Supplementary Material:

Laederach, A. Shcherbakova, I. Liang, M. Brenowitz, M. and Altman, RB. Local kinetic measures of macromolecular structure reveal multiple parallel folding pathways for a large RNA molecule.

Description: Tables S1-S3 summarize the rate constants for the different kinetic model presented in the manuscript. The standard error on the kinetic parameters is reported in these tables as determined by bootstrapping. If no error is reported, then the kinetic parameter was bound to the value during least squares minimization. Units are s^{-1}.

Table S1: Rate constants and standard errors for Mg solution condition, *k*=4, *l*=3.

	U	I1	I2	I3	F
U	0	0.70±0.06	0.78±0.09	0.34±0.03	0.01±0.003
I1	0	0	0.23±0.04	0.31±0.02	0.00±0.005
I2	0	0.01±0.002	0	0.07±0.01	0.12±0.01
I3	0	0.01±0.002	0.01±0.001	0	0.001±0.000
F	0.00±0.005	0	0	0	0

Table S2: Rate constants and standard errors for Na solution condition, *k*=3, *l*=2

	U	I1	I2	F
U	0	37.5±2.8	70.1±6.7	20.2±1.9
I1	0	0	0.00±0.0002	1.7±0.11
I2	0	0.001±0.004	0	0.25±0.01
F	0.0001±0.00003	0	0	0

Table S3: Rate constants and standard errors for Mg solution condition, *k*=3, *l*=2

	U	I1	I2	F
U	0	0.79±0.08	0.35±0.03	0.002±0.001
I1	0	0	0.03±0.005	0.11±0.004
I2	0	0.001±0.0002	0	0.003±0.0004
F	0.0001±0.0001	0	0	0

Description: Tables of clustered sites of protection. Numbers correspond to central residue (Full L-21 *Tetrahymena termophila* numbering)

Mg^{2+} mediated folding, 1998 data set

Green Cluster
185,152,140,110,163,170,175
Red Cluster
46, 57, 83, 94, 342, 204,105,120, 25, 29
Blue Cluster
330,282,273,300

Na^{+} mediated folding

Yellow Cluster
359, 369
Magenta Cluster
204, 215, 225, 256, 266, 273, 279, 299
Grey Cluster
59, 82, 96, 106, 111, 122, 126, 140, 154, 318, 328

Mg^{2+} mediated folding, 2004 data set

Green Cluster
111,126,139,154,257
Red Cluster
48, 59, 82, 96,105,119,203,283,344
Blue Cluster
273,303
Cyan Cluster
163,180

Description: Details of Gap Statistic for determination of *k*.

The Gap Statistic is a formalism that analyzes the relative within cluster dispersion (W_k) as a function of *k* (the number of clusters) for clustering data. For normally distributed data, W_k will decrease monotonically with *k* as is illustrated by the blue data in Figure S1a. If the data is not randomly distributed, but rather can be clustered into tight groups with small within cluster dispersion, then for small values of *k* the average within cluster dispersion will decrease rapidly with increasing *k*. This is the case with time-resolved hydroxyl radical footprinting data, as is illustrated by the red data in Figure S1a.

The Gap Statistic compares the expected relative decrease in cluster dispersion (blue line Figure S1a, W_k^*) for normally distributed data as a function of *k* with the actual decrease for the experimental data (red line Figure S1a, W_k). This makes it possible to determine a lower bound for a value of *k* above which clustering becomes ill-defined for the experimental data. This difference is normalized relative to the variance in W_k^* for multiple repeats of the clustering (indicated by the blue error bars). This is the principle behind the Gap Statistic, illustrated in Figure S1b. In this case, the Gap statistic picks a value of *k*=4, as indicated by a red vertical line. The reason for this choice is that at *k*=4 the condition $Gap(k) \geq Gap(k+1) - s_{k+1}$ is satisfied. For values of *k*>4 the clustering of the experimental data behaves like normally distributed data.

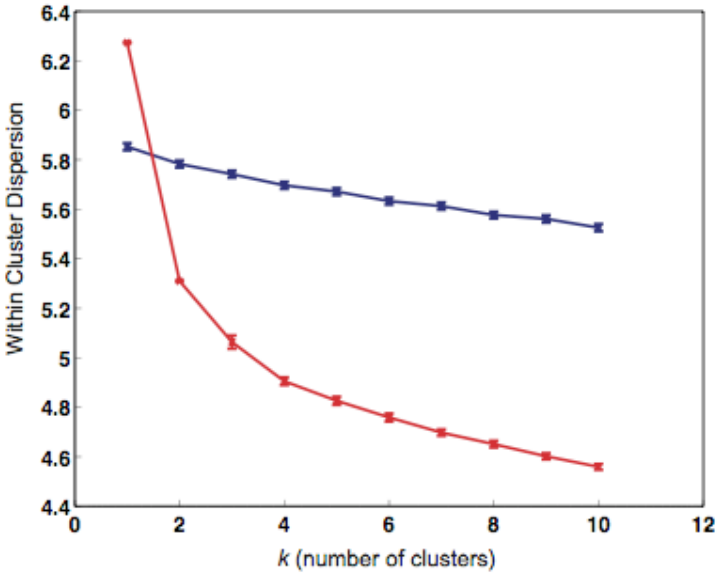
This formalism also allows us to estimate the relative signal to noise of the different data sets clustered in this manuscript on a cluster by cluster basis. The value of the *Gap(k)* parameter is a measure of relative tightness of the clusters compared to a random distribution. Therefore, larger values of *Gap(k)* indicate tighter clusters. As can be seen in Table S4, the more recent data sets (2004) have larger values of *Gap(k)*, indicating tighter clusters. This is a result of lower signal to noise in the data due to improved data collection methodology.

Table S4: Values of *Gap(k)* for the

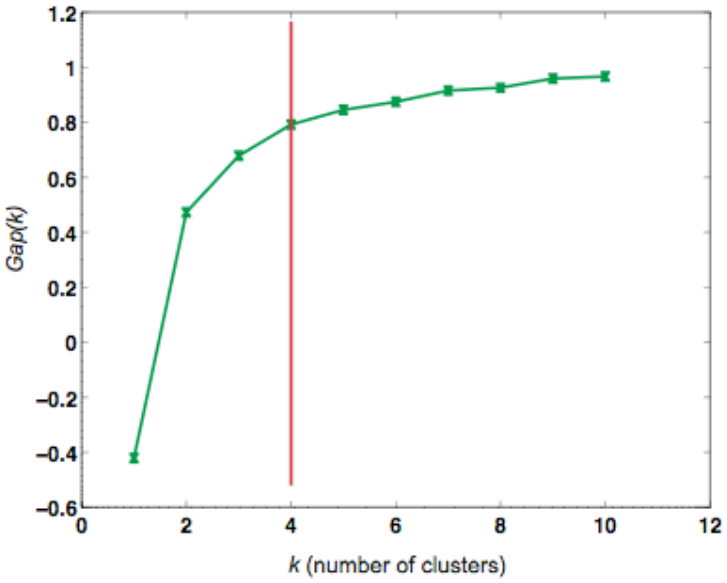
	Gap(k) 2004 Mg^{2+} data	*Gap(k)* 1998 Mg^{2+} data	*Gap(k)* 2004 Na^{2+} data
k=3	0.70±0.03	0.64±0.02	0.88±0.02
k=4	0.80±0.03	0.66±0.03	0.90±0.02
K=5	0.82±0.03	0.67±0.03	1.02±0.02

Description: Supplementary Material Figure 1: a.) Plot of the within cluster dispersion as a function of *k* for randomly distributed data (blue, W_k^*) and for the 2004 Mg^{2+} mediated folding data (red, W_k). Error bars represent variance when clustering is repeated 100 times. b.) Plot of *Gap(k)* (as defined in Equation 1 of the manuscript) as a function of *k* for the same data set. The vertical red line indicates the selection of *k*=4 as per the Gap statistic criterion.

a

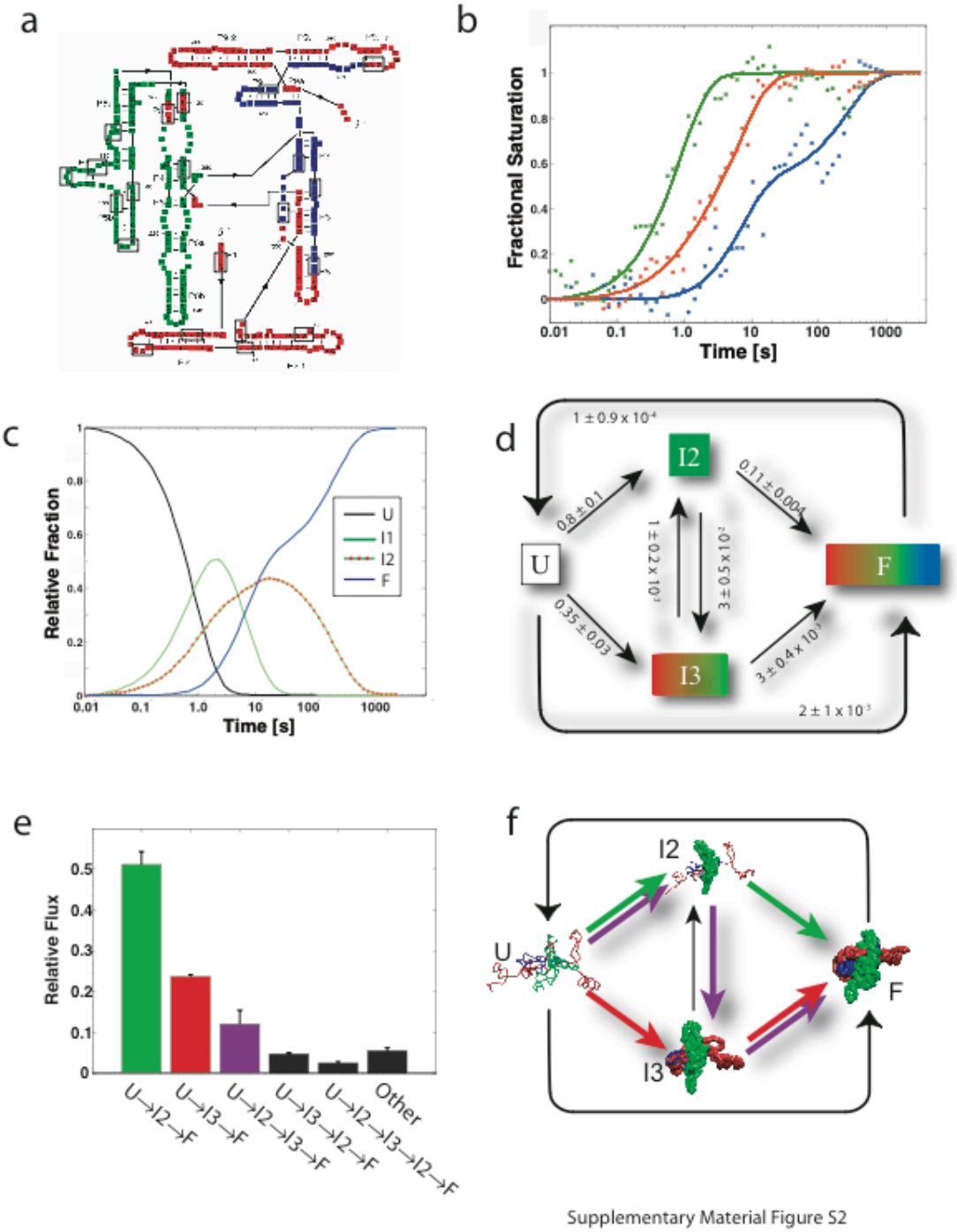


b



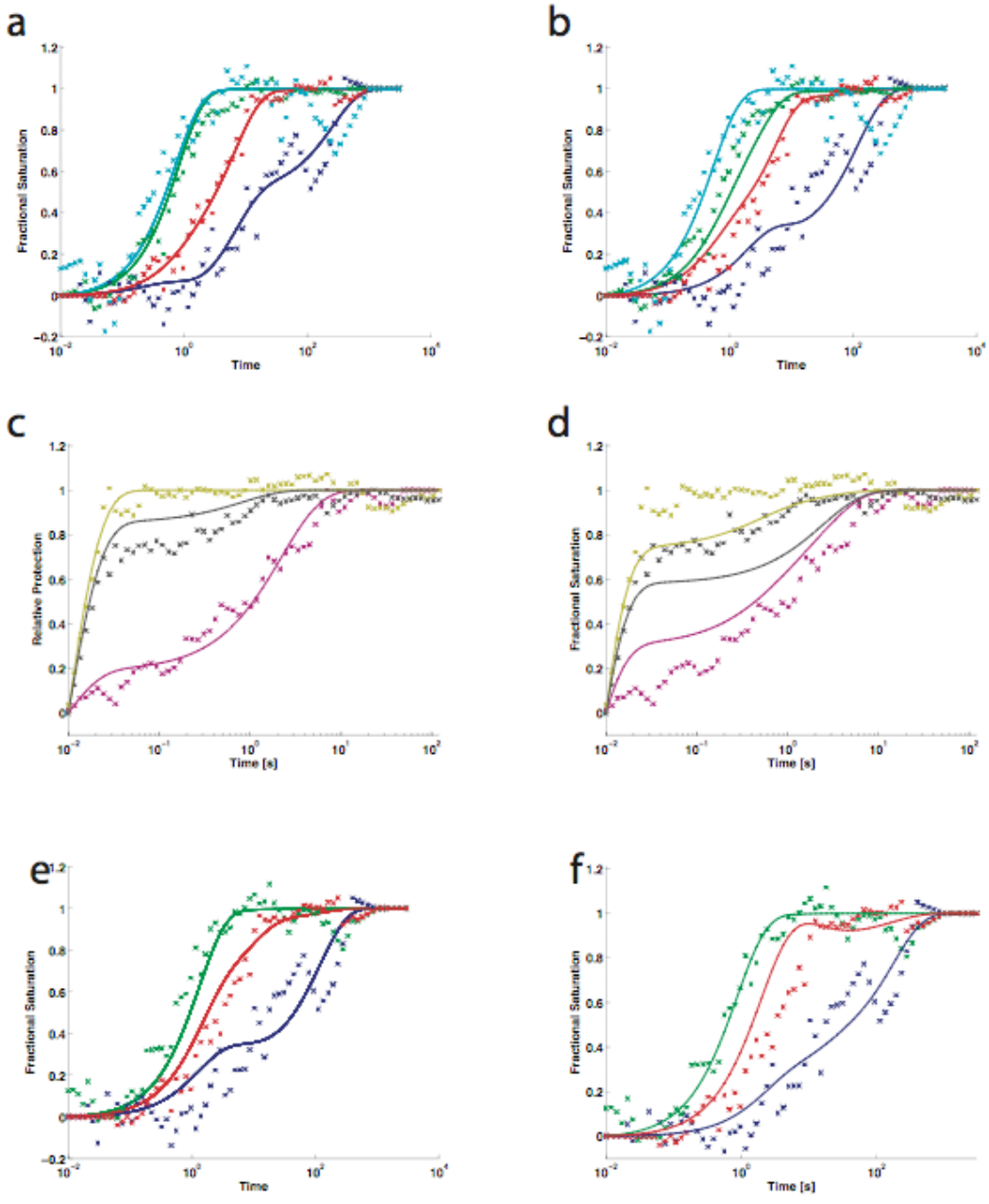
Supplementary Material Figure S1

Description: Supplementary Figure 2 Caption: Clustering and kinetic analysis of 1998 Mg^{2+} mediated folding data set with *k*=3. a) Colored secondary structure diagram representation of the L-21 *T. thermophila* group I intron. Colors represent regions of the molecule exhibiting similar time-progress as determined by k-means clustering of site-specific progress curves. Boxes indicate protection sites used. b) Best fitting kinetic model predictions (lines) to average time-progress curves (×). c) Time-evolution of the different species in solution. d) Best fitting kinetic model with rate constants given in s^{-1}. Reverse rates were constrained to zero as an equivalently good fit to the data was obtained with and without constraints. The I2 and I3 intermediates correspond structurally to I2 and I3 in Figure 3c. e) Summary histogram of the relative flux through the different folding possible folding pathways. f.) Cartoon representations of major flux through the different folding pathways of the ribozyme.



Supplementary Material Figure S2

Description Supplementary Material Figure S3 Caption. Plots of the best fits to the data for the second and third best models tested. These plots correspond to the RMSE calculations in Table 1 a.) and b.) are for the Mg^{2+} mediated folding reaction with k=4. c.) and d.) are for the Na^{+} mediated folding reaction, and e.) and f.) are for Mg^{2+} mediated folding but with k=3.



Supplementary Material Figure S3

Kinfold 1.0 Documentation

KinFold v1.0
Alain Laederach
Department of Genetics
Stanford University
February 2006

The software and data examples that accompany this documentation may be downloaded from http://simtk.org/home/KinFold

Introduction

This package contains a series of matlab scripts to analyze and model time-progress curves measured with local probes of structure. Local probes of macromolecular structure are measurements that are sensitive to the environment of a relatively small region within a macromolecule. These include, but are not limited to, NMR deuterium exchange and shift perturbation analysis, Fluorescence Resonance Energy Transfer (FRET), and RNA/DNA protein footprinting. The separate transitions reported by individual probes yield unique insight into folding intermediates. While simultaneous acquisition of many unique local transitions provides a cornucopia of information, creating an accurate global de-scription of folding that remains faithful to local details is very challenging.

The package requires Matlab which one can obtain from the Mathworks http://www.mathworks.com), although some of the scripts will also work with Octave (http://www.octave.org). This is not necessarily intended as "easy to use" software, but rather to provide the basic tools to carry out an analysis similar to the one reported in Laederach et al., "Local kinetic measures of macromolecular structure reveal partitioning among multiple parallel pathways from the earliest steps in the folding of a large RNA molecule," JMB 2006. This manuscript details the algorithms implemented in the accompanying code.

Any questions, bug reports maybe reported to Alain Laederach.
(alain@helix.stanford.edu)

Methods:

Below are function definitions and usage examples for the main scripts to carry out a complete analysis of a data set. By inputting the commands in matlab that are shown after >> you can get through the example data provided here. (The steps that require a supercomputer are noted.

Data entry:

```
function [time_bins,interp_data_ave,res_labels]=readxlsfootprintdata(filename);
```

This function will read in an plot data from an excel spreadsheet. The data is binned and vectorized for k-means clustering.

example:
```
>> [time_bins,interp_data_ave,res_labels]=readxlsfootprintdata('dataeg.xls');
```
Note the format of the data in the excel file dataeg.xls, it is important that this format is respected if you are inputting your own data.
The result will be (in this case) 21 figures showing the binned and extrapolated
data.

Gap Statistic:

The data can now be analyzed using the gap statistic as follows:

```
>> khat=compute_Gap(time_bins,interp_data_ave,[2 6],10,100)
```
This will take a couple of minutes to run, but will return a value of khat=3. This means that the Gap statistic has estimated that the data has three clusters
and this value should be used in the clustering, which is the next step. Note that for this and the next step you need to have the Statistics Toolbox installed. To check if you can type

```
>> kmeans
??? Error using ==> kmeans
At least two input arguments required.
```

If that is the error you get, then the Toolbox is installed, if you get a different error message, it will have instructions on how to install that toolbox.

If you do not want to install this toolbox or are using Octave you will need to skip the clustering step, I have stored the results of the clustering so for this demonstration you can continue.

Clustering:

Now we will cluster the data using kmeans clustering:

```
>> [IDX, C, SUMD, D] = kmeans(interp_data_ave',
3,'Distance','cityblock','Replicates',100,'EmptyAction','singleton')
```

Notice how the number of clusters is specified, it is the second argument of the
function.

You can see the results of the clustering using the following two commands:

```
>> load visualization.mat
>>
display_clusts(IDX,res_labels,C,exp(time_bins),1,interp_data_ave,imagex,offset,
residue_locations,1,2)
```

Note that this is specific to the L-21 tetrahymena ribozyme, as this function displays the data on the secondary structure.

Kinetic Model Fitting:

This step requires a supercomputer, but for the purposes of this example, it is possible to run the code like this:

```
>> search_kspace('17')
```

after a while you should get output like:

```
 Iteration  Func-count     f(x)          step          optimality   CG-
iterations
     0         13         39.4335                          2.54
```

This tells the code to test model number 17 (out of 28, given that k=3, and that the number of intermediates =2). The code is setup to try and optimize the model for approximately 15 hours of CPU time and this would need to be repeated for all models. This particular numerical solution is also the most computationally efficient, less efficient code is available upon request to Alain Laederach (alain@helix.stanford.edu) that can handle stiff systems. This is common of there are large differences (greater than 3 orders of magnitude) between the slowest and fastest forming sites).

Optimization visualization:

The results of a supercomputing run are stored in the analysis_all.mat file and can be visualized by typing the command:

```
>> make_plots_interest(1,1,4)
```

Best fits to the data are shown as well as the time-evolution of the different intermediates.

Note that the output values (mean_K) are the rate constants of the best fitting model with K(1,2) corresponding to the rate from U->I1, K(1,3)-> U->I2 etc.

Flux analysis

Flux analysis also requires a significant amount of time to compute accurately, but for the purposes of this demonstration the following commands can be issued:

```
>> get_fluxes_t(1)
```

This command should be repeated at least 100 times to get better sampling, each time incrementing the argument by one. (About 100 CPU hours)

Pathway Analysis:

This command analyzes the result of all the pathways data stored in the pathways
folder.

```
>> pathways=analyze_state_pathways;
```

Pathway Visualization:

The most common pathways can be vizualized by issuing the following command.

```
>> [clust_pathway,diff_pathway_vect]=pathway_analysis(pathway);
```

This will show a histogram of the fluxes through the major pathways.

The analysis presented above can be repeated with a different data set but will require the use of a super computer (between 200-300 nodes) for the computationally expensive steps. The scripts for generating massively parallel runs are nto included in this distribution as they are specific to the setup of the machine. For help with distributed computing setup, feel free to contact Alain Laederach (alain@helix.stanford.edu).