Visualization of Barrier- Tree- Sequences

Christian Heine and Gerik Scheuermann

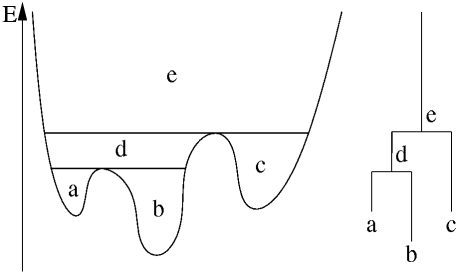
Image and Signal Processing Group
Department of Computer Science
University of Leipzig

EMBIO Project Meeting
2006-05-23

Barrier Trees

Barrier Trees

- use as topological simplification of folding landscapes
- present a useful reduction of the conformational space for kinetic folding



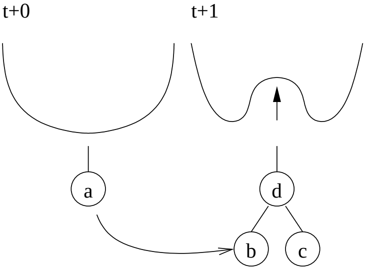
Barrier Tree Sequences

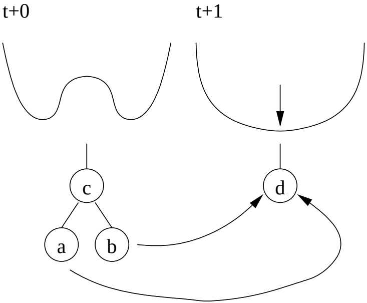
RNA molecules grow while folding

- assumed to have a strong impact on the native spatial structure of short lived RNA molecules
- for each molecule length there is a folding landscape, these are correlated
- barrier trees are correlated as well, it makes sense to identify vertices among them

Possible changes implying operations on the barrier trees

- local minimum refolds to local minimum
- more than one local minimum refold to the same local minimum (merge)
- new local minima are created
- local minima might also disappear from the set of relevant local minima





Ideas

Goal

- visualization of changing folding landscapes using barrier trees

Problems

- optimal visualization?
- complex changes in leaves
- identification of barriers

Possible Solutions

- **Barrier Tree animation**
- **Supergraph**
- Barrier Tree time-landscape

Barrier Tree Animation

. . . is a *dynamic graph drawing* problem

- it is not sufficient to layout and show each Barrier Tree independently
- in general, static aesthetic criteria (i.e. number of crossings, distribution of vertices and edges, etc.) fight dynamic aesthetic criteria (preservation of the *mental map*)
- there are only very specific or very general approaches

foresight layout with tolerance (Diehl and Görg '02)

- general approach for offline dynamic graph drawing problems
- blend between static and dynamic aesthetic criteria
- reduces dynamic graph drawing problem to a static one
- algorithm:
    - create a supergraph, i.e. a graph that contains all subgraphs
    - determine layout of the supergraph
    - determine layout of the subgraphs based on the layout of the supergraph, but make small improvements for static aestetic criteria
    - present the subgraphs and transitions between them (e.g. morphing)

Supergraph Construction

Problems

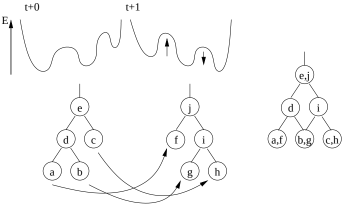
- complex leaf mapping schemes
- correspondence of barriers between successive barrier trees is not given, but must be inferred by graph theoretic properties of the trees

Supergraph Construction

- ignores energy (height) of the vertices
- preserves topological properties of the Barrier Trees
- is quite fast ($O(n \log n)$)

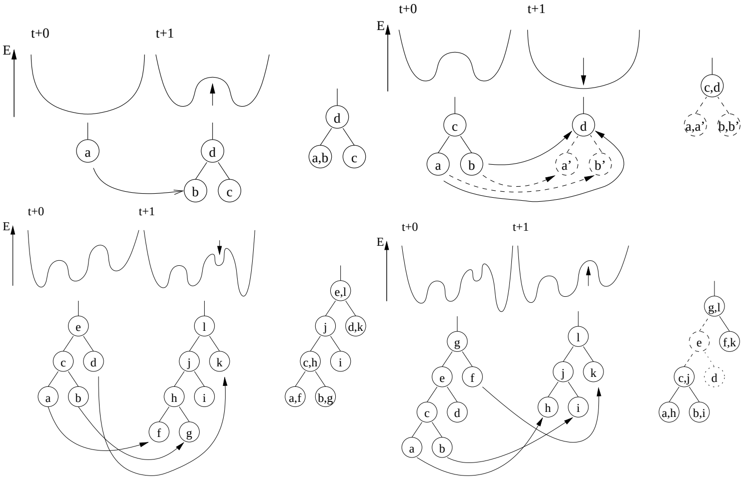
Algorithm (ideas)

- construct supergraph iteratively, i.e. add one tree at a time
- identify corresponding barriers using the set of leaves (local minima), that can be reached by descending in the Barrier Tree (folding landscape)

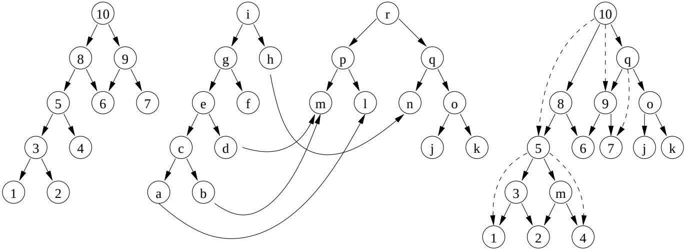


- resulting supergraph will be a DAG (directed acyclic graph)
- use previous supergraph structure to quickly determine sets of leaves

Supergraph Construction (example)



Supergraph Construction (example)



supergraph n tree n tree $n+1$ supergraph $n+1$

Supergraph Layout

uses the *dot* algorithm (Gansner et al. '93)

- part of popular GraphViz package
- used for static layout of DAGs
- best(?) heuristics for DAG specific layout problems

Algorithm

- assign vertices to layers
- find an order of vertices for each layer, that minimizes edge crossings
- assign coordinates to each vertex minimizing edge length but preserving the order

Modifications for dynamic case

- only minimize important edge crossings, weight of an edge crossing
- calculate only horizontal position, multiple strategies

Subgraph Presentation

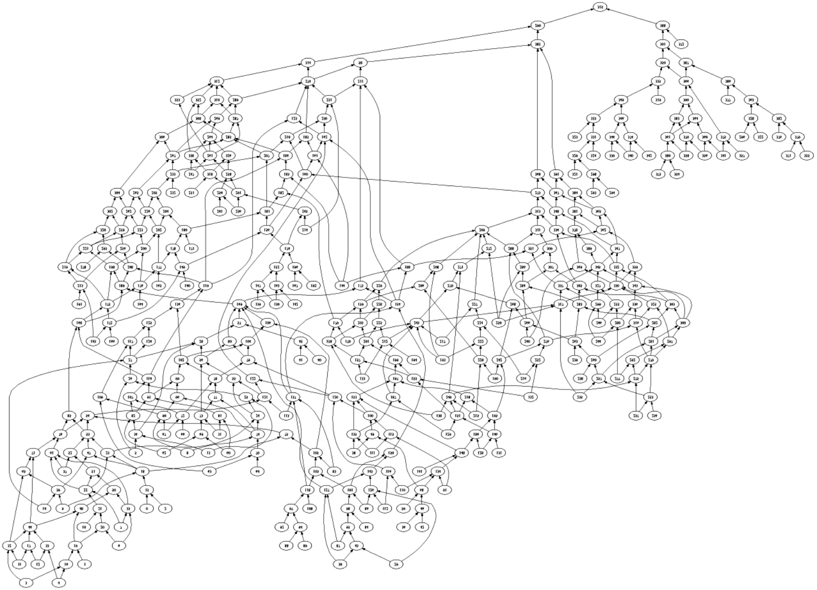
Subgraph Layout

- copy horizontal position from supergraph layout
- use the energy of a vertex as the vertical position
- follow the energy tree style, i.e. orthogonal edges
- edge crossings in subtrees can not be avoided

Transition between Subgraphs

- fade-in of created vertices
- fade-out of deleted vertices
- movement of vertices that change their energy
- edges change according to the vertices they connect

Results



Conclusions and Future Work

it works but. . .

- Barrier Tree sequences are highly dynamic
- resulting supergraph cannot reuse much information
- lots of edges and edge crossings

Solution: preprocessing

- reduction of barriers with small energy difference
- reduction of leaves of high energy

open questions

- optimal supergraph construction strategy?
  alternatives:
  - minimum graphtheoretic supergraph
  - maximum agreement tree

alternative approaches

- Barrier Tree time-landscape