Parsimonious Reconstruction of Ancestral Networks

Rob Patro, Emre Sefer, Justin Malin, Guillaume Marçais, Saket Navlakha, Carl Kingsford
Center for Bioinformatics and Computational Biology
University of Maryland
September 7, 2011
Ancestral Network Reconstruction

What?
Reconstruct the biological networks — regulatory, protein interaction or signaling pathways — of ancestral species

Why?

▸ Study the evolution of functional modules
▸ Learn what interactions are conserved
▸ Understand robustness & evolvability of biological networks
▸ Improve network-based alignment & phylogeny
Related Work

Reversing Network Growth:

Gibson and Goldberg (2009) – Multiple networks, not parsimony or ML
Navlakha and Kingsford (2011) – Single network, greedy model reversal

Ancestor Reconstruction (Maximum Likelihood, require total ordering):

Pinney et al. (2007)
Dutkowski and Tiuryn (2007)
Zhang and Moret (2008/10) – Used to improve regulatory inference

Metabolic Network Reconstruction:

Mithani et al. (2009) – Fixed node set; Gibbs sampling
Represent Network Evolution Histories

Leaf nodes exist in the extant network

Duplication tree specifies (partial) time constraints
  Child nodes exist after their ancestors

Edges between leaf nodes represent extant interactions

How do we encode ancestral interactions?
Encoding Ancestral Interactions

Assume a duplicate inherits its parents interactions.

Non-tree edges between ancestral nodes show how interactions flip on and off.
Encoding Ancestral Interactions

Assume a duplicate *inherits* its parents interactions.

Non-tree edges between ancestral nodes show how interactions *flip* on and off.
Assume a duplicate inherits its parents interactions.

Non-tree edges between ancestral nodes show how interactions flip on and off.

A set of flips that reconstructs the extant networks encodes a possible history of interaction gain and loss.
For any pair \((u, v)\) of nodes in the trees and paths \(p_u\) and \(p_v\) from \(u\) and \(v\) to their (possibly distinct) roots, the parity of flips between these paths encodes the state of the inferred edge.

Even \(\iff\) no edge, odd \(\iff\) edge
Not all sets of flips (histories) are **valid**

1-blocking loop

A ceases to exist here, after it duplicates

2-blocking loop

The duplication of A depends on the duplication of B and vice-versa

3-blocking loop

Blocking loops imply that the duplication events can't be consistently ordered while respecting the inferred interactions

A history $H$ is valid $\iff$ it contains no **blocking loops**
**Given:** a duplication forest $F$ and extant networks $G_1$ and $G_2$

**Find:** $H$ — a valid interaction history **reconstructing** $G_1$ and $G_2$, with a minimum cost set of edge flips (i.e. the most **parsimonious** solution).

Despite the exponential number of flip encodings constructing $G_1$ and $G_2$, we can discover a maximally parsimonious set of flips in $O(N^2)$ time.

### Duplication forest:
- Trees explain node duplication and node loss
- Leaves in extant networks, internal nodes in ancestors

### Interaction encoding:
- Non-tree edges represent interactions
- Edge gain/loss affects descendants
Basic idea: Recurse down the tree, finding the minimum cost set of edge flips that construct the extant networks.

At each internal node, decide:

Is it better (lower cost) to add an edge here or separately in subtrees?

We avoid 2-blocking loops by design.

Algorithm recurses into either the left or right subtree; never both simultaneously.
Handling Multiple Graphs

To infer the ancestral interactions using data from multiple graphs:
Lower cost to add an interaction in the ancestor or separately in the extant species?

Same as single-graph DP step, except don’t consider flips between species
Breaking Blocking Loops

Blocking loops of order $\geq 3$ handled post-hoc

If there are no blocking loops, we’ve found the optimal solution

\textbf{while} any blocking loop $\ell$ exists:

- $e = \text{some edge of } \ell$
- Forbid $e$
- Re-run the dynamic program

Gives us an upper bound on $\Delta(OPT)$

Loop-free solution is at least as costly as initial (loopy) solution
Benefits of Our Approach

- Can encode directed & undirected networks
  PPI and regulatory networks, signaling pathways
- Can encode networks both with and without self-loops
- Does not require branch lengths (total ordering of duplications)
- Can handle asymmetric edge creation and deletion costs
Consider 3 models to generate synthetic regulatory networks

1) Foster, Kauffman, and Socolar 2006:
   Based on node duplication
   In & Out edges removed probabilistically after duplication
   Nodes lost only when they have no incident edges

2a) Degree-independent variant
2b) Degree-dependent variant

General model:
   Arbitrary edge gain, loss
   Node duplication
   Arbitrary node loss

Compute F1-Score over 100 trials for each choice of parameters
Degree-dependent model (2b)
Summary of Performance on Synthetic Data

Performance is generally good.

Arbitrary node loss has the largest single effect:

This effect can be mitigated by considering more extant species.

Blocking loops of size $\geq 3$ are rare in practice:

Occurred in $< 2\%$ of all of our test cases.

Even when they occur, often find a loop-free sol. of the same cost.
Real bZIP PPI

bZIP PPI analyzed in the work of Pinney et al. (PNAS 2007)

“Ground truth”: ancestral interactions predicted using sequence reconstruction of ancestral Teleost network:

<table>
<thead>
<tr>
<th></th>
<th>Pinney et al. Maximum Likelihood</th>
<th>Our algorithm Parsimony</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.68</td>
<td>0.78</td>
</tr>
<tr>
<td>Recall</td>
<td>0.88</td>
<td>0.90</td>
</tr>
<tr>
<td>F1-Score</td>
<td>0.77</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Simple extension of our algorithm to arbitrary # of extant species
Comparison of Inferred Edges

Most predictions are the same
We make fewer total predictions:
But more of them are correct
Consider a larger space of histories
Not constrained by edge lengths
Conclusion & Future Work

Parsimony-based reconstruction performs well
On both real & synthetic data

Dynamic programming solution efficient & accurate
Doesn’t require phylogenetic branch lengths

Future Work:

- Room to improve both sensitivity & specificity
- Study the effect of noise
- Improve uncertain duplication histories (tree inference)
- How many (near) optimal solutions are there, how do they differ?
- Is avoiding general (i.e. $k \geq 3$) blocking-loops $\mathcal{NP}$-hard?
Thanks

Grants:

\{\text{EF-0849899, IIS-0812111, CCF-1053918}\}
\{\text{1R21AI085376, R01HG002945}\}
\{\text{2008-04049, 2010-15739-01}\}

People:

Emre Sefer  Justin Malin  Guillaume Marçais
Saket Navlakha  Carl Kingsford

Darya Filippova  Geet Duggal
Duplication History Framework