RNA folding kinetics

Marcel Kucharík

Institute for Theoretical Chemistry
University of Vienna

Bled, February 18th, 2012

Notation

We will use following notation further:

- *structure* – secondary structure of certain RNA sequence (often characterized with bracket notation `"..(...).."`)
  - number of different structures grows exponentially with length of sequence
- *structure distribution* – distribution of different structures belonging to single sequence
- *(RNA / folding) landscape* – structures connected with neighboring relation (move set) and evaluated by energy function
  - configuration space – all valid structures (for one sequence)
  - move set – open, close, (shift) base pair
  - energy function – function assigning energy to each structure

Folding kinetics – motivation

To understand better processes guided by RNA, equilibrium distribution is not enough to compute. Because:

- **time-scale** (some sequences will not achieve equilibrium in their lifetime)
- predicted **structure distribution changes over time** – this greatly influences the behavior and function of the RNA
- folding begins from one side of sequence – **co-transcriptional folding**

Input: sequence

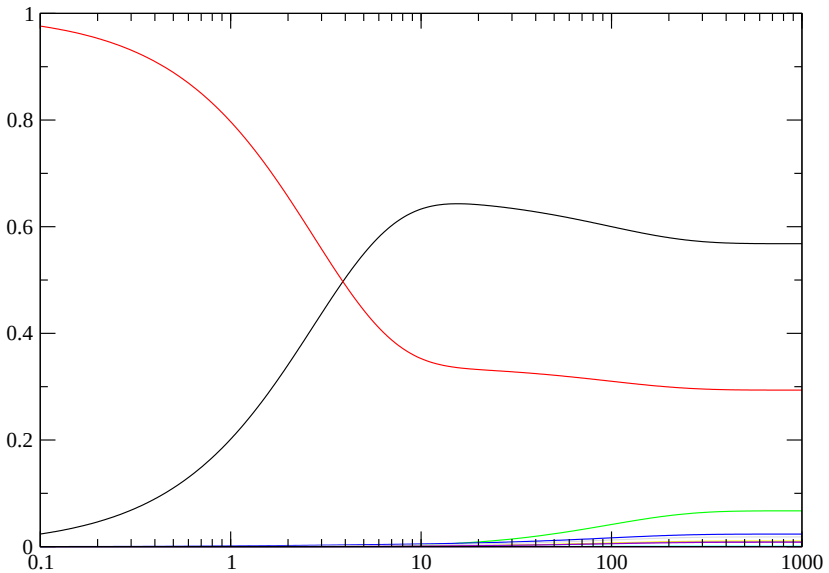
Intermediate: sequence, its structures, rates between these structures

```
   ACGAUCACGACGAGACGCGAAGAGAUCAGAGCAUACGACAGCAG
1  ..((((....((.....)).....))))...((.........)).   -3.20   0  12.20
2  ..((((.((.((...)))).....))))...((.........)).   -2.70   1   0.80
3  ....((.((.....)).))...........((.........)).   -1.90   1   3.00
4  .......((.((...)))).............((.........)).   -1.00   3   0.90
5  .((.((.....))..))..............((.........)).   -0.70   3   1.40
6  .((.((.......))))..............((.........)).   -0.70   1   1.90
7  ...............................((.........)).   -0.50   4   0.30
8  ..........((....((.............))...)).......   -0.20   3   0.20
9  .((.((.......))..))............((.........)).    0.70   1   0.50
10 .((((...........((.............)))).)).......    2.90   1   0.20
```

rates1.out

0.8765	0.1246	1.62e-05	0	0	0	0.001584	0	0	0
0.2411	1.125	0	0.001576	0	0	6.412e-08	0	0	0
0.000137	0	0.8239	0.0004405	0	0	0.06092	0	0	0
0	0.01963	0.001255	1.556	0	0	0.2158	0	0	0
0	0	0	0	1.009	0.03932	0.1536	0.02695	0	0
0	0	0	0	0.04407	0.8031	0.04327	0.01143	0.0355	0
0.04916	1.029e-06	0.2236	0.2779	0.08921	0.02242	2.182	0.1779	0.02164	0.0002229
0	0	0	0	0.03479	0.01317	0.3955	1.658	0	0.001656
0	0	0	0	0	0.3778	0.4448	0	0.6279	0
0	0	0	0	0	0	0.2248	0.7511	0	0.1028

Output: graph distribution of each structure over time



Approaches to folding kinetics

(How to get from "Intermediate" to "Output")

Simulation:

- Monte Carlo simulation on landscape
- simulation of a lot of structures gives us statistical information about refolding paths and distributions
- slow and probabilistic (lot of trajectories have to be collected to have reasonable statistics)

Markov process analysis:

- structures can be viewed as states in Markov process
- need to obtain rates (speed of change) between structures from landscape properties – different methods
- gives exact probability $P_x(t)$ of observing structure x at time t
- still slow when number of states is big

Whole Markov process is defined by initial distribution and rates, which forms infinitesimal generator matrix $Q = \{q_{ij}\}_{ij}$, where q_{ij} is rate from i to j and $q_{ii} = -\sum_{j \neq i} q_{ij}$

Change in probability distribution $P(t) = \{p_0(t), \ldots, p_n(t)\}$ in states is guided by "Kolmogorov's equations":

$$\frac{dp_i(t)}{dt} = \sum_{j \neq i} [p_j(t) q_{ji} - p_i(t) q_{ij}] = \sum_j p_j(t) q_{ji}$$

or, using definition of Q:

$$\frac{dP(t)}{dt} = P(t)Q$$

So the solution is:

$$P(t) = P(0)e^{Qt}$$

Problems: (treekin)

- Q is $n \times n$ in size where n is number of structures – this should be reasonably small ($n \approx 1000$ is computed about 40 secs.)
- computation of e^{Qt}
  - slow
  - efficient algorithms only for symmetric matrices
    - done in eigenspace of Q – need to obtain all eigenvectors of Q
    - cannot symmetrize matrix with one or more absorbing states
- numerical instability (in decomposition of Q)

How to get from "Input" to "Intermediate"

All structures:

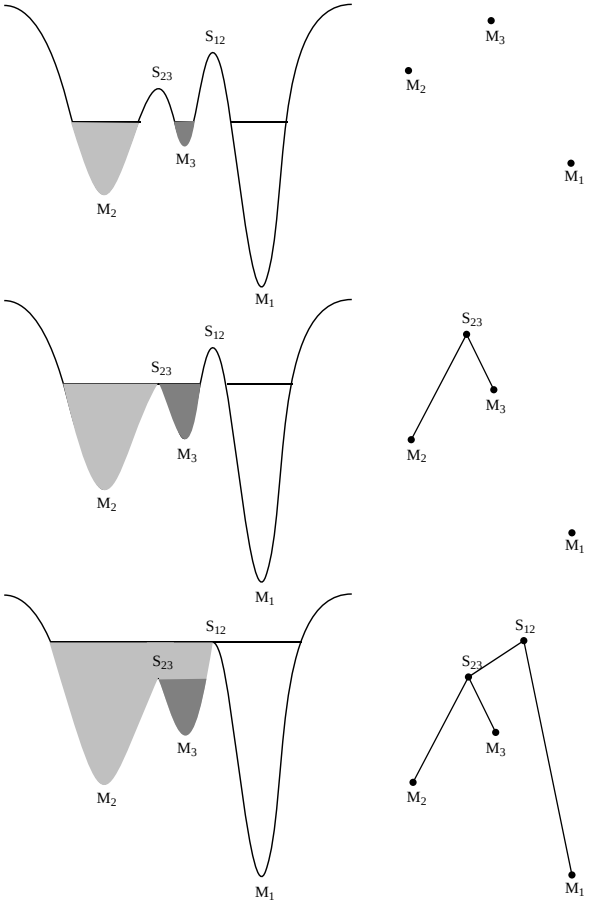
- unfeasible for normal RNA – number of structures grows exponentially
- rates are obtained easily from energy barriers between each two structures

Solution: **Coarse graining** = fuse bunch of structures (microstates) to one macrostate and compute only with that "bag" of structures as one

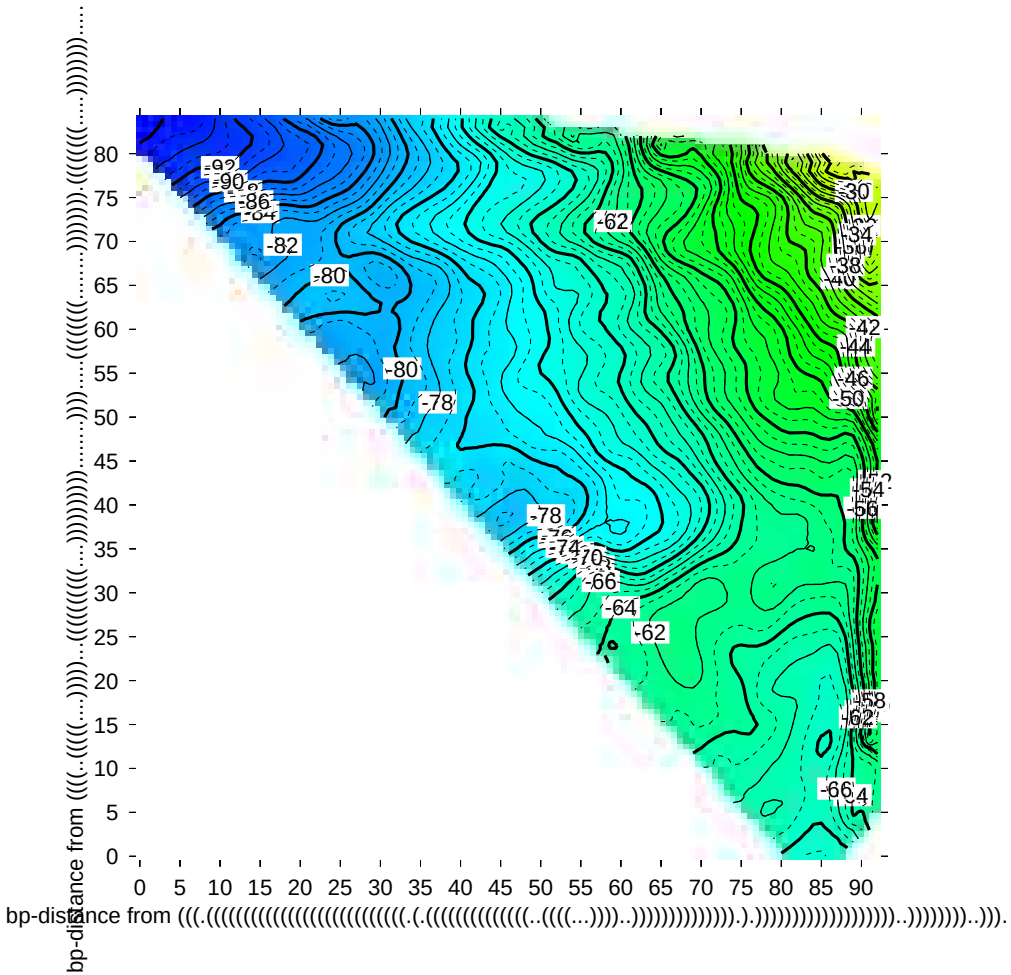
Coarse graining:

- reduces state space (to make time and memory requirements feasible, to have better insight, . . .)
- good CG – similar microstates are in one macrostate
- problem: how to get rates between macrostates
- examples of macrostates:
  - **local minima basins** – structures, whose gradient walk ends in same minimum belong to one macrostate
  - **equal base pair distance** – 2 reference structures are chosen, macrostates are then defined by distances from these 2 structures

Different methods for coarse graining:



(a) Barriers – local minima search

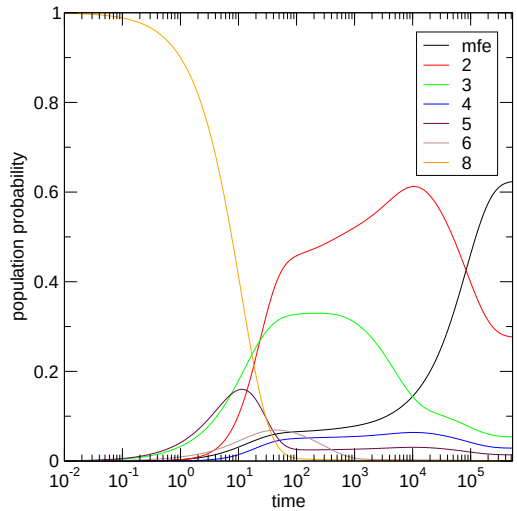


(b) RNA2Dfold – base pair distances

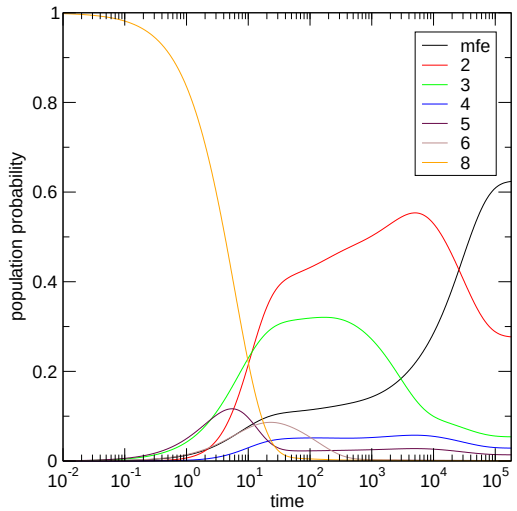
Rates generation:

- Arhenius kinetics: $r_{xy} = e^{-\frac{E(barr_{xy})-E(x)}{kT}}$
- recomputation from microrates (`bariers`)
- sampling + recomputation (`RNA2Dfold`)
- ...

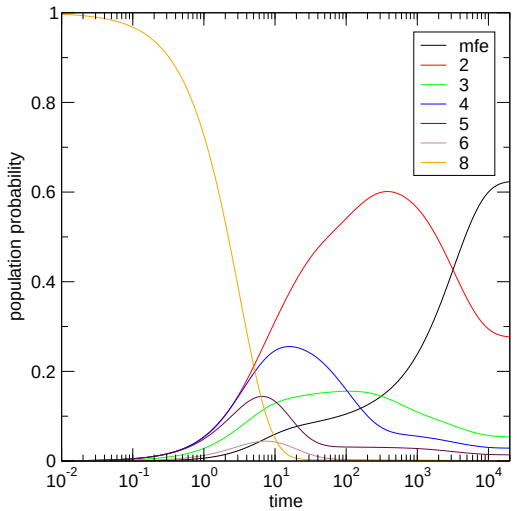
Comparison for different rates generation:



(a) No coarse graining



(b) Rates recomputed from microrates (barriers)



(c) Rates approximated from energy barriers (barriers)

Main problem of current local minima approach

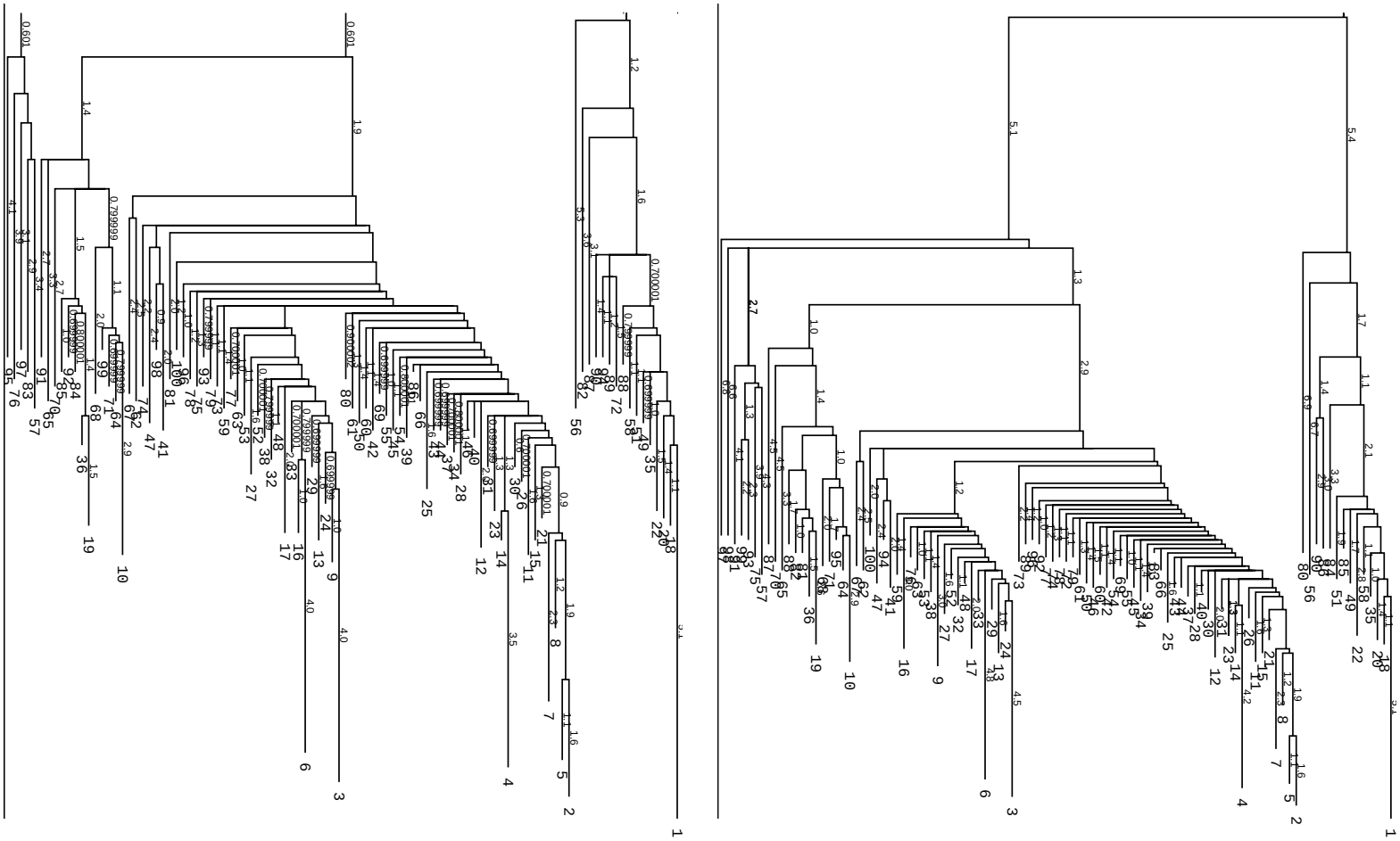
Pipeline: `RNAsubopt -e | barriers | treekin`

- `barriers` program **cannot compute minima and rates for longer sequences (more than 100nt)** due to exponential increase in the number of structures
  - compute only some local minima – especially those most energetically important

RNAlocmin

Pipeline: `RNAsubopt -p | RNAlocmin | treekin`

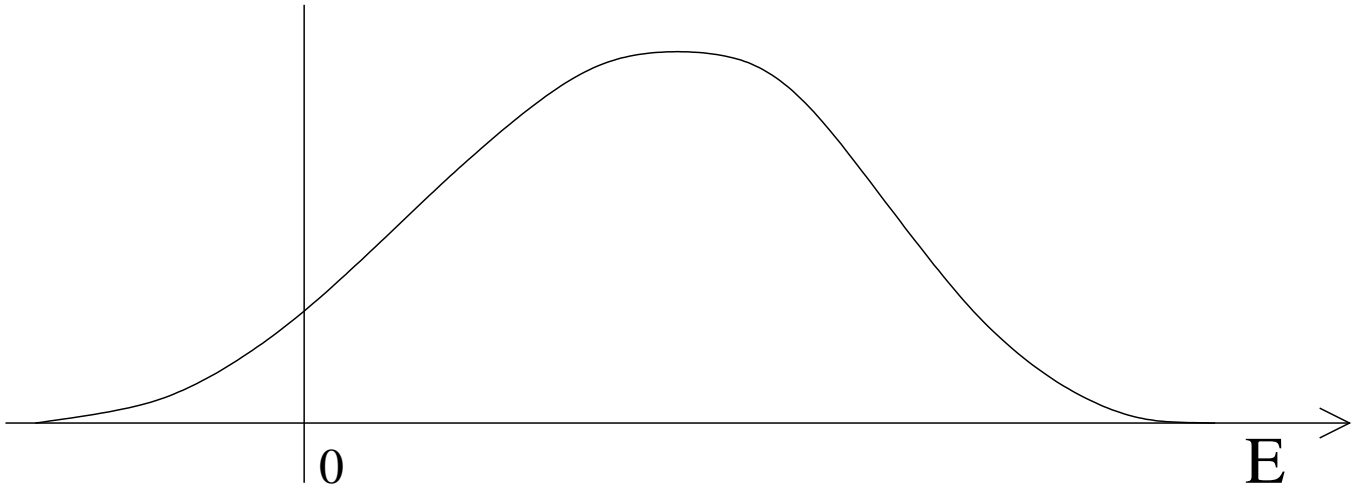
1. sample structures with low energy – stable structures (`RNAsubopt`)
2. simple deepest descend search of local minima from sampled structures
3. compute rates and approximate barrier tree from `findpath` algorithm – Arhenius kinetics (Vienna RNA package)
   - bottleneck of this approach – `findpath` is being run $O(n^2)$ times, where n is number of local minima found
   - 1000 minima would take approx. 15-20 minutes – still better than `barriers` in most cases
4. do Markov process analysis (`treekin`)



(a) Barriers (b) RNAlocmin

`barriers` program was stopped after 100 minima found, otherwise it would take $\approx$ 20 mins. to compute whole tree

Sampling



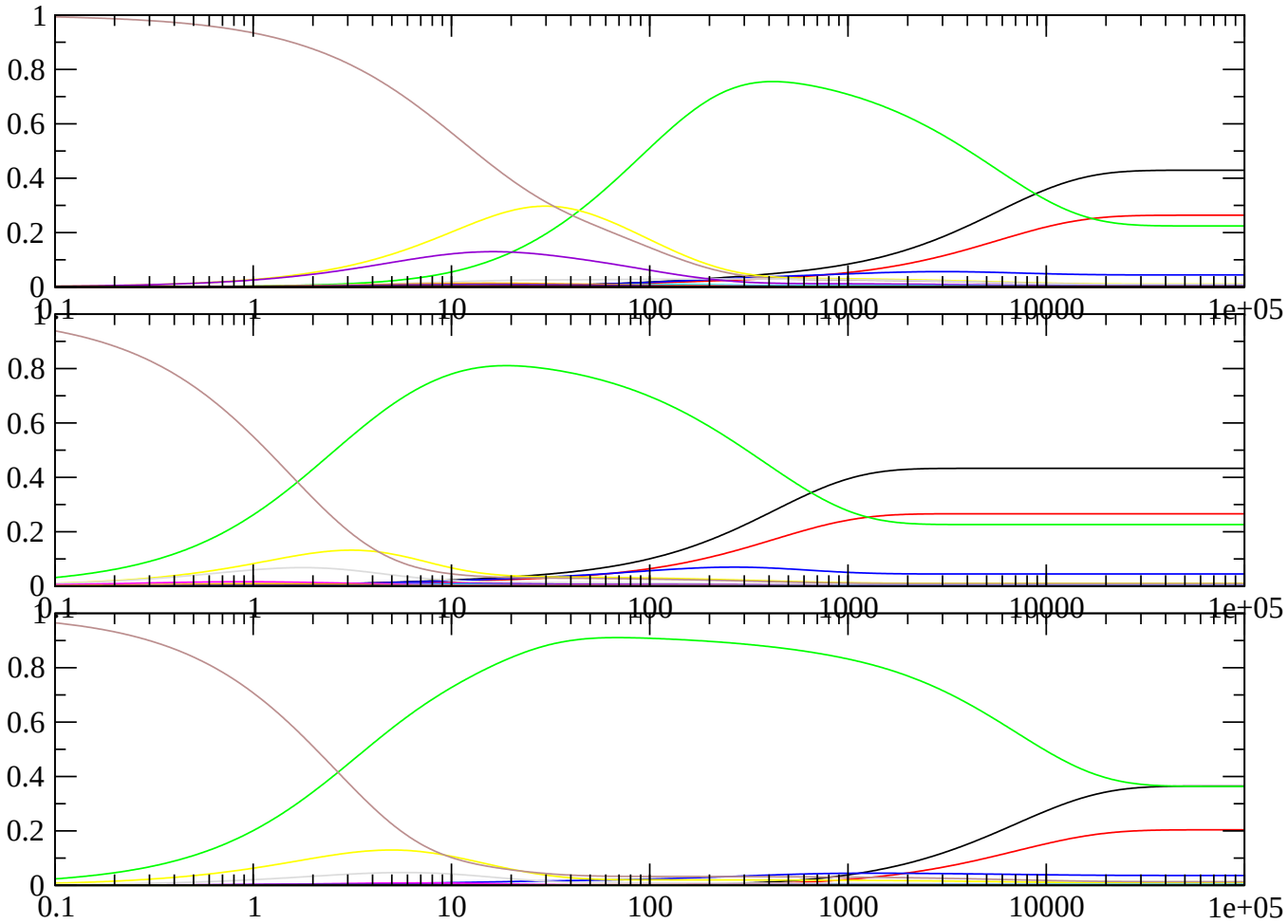
RNA structures distibution according to energy

- need to sample from left side of spectrum
- high entropy sampling – to capture a lot of minima
- **solution:** scale Boltzman parameters in sampling (`RNAsubopt` – thx to Ronny)

Good sampling is crucial:
(compared to `barriers` with threshold of 200 minima)

scale factor	1.0	1.5	1.8	2.0	2.2	2.5	3.0
# minima (40nt)	7	45	101	149	171	198	197
first lost (40nt)	3	16	41	81	125	190	156
# minima (89nt)	61	181	192	199	192	193	184
first lost (89nt)	28	93	73	165	73	43	34

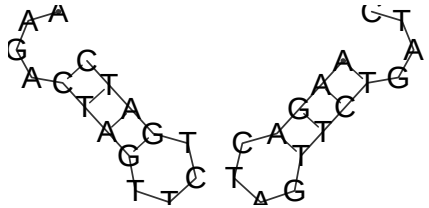
- sampled 10000 structures (very small sample) – `barriers` had to crawl through more than 92000 structures (89nt)
- missed minima are mainly minima with small basins



Top: RNAlocmin (rates from findpath)
Middle: Barriers (rates from energy barriers)
Bottom: Barriers (rates computed)

Ways to go

- how to efficiently incorporate co-transcriptional folding into these models
  - construct barrier trees for each sub-structure and then find transitions (`barmap` approach)
- coarse grain according to the abstract structures
  - does not matter how long is helix (or where it is)



  - different levels of the abstraction
  - how to get rates?
- other coarse graining

Acknowledgements

Many thanks to:

- Ivo
- Ronny
- all other great guys at TBI
- . . . and . . . wait for it . . .

You for attention!

Questions? Suggestions?