Reconstructing Networks
Part II: Computational Aspects

C. Randal Linder, Bernard M.E. Moret,
Luay Nakhleh, and Tandy Warnow

U. of Texas at Austin and U. of New Mexico

Overview

- Review of tree reconstruction
- Main tasks in reconstruction
- Preprocessing the data
- Building a network
- Evaluating reconstructions: criteria and simulations
- Maddison's idea and our method
- Conclusions

Reconstructing Trees: Overview

- Data
- Methods
- Evaluation
- Conclusions

Reconstructing Trees: Data

- **All kinds of data have been used: behavioral, morphological, metabolic, etc.**
- **Preferred choice today is molecular data.**
- **Two main kinds of molecular data:**
  - **sequence data**
    (DNA sequence on genes)
  - **gene content and order data**
    (gene list or sequence on chromosomes)
- **Data elements that can assume new values (according to the model) independently of others are called *characters*.**

Sequence Data

Typically the DNA sequence of a few genes.

Characters are individual positions in the string and can assume four states.

Evolves through point mutations, plus insertions (incl. duplications), and deletions.

- Find homologous genes across all organisms.
- Align gene sequences for the entire set (to identify gaps – insertions and deletions – and point mutations).
- Decide whether to use a single gene for each analysis or to combine the data.
- Lengths limited by size of genes
  (typically several hundred base pairs)

Sequence Data: Attributes

- **Advantages:**
  - Large amounts of data available.
  - Accepted models of sequence evolution.
  - Models and objective functions provide a reasonable computational framework.
- **Problems:**
  - Sequencing errors (down to $\sim$1%).
  - Fast evolution restricts use to a few million years.
  - Gene evolution need not be identical to organism evolution.
  - Multiple alignments are not well solved.
  - Reconstruction methods do not scale well
    (in terms of accuracy and running time).

Gene Content and Order Data

The list (or ordered sequence) of genes on one or more chromosomes.

Entire gene order is one character (huge number of states).
Content evolves through insertions (incl. duplications), deletions, and translocations (between chromosomes).
Order also evolves through inversions and transpositions.

- Identify homologous genes, including duplications.
- Refine model for collection of organisms (e.g., handle bacterial operons or eukaryotic exons explicitly).

Gene Content and Order: Attributes

- **Advantages:**
  - Low error rate (recognize homologies).
  - No gene tree/species tree problem.
  - Rare evolutionary events and unlikely to cause “silent” changes—so can go back hundreds of millions years.
  - Synteny (on same chromosome) and adjacency are powerful tools to distinguish orthologs from paralogs and to identify horizontal gene transfers.
- **Problems:**
  - Mathematics *much more complex* than for sequence data.
  - Models of evolution not well characterized.
  - Limited data (mostly organelles).

Reconstructing Trees: Methods

Three categories of methods:

- Distance-based methods, such as neighbor-joining.
- Parsimony-based methods (such as implemented in PAUP*, Phylip, Mega, TNT, etc.)
- Likelihood-based methods (including Bayesian methods, such as implemented in PAUP*, Phylip, FastDNAML, MrBayes, GAML, etc.)

In addition:

- Meta-methods (quartet-based methods, disk-covering method) decompose the data into smaller subsets, construct trees on those subsets, and use the resulting trees to build a tree for the entire dataset.

Distance-Based Methods

- Use edit or expected true evolutionary distances.
- Usually run in *low polynomial time*.
- Reconstruct *only topologies*: no ancestral data.
- Prototype is Neighbor-Joining; BioNJ and Weighbor are two improvements.
- NJ is optimal on additive distances (where the distance along a path in the true tree equals the pairwise distance in the matrix).
- NJ is statistically consistent (produces the true tree with probability 1 as the sequence length goes to infinity).

Parsimony-Based Methods

- Aim to minimize total *number of character changes* (which can be weighted to reflect statistical evidence).
- Assume that characters are *independent*.
- Reconstruct *ancestral data*.
- Are known not to be statistically consistent with sequence data (but examples are fairly contrived).
- Finding most parsimonious tree is computational very expensive (NP-hard).
- Optimal solutions limited to sizes around 30; heuristic solutions appear fairly good to sizes of 500.

Likelihood-Based Methods

- Are based on a specific model of evolution and *must estimate all model parameters*.
- Produce *likelihood estimate* (prior or posterior conditional) for each tree.
- Are statistically consistent.
- Reconstruct *only topologies* (Bayesian methods may reconstruct ancestral data).
- Are prone to numerical problems: likelihood of typical tree on 20 items is around 10^{-21}; on 50 items, around 10^{-75}; . . .
- Even *scoring* one tree is very expensive.
- Optimal solutions limited to sizes below 10; heuristic solutions appear fairly good to sizes of 100.

Reconstructing Trees: Evaluation

The standard metric used in comparing two trees is the ***Robinson-Foulds*** metric, based on comparing the leaf bipartitions induced by internal edges:

U: {A, B},{C, D, E} (correct) U': {A, B},{C, D, E}

V: {A, B, C},{D, E} (incorrect) V': {A, B, E},{D, C}

Difference in V and V' ⟶ 50% error

Reconstructing Trees: Conclusion

- Reconstruction methods based on sequence data (the current standard) can do well up to a few hundred taxa when using good data (many genes and with good ortholog identification).
- Gene-content and gene-order data are crucial in the identification of orthologs and horizontal transfers.
- Gene-order data can be used for tree reconstruction when studying older events—the methods are relatively new.
- Scaling up to larger datasets may be possible through metamethods such as disk-covering methods.

Reconstructing Networks

The main tasks are:

- **Eliminate noisy signals**
- **Detect reticulation events**
- **Build a network**
- **Assess the quality of a reconstruction**

Noisy Signals

- Data errors, biased sampling, differential selection pressures, etc.
- Gene tree / species tree problem
- Other confounding data (e.g., many genes acquired through horizontal transfer)
- Poor phylogenetic tree reconstruction (for methods based on trees—the vast majority)

Gene Tree / Species Tree

- Depending on the data used (DNA sequences, syntenies, gene orders), reconciliation of gene trees within a single species tree is possible.
- Reconciliation must be done early to avoid later errors, but it usually requires fairly good phylogenetic trees—a chicken and egg problem!

Confounding Data

Horizontal gene transfer and retrogression

- Legitimate reticulation events, but can lead tree reconstruction astray.
- Before attempting reconstruction, can identify and eliminate genes acquired through HGT or retrogression using an intrinsic method
  (based on GC content, codon usage statistics, gene content or order data).

Poor Phylogenetic Trees

- Due to confounding data? Gene tree/species tree, horizontal transfer?
- All precautions taken? Due to poor choice of reconstruction method?
- No method works? Insufficient data or too many errors? Can a reconstruction method not based on trees work?

Build the Network

Three main approaches:

Add to a Tree:
Infer a tree, then add extra edges to optimize some criterion

Combine Trees:
Infer a collection of optimal trees, then reconcile the trees into a network

Compute Splits:
Precompute incompatibilities in the dataset and introduce network edges to account for them

Add to a Tree: Description

- Reconstruct a tree T
- Add non-tree edges to T until stopping rule is triggered
- Stopping rule is some optimization criterion (e.g., minimize weighted sum of evolutionary events)

Requires good reconstruction of the first tree!

Non-tree edges are typically chosen on the basis of incongruence between segments of sequences.

Add to a Tree: Rationale

- If data are “tree-like”, no extra edges will be needed.
  (Parsimony again: do not add unnecessary edges.)
- If data imply a network, various segments of the sequences will require nontree edges.
- Candidate nontree edges are selected to optimize some tree-style criterion.

Add to a Tree: Methods

- Statistical parsimony (Templeton *et al.*)
- Reticulogram (Makarenkov and Legendre)
- Horizontal Gene Transfer (Hallett and Lagergren)

Add to a Tree: Codes

- TCS
  - statistical parsimony method
  - written by Clement *et al.* in Java
- T-REX
  - Reticulogram method
  - written by Makarenkov, binaries for Windows and Mac
- Horizontal Transfer Code
  - Hallett & Lagergren method
  - written by Addario-Berry *et al.* in Java

Combine Trees: Description

- Compute best trees (e.g., in terms of parsimony scores) for various segments of the sequences and combine all into a network; or
- Compute all minimum spanning trees for the data and combine them into an minimum spanning network.

Confounding events (gene/species tree) must be removed!

Less sensitive than “Add to a Tree” to the quality of reconstructed trees, but a single error in one tree can still force the introduction of a reticulation event.

Combine Trees: Rationale

- Different segments of the sequences evolved down different trees.
- The trees are reliable, so conflicts between the trees represent reticulation events.
- The different trees can be combined into a network.

Combine Trees: Methods

- Netting (Fitch)
- Median Networks (Bandelt *et al.*)
- Median-Joining networks (Bandelt *et al.* and Foulds *et al.*)
- Molecular-variance parsimony (Excoffier and Smouse)

Combine Trees: Codes

- NETWORK
  - Median-Joining and Reduced Median methods
  - written by Röhl *et al.*, binaries for Windows
- ARLEQUIN
  - Molecular-variance parsimony
  - written by Schneider *et al.* in Java

Compute Splits: Description

- Compute a distance matrix.
- Find all possible splits.
- Construct a network that contains all splits.

Requires an accurate estimate of pairwise distances. Does not normally return a network, but a superset that contains all plausible networks.

Compute Splits: Rationale

- When the distance computation is accurate, the method returns the set of true splits.
- With reticulation events, the set of splits of the network equals the set of splits of all trees inside the network.

Any deviation from optimal conditions means that the method returns too many splits.

Compute Splits: Methods

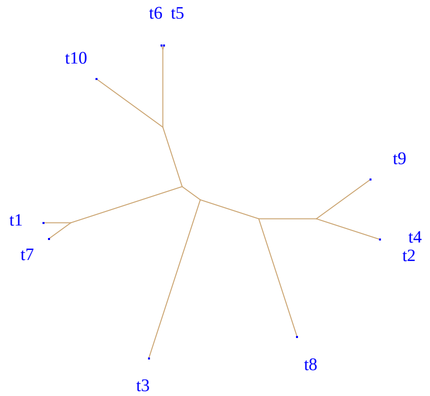
- splits decomposition (Bandelt and Dress)
- neighbornet (Bryant and Moulton)

Compute Splits: Codes

- SplitsTree
  - splits decomposition
  - written by D. Huson, binaries for Mac and Unix
- SpectroNet
  - analysis and visualization of data based on splits
  - written by M. Hendy *et al.* in C++ (source)
- NeighborNet
  - neighbornet
  - written by D. Bryant for Unix (source)

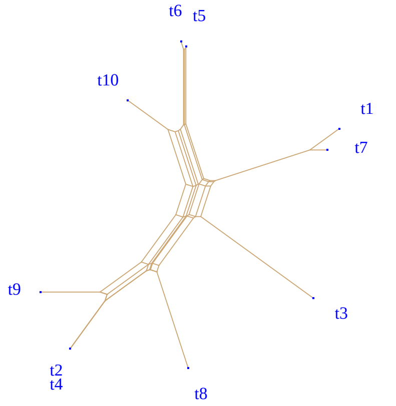
NeighborNet Example: Model Tree

The model "network" is a tree of 10 taxa, generated by `r8s` with 2000-nucleotide sequences under the GTR+Gamma model with invariant sites.



NeighborNet Example: Output

The output of NeighborNet is a complex network giving many choices of reticulations, in spite of the fact that the model network was just a tree.



Other Methods

- STATGEOM
  - statistical geometry
  - written by K. Nieselt-Struwe, source and Sun binaries
- GEOMETRY
  - statistical geometry
  - written by Kuznetsov and Morozov, binaries for DOS
- PYRAMIDS
  - the Pyramids method
  - written by J.C. Aude *et al.*, binaries for Windows and Unix

Evaluating Reconstructions

- Generate a model phylogeny, evolving simulated molecular data along the paths and creating reticulation events according to a predefined model.
- Infer a phylogeny using the method under test.
- Compare the inferred phylogeny to the model phylogeny.

Generating a Model Phylogeny

- **What data must be used?**
  Sequence data is always required.
- **How to evolve the data?**
  DNA models, models for genome rearrangement, mixing genes or DNA sequences in hybridization.
- **Does topology depend on the data?**
  Do reticulation events depend on the characteristics and history of the putative "parents"?
- **How are pure speciation events chosen?**
  What is the model for creation of tree topologies?
- **How are rates of evolution determined?**
  Random, inherited, models of external pressures, etc.

Topological Accuracy

We developed two measures of the topological error between two networks:

- ***Tree-based:* use splits as for trees.**
- ***Network-based:* generalize bipartitions (Robinson-Foulds) to tripartitions.**

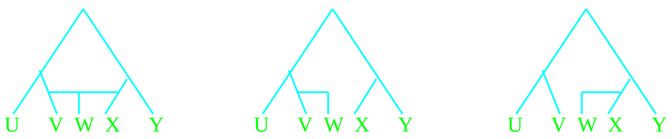
Tree Splits: Concept

- **A network N induces a set $\mathcal{T}(N)$ of trees**
- **Each tree T defines a set $\mathcal{C}(T)$ of splits**
- **Define the splits of a network as**

$$\mathcal{C}(N) = \cup_{T \in \mathcal{T}(N)} \mathcal{C}(T)$$

- **Define**
  - $FP(N, N') = |\mathcal{C}(N') - \mathcal{C}(N)|$ false positives
  - $FN(N, N') = |\mathcal{C}(N) - \mathcal{C}(N')|$ false negatives

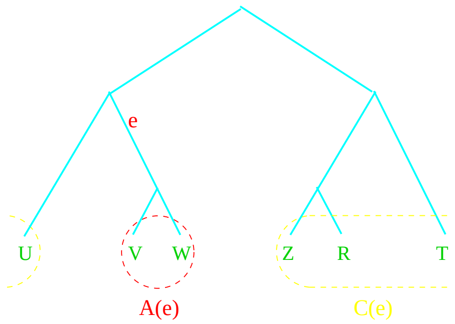
Tree Splits: Example



V,W	V,W	U,V
U,V,W	U,V,W	X,W
X,Y	X,Y	X,Y,W
U,V		
X,W		
X,Y,W		

Tripartitions: Illustration

Edge *e* defines a bipartition of the set of taxa.



Edge *e* defines a tripartition of the set of taxa.

Tripartitions: Definitions

For a network edge *e*, define

- ***A(e)*: taxa reachable from root *only* by using *e***
- ***B(e)*: taxa reachable from root *both* by using *e* and by not using *e***
- ***C(e)*: taxa unreachable from root by using *e***

Define $e_1 \equiv e_2$ to mean

$A(e_1) = A(e_2)$ and $B(e_1) = B(e_2)$ and $C(e_1) = C(e_2)$

Tripartitions: An Error Measure

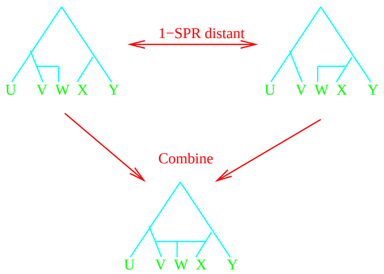
Given a model network N and an inferred network N', we define

- $FP(N, N') = |\{e' \in N' : \nexists e \in N,\ s.t.\ e \equiv e'\}|$
- $FN(N, N') = |\{e \in N : \nexists e' \in N',\ s.t.\ e \equiv e'\}|$

This is exactly the definition of the Robinson-Foulds metric—we simply generalized the notion of equivalence of edges through the tripartitions.

Wayne Maddison's Observation

Two gene trees that are related by one reticulation event are 1-SPR distant.



SPR (Subtree Prune and Regraft) is a local change operator for trees: disconnect a subtree and reattach it somewhere else in the tree.

Maddison's Algorithm

- **Construct gene trees T_1 and T_2**
- **Compute $d_{rSPR}(T_1, T_2)$, the number of SPR operations needed to transform T_1 into T_2.**
  - **If $d_{rSPR}(T_1, T_2) = 0$, return a tree.**
  - **If $d_{rSPR}(T_1, T_2) = 1$, return a network .**
  - **Otherwise, return nothing.**

Maddison's Algorithm: Challenges

- Computational:
  - How to compute the SPR distance between two trees efficiently.
- Systematic:
  - How to infer the two gene trees with no topological error.

Computational Challenge: Approach

We generalized Maddison's algorithm to obtain a new network reconstruction algorithm.

- It runs fast (in polynomial-time).
- It works for any fixed number of reticulations.
- It works for a class of constrained network topologies (galled trees).

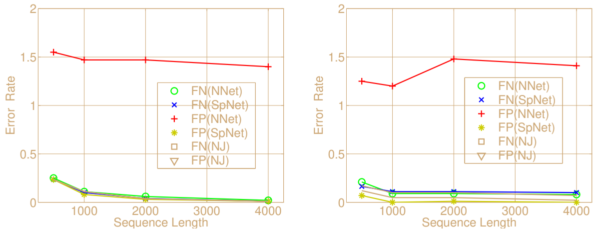
Systematic Challenge: Approach

How to infer the gene trees with no topological error?

- The problem is *positive* errors —unresolved edges (polytomies) are acceptable in the input.
- We use the consensus of sets of "good" gene trees.
- Experimental studies show significant improvements over using the "best" gene tree.

Experimental Results

Simplest case: tree vs. network with one hybridization.
Our method (SpNet), NeighborNet (NNet), Neighbor-Joining (NJ).
20 taxa, birth-death generation with 0.1 scaling and 2.0 deviation
Hybrid is a leaf with parents selected at random, using bias toward short tree distances.

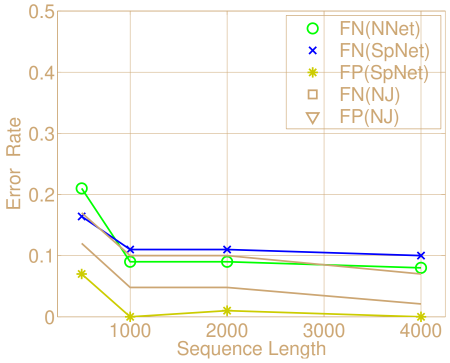


Tree Network

FP/FN rates: splits measure normalized by the number of splits in the model network.

Experimental Results (cont'd)

A more detailed look at relative error again using splits-based FP and FN.



Conclusions

- **Reconstructing networks is necessary when hybridization took place.**
- **Data preparation is crucial.**
- **We gave metrics with which to compare networks and thus evaluate reconstructions.**
- **We wrote network generators that produce good simulations, enabling us to test the accuracy of reconstruction algorithms.**
- **We proposed a reconstruction algorithm that works well for small numbers of reticulations and for special networks.**