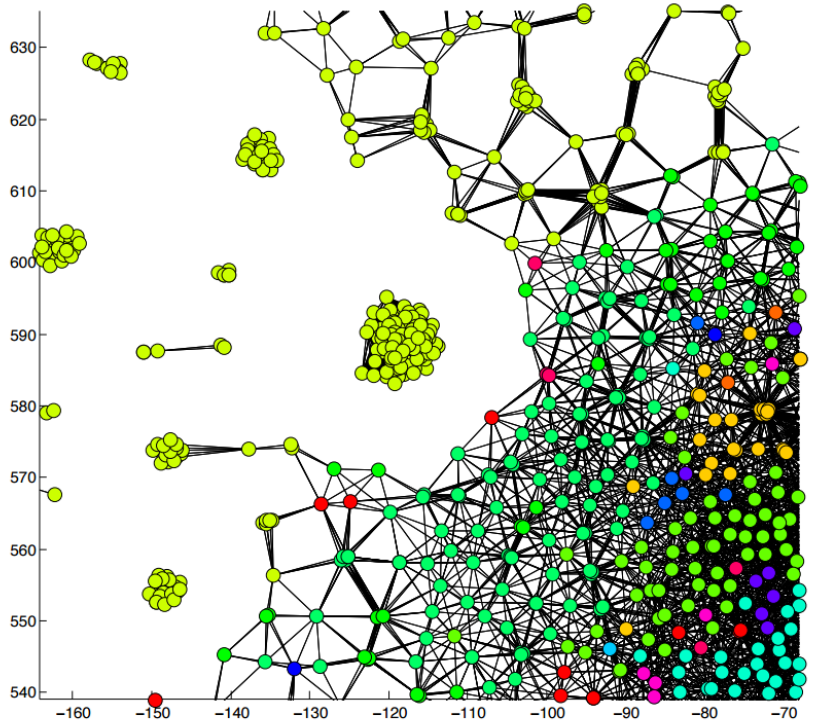
# TreeScaper: Using Networks to Explore, Quantify, and Summarize Phylogenetic Tree Space

NC STATE UNIVERSITY




**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

Jeremy Ash

Wen Huang

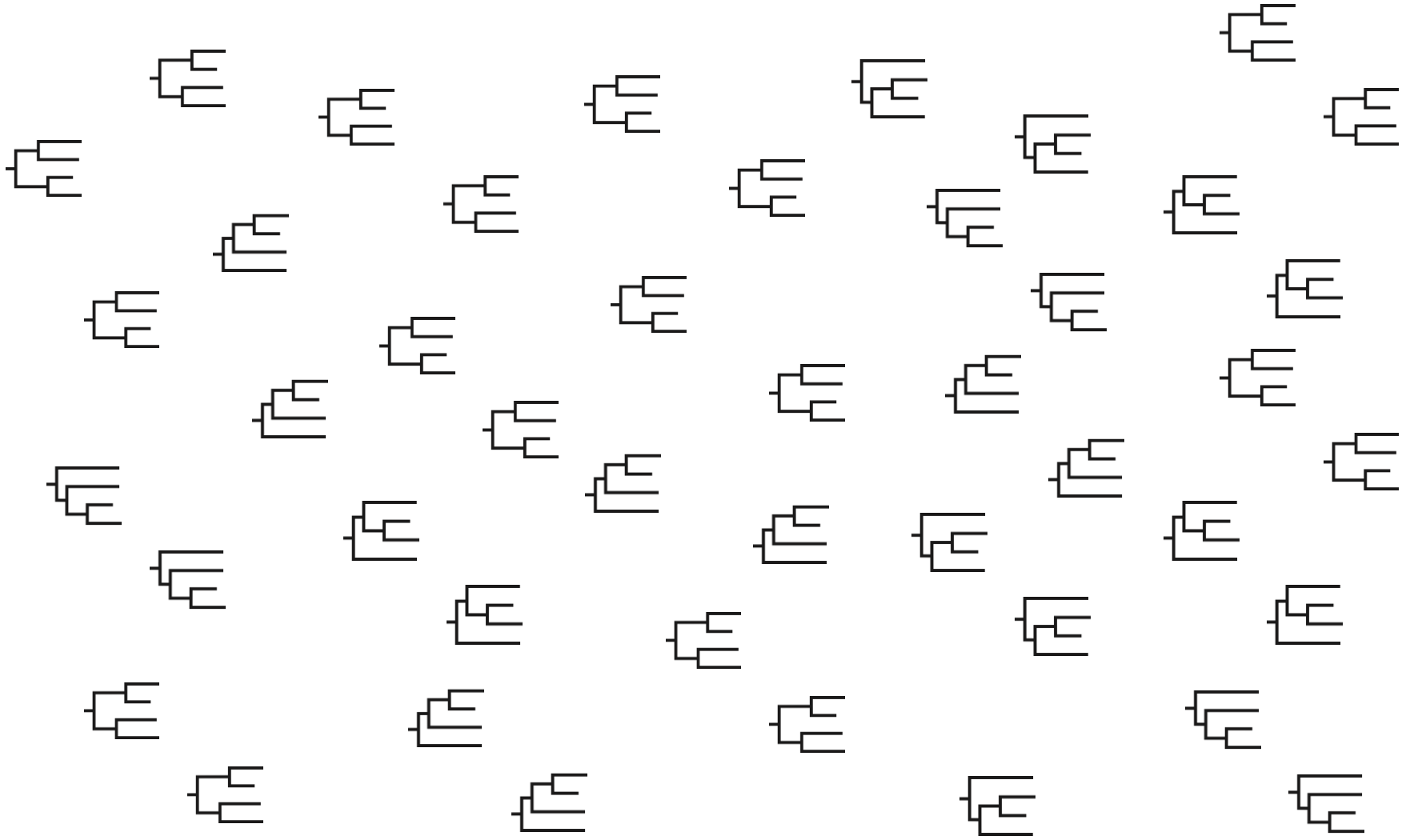
Paul Van Dooren

Jim Wilgenbusch

# Part 1

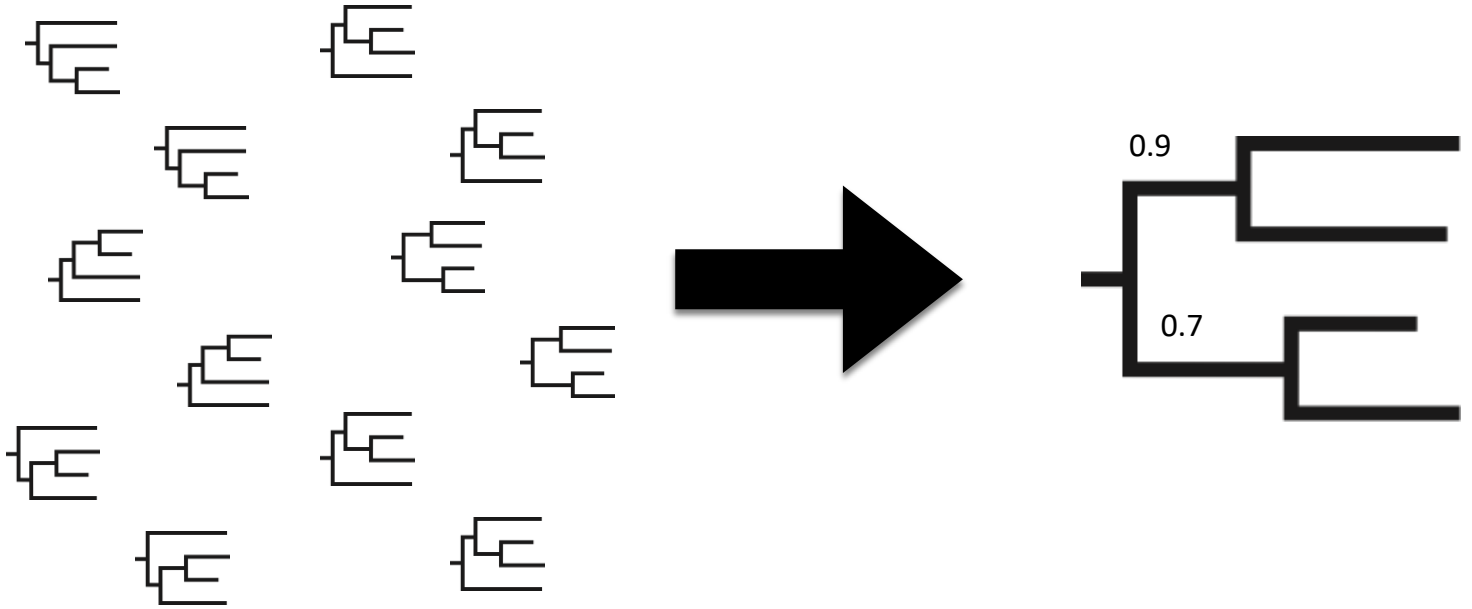
Non-Linear Dimensionality Reduction in TreeScaper

# Tree Sets



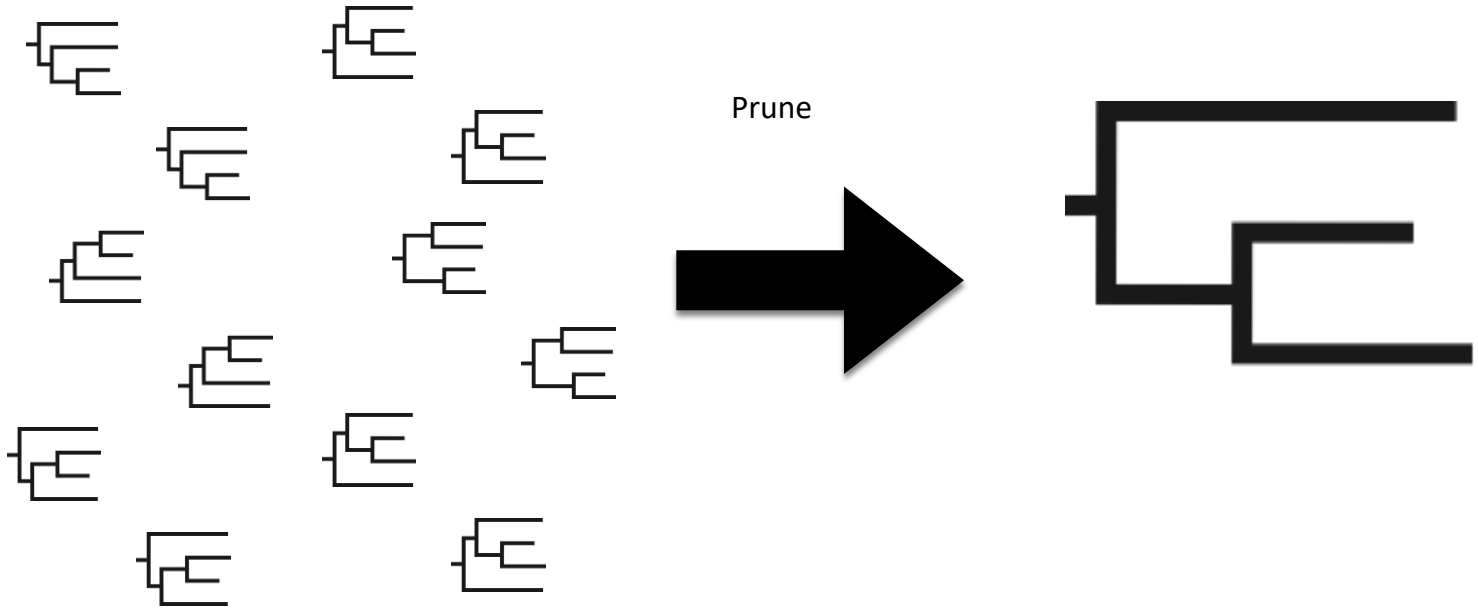
# Summarizing Tree Sets

- Consensus trees



# Summarizing Tree Sets

- Consensus trees
- Agreement subtrees



# Summarizing Tree Sets

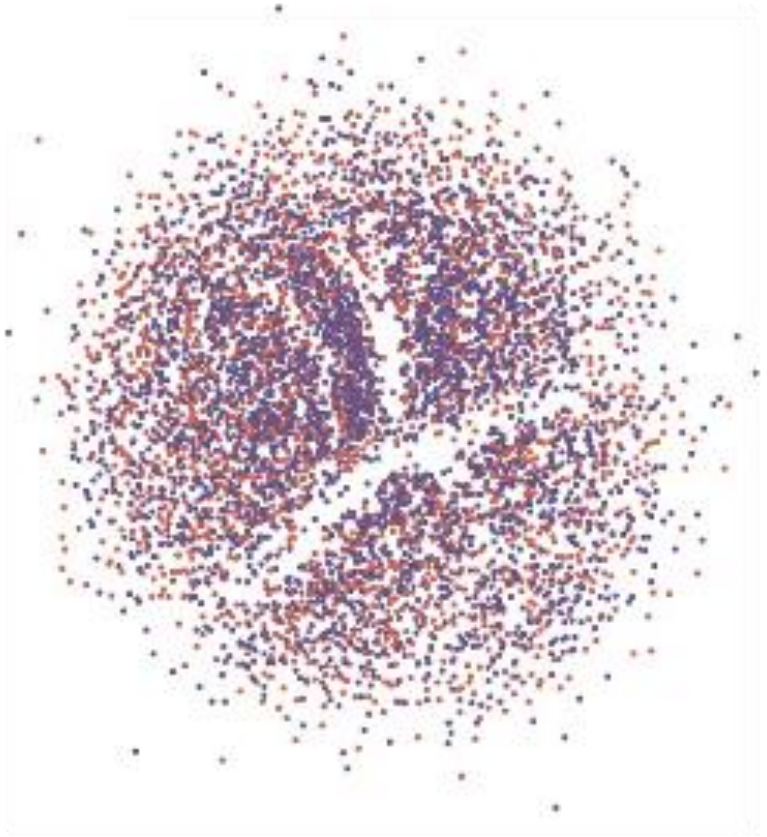
- Consensus trees
- Agreement subtrees
- Clustering

**Clustering genes of common evolutionary history**

Kevin Gori[1], Tomasz Suchan[2], Nadir Alvarez[2], Nick Goldman[1,*] and Christophe Dessimoz[1,2,3,4,5,*]

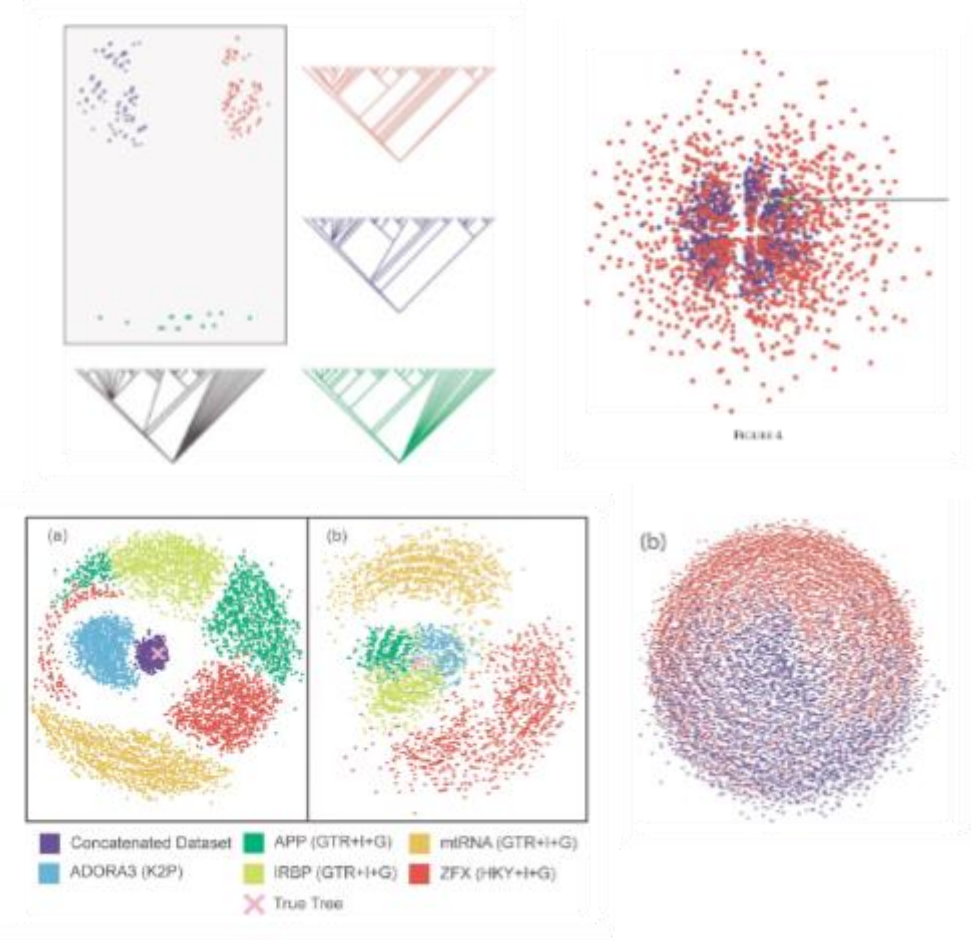
# Summarizing Tree Sets

- 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.

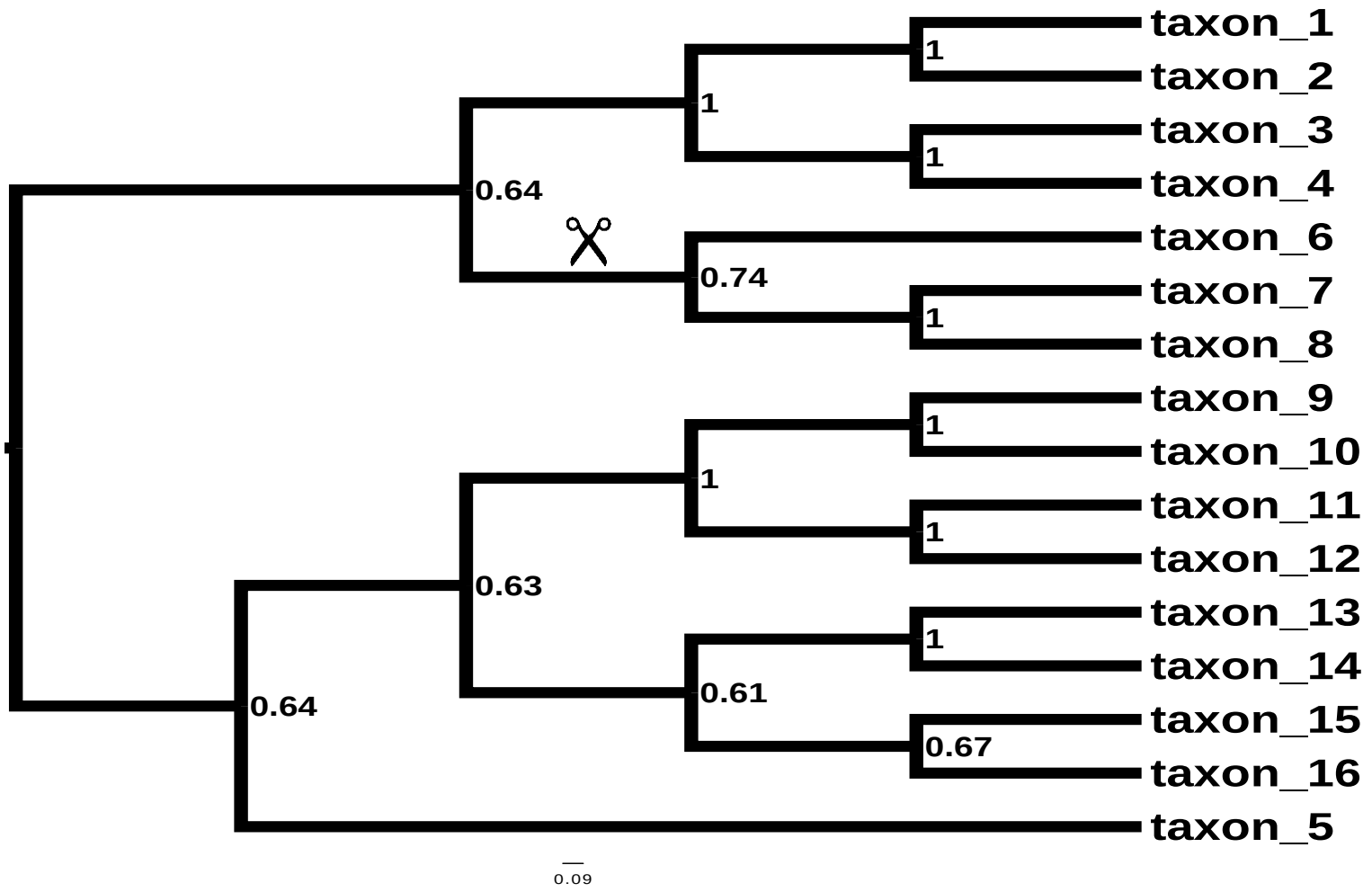
# Our Contribution

- Does the Nonlinear Dimensionality Reduction (NLDR) method matter?
  - How to measure the quality of the NLDR visualization?
- What is the intrinsic dimensionality of the data?
  - Is 3D needed to fit tree-to-tree distances?
- Develop software for exploring large sets of trees.
  - Written in C++ with efficient computational methods

# Phylogenetic Bipartitions

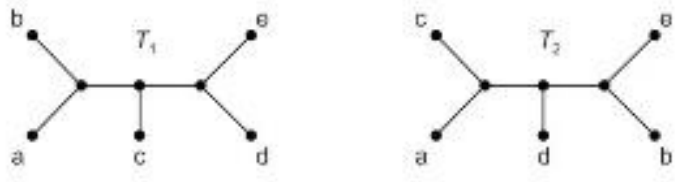
**Bipartition:**

- **A division of the elements of the set into two groups**
- **In this case:**
  - **(6,7,8)|(the rest)**



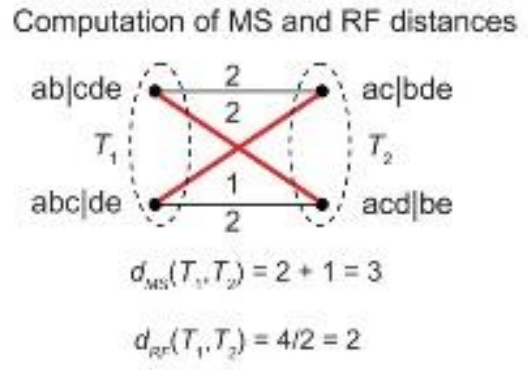
# Plot Trees as a "Phylogenetic Landscape"

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



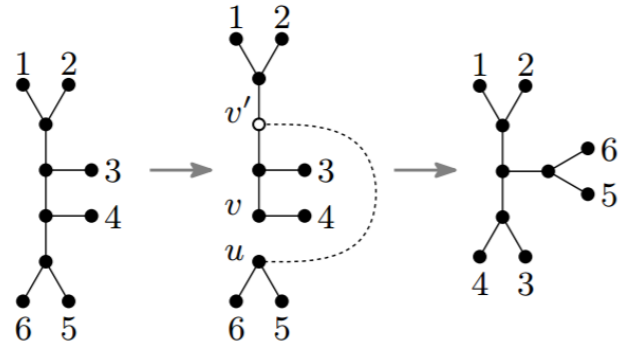

***RF and Matching***

*Bogdanowicz et al. 2012.*



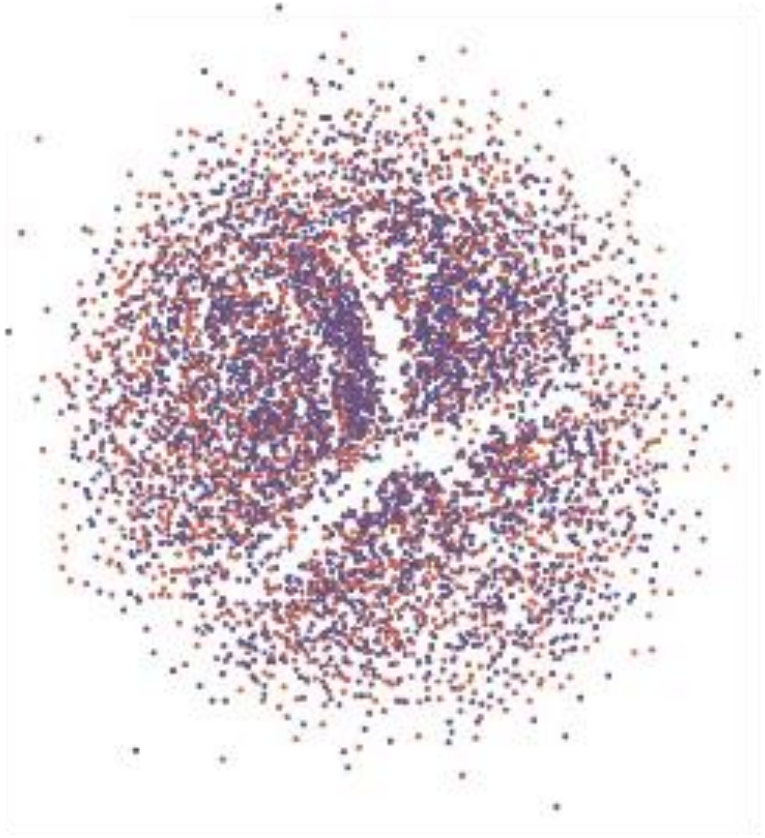

***SPR***

*Whidden and Matsen 2015.*



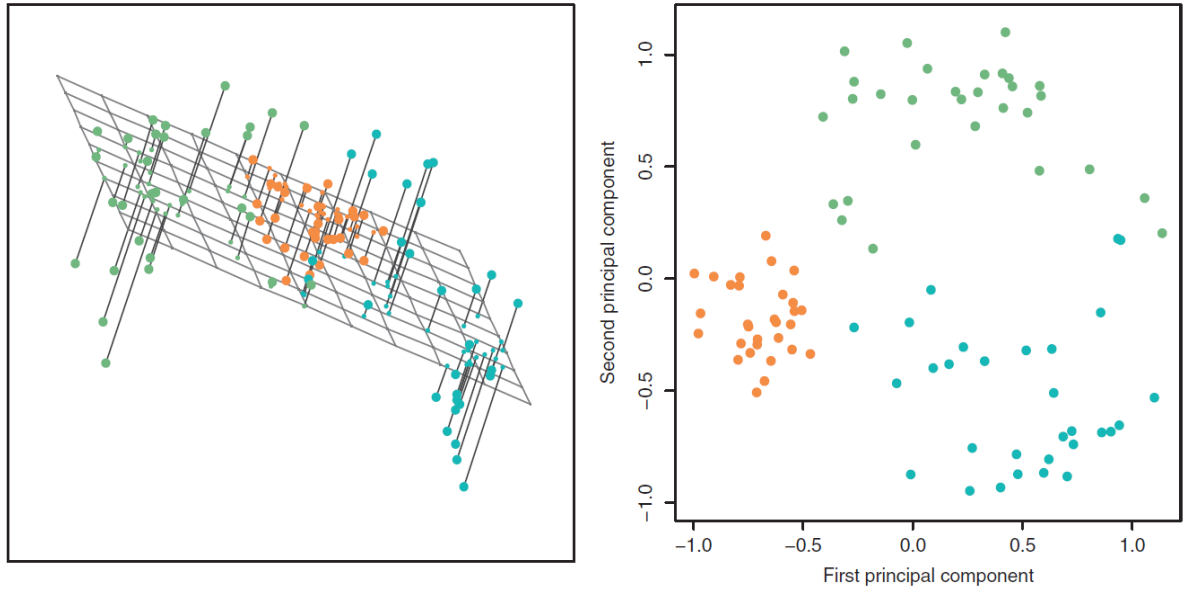
# Plot Trees as a “Phylogenetic Landscape”

- 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

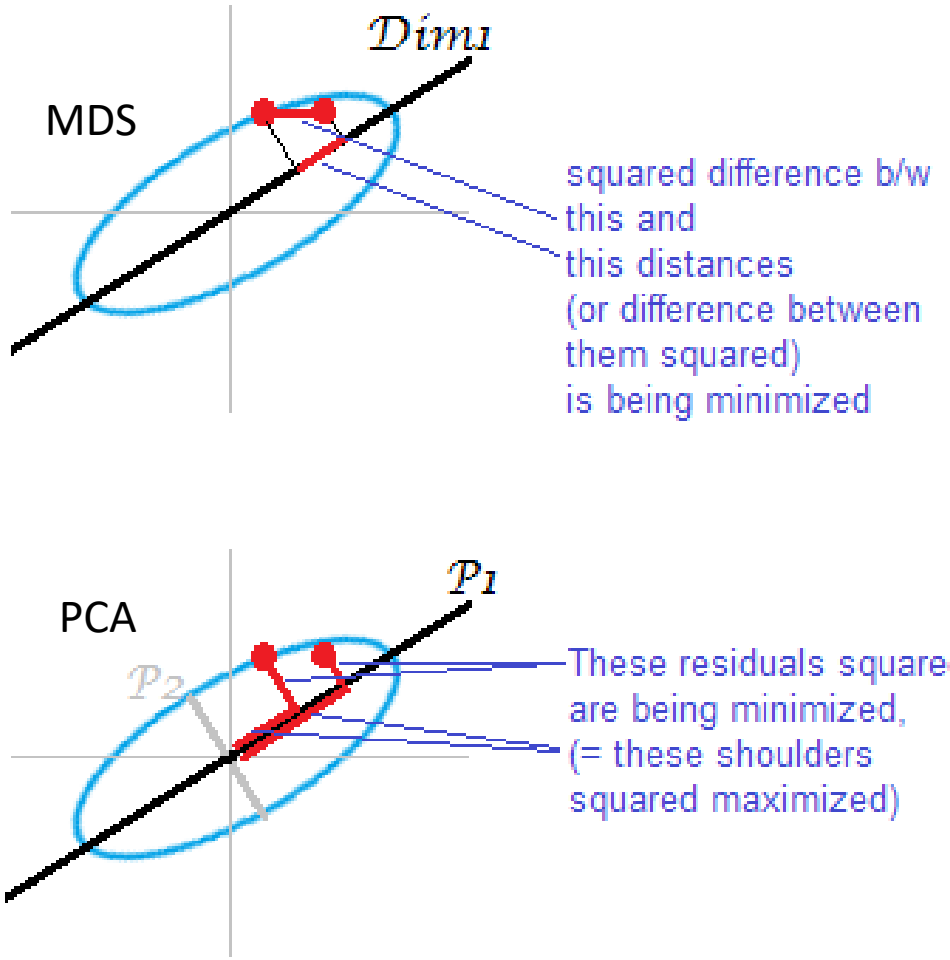



From “An Introduction to Statistical Learning” by Hastie et al.

# Multidimensional Scaling

- Given a n×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:

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



# 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.

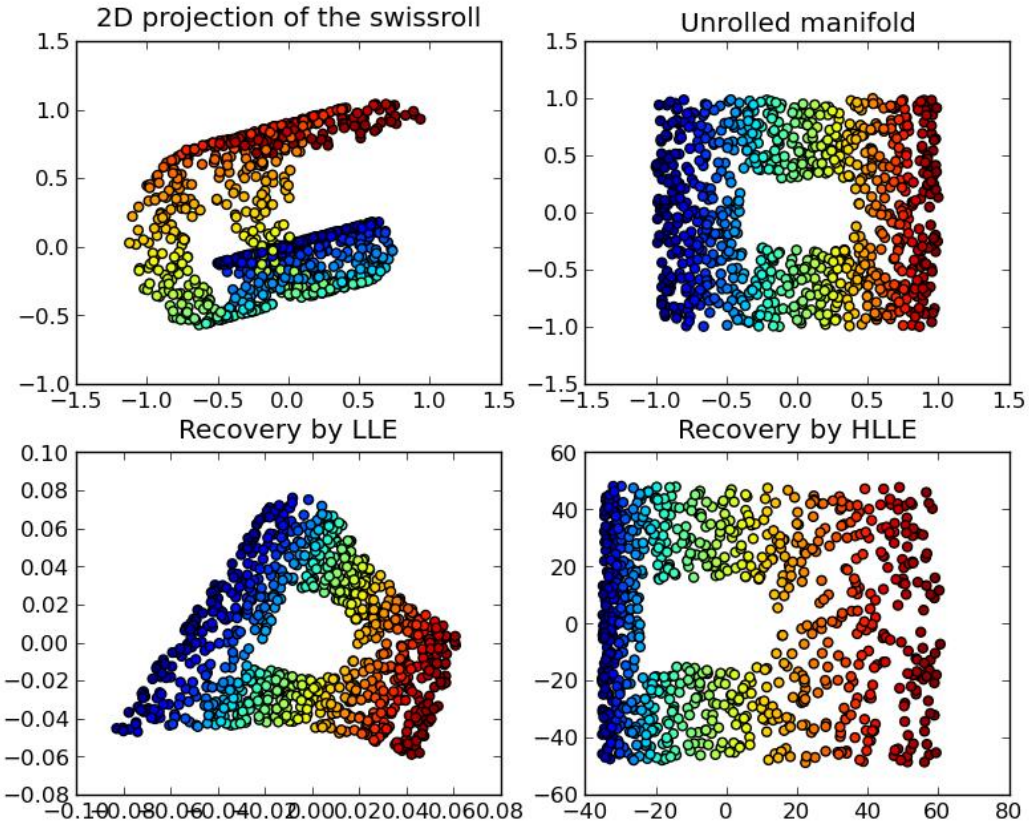



Figure By Olivier Grisel

# TreeScaper NLDR

**Cost Functions**

- Sammon's Nonlinear Mapping

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

*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 3 mtDNA genome alignments (Salamanders, Mammals, and Fishes)
- Evaluate trees with different NLDR Methods

# Projections in 2D

## Bootstrap Trees from 15 mtDNA Genes

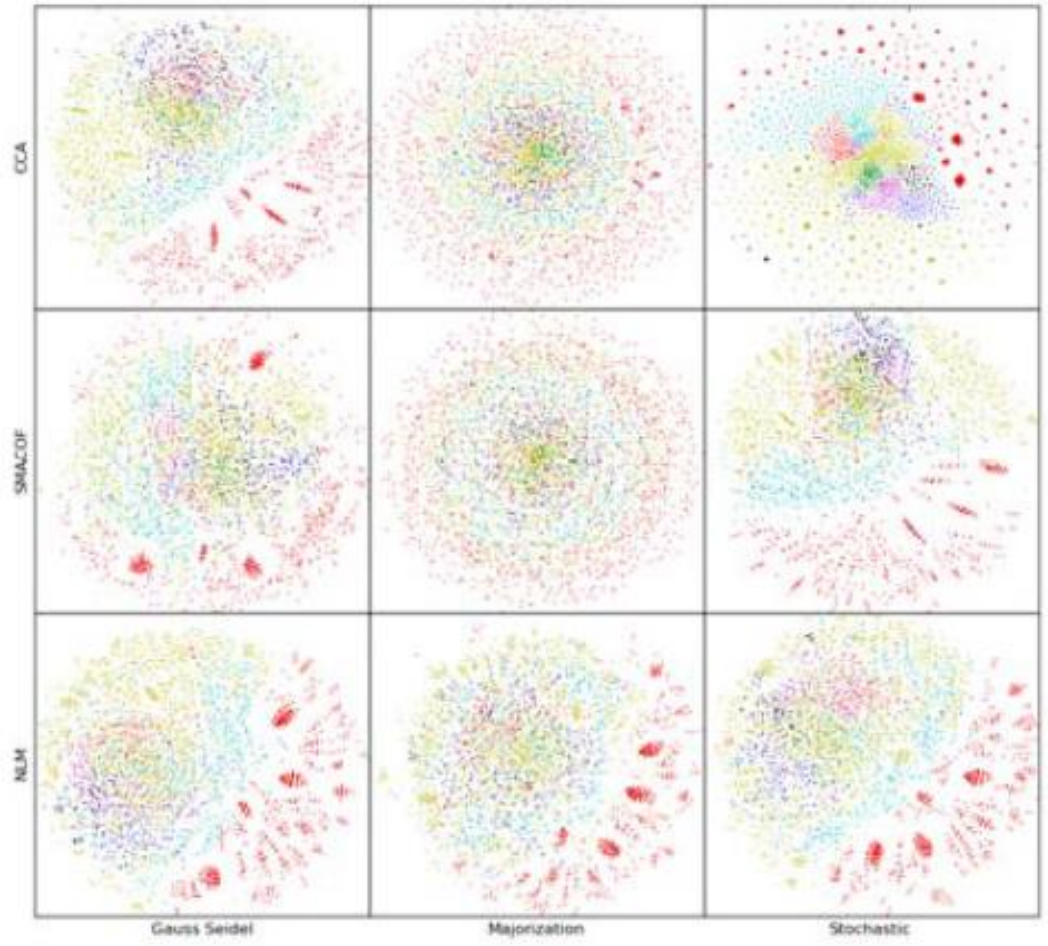
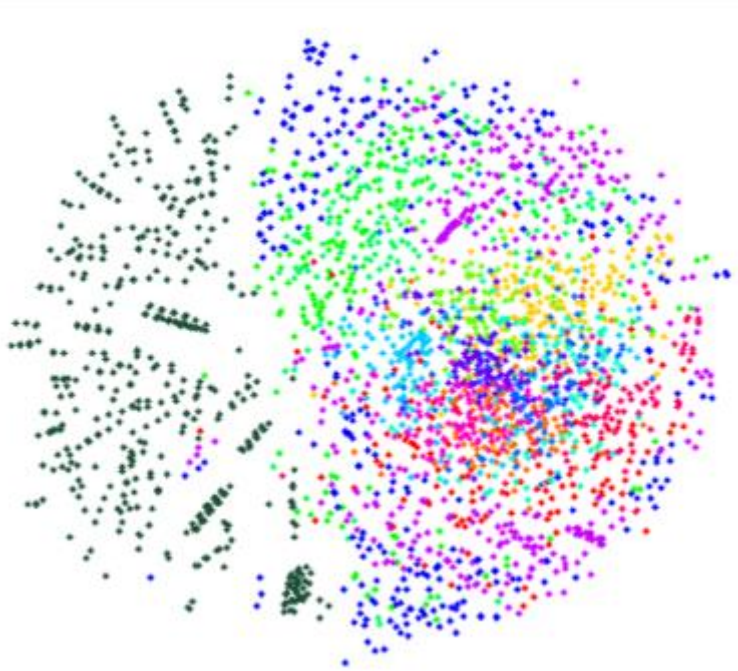



Figure by Wen Huang

# Projections in 2D

## Bootstrap Trees from 15 mtDNA Genes

**Kruskal-1 & Linear Iteration**



**Colors = trees from different genes**

**CCA & Stochastic Gradient**

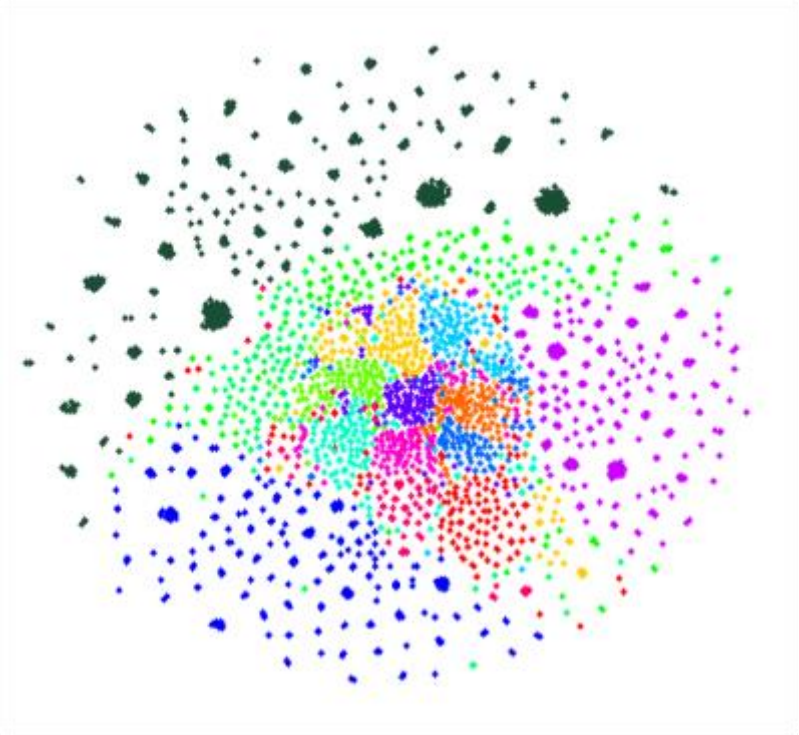
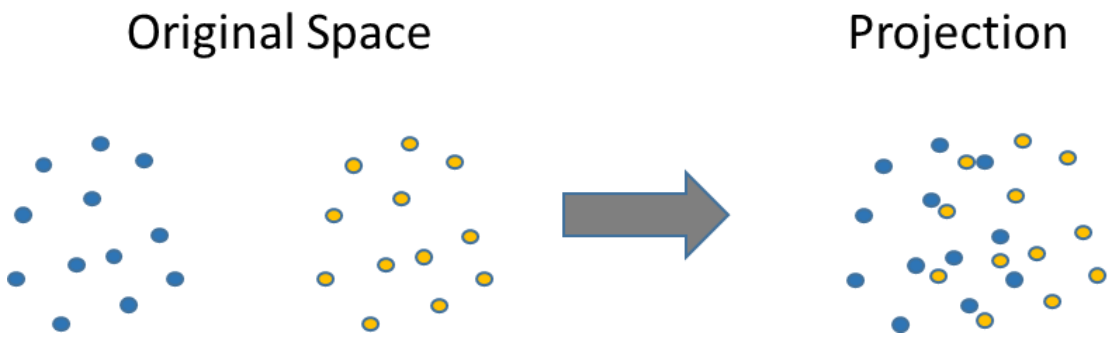


Figure by Wen Huang

# Measures of NLDR Quality

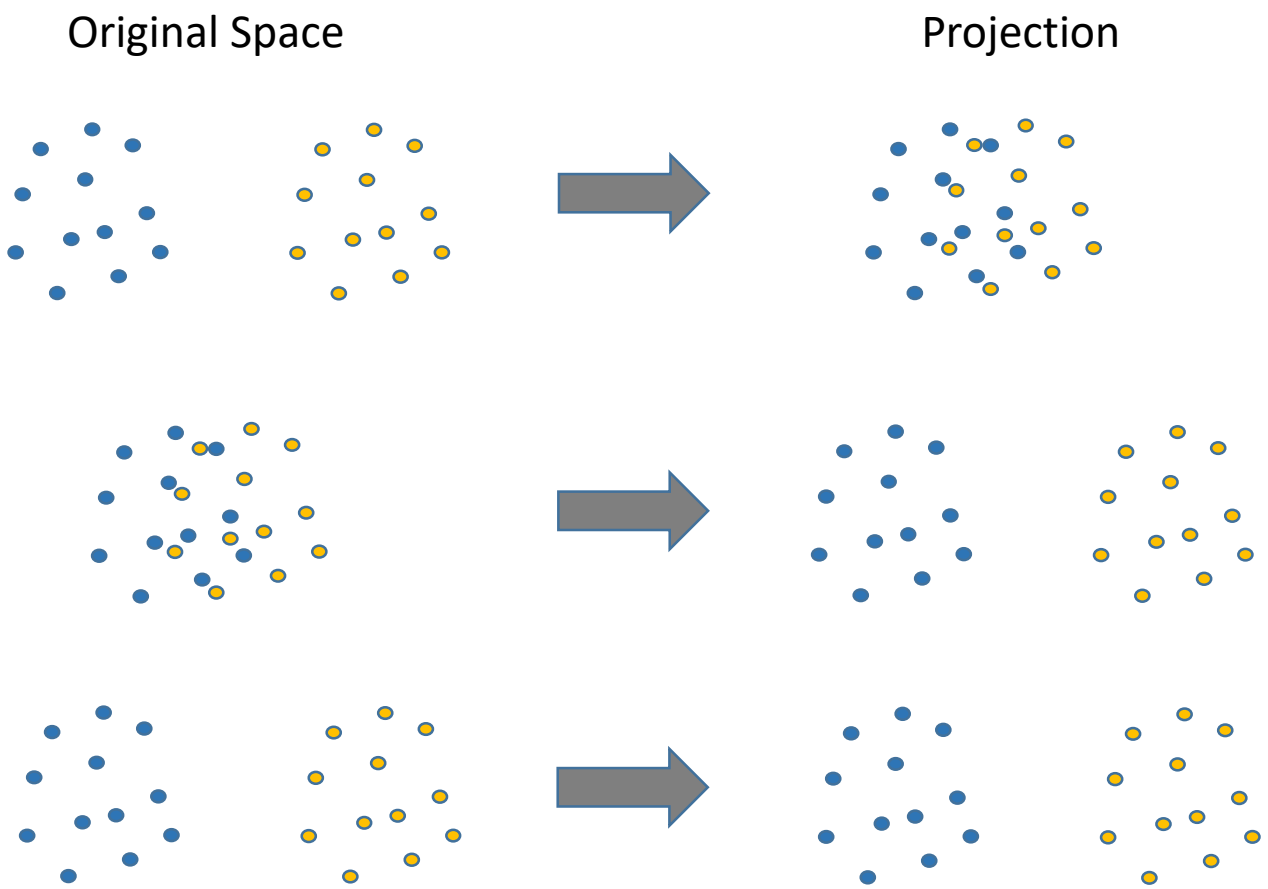



$$Tru = 1 - \frac{2}{nk(2n-3k-1)} \sum_{i=1}^{n} \sum_{j \in U_k(i)} (r(i,j) - k).$$

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

- *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




- High Trustworthiness
- Low Continuity

- Low Trustworthiness
- High Continuity

- High Trustworthiness
- High Continuity

# Measures of NLDR Quality

**CCA & Stochastic Gradient**

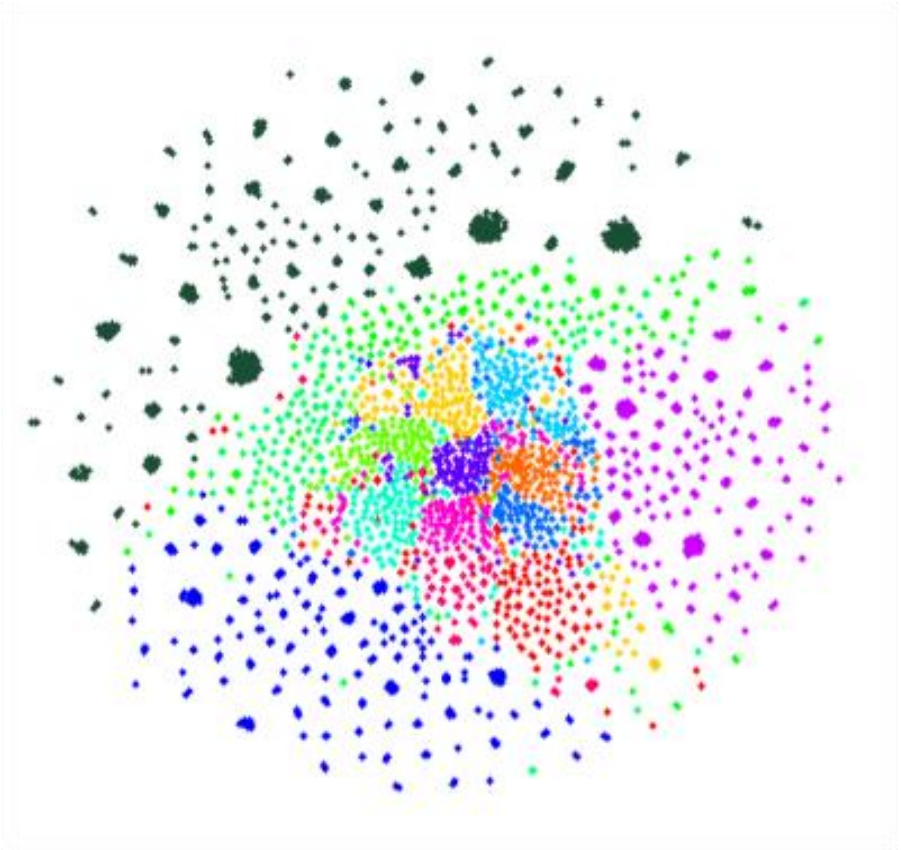
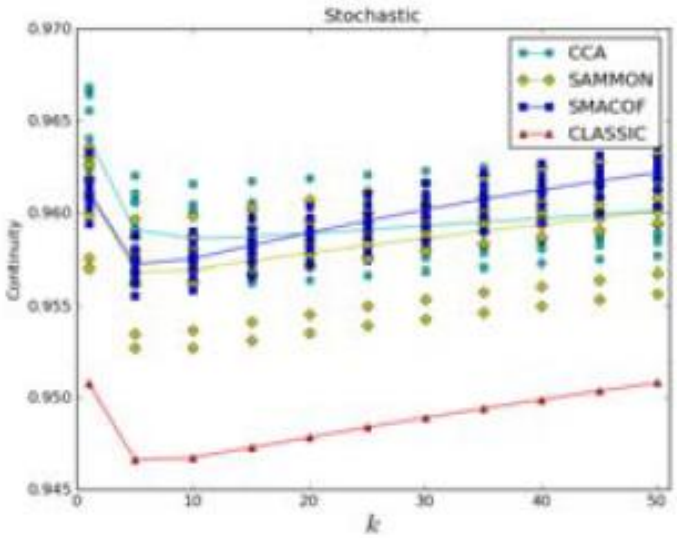
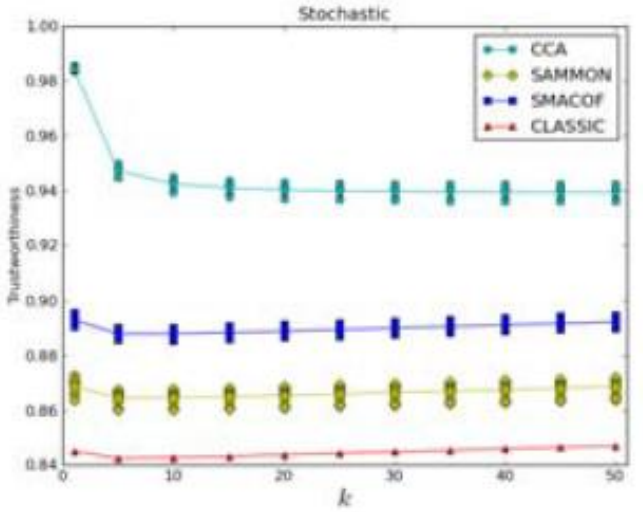
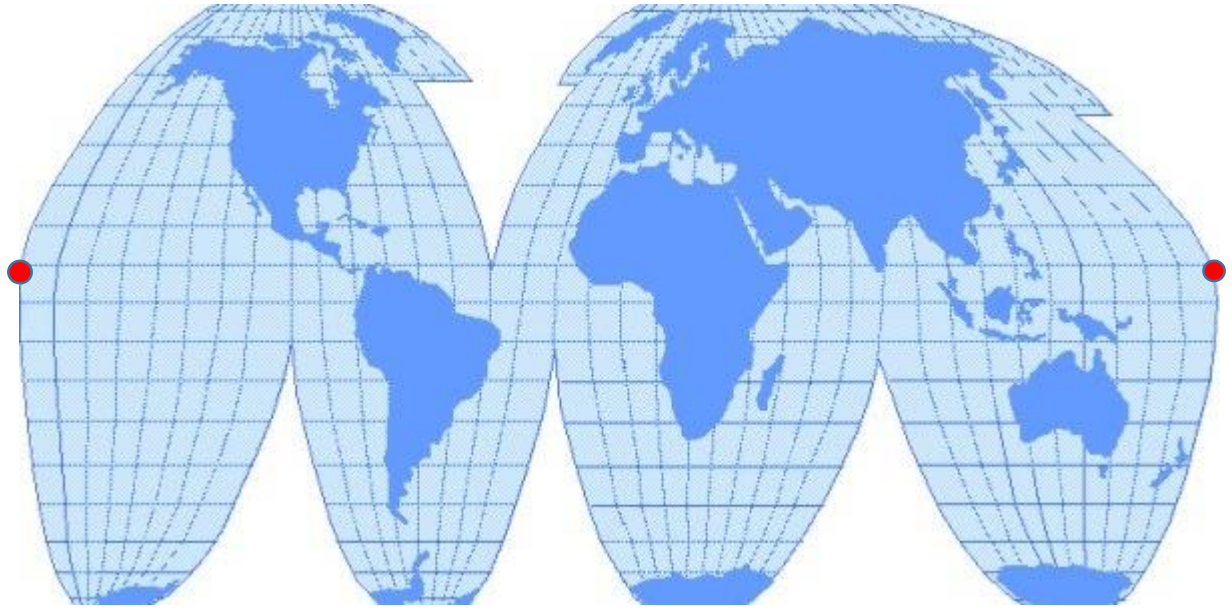


Figure by Wen Huang



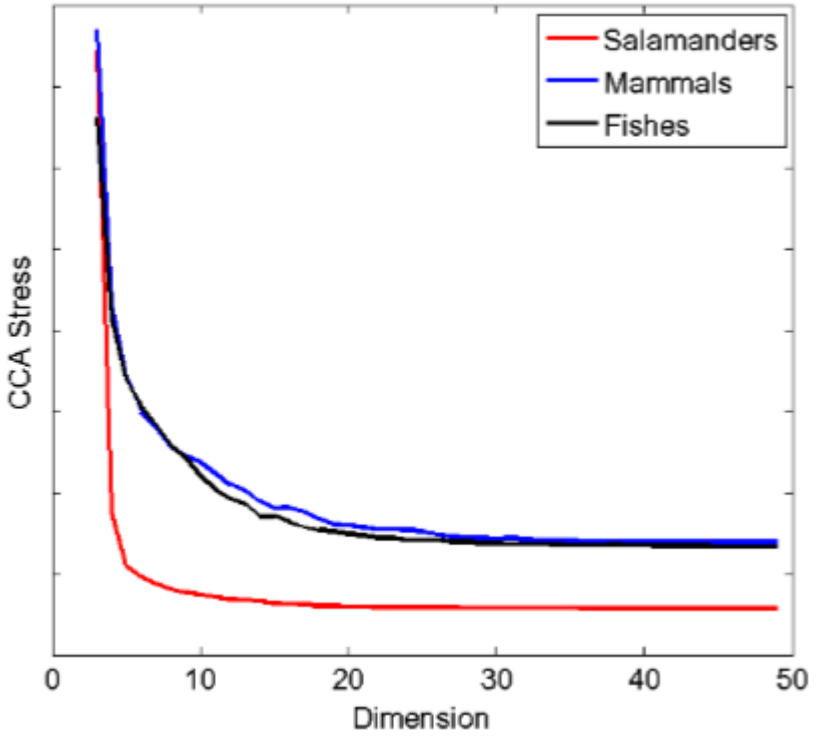
# Discontinuity in NLDR: Flattening of a Globe



# Is 3D Needed?

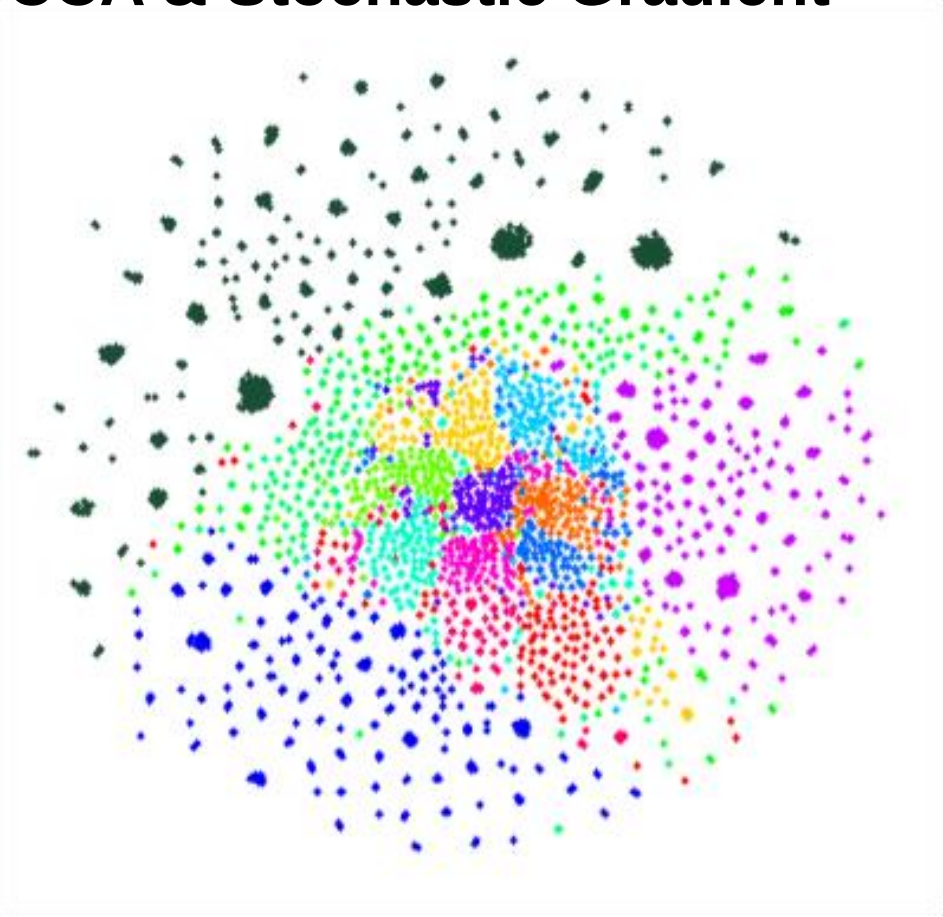
# **YES**

- 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

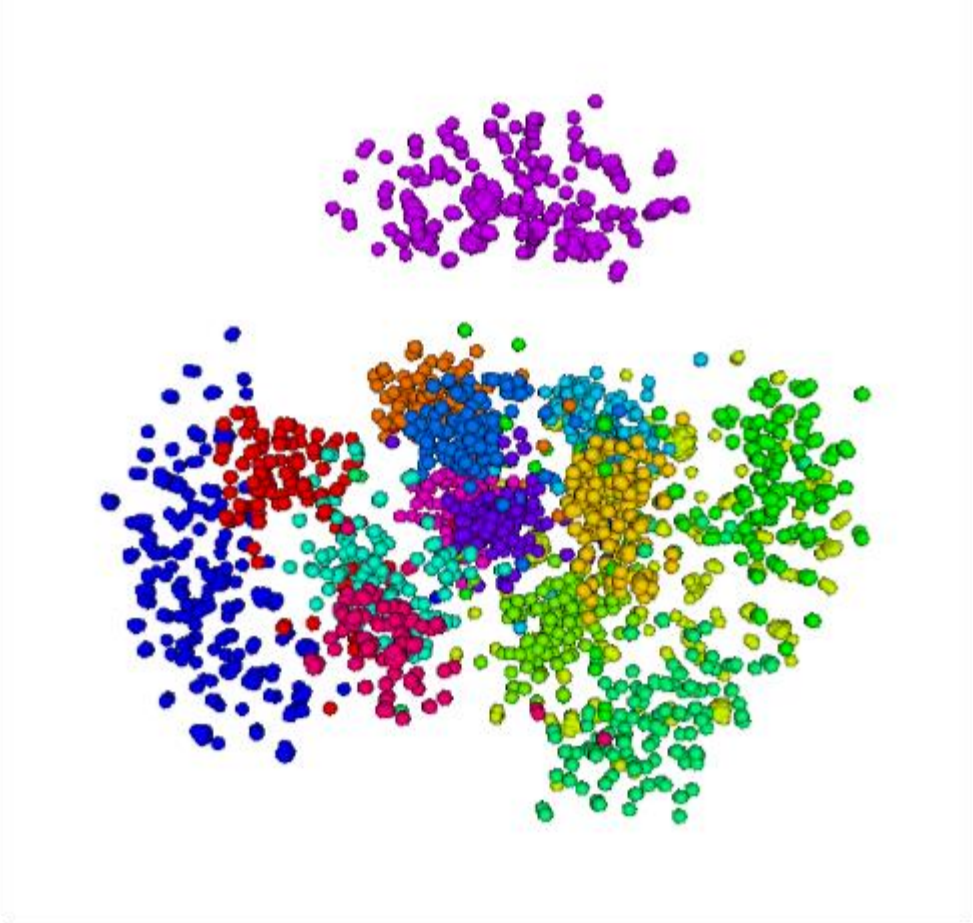
**CCA & Stochastic Gradient**



| Dimensionality Test | Unfiltered |
|---|---|
| NN | 6.91808 |
| COR | 4.53759 |
| ML | 17.2628 |

# Projection in 3D

## Bootstrap Trees from 15 mtDNA Genes

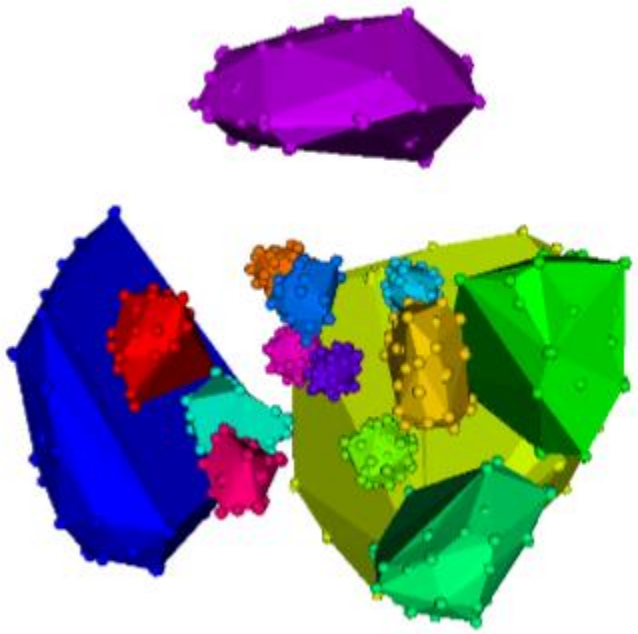


**Colors = trees from different genes**

# Projections in 3D

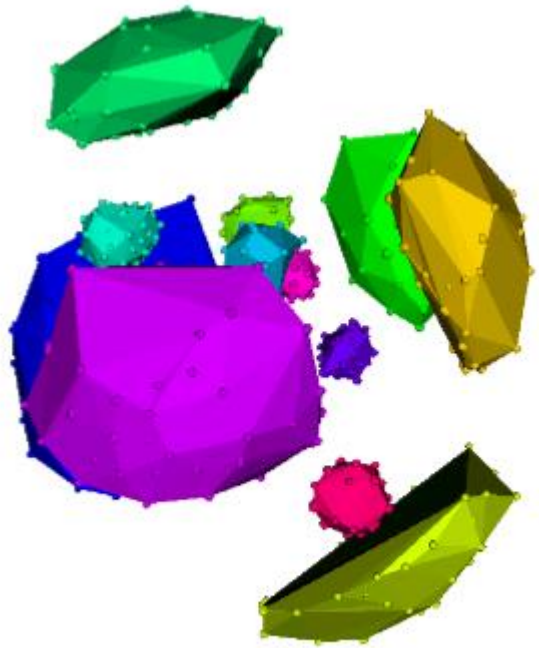
## Bootstrap Trees from 15 mtDNA Genes

**Salamanders**



**Colors = trees from different genes**

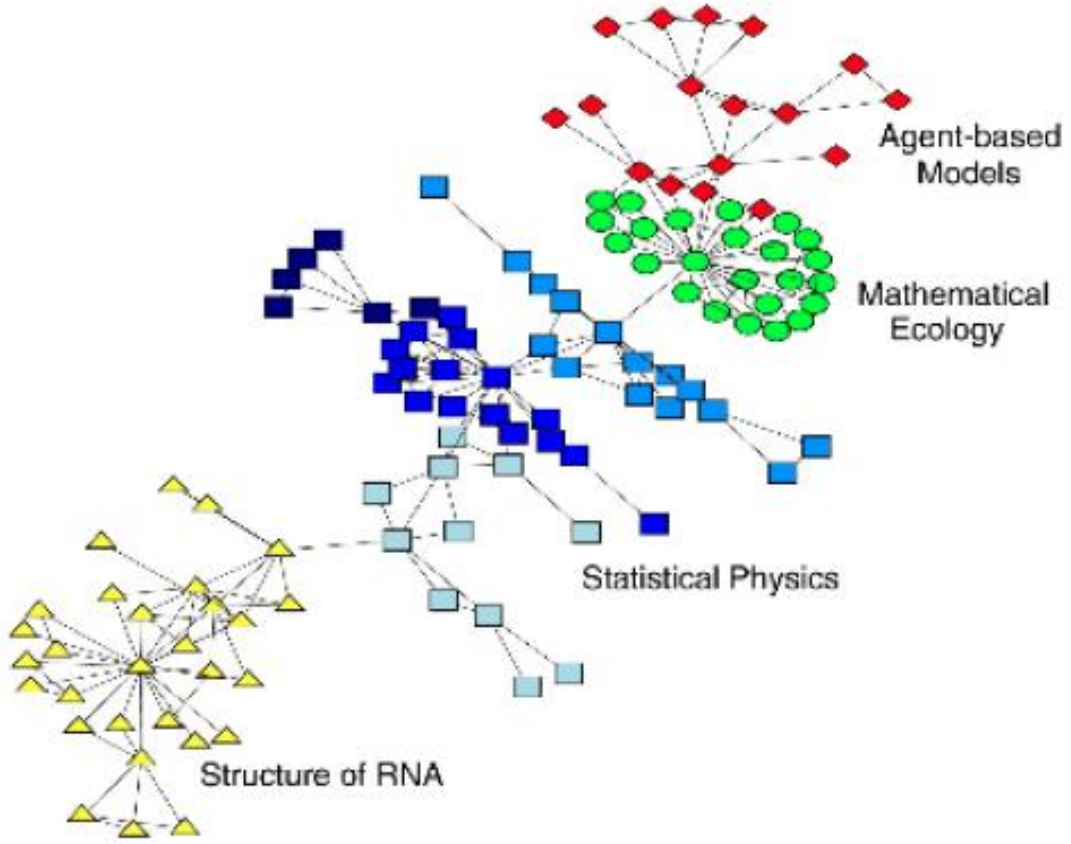
**Mammals**



# Part 2

## Network Community Detection Methods in TreeScaper

# Networks

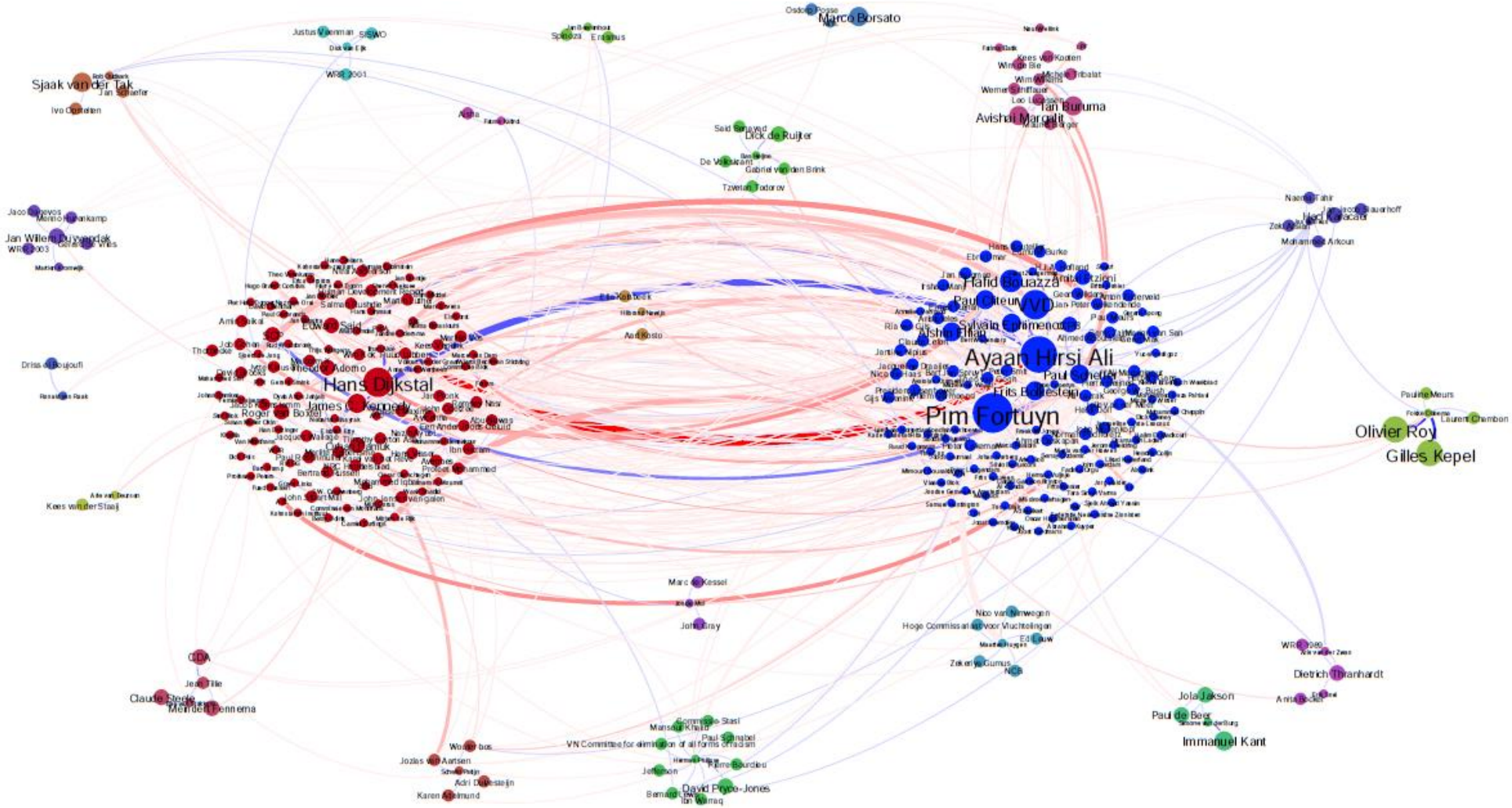



Collaboration network between scientists working in Santa Fe Institute

# Biological Networks

Yeast Protein Interaction Network

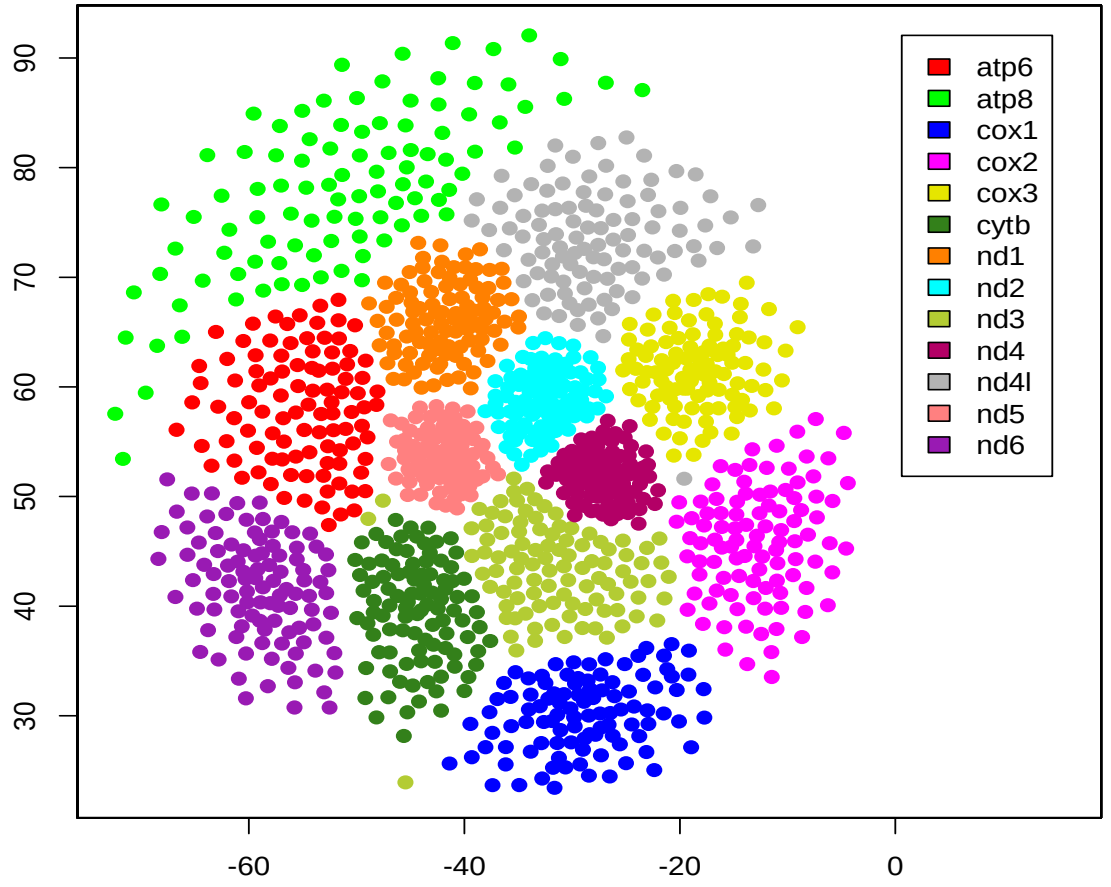
# Social Networks: Community Detection




Author Networks (by Vincent Traag)

# Summarizing Tree Sets

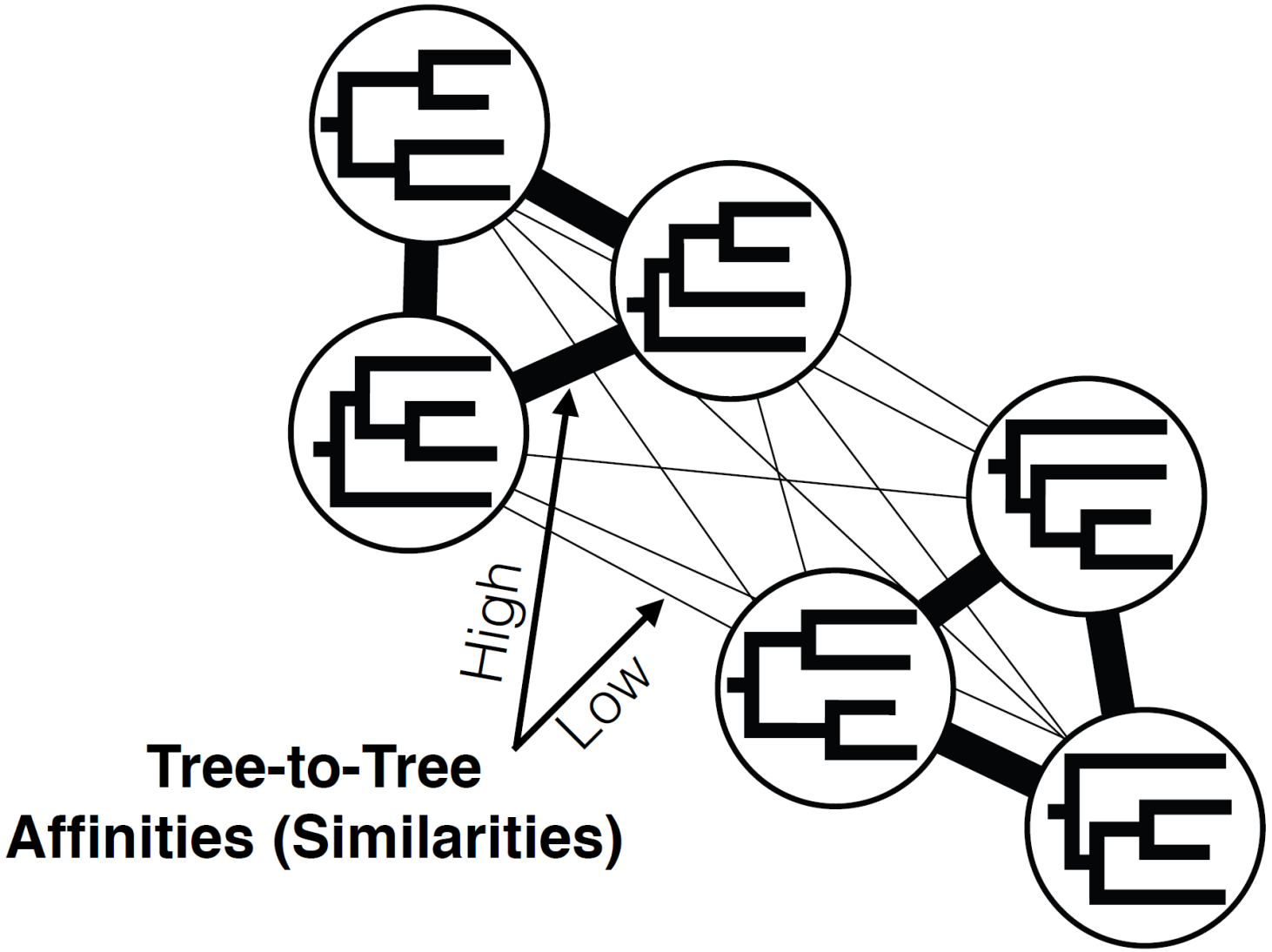
- Dimensionality Reduction
  - No objective way to discern distinct groups of trees



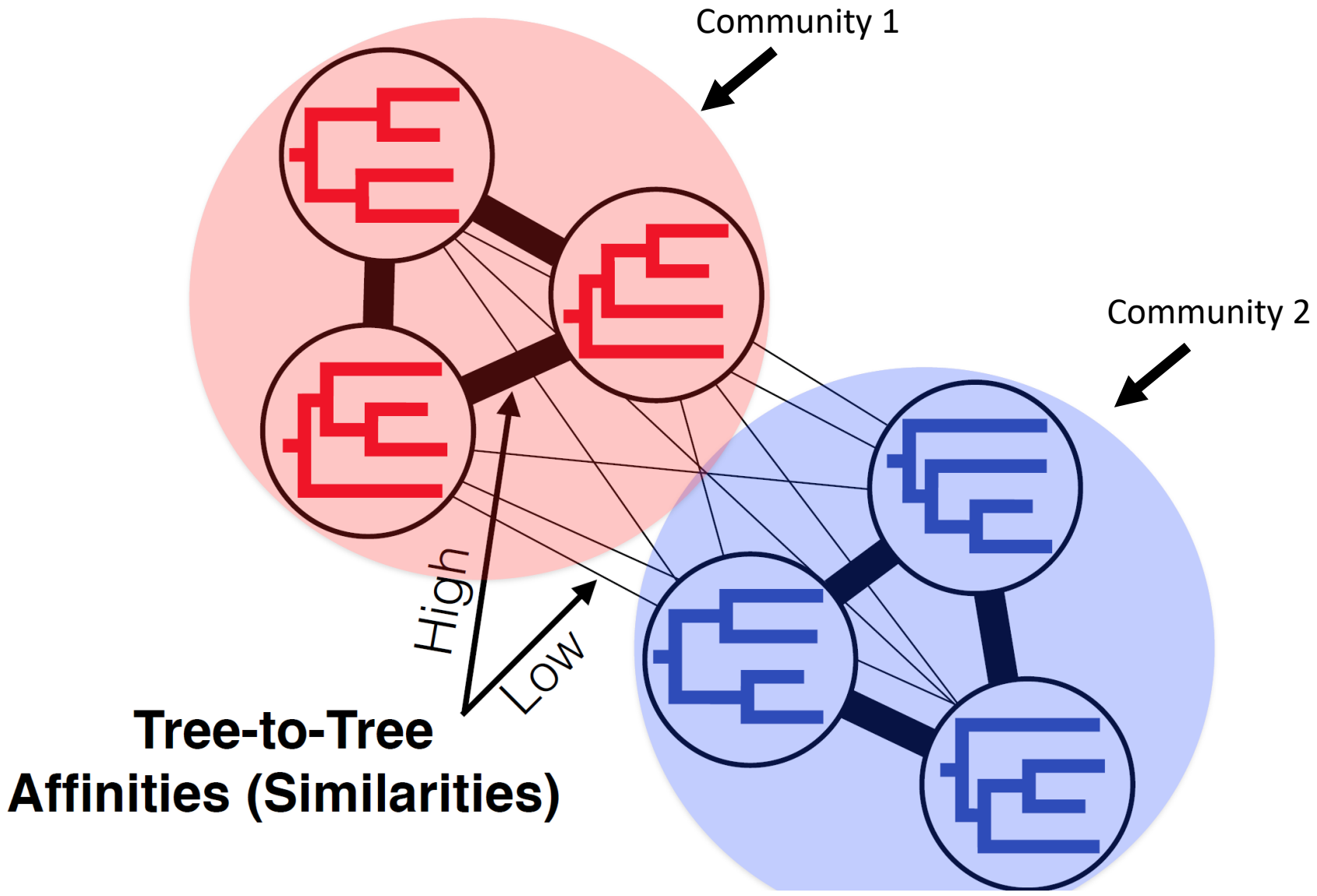
# A New Approach: Phylogenetic Networks

- 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

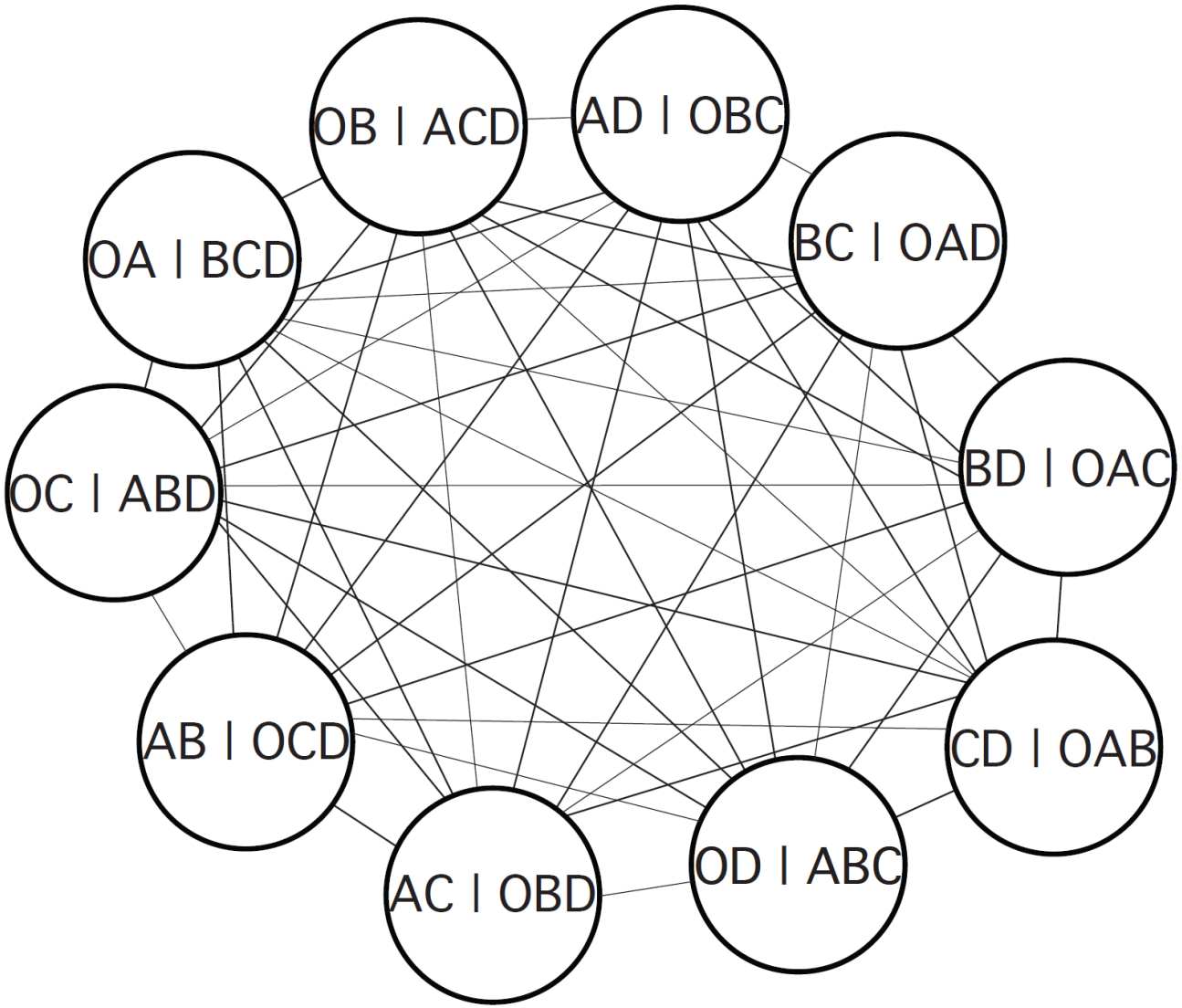
# Networks of Trees



# Networks of Trees: Community Detection



# Networks of Bipartitions



# 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 $p_1$ be the probability that $a$ is in a tree topology. We take this to be the fraction of the trees in the tree set that do not contain $a$.

Let $p_2$ be the probability that $a$ is not in a tree topology. We take this to be the fraction of the trees in the tree set that contain $a$.

# Networks of Bipartitions: Covariances

Recall the formula for the expectation of a random variable: $E[X] = x_1p_1 + x_2p_2 + ... + x_np_n$. Therefore:

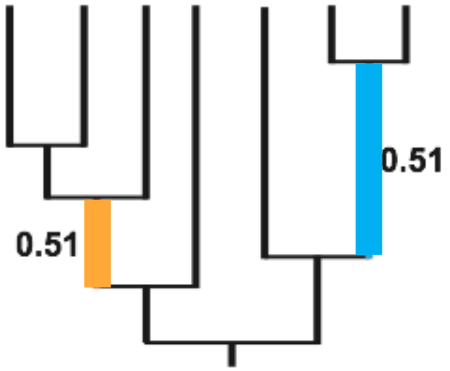
$$\begin{aligned} E[X] &= (0)p_1 + (1)p_2 \\ &= p_2 \end{aligned}$$

Let $Y$ be a variable that encodes the presence/absence of bipartition $b$ in the tree set. Similarly, $E[Y]$ is the fraction of trees containing biparititon $b$, and E[XY] is the fraction of trees that contain $a$ and $b$.

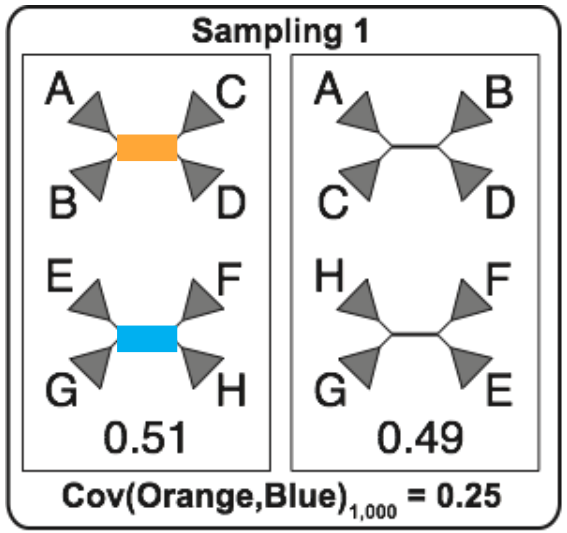
Thus, using the formula for covariance:

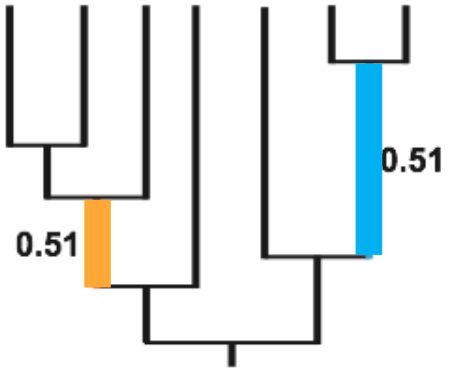
$$cov(X, Y) = E[XY] - E[X]E[Y]$$

# Bipartition Covariances

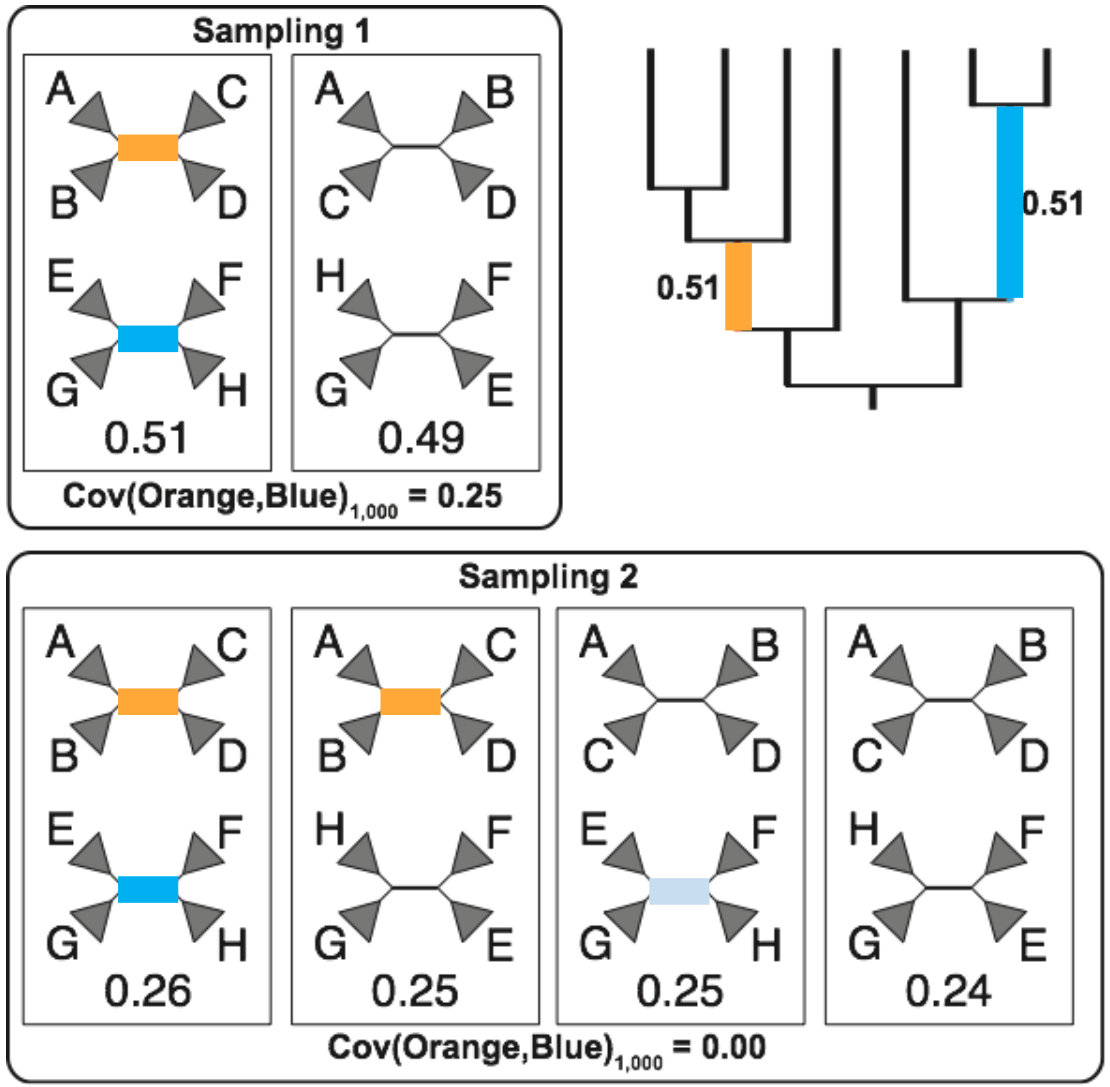


# Bipartition Covariances

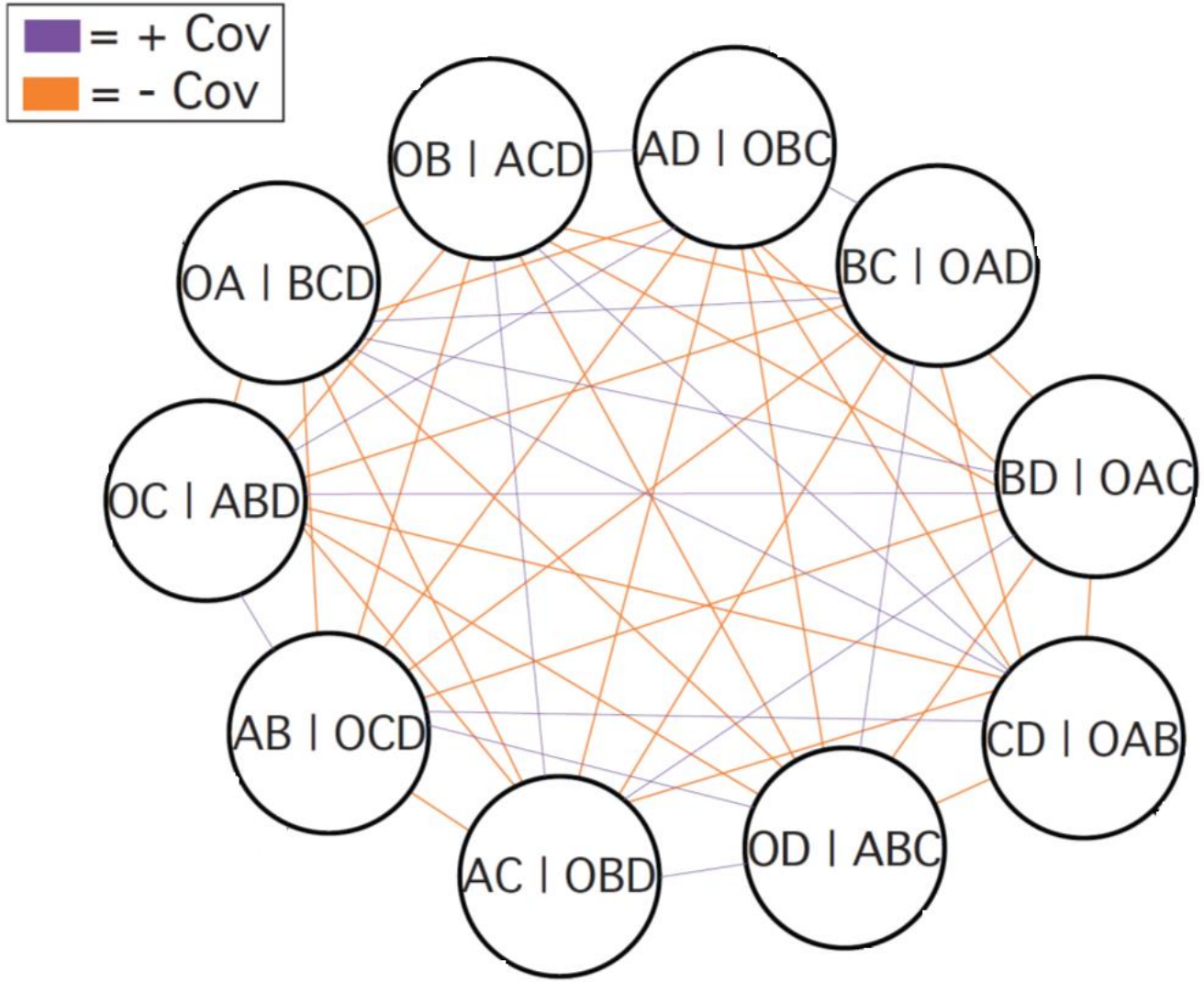


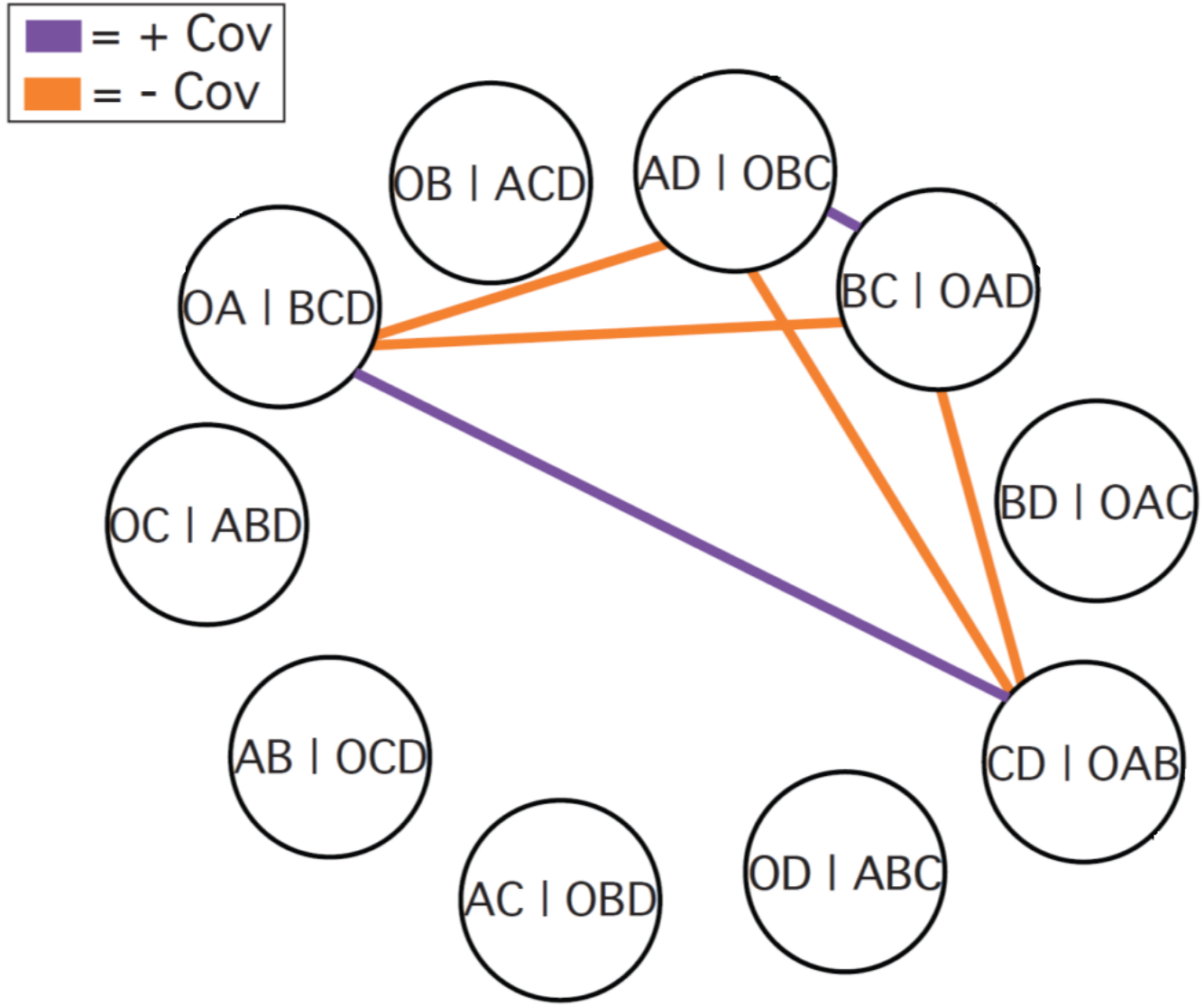
# Bipartition Covariances



# Networks of Bipartitions: Covariance Edge Weights

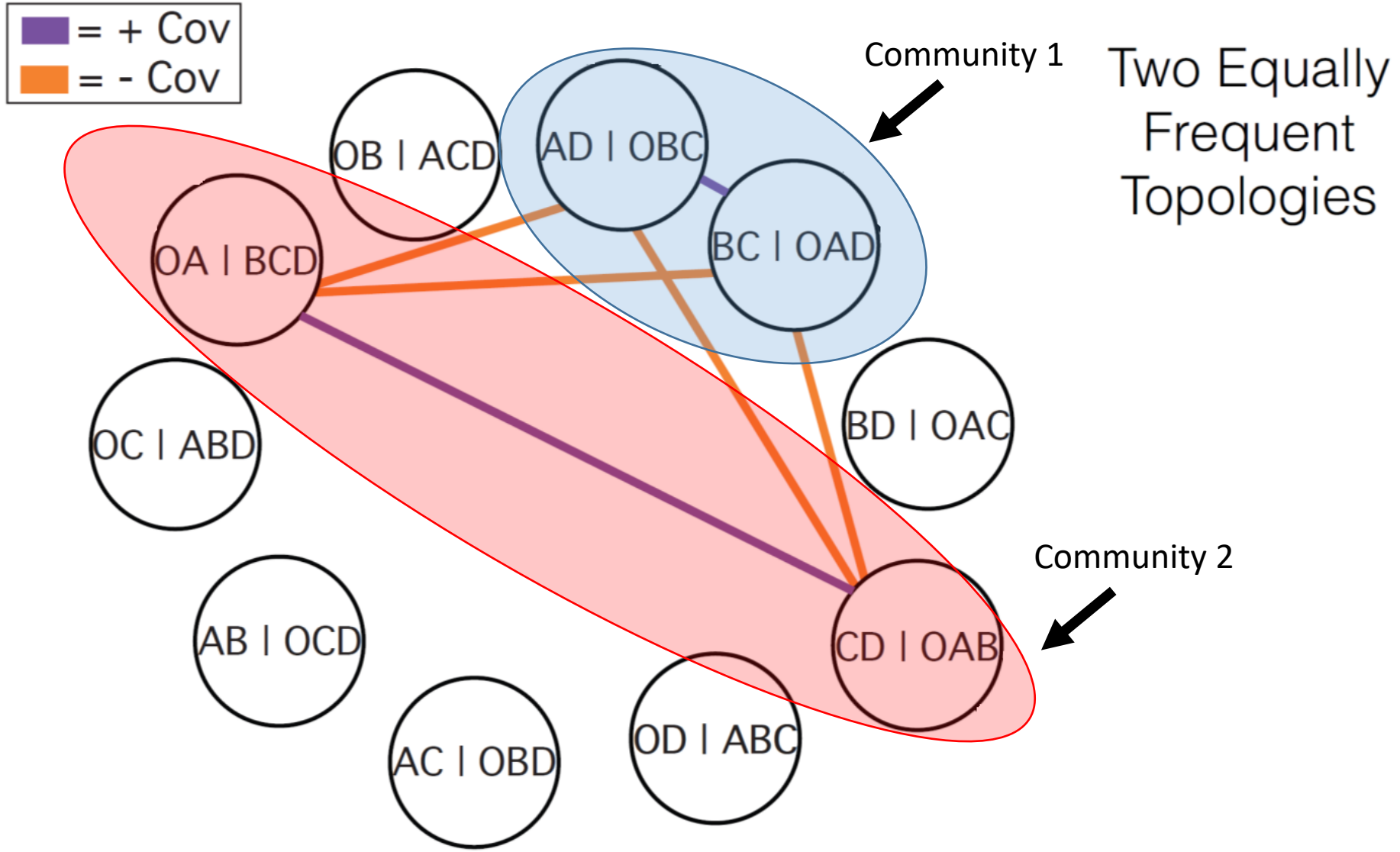


# Networks of Bipartitions: Covariance Edge Weights




Two Equally Frequent Topologies

# Networks of Bipartitions: Community Detection



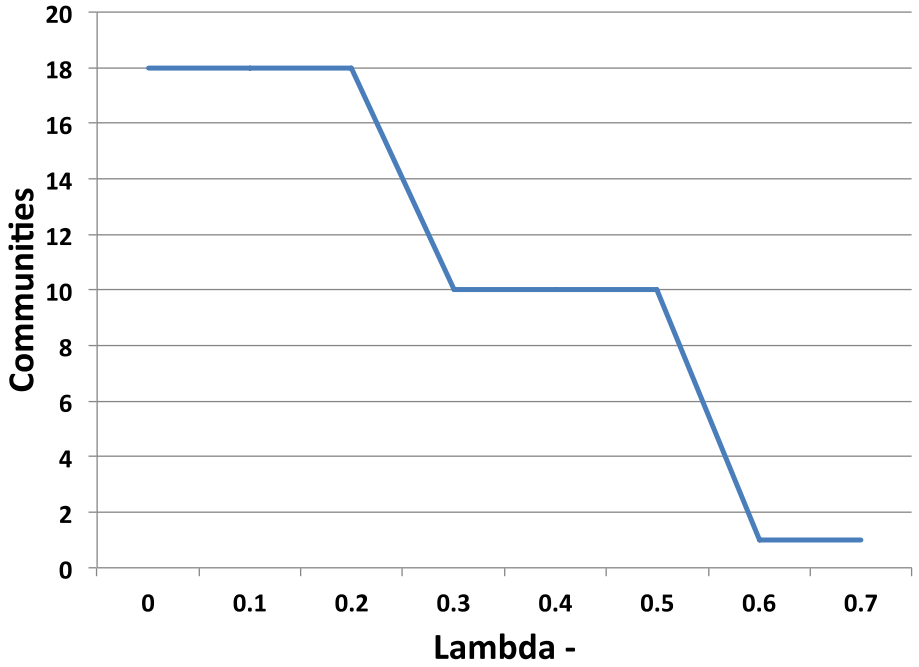
# Method for Community Detection

- Constant-Potts Model(CPM):

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

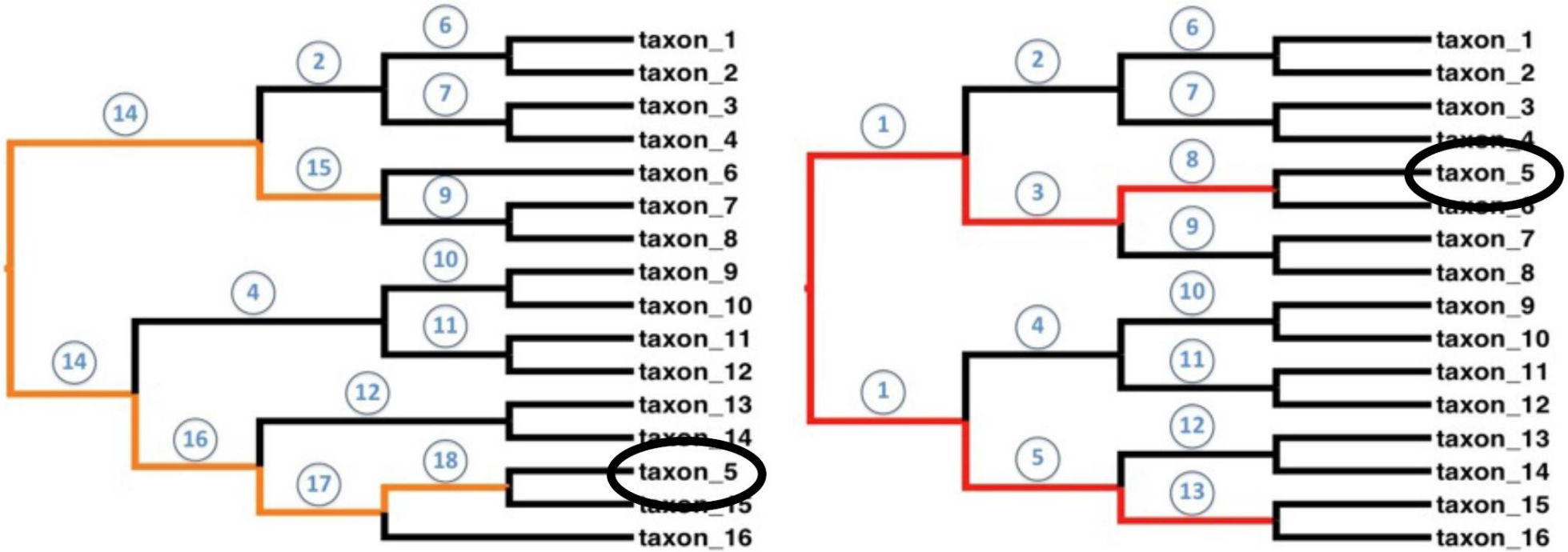
- $A_{ij}$ is the edge weight between node i and node j
- $\sigma_i$ is the community bipartition i belongs to
- $\delta(\sigma_i, \sigma_j)$ is zero unless $\sigma_i = \sigma_j$
- $c^2$ is community size squared
- Lambda values are tuning parameters
  - Only the difference matters in this case
- Hamilitonian,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

# Method for Community Detection



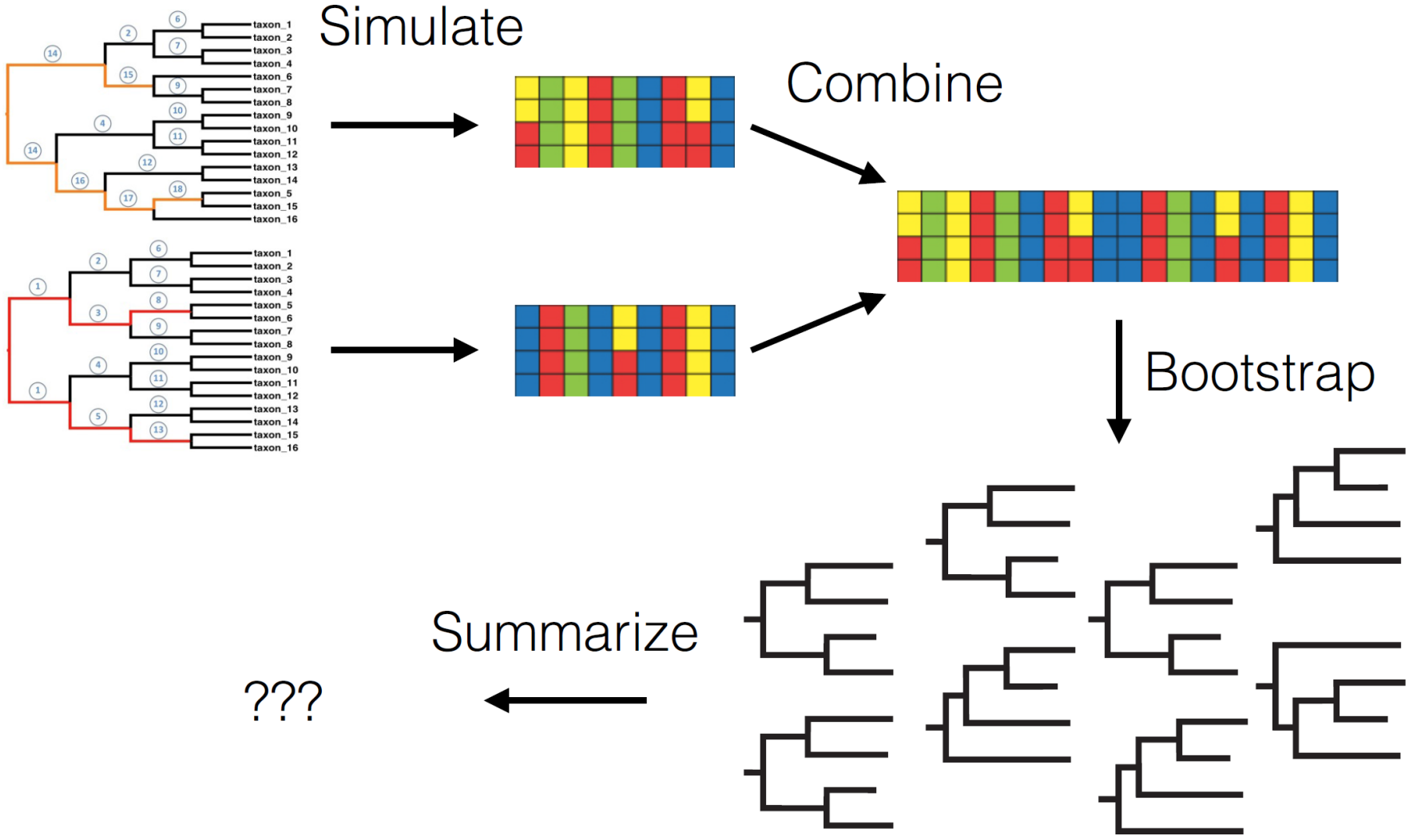

- $\lambda^+$ is fixed at some value and $\lambda^-$ is varied
- The largest range of lambda values that produces a single community structure (and that is between the two extremes) is the most stable community structure in our dataset

# Example: Simulation Experiment

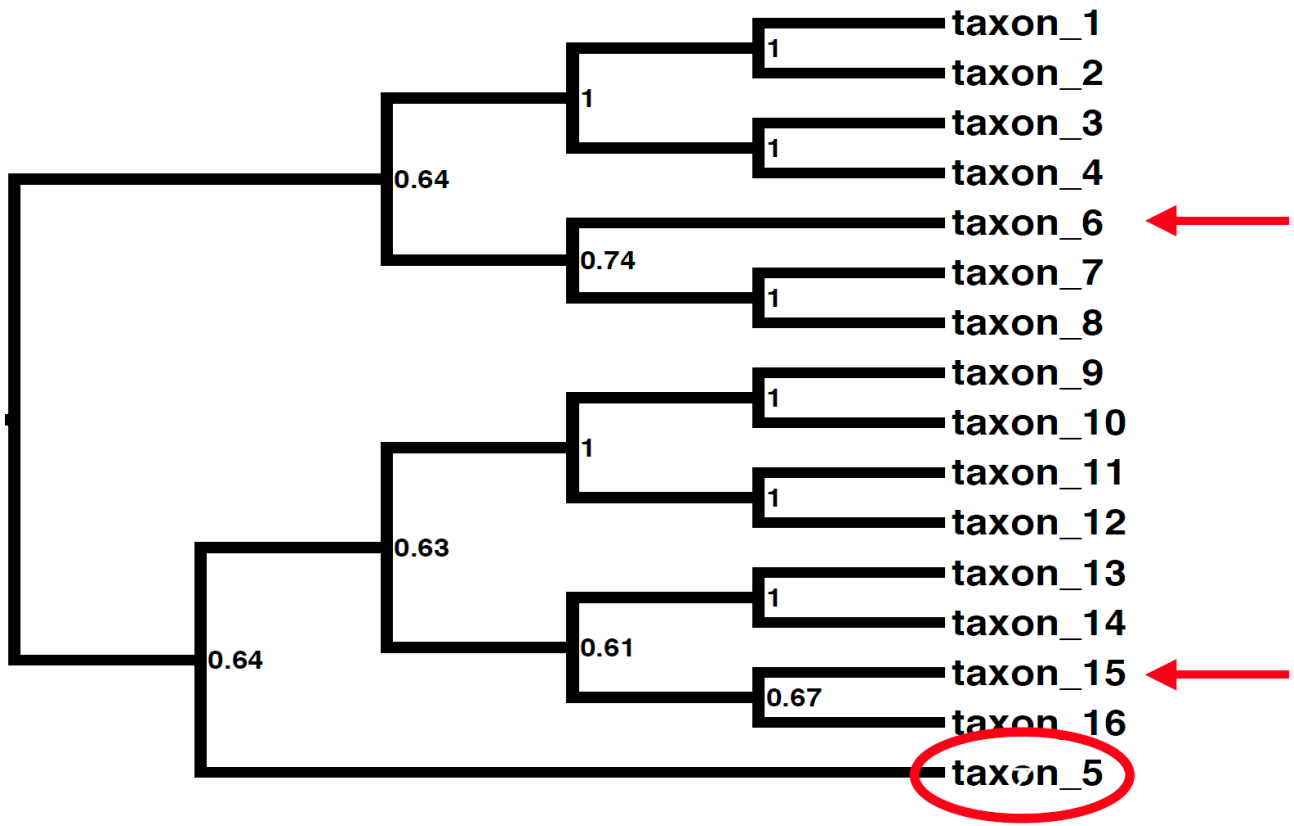



Topologies used for simulating two halves of an alignment.

# Example: Simulation Experiment

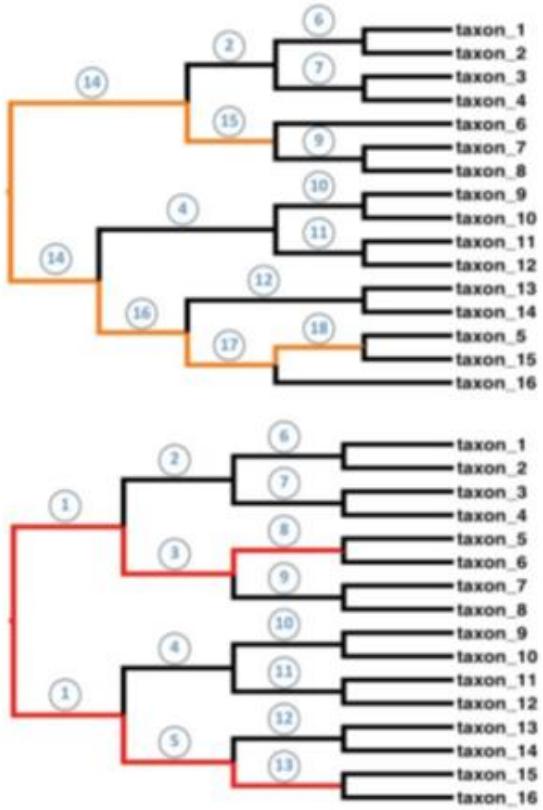


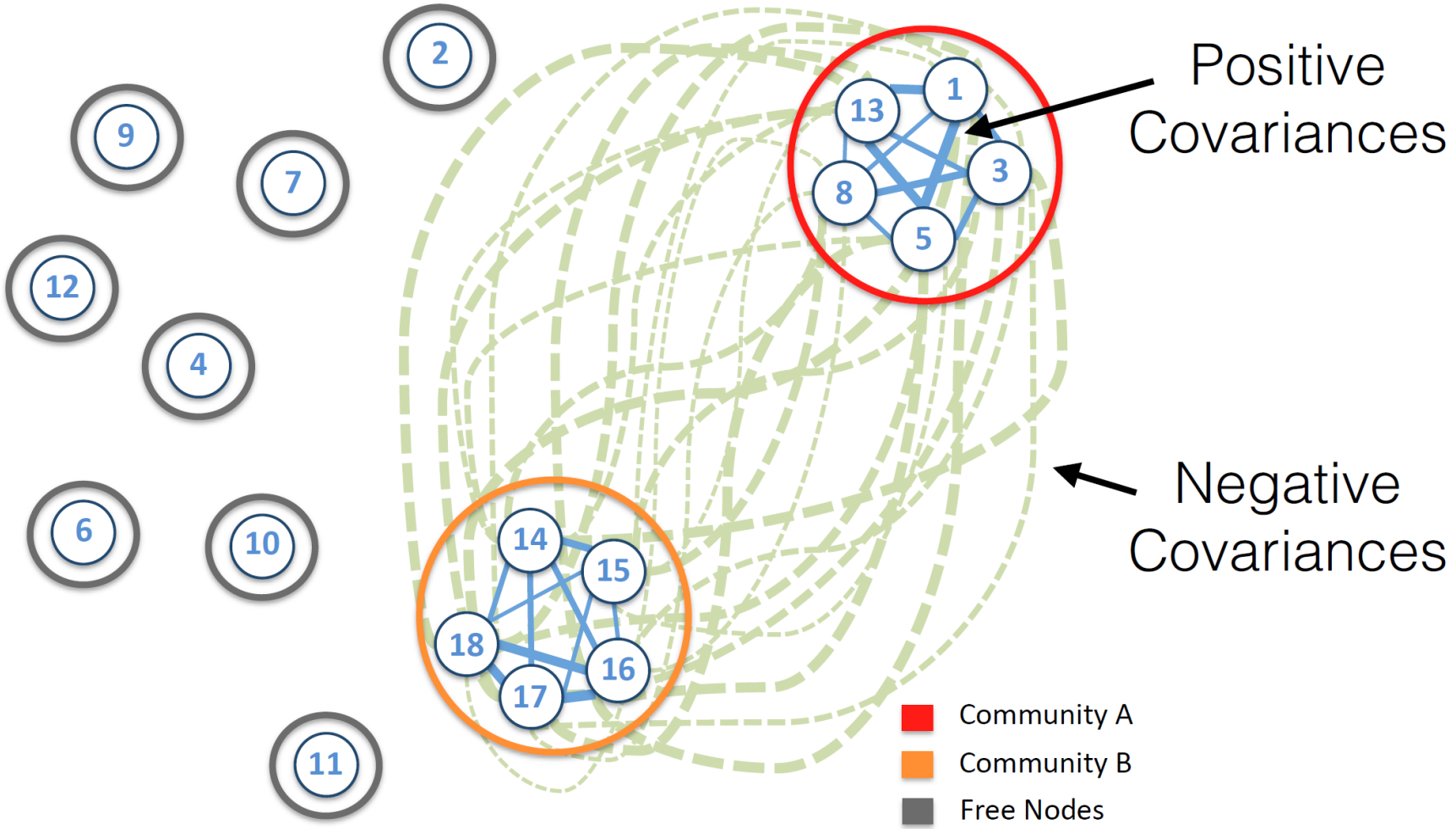
# Consensus Tree Contains Misleading Information



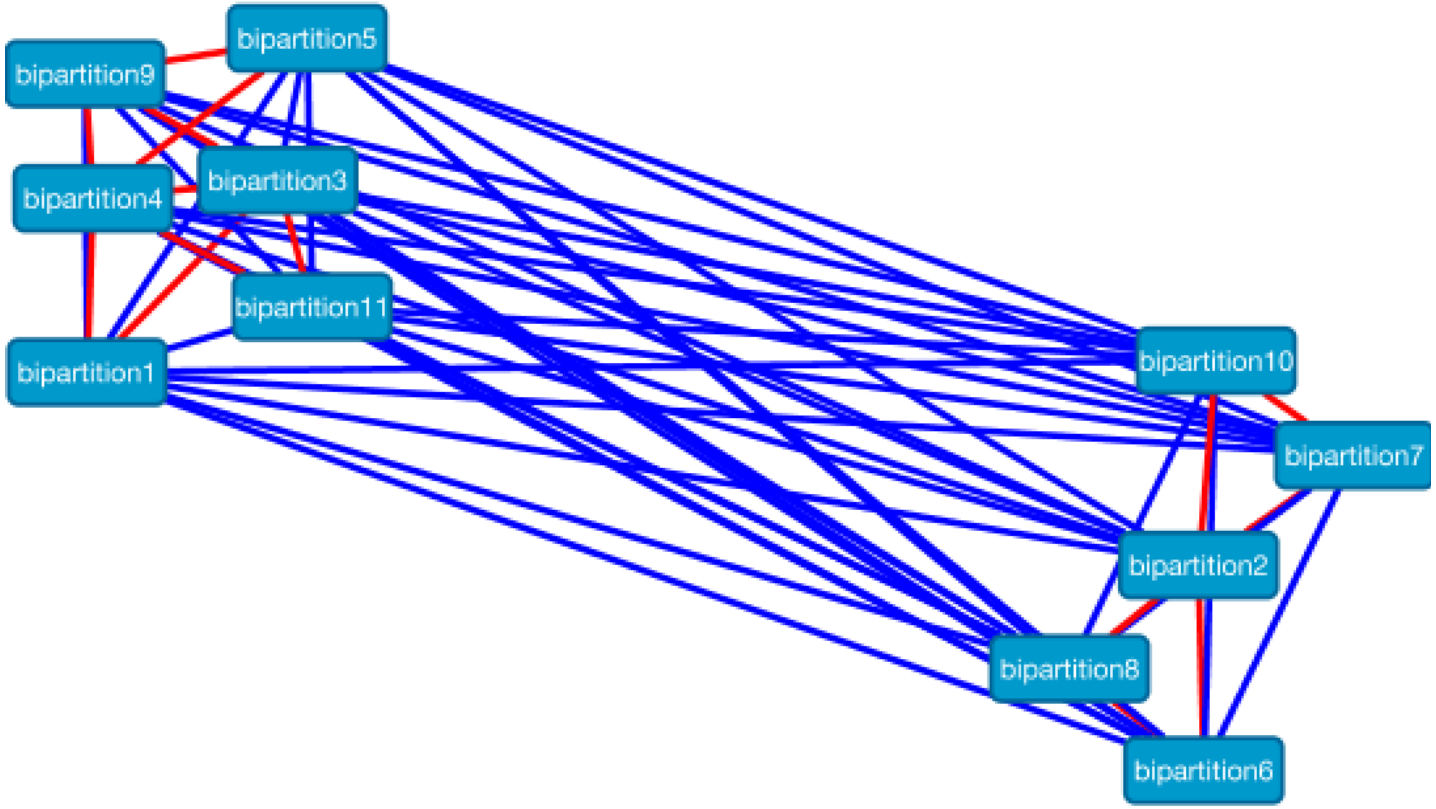

Majority-Rule Consensus Tree

# Network of Bipartitions

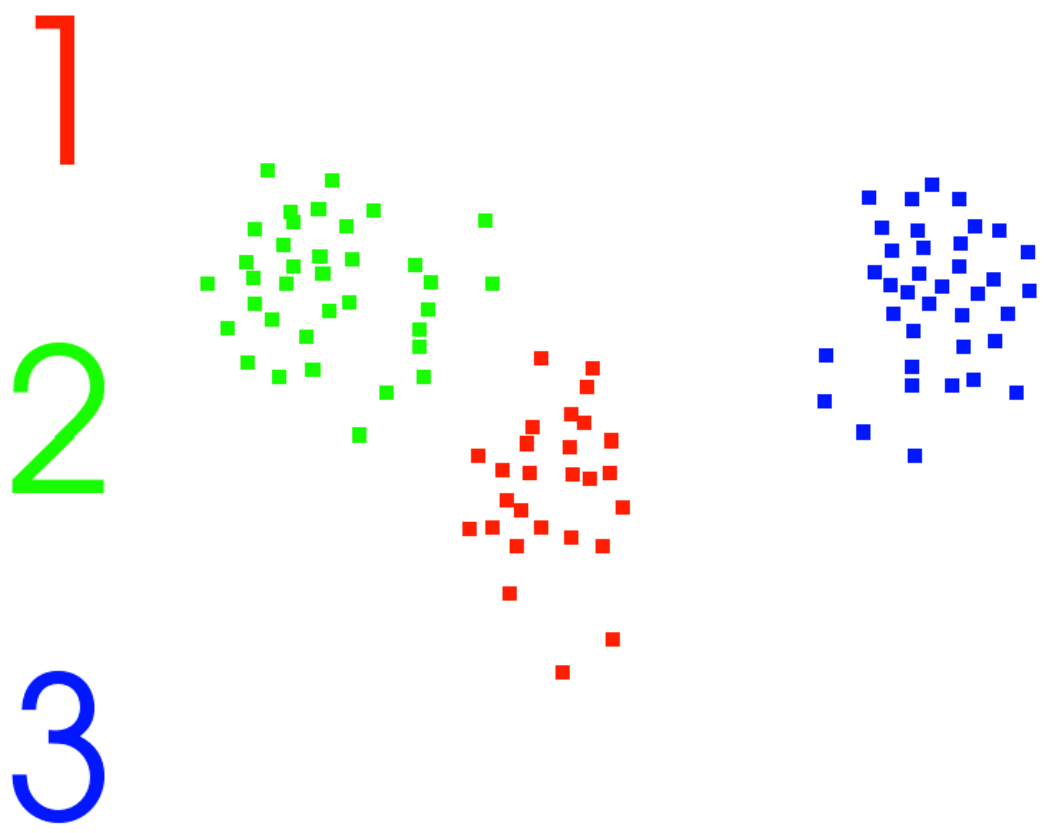


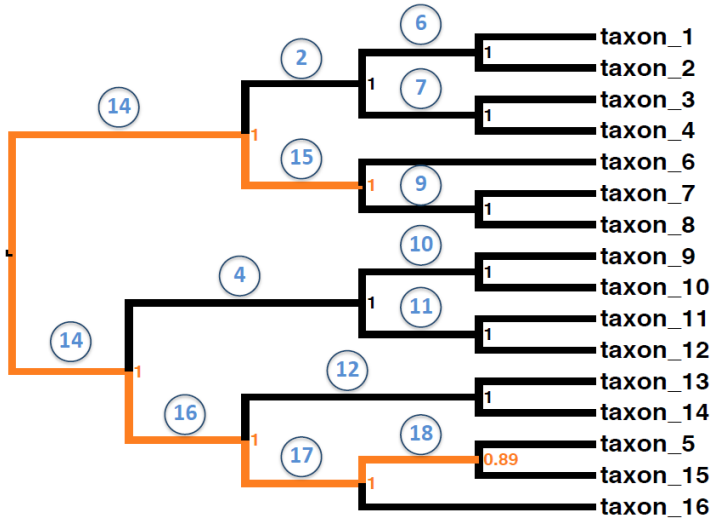
# Networks of Bipartitions (in Cytoscape!)

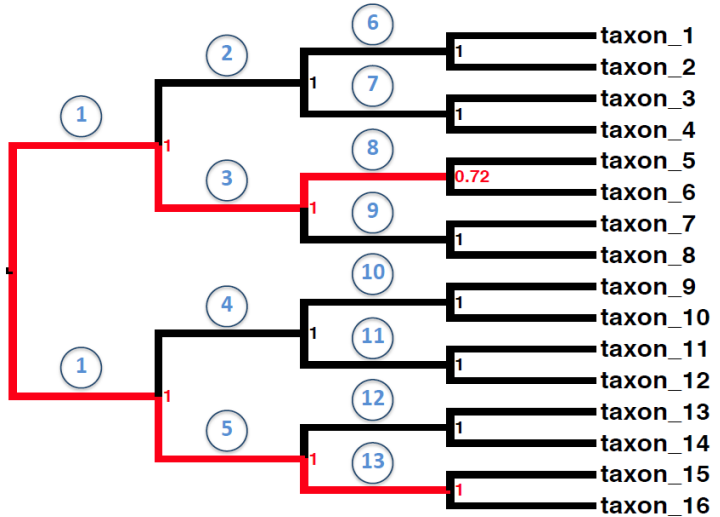


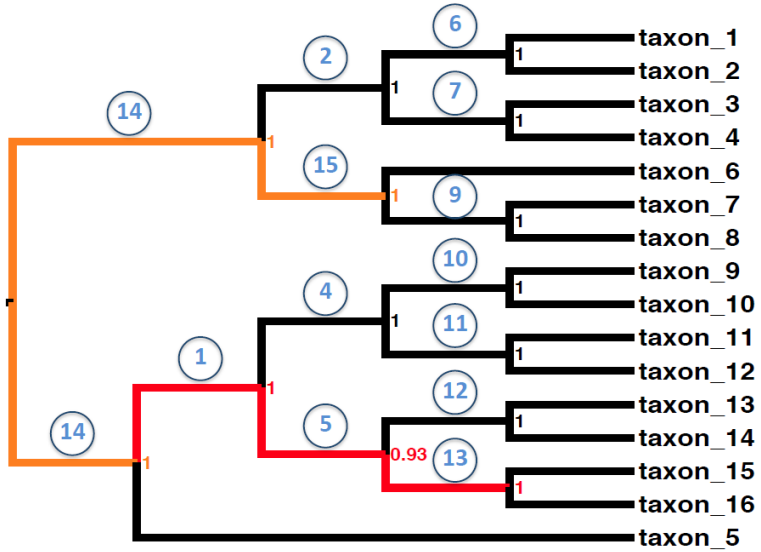
# Topology Based Community Detection



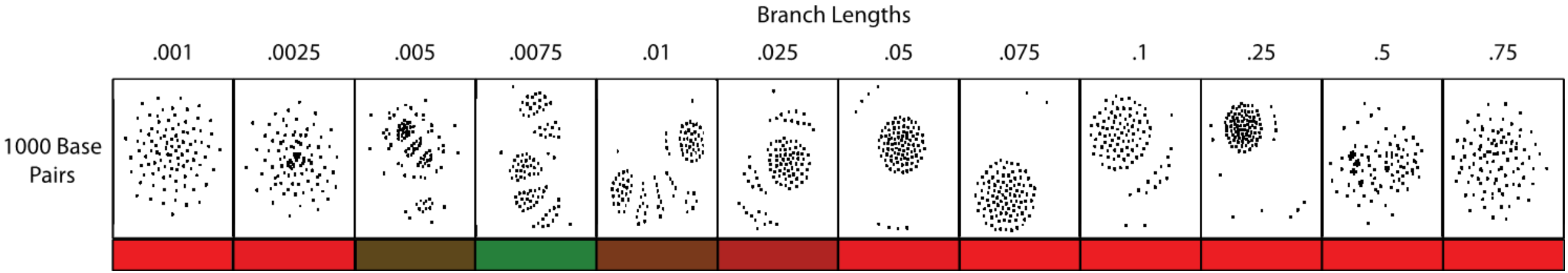
# Networks Detect Strong Conflict

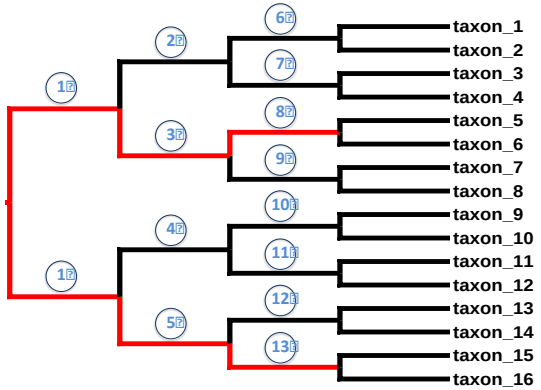


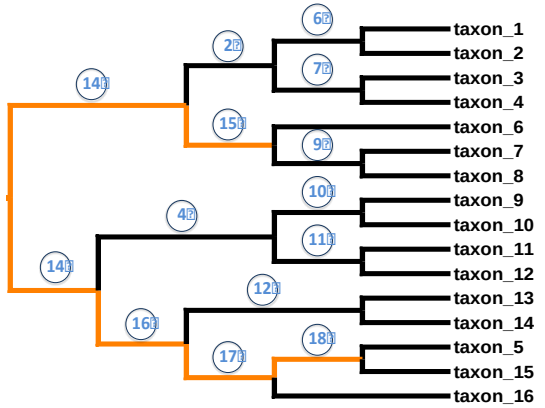



# Variation of Branch Length

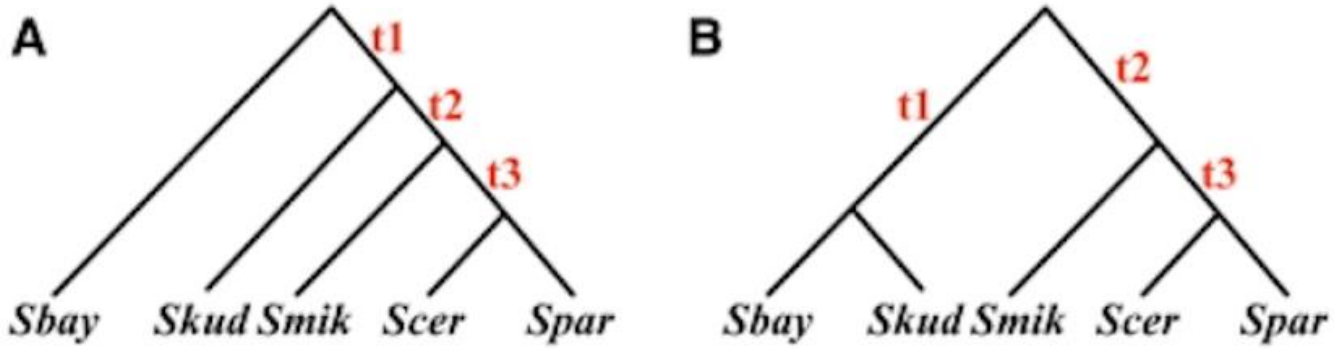


# Example: Empirical Data

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



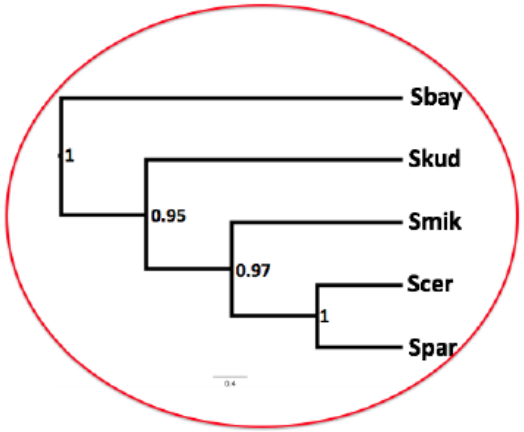

The Probability of a Gene Tree Topology within a Phylogenetic Network with Applications to Hybridization Detection

Yun Yu, James H. Degnan, Luay Nakhleh

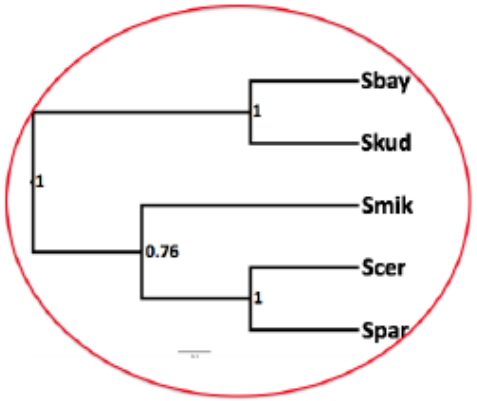
Published: April 19, 2012 • http://dx.doi.org/10.1371/journal.pgen.1002660

# Topology-based Network Analysis

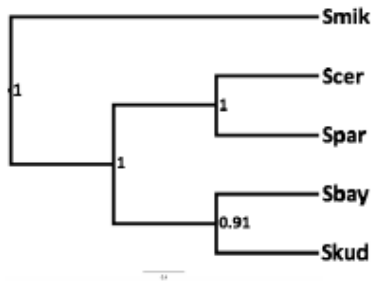
- Affinity matrix
  - Reciprocal of Tree-to-Tree distances
- Detect communities
  - Discovered 11 communities
- Consensus trees for each community
  - Consensus trees of two largest communities recovers 2 candidate species trees.



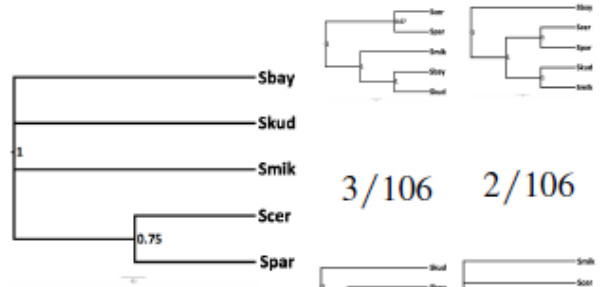

62/106




17/106




11/106




4/106



3/106



2/106

2/106

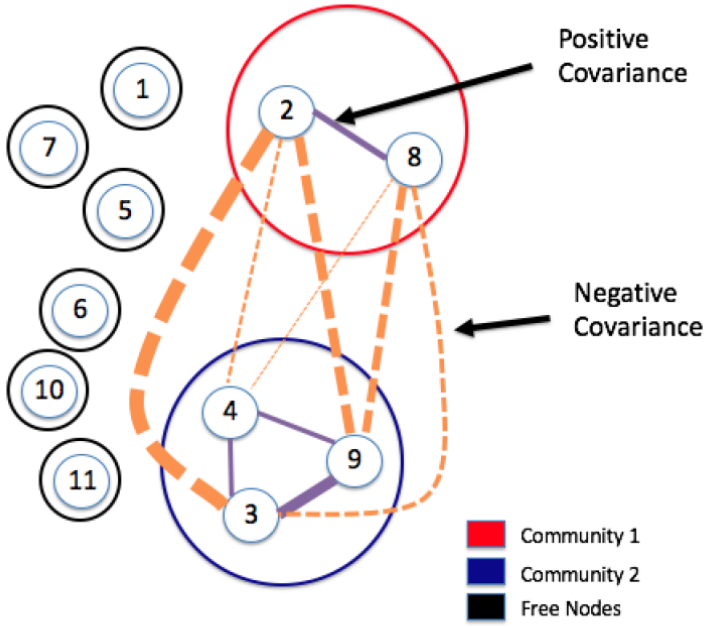


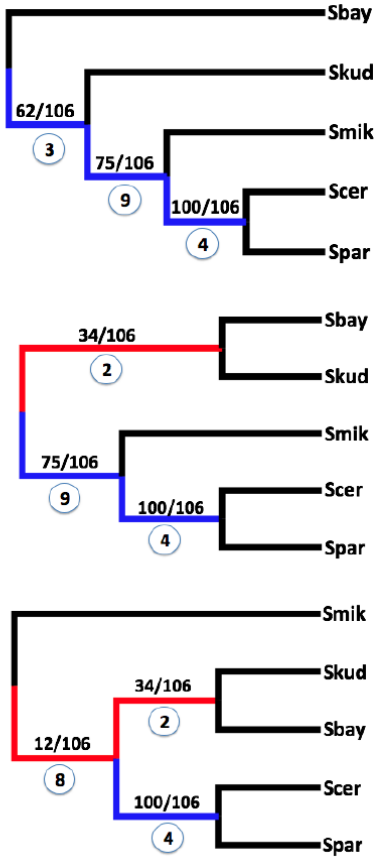
2/106

• • •

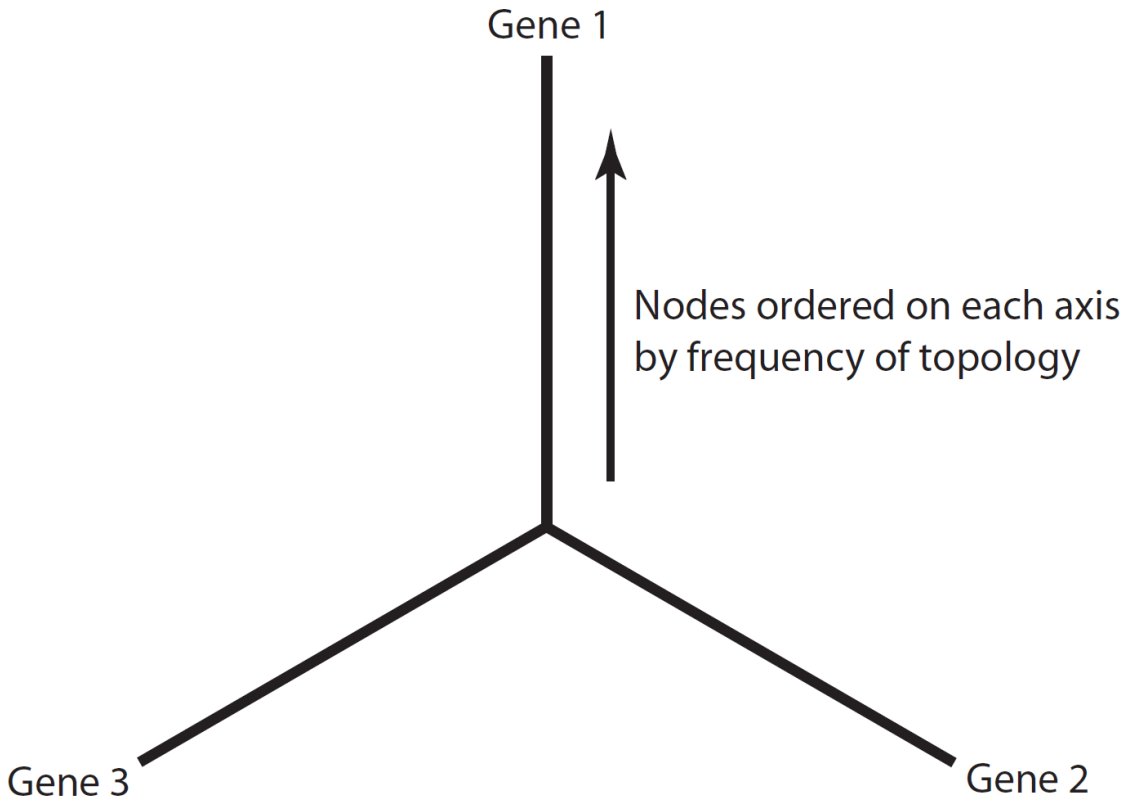
# Bipartition-based Network Analysis

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

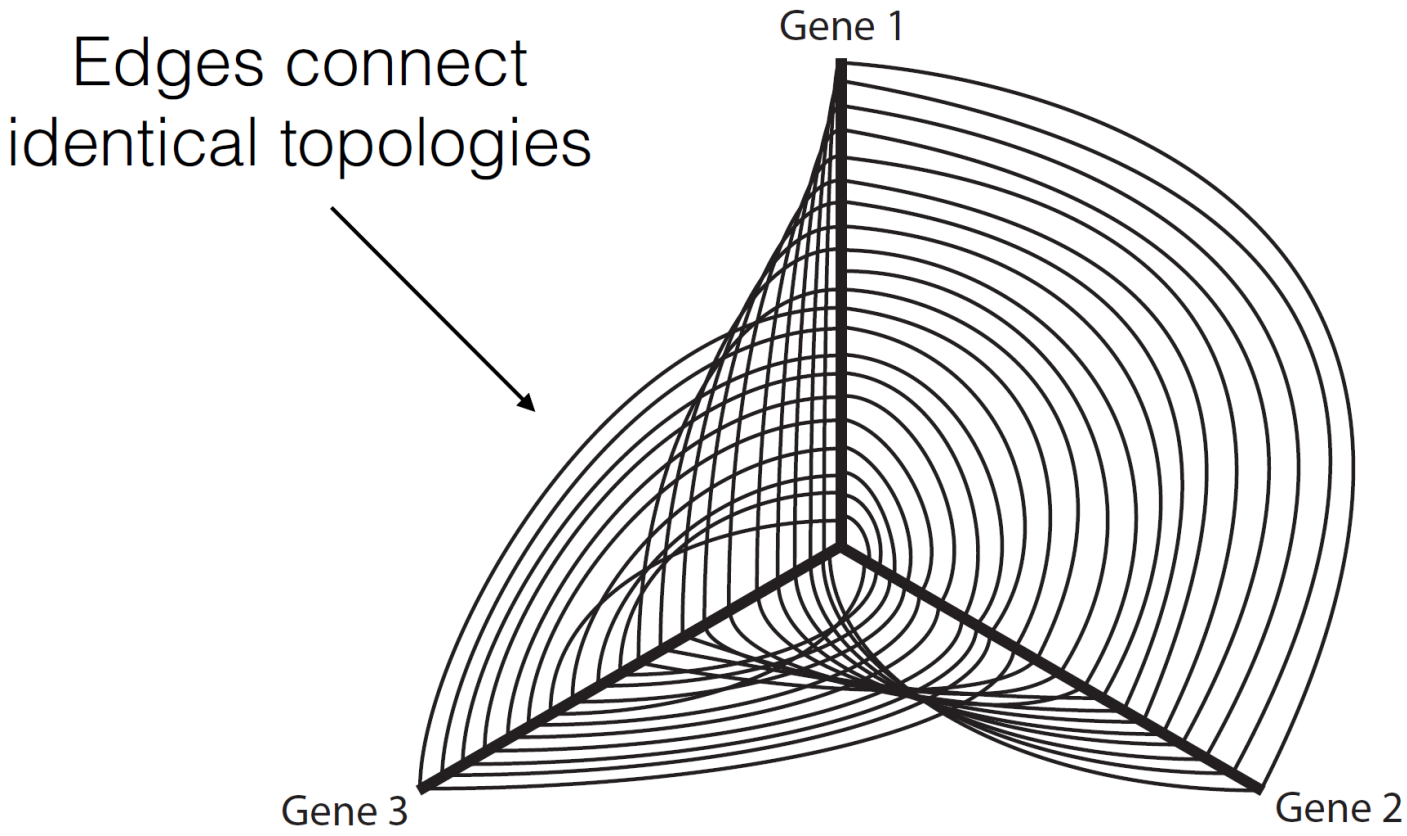


# Other Network Visualizations

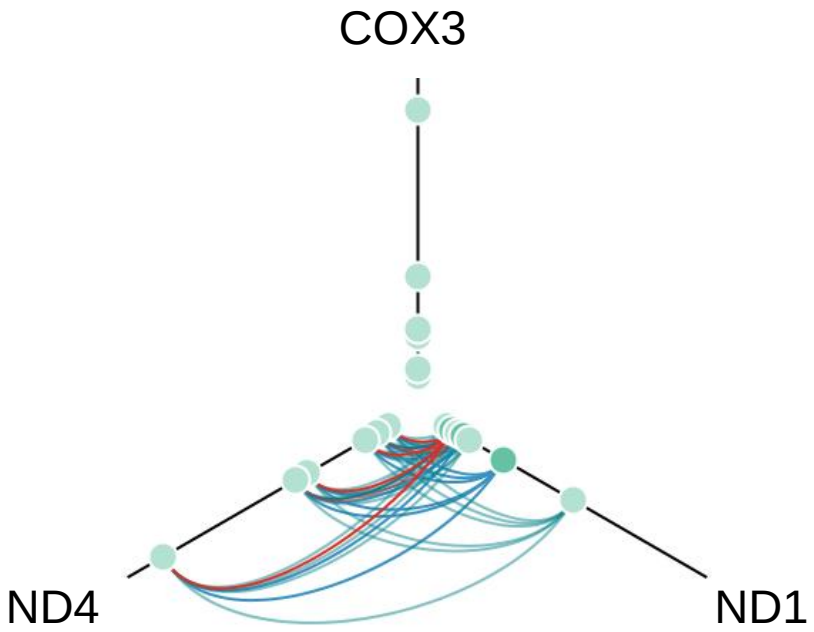


# Other Network Visualizations

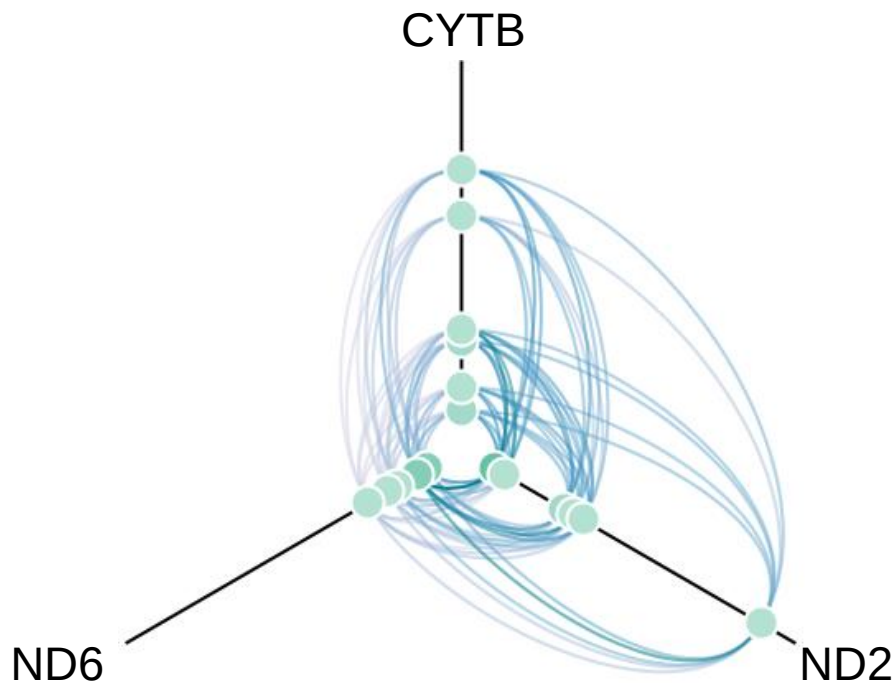


# Network Visualizations

Censored
RF distances below 4 removed



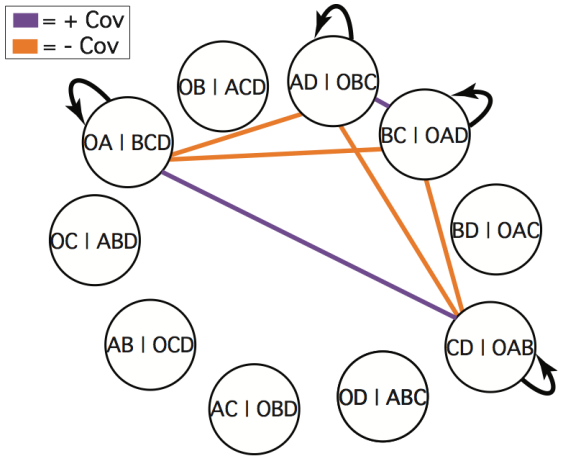

Uncensored



# Assessing Model Fit

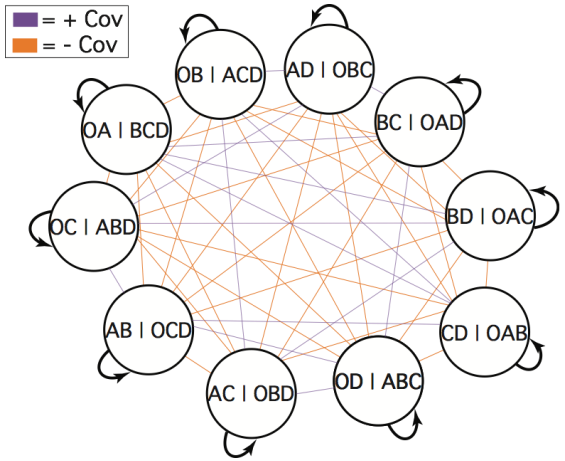
Using parametric bootstrapping or posterior prediction, we can compare network structures between observed and simulated datasets.

Empirical

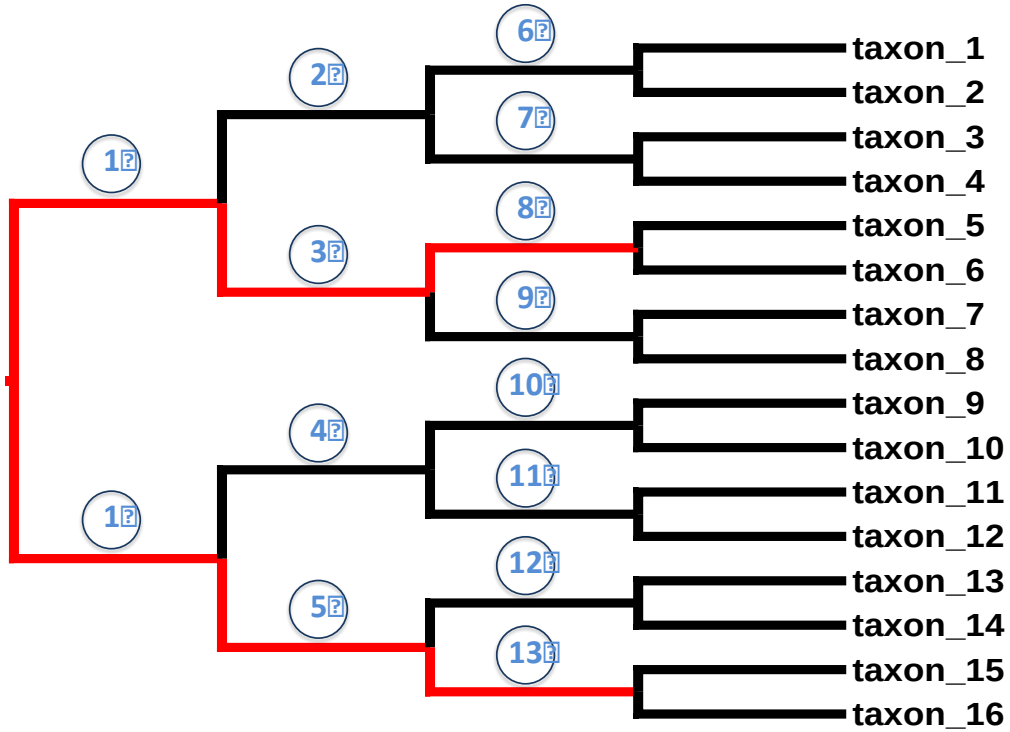



$\not\simeq$

Simulated



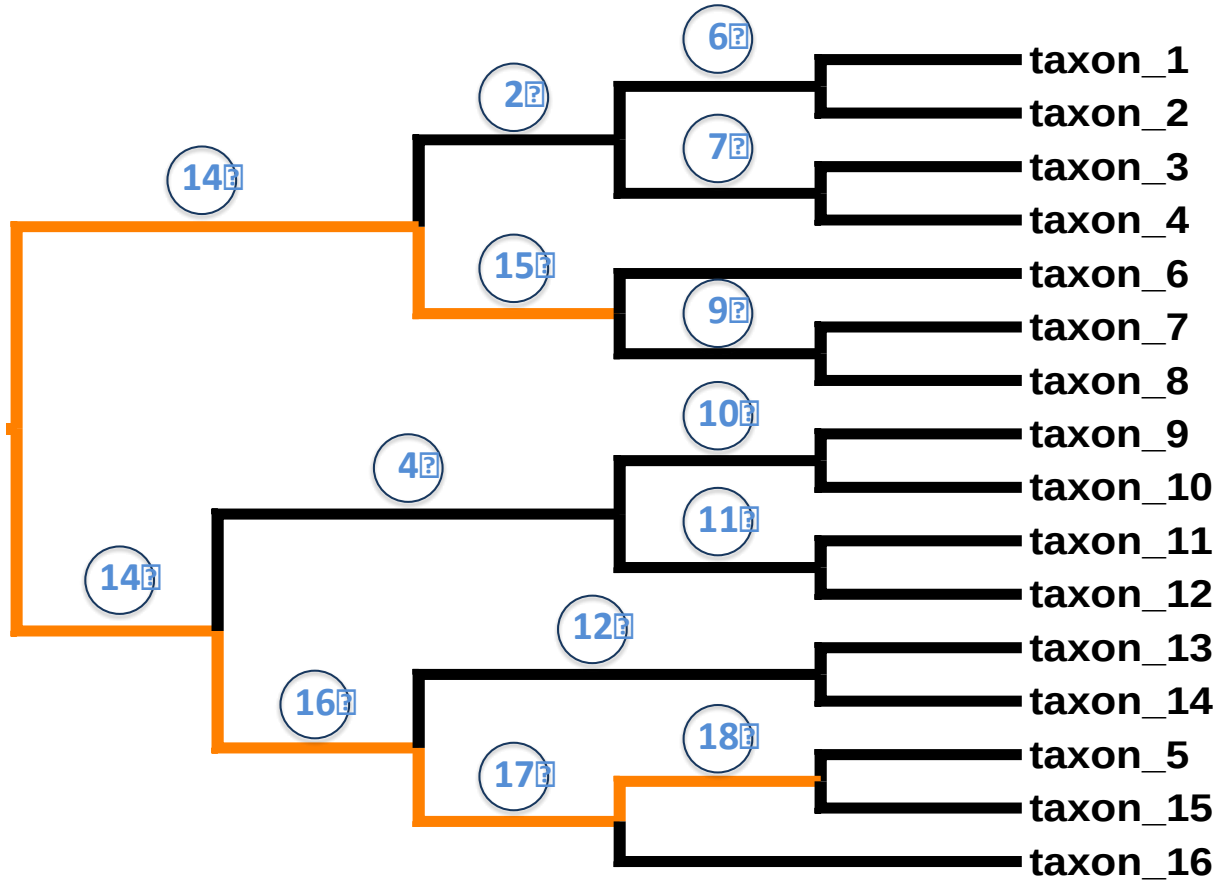
# The “Perfect Case”: 50 Trees




Cov(red, black) = (.50) – (.50*.50)
= .25

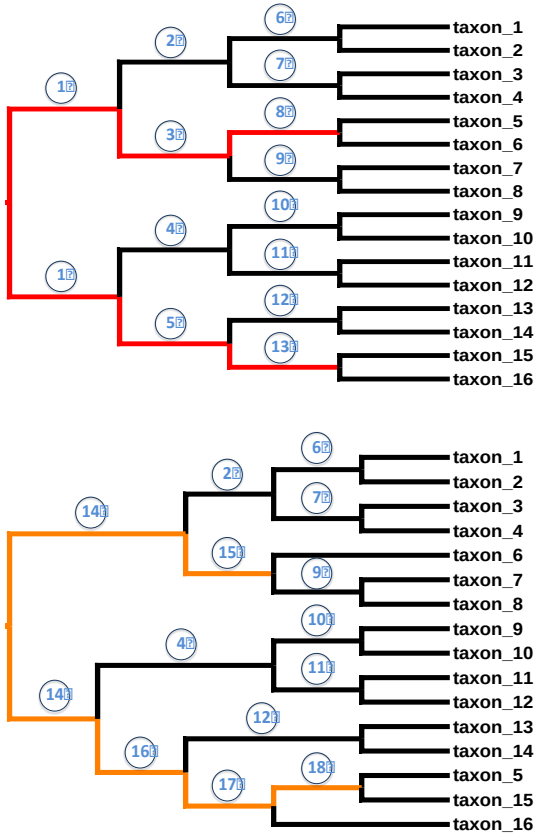
Cov(red, red) = (1) – (1*1)
= 0

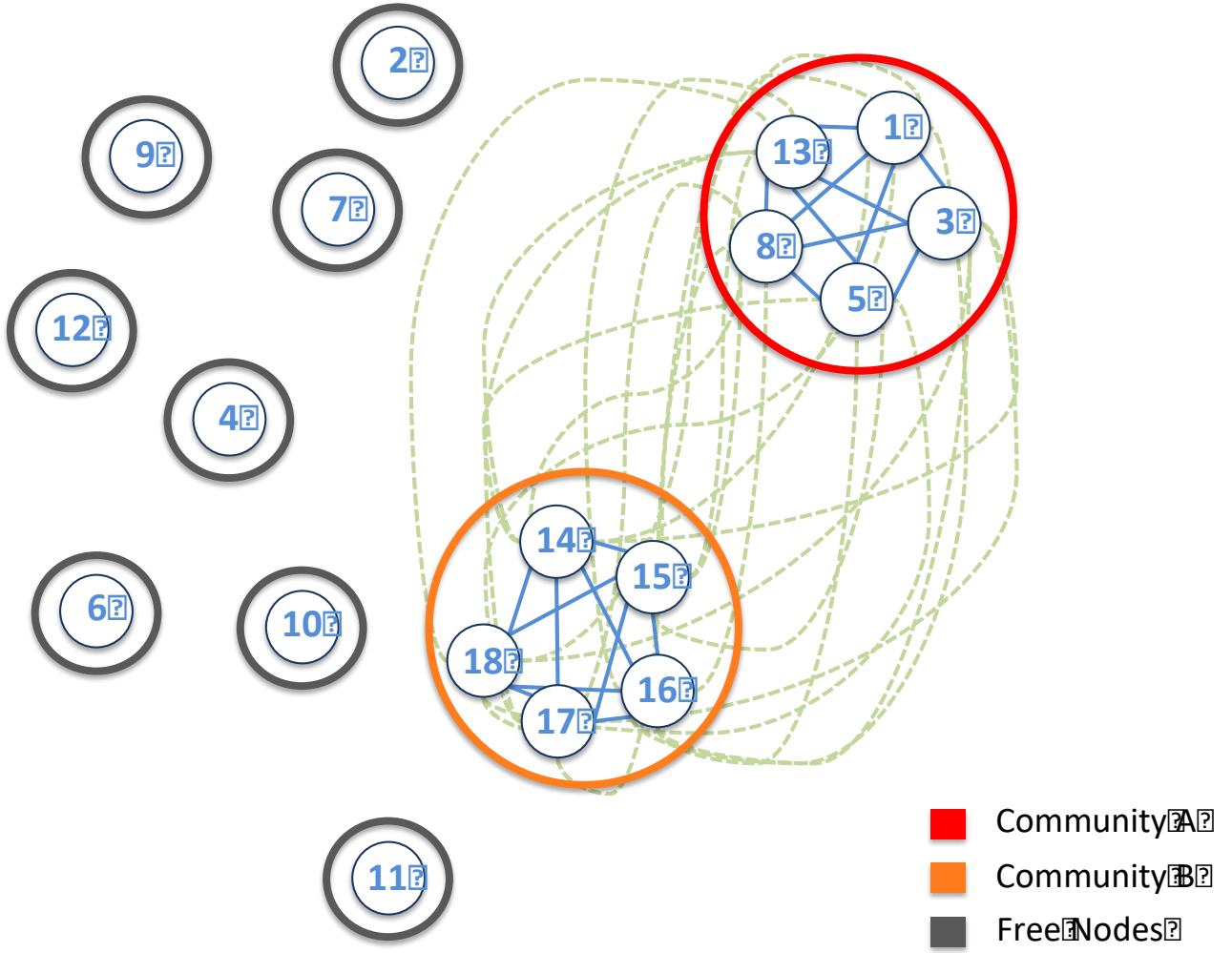
# The “Perfect Case”: 50 Trees




Cov(red, orange) = (0) – (.50*.50)
= -.25

# The Network


taxon_1
taxon_2
taxon_3
taxon_4
taxon_5
taxon_6
taxon_7
taxon_8
taxon_9
taxon_10
taxon_11
taxon_12
taxon_13
taxon_14
taxon_15
taxon_16


Community A
Community B
Free Nodes

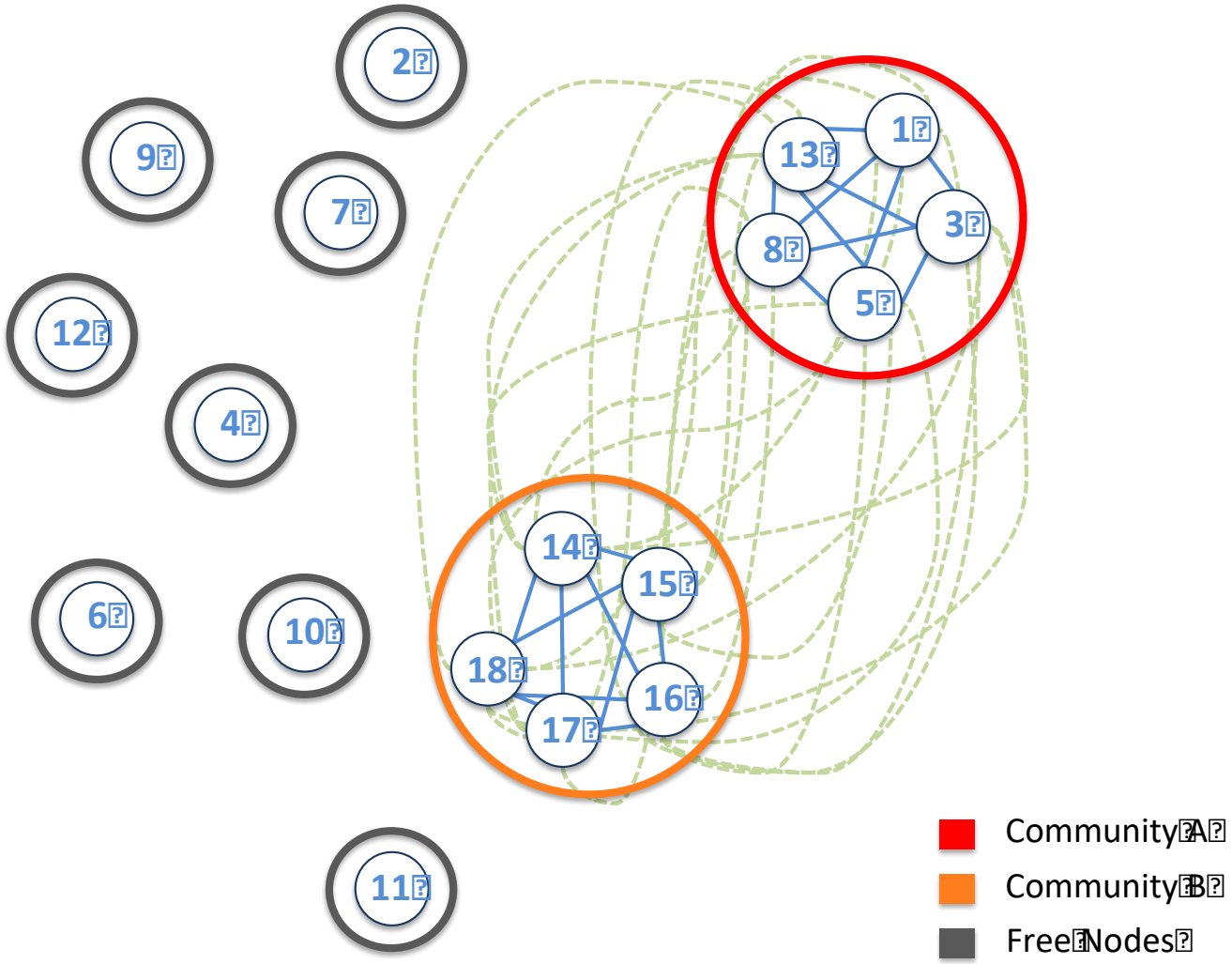
# Calculating the Hamiltonian

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

$H_a$ = -(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

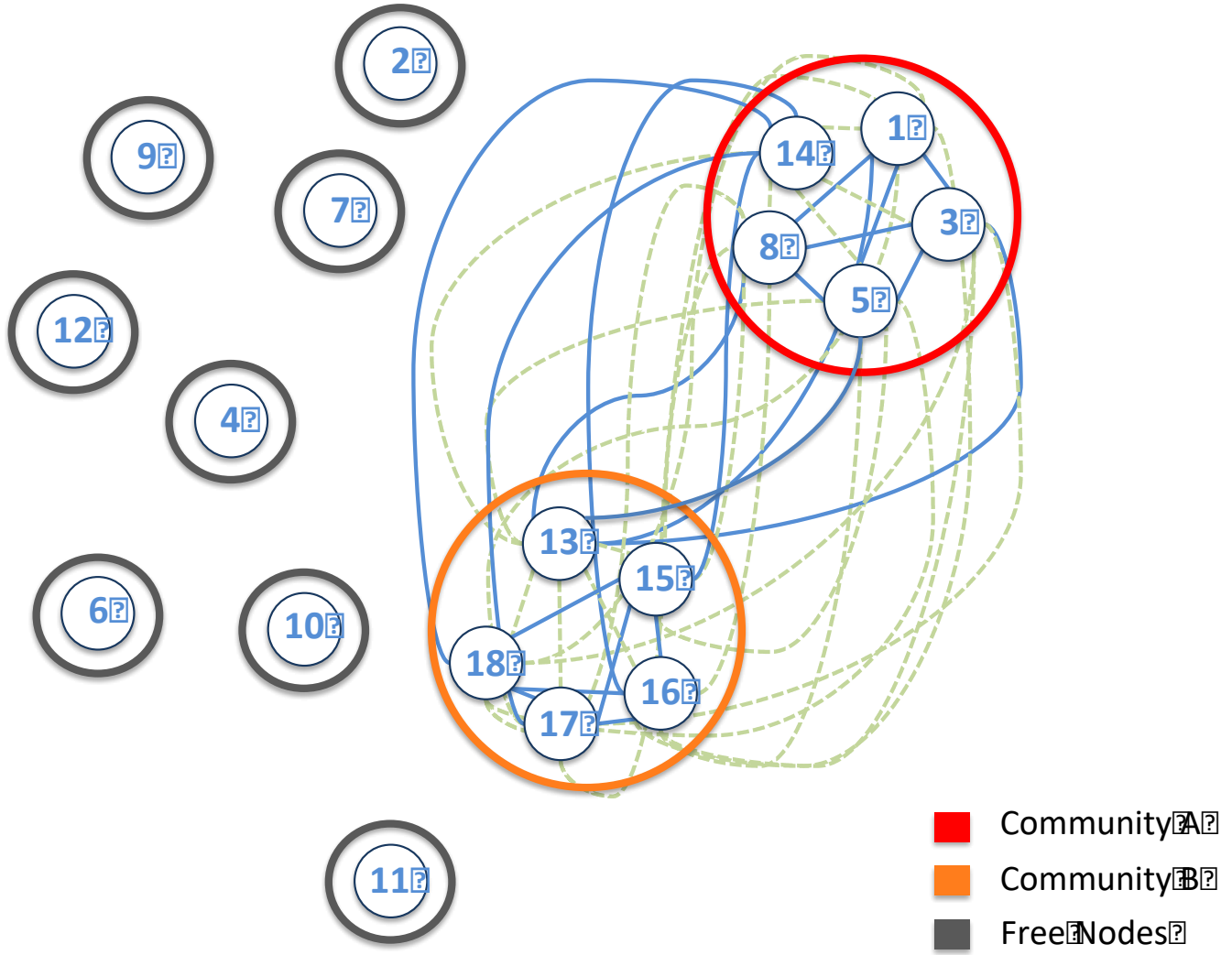
$$\mathcal{H}(\{\sigma\}) = -\sum_{ij}[\mathcal{A}_{ij} - c^2(\lambda^+ - \lambda^-)]\delta(\sigma_i, \sigma_j)$$

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

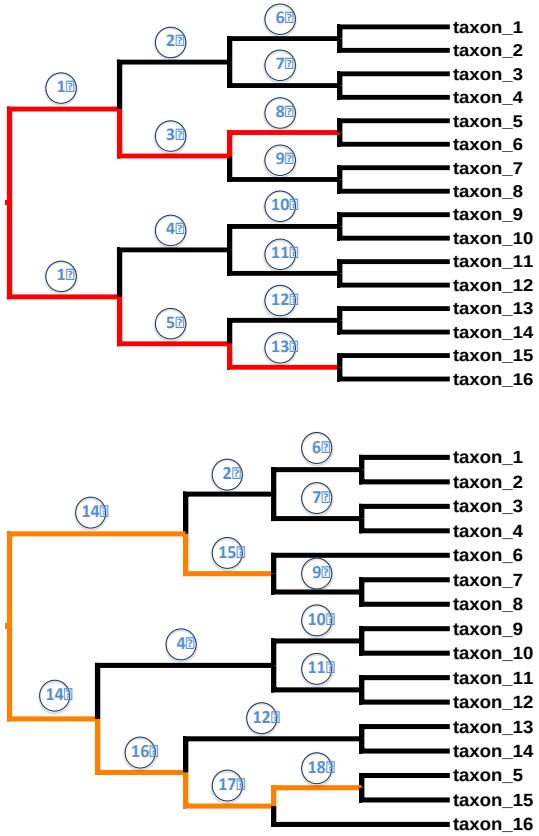
$H_a + H_b = H$ = -4.5

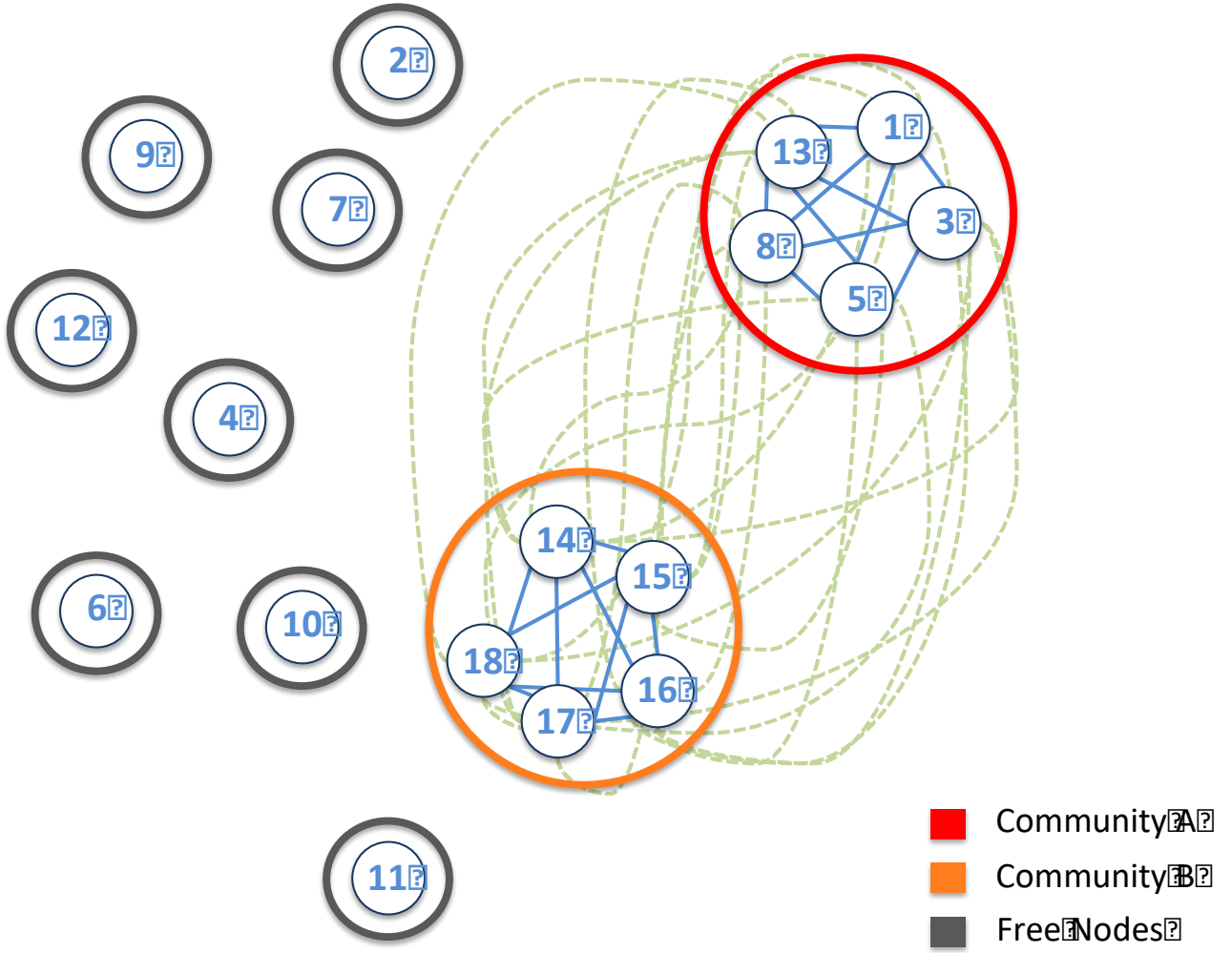
**Hamiltonian not minimized**

**No closed form solution!**
**Need optimization algorithm**
**(simulated annealing)**



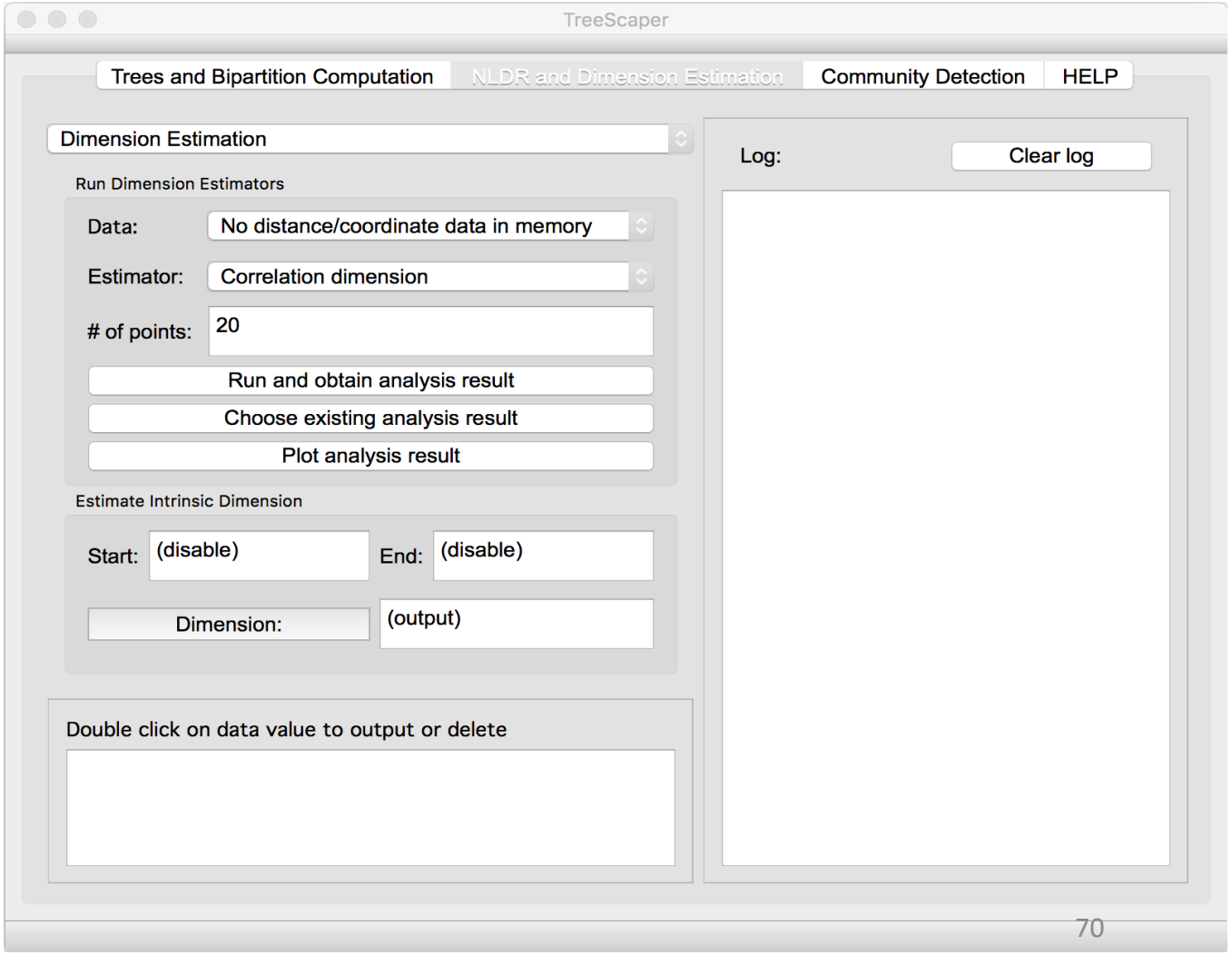
# Hamiltonian Minimum



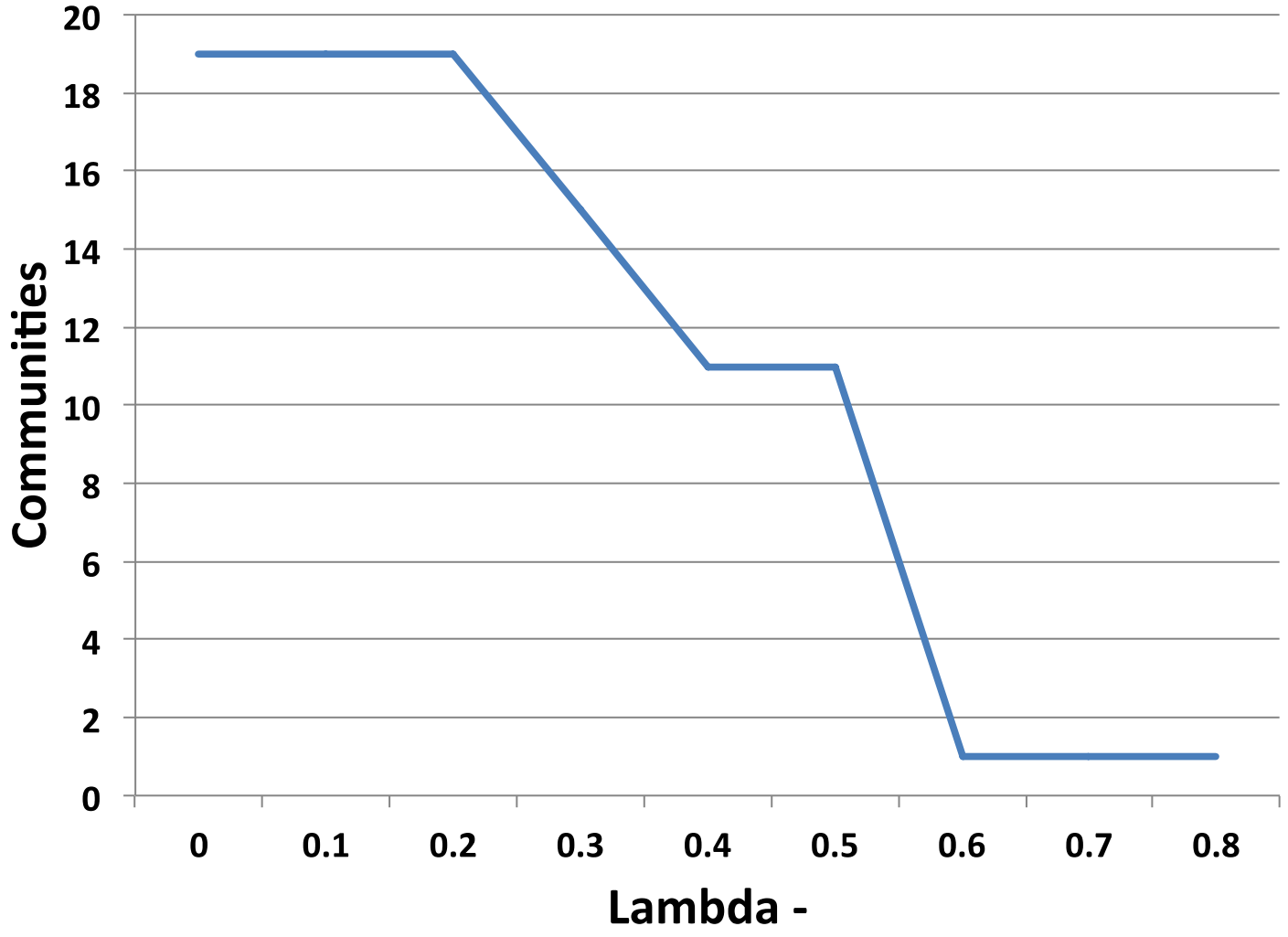



# TreeScaper Software

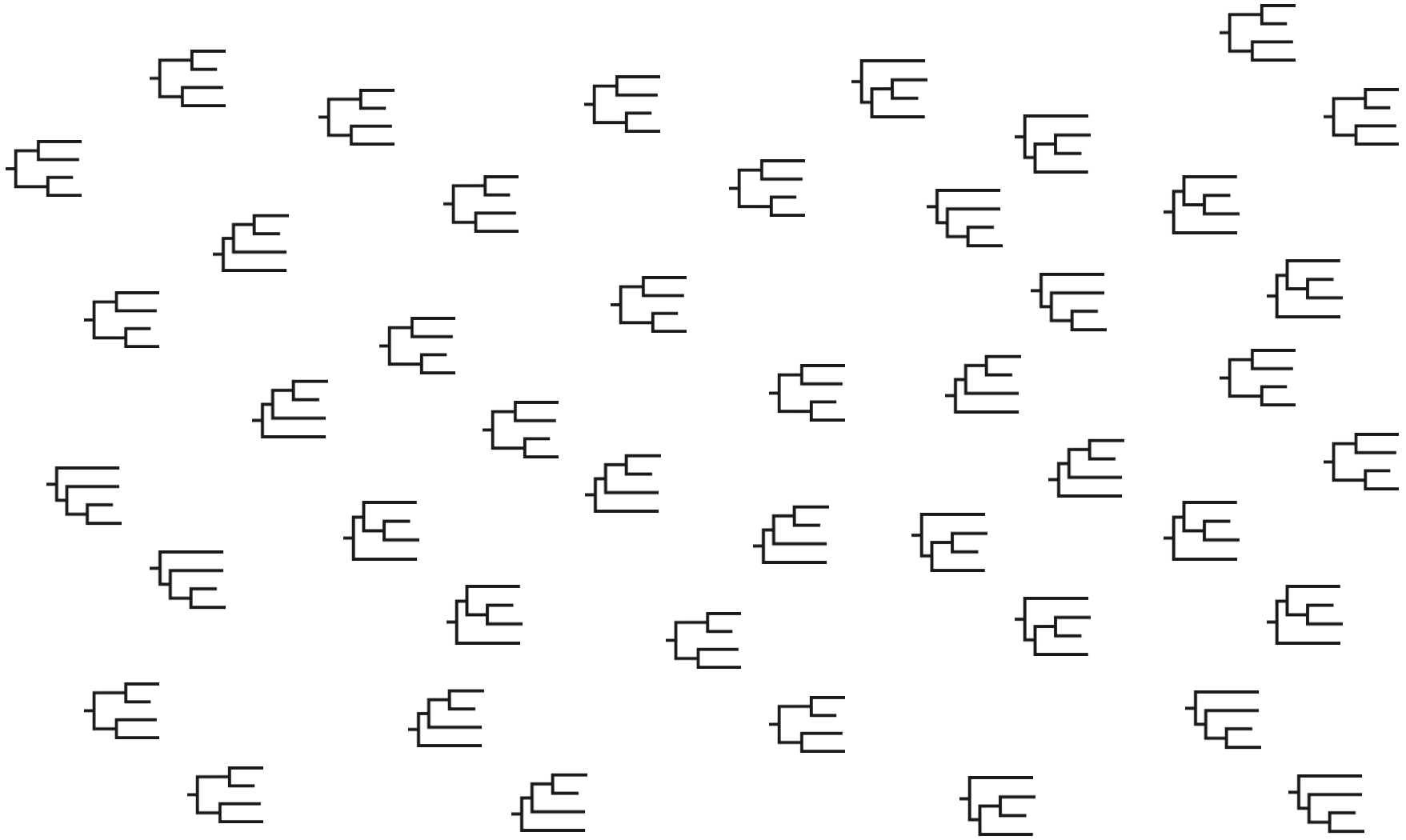
- Available on GitHub (developer version)
  - https://github.com/whuang08/TreeScaper
- Available on SourceForge:
  - https://sourceforge.net/projects/treescaper/
- TreeScaper Website:
  - http://www.math.fsu.edu/~whuang2/Indices/index_TreeScaper.html



# Finding the Natural Community Structure

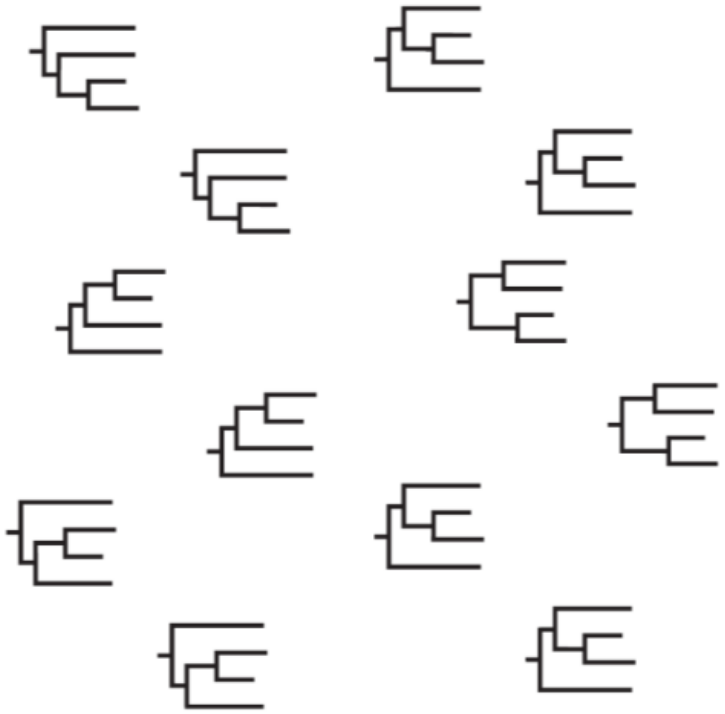


# Tree Sets



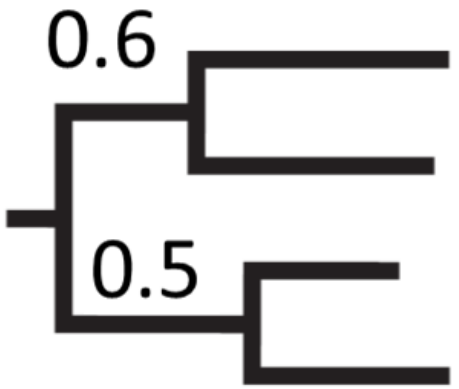
# Summarizing Tree Sets

Tree Set

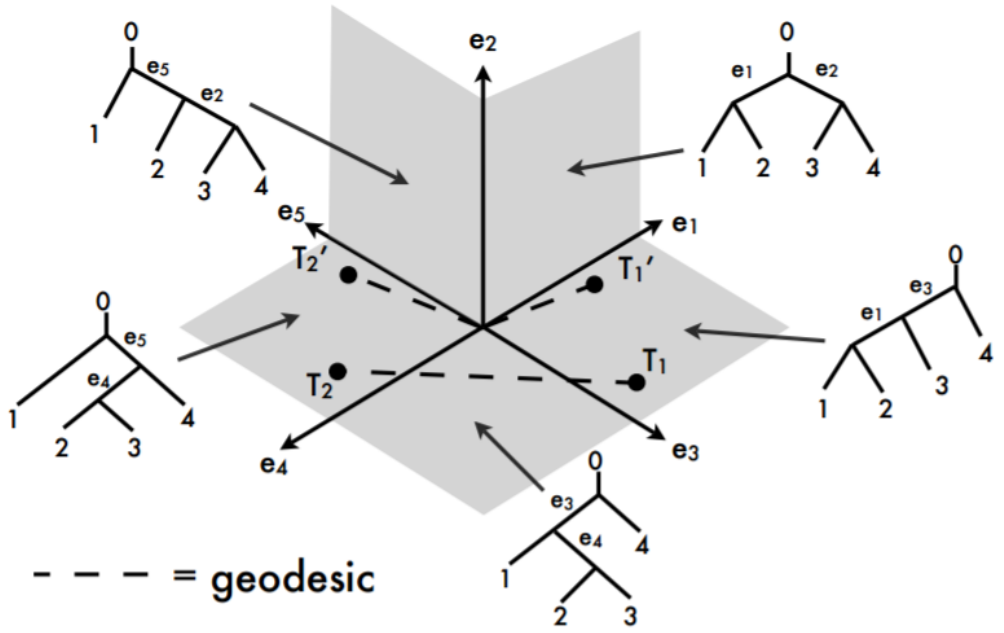


Loss Of Information!

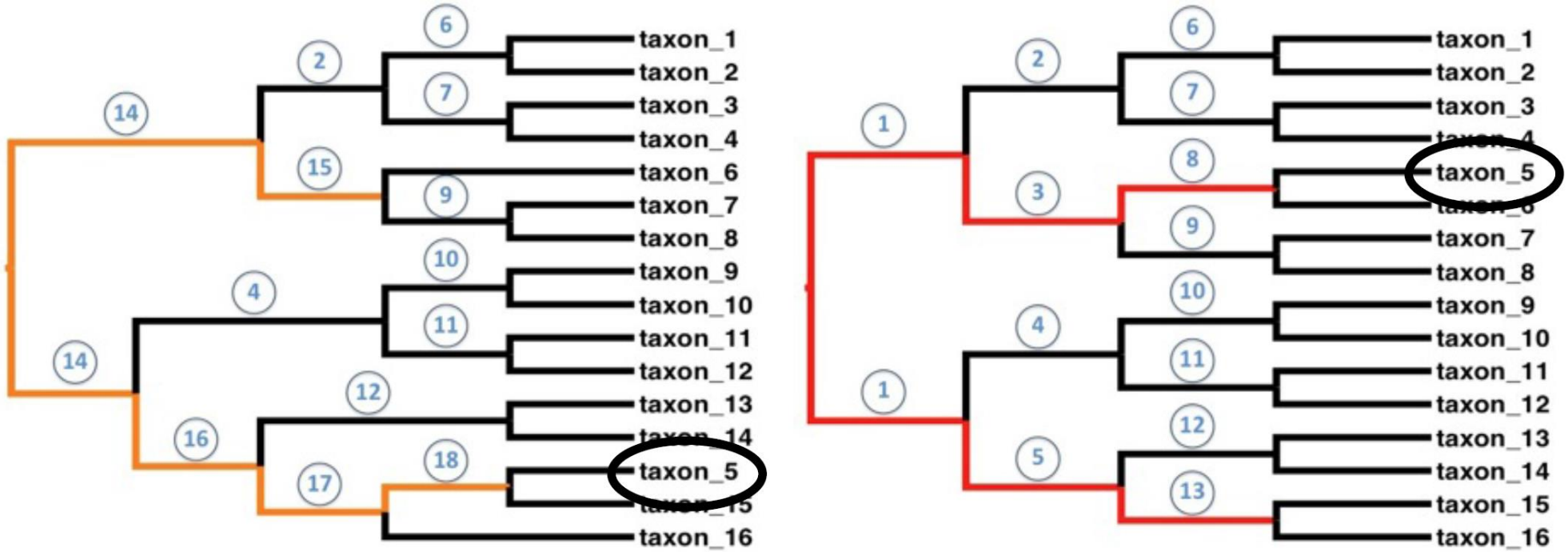
Majority Rule Consensus Tree

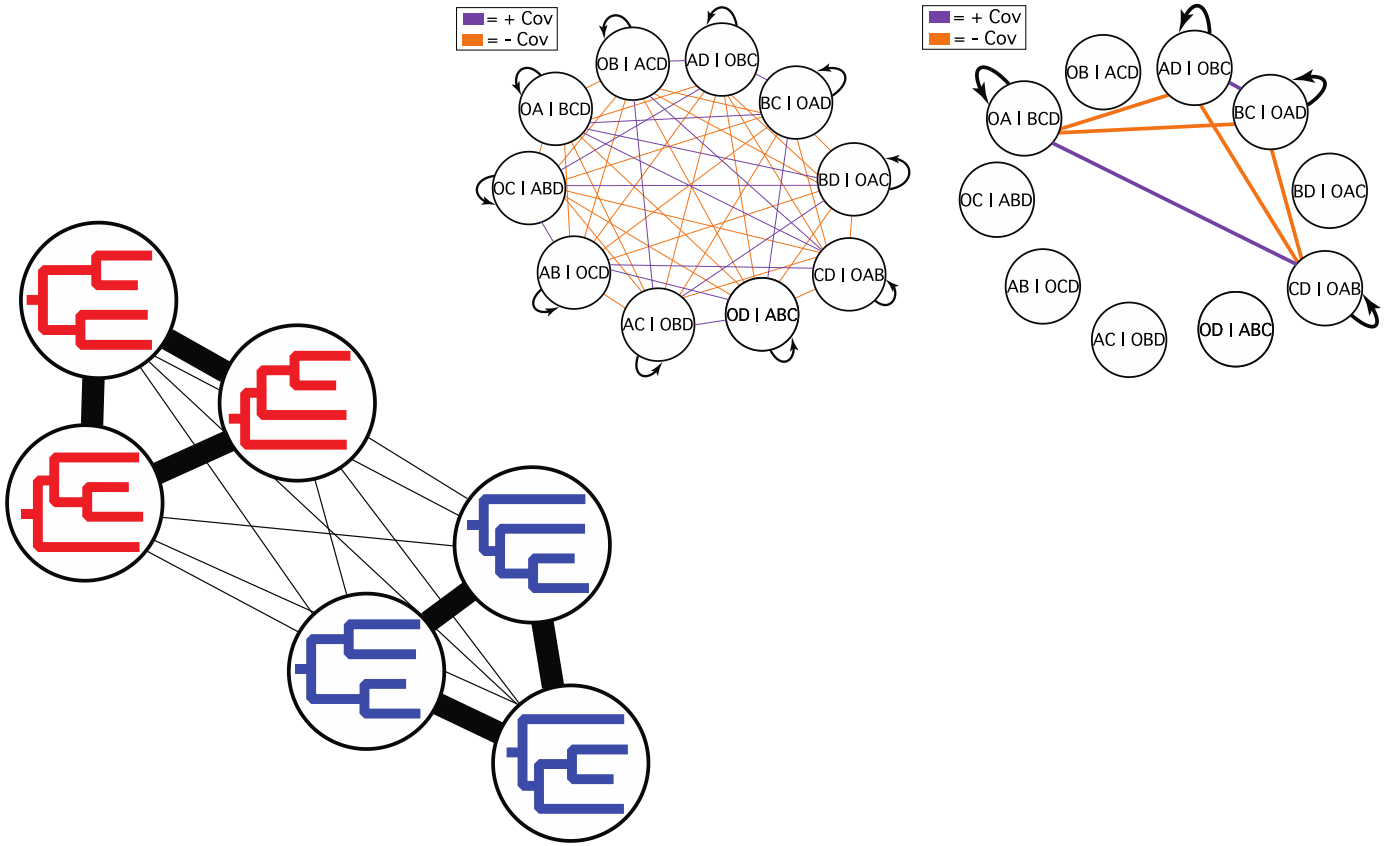



T. Margush and F. R. McMorris, Consensus n-trees, B. Math. Biol. 43 (1981), no. 2, 239–244




Megan Owen 2011


6
2
7
14
15
9
10
4
11
14
12
16
18
17
taxon_1
taxon_2
taxon_3
taxon_4
taxon_6
taxon_7
taxon_8
taxon_9
taxon_10
taxon_11
taxon_12
taxon_13
taxon_14
taxon_5
taxon_15
taxon_16
1
2
6
7
1
3
8
5
9
4
10
11
1
12
13
taxon_1
taxon_2
taxon_3
taxon_4
taxon_5
taxon_6
taxon_7
taxon_8
taxon_9
taxon_10
taxon_11
taxon_12
taxon_13
taxon_14
taxon_15
taxon_16


= + Cov
= - Cov
OB | ACD
AD | OBC
OA | BCD
BC | OAD
OC | ABD
BD | OAC
AB | OCD
CD | OAB
AC | OBD
OD | ABC
= + Cov
= - Cov
OB | ACD
AD | OBC
OA | BCD
BC | OAD
OC | ABD
BD | OAC
AB | OCD
CD | OAB
AC | OBD
OD | ABC

# Method for Community Detection

- Constant-Potts Model(CPM):

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

# Non Linear Dimensionality Reduction