A collection of programs for one-dimensional Ising analysis of linear repeat proteins with point substitutions

Jacob D. Marold[1], Kevin Sforza[1,2], Kathryn Geiger-Schuller[1,3], Tural Aksel[1,4], Sean Klein[1,5], Mark Petersen[1], Ekaterina Poliakova-Georgantas[1,6], and Doug Barrick[1*]

[1]T.C. Jenkins Department of Biophysics and Program in Molecular Biophysics, Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218.

[2]Current address: US District Court for the District of Delaware, 844 N. King Street, Wilmington, DE.

[3]Current address: Broad Institute of Harvard and Massachusetts Institute of Technology, Cambridge, MA 02142.

[4]Current address: Department of Cellular and Molecular Pharmacology, University of California, San Francisco, 600 16th Street, San Francisco, CA 94143

[5]Current address: US Department of Health and Human Services, 200 Independence Ave. SW, Washington, DC 20201.

[6]Current address: University of Maryland, College Park, MD, 20742,

*Corresponding Author: barrick@jhu.edu

Key words: Ising model, repeat protein, folding, cooperativity, coupling.

Abstract

A collection of programs is presented to analyze the thermodynamics of folding of linear repeat proteins using a 1D Ising model to determine intrinsic folding and interfacial coupling free energies. Expressions for folding transitions are generated for a series of constructs with different repeat numbers and are globally fitted to transitions for these constructs. These programs are designed to analyze Ising parameters for capped homopolymeric consensus repeat constructs as well as heteropolymeric constructs that contain point substitutions, providing a rigorous framework for analysis of the effects of mutation on intrinsic and directional (i.e., N- versus C-terminal) interfacial coupling free-energies. A bootstrap analysis is provided to estimate parameter uncertainty as well as correlations among fitted parameters. Rigorous statistical analysis is essential for interpreting fits using the complex models required for Ising analysis of repeat proteins, especially heteropolymeric repeat proteins. Programs described here are available at https://github.com/barricklab-at-jhu/Ising_programs.

1. INTRODUCTION

One of the goals of protein folding studies is to quantify the contributions of specific structural features within the native state to the overall free energy of folding. As a result of the high level of cooperativity in protein folding, the relative contributions of such structural features cannot easily be determined because they are hidden in an all-or-none, or "two-state" transition. Though two-state behavior makes overall protein folding energetics easy to quantify, it prevents an energetic dissection of the whole into its parts.

Over the last 15 years, a class of proteins with rough translational symmetry, termed “linear repeat proteins”, has been recognized as having an architecture that permits quantification of local stabilities, long-range coupling energies, and cooperativity (Mello and Barrick, 2004; Kajander et al., 2005; Wetzel et al., 2008; Aksel et al., 2011; Marold et al., 2015; Geiger-Schuller and Barrick, 2016). Much like the helix-coil transitions of simple polypeptides, the unfolding of linear repeat proteins can be analyzed using a one-dimensional Ising model, where the overall folding free energy can be broken down into intrinsic folding energies of individual repeats (ΔG_i) and coupling energies between folded nearest-neighbor repeats ($\Delta G_{i-1,i}$).

To quantify cooperativity in linear repeat proteins, a series of equilibrium unfolding transitions (often from chemical denaturation) are obtained for proteins containing different numbers of repeats, and the transitions are globally fitted using a model that relates the extent of folding to ΔG_i and $\Delta G_{i-1,i}$. Varying the number of repeats is essential to resolve the values of ΔG_i and $\Delta G_{i-1,i}$ (Aksel and Barrick, 2009). Although this analysis is simplest when applied to arrays with identical repeat sequence, solubility considerations usually require sequence modification to one or both terminal repeats (Figure 1A). The effects of these terminal modifications on stability need to be accounted for by including intrinsic folding energy parameters for the substituted terminal repeat (ΔG_N and ΔG_C).

In addition, by combining point mutation and length variation, Ising analysis of linear repeat proteins can be extended to analyze the effects of sequence substitutions on the underlying parameters (Figure 1B). Although these sequence perturbations increase the complexity of the fitting model, Ising analysis of constructs containing sequence substitutions along with the capped homopolymer constructs in Figure 1A allows the energetics of specific structural features to be resolved into intrinsic and interfacial stabilities (Aksel et al., 2011; Cortajarena et al., 2011).

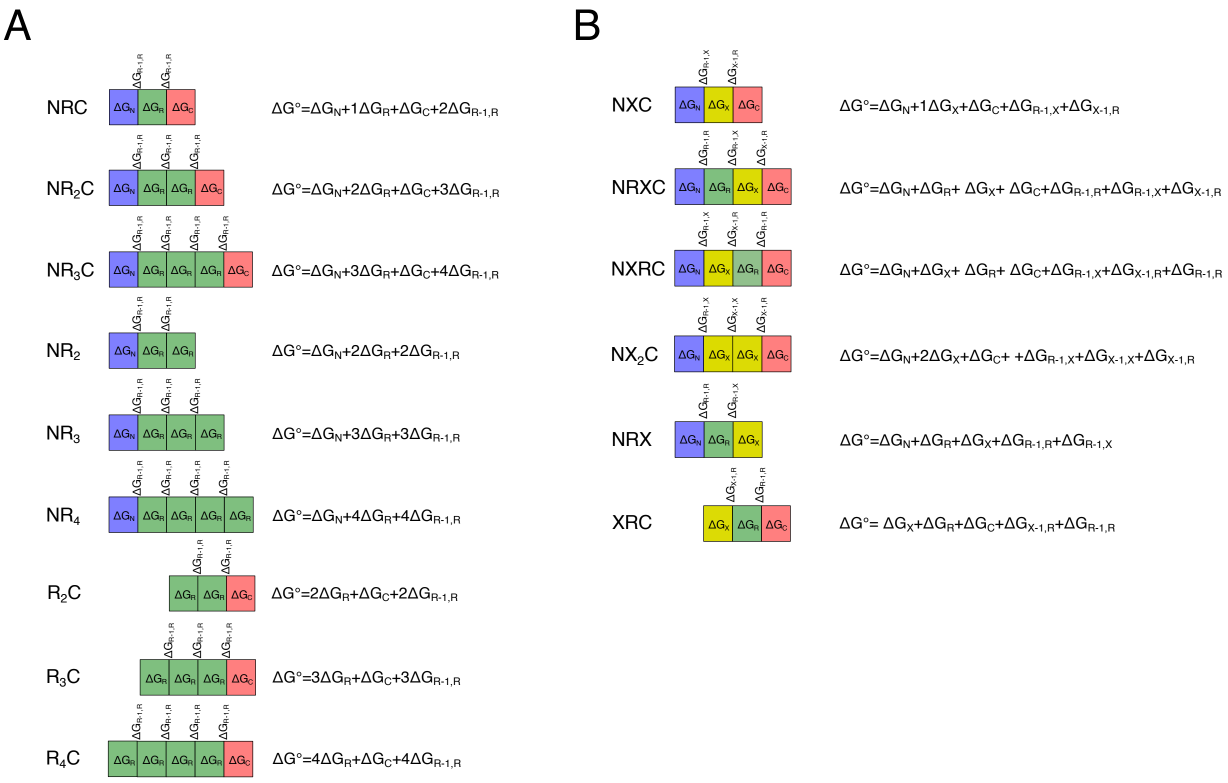

A
NRC $\Delta G°=\Delta G_N+1\Delta G_R+\Delta G_C+2\Delta G_{R-1,R}$
NR_2C $\Delta G°=\Delta G_N+2\Delta G_R+\Delta G_C+3\Delta G_{R-1,R}$
NR_3C $\Delta G°=\Delta G_N+3\Delta G_R+\Delta G_C+4\Delta G_{R-1,R}$
NR_2 $\Delta G°=\Delta G_N+2\Delta G_R+2\Delta G_{R-1,R}$
NR_3 $\Delta G°=\Delta G_N+3\Delta G_R+3\Delta G_{R-1,R}$
NR_4 $\Delta G°=\Delta G_N+4\Delta G_R+4\Delta G_{R-1,R}$
R_2C $\Delta G°=2\Delta G_R+\Delta G_C+2\Delta G_{R-1,R}$
R_3C $\Delta G°=3\Delta G_R+\Delta G_C+3\Delta G_{R-1,R}$
R_4C $\Delta G°=4\Delta G_R+\Delta G_C+4\Delta G_{R-1,R}$
B
NXC $\Delta G°=\Delta G_N+1\Delta G_X+\Delta G_C+\Delta G_{R-1,X}+\Delta G_{X-1,R}$
NRXC $\Delta G°=\Delta G_N+\Delta G_R+ \Delta G_X+ \Delta G_C+\Delta G_{R-1,R}+\Delta G_{R-1,X}+\Delta G_{X-1,R}$
NXRC $\Delta G°=\Delta G_N+\Delta G_X+ \Delta G_R+ \Delta G_C+\Delta G_{R-1,X}+\Delta G_{X-1,R}+\Delta G_{R-1,R}$
NX_2C $\Delta G°=\Delta G_N+2\Delta G_X+\Delta G_C+ +\Delta G_{R-1,X}+\Delta G_{X-1,X}+\Delta G_{X-1,R}$
NRX $\Delta G°=\Delta G_N+\Delta G_R+\Delta G_X+\Delta G_{R-1,R}+\Delta G_{R-1,X}$
XRC $\Delta G°= \Delta G_X+\Delta G_R+\Delta G_C+\Delta G_{X-1,R}+\Delta G_{R-1,R}$

Figure 1. Schematic of a series of tandem repeat constructs for Ising analysis. Each repeat is indicated by a rectangle. (A) A capped homopolymeric series. Identical repeats (R) are colored green; to maintain solubility, these repeats are flanked on the N- or C-terminus (or both) by polar capping repeats (N, blue and C, red, respectively). (B) A capped heteropolymeric series, in which repeats harboring a sequence substitution (X, yellow) are combined with R, N, and C repeats. Each repeat has an associated intrinsic folding energy (ΔG_i), depending on its type (N, R, X, or C). Folded repeats couple with their neighbors through an interaction energy ($\Delta G_{i-1,i}$) that is assumed to the same for NR, RR, and RC interfaces. The equations on the right give the free energy difference between the fully folded and fully unfolded states as a sum of these intrinsic and interfacial terms. Global Ising analysis of the series in (A) allows the N, R, and C intrinsic and interfacial parameters to be determined. By combining the series in (A) and (B), the effects of mutation (X) on Ising parameters can be determined (see Table 1).

Here we present a collection of programs that perform 1D-Ising analysis on repeat protein arrays. The programs return fitted parameter values for ΔG_i and $\Delta G_{i-1,i}$, and their denaturant dependences. In addition, bootstrap analysis (Efron, 1979; Johnson, 2008) is used to determine confidence intervals for each parameter and their correlations with each other. We use these programs to analyze a series of unfolding transitions of homopolymeric repeat arrays with modified caps, and to analyze a larger data set that includes point substitutions in internal repeats. Analysis of point substitutions allows the energetics associated with atomic-level structural features to be resolved into intrinsic and coupling energies. Moreover, by creating asymmetric interfaces, the coupling contributions of these structural features can be apportioned into the N- (*i-1, i*) and C-terminal (*i, i+1*) interfaces. This suite of programs, which is available on github (https://github.com/barricklab-at-jhu/Ising_programs) as a set of python programs and Jupyter notebooks, significantly extends an earlier analysis suite of programs (Lowe et al., 2018) by providing analysis of mutational data, bootstrapped confidence intervals, and full and partial parameter correlation analysis.

2. REPRESENTATION OF FOLDING TRANSITIONS OF REPEAT PROTEINS USING AN ISING MODEL

To determine ΔG_i and $\Delta G_{i,\ i-1}$ from Ising analysis, equilibrium folding transitions (colloquially, "melts") need to be acquired for a series of constructs of varying repeat number (e.g., Figure 1). These melts are most easily obtained using chemical denaturation with urea or guanidine hydrochloride. Although thermal unfolding provides access to important quantities such as enthalpy and entropy, thermal denaturation is much less likely to be reversible than chemical denaturation, preventing the determination of true equilibrium thermodynamic parameters. Folding is often monitored by circular dichroism (CD) spectroscopy, which is well suited for α-helical repeats. Other spectroscopic signals, such as intrinsic tryptophan fluorescence, may also be suitable for monitoring unfolding. Typically, multiple melts are measured for each type of construct. The resulting data set, which contains M melts for all constructs, is globally fitted to estimate Ising parameters using the equations below.

2.1 *A matrix approach to computing partition functions for linear repeat proteins*

Unlike equilibrium two-state models for protein folding, the Ising model includes intermediate states where some repeats are folded and others are unfolded. Analysis of unfolding transitions using this model requires an expression for folding that includes all 2^{ℓ} conformations (where ℓ is the number of repeats in a particular construct). Such an expression can be obtained from a partition function in which folding is represented at the level of individual repeats using two-by-two correlation matrices (*W*). For a general heteropolymer with ℓ repeats,

$$q = n \times W_1 \times W_2 \times \ldots \times W_{\ell} \times c \tag{1}$$

where

$$W_i = \begin{pmatrix} e^{-(\Delta G_i + \Delta G_{i-1,\, i})/RT} & 1 \\ e^{-\Delta G_i/RT} & 1 \end{pmatrix} = \begin{pmatrix} \kappa_i \tau_{i-1,\, 1} & 1 \\ \kappa_i & 1 \end{pmatrix} \tag{2}$$

The top and bottom rows correspond to repeat *i-1* being folded and unfolded. The left and right columns correspond to repeat *i* being folded and unfolded. The quantities κ_i and $\tau_{i\text{-}1,\, i}$ are equilibrium constants for intrinsic folding of repeat *i*, and for interface formation between repeats *i-1* and *i*. The *n* and *c* terms in equation 1 are vectors that convert the *W* matrix product (itself a two-by-two matrix) into a scalar:

$$n = \begin{pmatrix} 0 & 1 \end{pmatrix} \tag{3A}$$

$$c = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{3B}$$

The zero in equation 3A eliminates terms that include a folded ghost repeat[1] at position 0 (corresponding to the top row of the matrix product in equation 1).

2.2 *Simplification of the partition function for capped homopolymeric repeat proteins*

The product of the W_i matrices can be simplified considerably if repeats are identical. For constructs with $\ell - 2$ identical internal R repeats and terminal N- and C-caps (see the capped homopolymeric series, Figure 1A), the partition function can be written

$$q = n \times W_N \times W_R^{\ell-2} \times W_C \times c \tag{4}$$

Although the same number of matrix multiplications are needed to compute the homopolymeric and the general partition function (equations 4 and 1, respectively), the homopolymeric version involves many fewer free energy terms. Here we refer to a model used to analyze capped homopolymeric repeat proteins as "model 1" (Table 1). Since the solubilizing substitutions on the capping repeats are likely to affect the intrinsic free energy, separate free energy terms (ΔG_N, ΔG_R, and ΔG_C) are included in the model. However, since these solubilizing substitutions are distant from the interfaces with the internal R repeats, a single interfacial free energy is used to model all interfaces (NR, RR, and RC interfaces), which we will represent as $\Delta G_{R-1,R}$.

2.3 *Modification of the partition function for capped heteropolymeric repeat proteins*

For several decades, point substitutions have been used in protein folding studies to determine how specific interactions contribute to protein stability (Alber and Matthews, 1987) and to the kinetics of folding (Matthews and Hurle, 1987; Fersht et al., 1992).

[1] As with ghosts, there is no such thing as a zeroth repeat.

Combining targeted sequence substitutions with Ising analysis of repeat proteins permits the effects of structural perturbation on stability to be resolved into intrinsic and interfacial free energies. One way to quantify the effects of point substitution on intrinsic and interfacial stability is to prepare and analyze capped homopolymeric sequences with the same substitution in each repeat, and compare fitted Ising parameters to those determined from unsubstituted repeats. In this approach, ΔG_R and $\Delta G_{R-1,R}$ values are fitted separately for each series using the capped homopolymer partition function (equation 4). Comparing these values reveals the effects of substitution on intrinsic folding and interfacial coupling. This approach was used to determine the effects of surface substitution within the helices of consensus TPR arrays (Cortajarena et al., 2011); substitutions were found to perturb ΔG_R (although $\Delta G_{R-1,1}$ was not varied in fits of the substituted arrays).

A more informative approach to quantify the effects of point substitution on Ising parameters involves constructing arrays in which unsubstituted and substituted repeats are combined in the same constructs (Figure 1B). In this heteropolymeric variation, the partition function contains different *W* matrices for unsubstituted versus unsubstituted repeats (which we will refer to as X and R repeats), which differ in their intrinsic folding free energy parameters (ΔG_X and ΔG_R). Because the heteropolymer approach creates asymmetric interfaces (RX, XR, and XX, in addition to RR; see Figure 1B), these *W* matrices are further subdivided depending on the identity (X versus R) of their *i-1* repeat, which differ in their interfacial coupling energies ($\Delta G_{R-1,X}$, $\Delta G_{X-1,R}$, $\Delta G_{X-1,X}$, and $\Delta G_{R-1,R}$). Although this approach increases the complexity of the model (compare model 2 with model 1 in Table 1), additional structural information is accessible using this approach. Specifically, comparing $\Delta G_{R-1,X}$ and $\Delta G_{X-1,R}$ values to $\Delta G_{R-1,R}$ resolves whether the substitution in the X repeat at position *i* affects the stability of the *i-1* interface or the *i+1* interface. Moreover, comparing $\Delta G_{X-1,X}$ with the sum of $\Delta G_{R-1,X}$ and $\Delta G_{X-1,R}$ provides a measure of thermodynamic coupling between adjacent interfaces.

2.4. *Using the partition function to model denaturant-induced unfolding transitions*

Unfolding transitions can be modeled using the fraction of repeats that are folded as a function of denaturant concentration. The fraction of repeats that are folded ($f_f = n_f/(n_f + n_u)$) can be calculated by partial differentiation of the partition function with respect to intrinsic equilibrium constants of each type. For model 1 (Table 1),

$$\begin{aligned} f_f &= \frac{1}{\ell q}\sum_j \kappa_j \frac{\partial q}{\partial \kappa_j} \\ &= \frac{1}{\ell q}\left[\kappa_N\left(\frac{\partial q}{\partial \kappa_N}\right) + \kappa_R\left(\frac{\partial q}{\partial \kappa_R}\right) + \kappa_C\left(\frac{\partial q}{\partial \kappa_C}\right)\right] \end{aligned} \tag{5}$$

For model 2, equation 5 contains an additional partial derivative (with respect to κ_X). Denaturant dependencies are typically built into the intrinsic (κ_i) but not the interfacial ($\tau_{i, i-1}$) terms,[2] assuming a linear dependence of free energy on denaturant concentration,

$$\Delta G_i(x) = -RT\ln\kappa_i = \Delta G_{i,H_2O} + m_i[x] \tag{6}$$

where [x] is the molar denaturant concentration, and $\Delta G_{i,H2O}$ and m_i are constants determined from the fit. The H_2O subscript associated with the free energy on the right-hand side of equation 6, which indicates an extrapolated free energy in the absence of denaturant (Greene and Pace, 1974), will be omitted below for brevity. For analysis of heteropolymeric repeat arrays, a denaturant dependence for both types of repeats can be treated separately (m_R and m_X; model 2).

Using f_f, the observed signal for each of the M melts can be calculated as a population-weighted average:

[2] In principle, denaturant dependence of the interfaces could be included by introducing an interfacial m-value through an equation analogous to (6), although in practice, m_i and $m_{i-1,i}$ are strongly coupled.

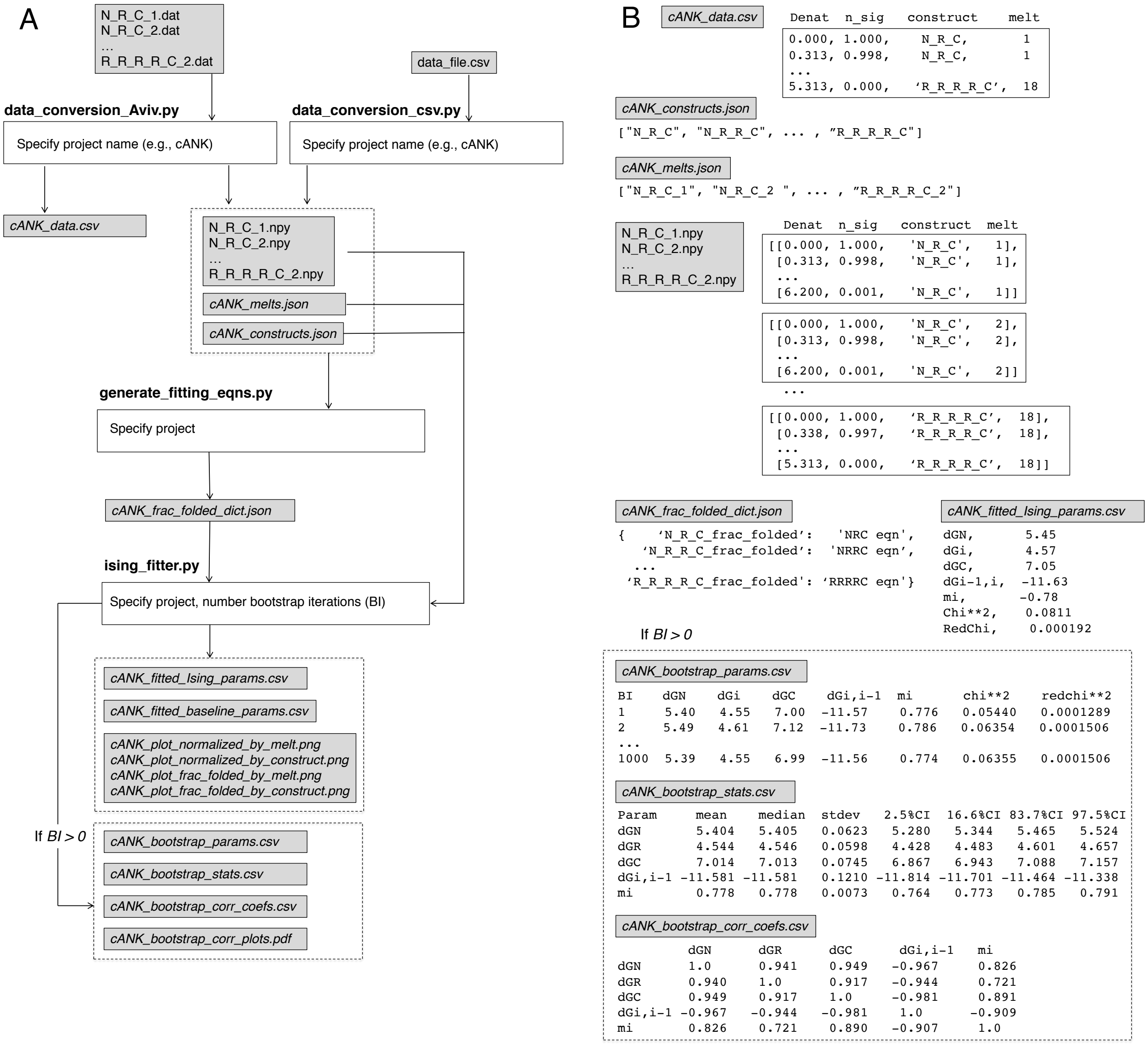
$$Y_{calc,\ k\in M} = f_f Y_{n,\ k\in M} + (1-f_f)Y_{d,\ k\in M} \tag{7}$$

where $Y_{n,\ k\in M}$ and $Y_{d,\ k\in M}$ are spectroscopic signals of the fully folded and fully unfolded states (see equation 9 below) for the k^{th} of M melts[3]. Note that equation 7 assumes equal spectroscopic contributions of all repeats, which is likely to be the case for a signal such as CD, which measures secondary structure in each repeat. Equation 7 can be modified to account for site-specific signals such as tryptophan fluorescence (see Aksel and Barrick, 2014).

3. PROGRAMS FOR ISING ANALYSIS OF FOLDING TRANSITIONS OF REPEAT PROTEINS

The overall workflow for our 1D-Ising analysis programs is shown in Figure 2A. Analysis is performed using three sequential python programs: (1) a program that converts data files to files containing numpy arrays, (2) a program that generates expressions for partition functions (equations 1-4) and fractional folding (equation 5) for each construct, and (3) a program that fits the converted data from the first program with a normalized expression based on fractional folding expressions generated from the second program. The third program also plots fitted data, performs bootstrap analysis, and analyzes parameter correlation. Each program is available on github (barricklab-at-jhu/Ising_programs) as a .py file that can be run directly in a terminal command line or in an IDE such as Anaconda (2016). In addition, the programs are combined in an interactive python notebook (.ipynb; Perez and Granger, 2007) that can be run in Jupyter (Randles et al., 2017).

[3] In subsequent expressions, it will be assumed that the kth melt belongs to the set of M total melts, e.g., $Y_{calc,\ k\in M} \rightarrow Y_{calc,\ k}$.


A
N_R_C_1.dat
N_R_C_2.dat
…
R_R_R_R_C_2.dat
data_file.csv
data_conversion_Aviv.py
Specify project name (e.g., cANK)
data_conversion_csv.py
Specify project name (e.g., cANK)
cANK_data.csv
N_R_C_1.npy
N_R_C_2.npy
…
R_R_R_R_C_2.npy
cANK_melts.json
cANK_constructs.json
generate_fitting_eqns.py
Specify project
cANK_frac_folded_dict.json
ising_fitter.py
Specify project, number bootstrap iterations (BI)
cANK_fitted_Ising_params.csv
cANK_fitted_baseline_params.csv
cANK_plot_normalized_by_melt.png
cANK_plot_normalized_by_construct.png
cANK_plot_frac_folded_by_melt.png
cANK_plot_frac_folded_by_construct.png
If BI > 0
cANK_bootstrap_params.csv
cANK_bootstrap_stats.csv
cANK_bootstrap_corr_coefs.csv
cANK_bootstrap_corr_plots.pdf
B
cANK_data.csv
cANK_constructs.json
cANK_melts.json
N_R_C_1.npy
N_R_C_2.npy
…
R_R_R_R_C_2.npy
cANK_frac_folded_dict.json
cANK_fitted_Ising_params.csv
If BI > 0
cANK_bootstrap_params.csv
cANK_bootstrap_stats.csv
cANK_bootstrap_corr_coefs.csv

```
Denat  n_sig   construct    melt
0.000, 1.000,    N_R_C,       1
0.313, 0.998,    N_R_C,       1
...
5.313, 0.000,   'R_R_R_R_C',  18
```

```
["N_R_C", "N_R_R_C", ... , ″R_R_R_R_C"]
```

```
["N_R_C_1", "N_R_C_2 ", ... , ″R_R_R_R_C_2"]
```

```
  Denat  n_sig   construct  melt
[[0.000, 1.000,   'N_R_C',   1],
 [0.313, 0.998,   'N_R_C',   1],
 ...
 [6.200, 0.001,   'N_R_C',   1]]

[[0.000, 1.000,   'N_R_C',   2],
 [0.313, 0.998,   'N_R_C',   2],
 ...
 [6.200, 0.001,   'N_R_C',   2]]
  ...
[[0.000, 1.000,   'R_R_R_R_C',  18],
 [0.338, 0.997,   'R_R_R_R_C',  18],
 ...
 [5.313, 0.000,   'R_R_R_R_C',  18]]
```

```
{    'N_R_C_frac_folded':   'NRC eqn',
   'N_R_R_C_frac_folded':  'NRRC eqn',
  ...
 'R_R_R_R_C_frac_folded': 'RRRRC eqn'}
```

```
dGN,       5.45
dGi,       4.57
dGC,       7.05
dGi-1,i,  -11.63
mi,       -0.78
Chi**2,    0.0811
RedChi,     0.000192
```

```
BI    dGN    dGi    dGC    dGi,i-1  mi       chi**2    redchi**2
1      5.40   4.55   7.00  -11.57    0.776   0.05440   0.0001289
2      5.49   4.61   7.12  -11.73    0.786   0.06354   0.0001506
...
1000  5.39   4.55   6.99  -11.56    0.774   0.06355   0.0001506
```

```
Param      mean    median  stdev   2.5%CI  16.6%CI 83.7%CI 97.5%CI
dGN        5.404   5.405   0.0623   5.280   5.344   5.465   5.524
dGR        4.544   4.546   0.0598   4.428   4.483   4.601   4.657
dGC        7.014   7.013   0.0745   6.867   6.943   7.088   7.157
dGi,i-1 -11.581 -11.581   0.1210 -11.814 -11.701 -11.464 -11.338
mi         0.778   0.778   0.0073   0.764   0.773   0.785   0.791
```

```
          dGN      dGR      dGC      dGi,i-1   mi
dGN       1.0      0.941    0.949   -0.967     0.826
dGR       0.940    1.0      0.917   -0.944     0.721
dGC       0.949    0.917    1.0     -0.981     0.891
dGi,i-1 -0.967   -0.944   -0.981     1.0      -0.909
mi        0.826    0.721    0.890   -0.907     1.0
```

Figure 2. Flow-chart for 1D-Ising analysis of tandem-repeat unfolding transitions. Here, analysis of a capped homopolymeric series with three .py programs is illustrated. (A) Python programs are shown in white boxes along with required control parameters, input and output files are shown in grey boxes. (B) The format of input and output files. The first program (*data_preprocessing_Aviv.py*) converts data files to files containing numpy arrays and generates outputs lists of constructs and melts. The second program (*generate_fitting_equations.py*) uses the list of constructs to generate and output a dictionary of expressions for fraction folded (equation 5) for each construct. The third program (*ising_fitter.py*) uses these fraction-folded expressions, along with the melt and construct lists, to fit the expressions to the melt.npy data using nonlinear least squares. *ising_fitter.py* outputs fitted parameters to a csv file, as well as plots of raw and transformed data with fitted curves (see Figure 3). Following the fit, the user can run *BI* iterations of bootstrap analysis, which returns parameter values for each bootstrap iteration, statistics for bootstrap parameter distributions, a correlation matrix for bootstraped parameters, and plots of bootstrap histograms and parameter correlation (see Figure 4). Alternatively, all three programs can be run from a single combined Jupyter notebook.

3.1 ***Preprocessing data: data_preprocessing.py***

The data preprocessing program generates a series of numpy arrays for each of the M melts and saves each to an *npy* file. In the process, this program writes out a list of constructs and a list of melts, which are used by the subsequent programs. To help identify output files, a "project" name should be specified in the data preprocessing program. This project name is used to name output files. It is convenient to use the name of the series to which the repeat constructs belong (such as "cANK" in the example in Figure 2).

There are two programs for data preprocessing that can be used for different types of input data. *data_preprocessing_csv.py* takes data in .csv format, which often requires some manual pre-processing, whereas *data_preprocessing_AVIV.py* takes input data in .dat format generated by Aviv spectrometers (the format of most of our denaturation data). In addition, *data_preprocessing_AVIV.py* outputs a single data csv file containing all melts, which can be combined with other csv data sets.

Input data files should be placed in the same folder as the jupyter notebook or the preprossessing program (and subsequent programs). For Aviv data files, each unfolding transition is in a separate file identified by the .dat extension. The root of each file name should give the structure of the repeat array for that melt, followed by an integer used to distinguish multiple melts of the same construct. For example, N_R_R_C_1.dat is the "first" melt for the construct NR_2C (two internal R repeats flanked by N- and a C- capping repeats; Figure 1A). Likewise, R_X_R_C_2.dat is the "second" melt for the construct RXRC. For unfolding transitions in other formats, melts should be manually combined into a single .csv file containing four columns. The first column contains the denaturant concentrations for each melt, the second column contains spectroscopic values, the third column gives the construct in the format described above (e.g., N_R_R_C), and the fourth column gives an ID number identifying the data set, ranging from 1 to M (Figure 2B).

In the conversion process, data are normalized such that for each of the M melts the point with the largest signal intensity[4] in each melt ($Y_{max,\,k}$) is set to one, and the point

[4] For CD measurements in the far-UV, this is typically the largest *negative* value.

with smallest intensity ($Y_{min,\ k}$) is set to zero. The other data points in the k^{th} melt are scaled to these values:

$$Y_{norm,\,j} = \frac{Y_{\max,\,k} - Y_{obs,\,j}}{Y_{\max,\,k} - Y_{\min,\,k}} \tag{8}$$

This normalization ensures that each melt has a similar influence in the fit (assuming each melt has the same number of points), which is important in cases where different melts have very different *Y* values. One potential pitfall is that if the starting *Y* values vary significantly from melt to melt, this scaling may produce significant differences in the absolute uncertainties in the normalized *Y* values of different melts. Such differences in uncertainties should be accounted for using weighting terms in the fitting program.

3.2. *Generating fitting equations: generate_fitting_equations.py*

Using the construct list generated by the data preprocessing program, the second program (*generate_fitting_equations.py*) builds a dictionary of partition functions (equation 4). From these partition functions, a dictionary of fraction folded expressions (equations 5 and 7) is built using the python *SymPy* module (which enables symbolic math manipulations, Meurer et al., 2017). A final and very important step in generating fraction folded expressions is the use of the *SymPy* command "simplify", which significantly shortens the resulting expressions. For example, the simplify command reduces the expression for the fraction of folded NR_3C from 467 to 227 characters. This reduction provides a significant speed-up during nonlinear least squares, which is especially important for bootstrap analysis (requiring hundreds to thousands of nonlinear-least squares optimizations). The dictionary of fraction folded expressions is output as a .json file for nonlinear least-squares fitting in the third program. Another feature built into *generate_fitting_equations.py* is a calculation of the rank of the coefficient matrix, which is useful evaluating whether the series of constructs being analyzed provides adequate constraints to fit the thermodynamic parameters (see Appendix 2). Results from the rank analysis are outputted to the screen, alerting the user to potential problems with the fit.

3.3. *Fitting with an Ising model: ising_fitter.py*

Using the converted data, construct list, and melt list generated by the data preprocessing program along with the fitting equations generated by the fitting equation program, the third program (*ising_fitter.py*) globally fits all of the folding transitions using nonlinear least-squares, by minimizing the sum of the squares of the residuals between normalized experimental $Y_{norm,\,j}$ values from each of the M melts (equation 8) and the calculated values $Y_{calc,k}$ (equations 5 and 7):

$$SSR = \sum_{k=1}^{M\ melts} \sum_{j=1}^{n_k} \left\{ Y_{norm,\,j,\,k} - Y_{calc,\,k}\left([x]_j;\ \Delta G_N,\ \Delta G_R,\ \Delta G_C,\ \Delta G_{R-1,R},\ m_R,\ Y_{n,\,k},\ Y_{d,\,k}\right)\right\}^2 \qquad (10)$$

The terms in parentheses are the parameters to be optimized; in equation 10, they are those of model 1 (Table 1). The outer sum in equation 10 is over each of the M melts, and the inner sum is over each of the n_k denaturant concentrations in each melt. The fit is performed with the python *lmfit* module (Newville et al., 2019), which provides a number of useful fitting options including setting bounds for adjustable parameters.

The parameters that are optimized in equation 10 include global thermodynamic parameters, which apply to all melts and all constructs,[5] and local baseline parameters, which apply individually to each melt (Table 1). For model 1, the thermodynamic parameters include intrinsic free energies for the N, internal R, and C repeats (ΔG_N, ΔG_R, and ΔG_C), an interaction free energy between adjacent repeats ($\Delta G_{R-1,R}$), and a denaturant dependence for each repeat (m_R, eq 5). The local parameters model a total of 2M baselines (a native and denatured baseline for each melt; $Y_{n,\,k}$ and $Y_{d,\,k}$ in equation 7). Each baseline is assumed to vary linearly with denaturant concentration; thus, each baseline requires two locally fitted parameters:

[5] An exception is the intrinsic free energies of folding of the capping repeats (ΔG_N and ΔG_C), which apply to all repeats that contain N- and C-terminal caps.

$$Y_{n,k} = a_{n,k} + b_{n,k}[x] \tag{11A}$$

$$Y_{d,k} = a_{d,k} + b_{d,k}[x] \tag{11B}$$

As is typical of nonlinear least-squares optimizations, initial guesses must be supplied to begin the search for optimal parameters. Finding satisfactory initial guesses for the thermodynamic parameters may require some care so that unfolding midpoints and slopes are reasonably approximated at the start of the search. Initial guesses for the baseline parameters are easily determined due to the normalization provided by the data preprocessing program: fitted values of $a_{n,k}$ and $a_{d,k}$ should be close to one and zero, respectively, and baseline slopes are often quite similar for all normalized melts. Thus, guesses for the $4m$ baseline parameters can usually be provided with just four parameters (a_n, b_n, a_d, and b_d).

From these initial guesses, *lmfit* uses the Marquardt-Levenberg algorithm to optimize the local and global parameters. The parameter set that minimizes the sum of the square of the residuals is then returned to the user as are plots of fitted data in various formats. Fitted thermodynamic and baseline parameters are also written to csv files, as are plots of data and fits (Figure 2).

3.4. *Bootstrap analysis*

To estimate uncertainties on fitted parameters and to explore parameter correlations, bootstrap analysis can be performed after the data have been fit. Bootstrap analysis is a resampling method in which sets of synthetic, or "bootstrapped", data are generated with errors derived from the original data set (Efron, 1979; Johnson, 2008). Here we apply the bootstrap to the residuals of the fit, generating bootstrapped data sets by (1) using the best fitted parameters to estimate "true" values for each melt at each measured denaturant concentration (the $Y_{calc,j,k}$ values in equation 10, using best-fitted parameters), (2) calculating residuals based on these $Y_{calc,j,k}$ values (the expression in curly braces in equation 10), and (3) adding randomly selected residuals[6] from step 2 to the true $Y_{calc,\,j,\,k}$ values. Assuming the fitted Ising model provides a good description of the measured $Y_{norm,\,j,\,k}$ values (specifically, that the $Y_{calc,\,j,\,k}$ provides an good estimate of the error-free value of the measured Y value at denaturant concentration j, melt k, and that the errors in $Y_{norm,\,j,\,k}$ are independent of j and k), these bootstrap data sets are an adequate approximation of independently determined data sets with the same structure as the observed data. Fitting each bootstrapped data set and comparing fitted parameter values provides an estimate of uncertainties for each fitted parameter.

One important parameter in bootstrap analysis is the number of bootstrap iterations performed. Ideally, thousands of iterations are performed, although for large data sets and complex models, performing thousands of bootstrap iterations may require more CPU time than is practical, especially on a desktop computer.[7] The length of time required for bootstrap analysis can be significantly shortened by using parallel processing, either running on multiple cores within a laptop or desktop computer, or through shared processors within a cluster. We have created a version of the *ising_fitter.py* that runs bootstrapping in parallel using the python "*multiprocessing*" module (*ising_fitter_parallel.py,* also available on github). To help estimate the amount of time a given bootstrap analysis will take, the time required for the fit of the observed data is provided by *ising_fitter.py*, which is a good approximation of the time required for a single bootstrap iteration.

Upon completion of the fit of the data, *ising_fitter.py* prompts the user for the number of bootstrap iterations to perform. The fitter then performs the specified number of fits of bootstrap data, using the same initial guesses that were used for the initial fit. Thermodynamic parameters from each bootstrap iteration are written to a *bootstrap_params.csv* file, and a statistical summary of bootstrapped thermodynamic

[6] Random selection with replacement.

[7] Factors that increase the amount of bootstrap CPU time are the number of melts, which determines the number of fitted baseline parameters), the number of thermodynamic parameters (e.g., compare Mode 2 with Model 1, Table 1), and the extent of parameter correlation.

parameter values, including averages and measures of variation, are written to a *bootstrap_stats.csv* file (Figure 2).[8]

3.5. *Correlation analysis*

One advantage of the bootstrap method is that it permits visual inspection of parameter correlation. Analysis of parameter correlation reveals not only how much uncertainty is associated with fitted parameters, but the origins of that uncertainty. After the bootstrap analysis is completed, a grid of correlation plots is generated for all pairs of bootstrapped parameters along with a histogram of bootstrap values for each selected parameter. Linear Pearson correlation coefficients are calculated from each plot and are written to a *bootstrap_corr_coef.csv* file. Comparing the correlation plots for each parameter to the histogram of bootstrapped values emphasizes the important point that although parameter correlation increases parameter uncertainty, the uncertainty values reported from the bootstrap analysis include uncertainties resulting from correlation with all other parameters. Though a particular pair of parameters may show strong correlation, as long as the confidence intervals estimated from bootstrap analysis are tolerably low (i.e., the parameter histogram is suitably narrow), the correlation can be considered acceptable.

Rather, it is when fitted parameter values have intolerably large confidence intervals that correlation analysis is useful. In such cases, identifying the underlying correlations (in particular, pairs of parameters with large partial correlation coefficients; see below) that lead to large parameter uncertainties can be used to modify the model to avoid high correlation, or to include parameter constraints that are informed by independent information.

Although the correlation analysis described above reveals the extent to which pairs of parameters covary, these absolute correlations are sometimes indirect. Such indirect

[8] Although all fitted parameters (thermodynamic and baseline parameters) are optimized in each bootstrap iteration, we are typically interested in uncertainties and correlations among the thermodynamic parameters. If the uncertainties in fitted baseline parameters are desired, the bootstrap portion of *ising_fitter.py* can easily be modified to include these values for statistical analysis.

correlations result when each variable in a pair is each strongly correlated with a third. As a result, the two parameters in the pair show correlation with each other. To reveal the direct correlation among parameter pairs, "partial" correlation coefficients can be calculated in which correlation between a pair of parameters is calculated after correlation to all the other variable are accounted for. For a pair of variables X and Y in a data set with *n*+2 total variables, partial correlation coefficients are calculated by performing linear regressions separately for each variable X and Y with the *n* other variables $\mathbf{Z} = \{Z_1, Z_2, \ldots, Z_n\}$, where $\{X,Y\} \not\subset \mathbf{Z}$, determining the residuals for these two regressions ($r_{X,\mathbf{Z}}$ and $r_{Y,\mathbf{Z}}$), and determining the correlation coefficients for the residuals $r_{X,\mathbf{Z}}$ and $r_{Y,\mathbf{Z}}$ (referred to as $\rho_{X,Y\cdot Z}$):

$$\rho_{XY\cdot Z} = \frac{\sum_{i=1}^{N}\left(r_{XZ,i}\, r_{YZ,i}\right)}{\sqrt{\sum_{i=1}^{N} r_{XZ,i}^{\;2}}\sqrt{\sum_{i=1}^{N} r_{YZ,i}^{\;2}}} \tag{12}$$

Using the values of bootstrapped thermodynamic Ising parameters in *boostrap_params.csv*, partial correlation coefficients are calculated using the program "*partial_correlation.py*". This program uses the *Pingouin* statistical package (Vallat, 2018) to calculate partial correlation coefficients, and outputs them to the file *bootstrap_partial_corr.csv*.

4. A FIVE-PARAMETER (MODEL 1) FIT OF A CAPPED HOMOPOLYMER SERIES

Using the fitting programs above, we have performed Ising fitting and bootstrap analysis on unfolding transitions of a series of capped homopolymeric consensus ankyrin repeat constructs described previously (Aksel and Barrick, 2009). The constructs in this series match those in Figure 1A (model 1, Table 1), ranging in length from three to five repeats, and containing either N-terminal caps, C-terminal caps, or both. For each construct, there are two melts (technical replicates), such that each construct has a similar impact on the fit. Likewise, for each capping pattern (NR_x, R_xC, NR_xC), there are three constructs of different length (x=3, 4, 5), so that the two caps have a similar impact on the fit. As a whole, the data set comprises 18 melts with a total of 499 observations. There are five fitted global thermodynamic parameters (ΔG_N, ΔG_R, ΔG_C, $\Delta G_{R-1,R}$, and m_R), and 72 local baseline parameters (four for each melt). Thus, there are 499–5–72 = 422 degrees of freedom (DOF). The fit took about 5 seconds of wall time on a MacBook Pro using a single 2.3 GHz Intel Core i9 processor (although this time depends on values of the initial guesses).

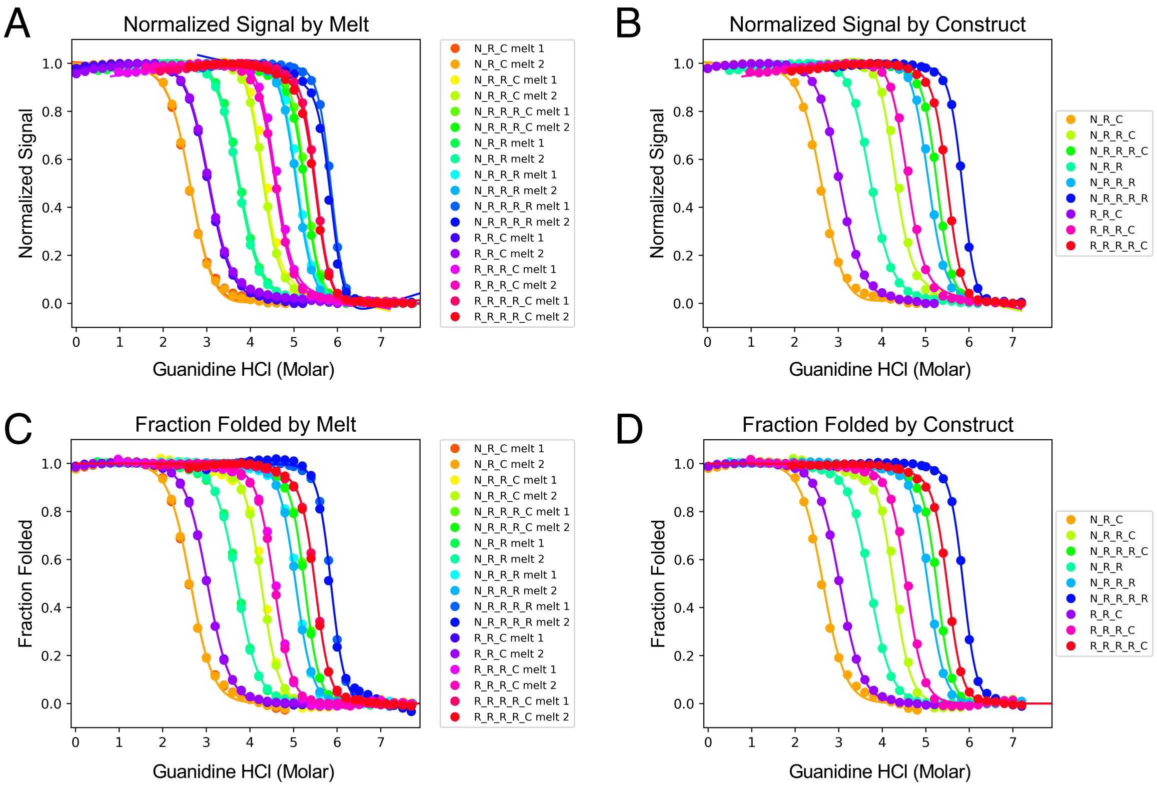


Figure 3. Fitted unfolding transitions for capped homopolymeric ankyrin arrays. Data are from (Aksel et al., 2011) and are fitted with model 1 (Table 1). Plots are direct outputs from the plotting section of the fitter. Data and fitted curves are plotted both as normalized signal (A and B, the space in which least-squares minimization is carried out) and as fraction folded (C and D). In the fraction folded representation, fitted curves for different melts of the same construct are identical (because a single set of thermodynamic parameters is fitted globally), but the data are not (due to random errors). In contrast, fitted curves to normalized data differ, since each curve has its own baseline parameters. Plots (A) and (C) show all fitted melts; plots (B) and (D) show one representative melt from each construct.

Overall, the fit of model 1 to the data is quite good. This can be seen in Figure 3A and B, where the normalized data and fitted curves are in good agreement. Inspection of normalized curves is important because this is the space in which the fit is performed. Highly sloped baselines, which are often a result of inadequate baseline sampling (and which often increase uncertainties on fitted thermodynamic parameters), are apparent upon inspection of normalized fits. In contrast, baseline slopes are hidden in fraction folded curves (Figure 3C and D). For melts in which fitted baseline slopes are unreasonably high, limits can be imposed on baseline slopes using the *lmfit* module by providing *min* and *max* values during initial parameter assignments.[9] The quality of the fit in Figure 3 is also indicated by the reduced sum of squares of the residuals (RSSR = SSR/DOF = 0.00019 in Figure 3A). Since the normalized unfolding transitions span one unit of signal change, this RSSR value reflects an average squared residual of about 0.019 percent, or an average residual of about 1.4 percent. This residual is within the error resulting from the circular dichroism measurements from which the normalized signal is derived.

To estimate uncertainties on the fitted thermodynamic parameters, 1000 bootstrap iterations were performed. In absolute terms, uncertainties on the thermodynamic parameters are quite small for the fit in Figure 3: 95 percent confidence intervals[10] of bootstrapped free energies are about 0.2 kcal mol^{-1} (Table 2). Bootstrap estimates for all thermodynamic parameters have unimodal and roughly normal distributions (diagonal histograms, Figure 4), consistent with a single well-defined minimum in SSR over the fitted parameter space. For each parameter, uncertainties can be estimated from the standard deviation of bootstrap values or from locating the parameter values that separate the low and high tails of the bootstrap distribution at 67 or 95 percent (Table 2). An advantage of the latter approach is that no assumptions made on the symmetry of the error. For the five thermodynamic parameters in model 1, the 67 and 95 percent

[9] In most cases, a preferable remedy for a poor baseline is to collect a melt with a better baseline. However, this is not always possible for marginally or highly stable constructs.

[10] Given that the bootstrap approach misses some sources of error, such as systematic variations among different data sets, 95% confidence intervals seem a safer and more appropriate estimate of uncertainties in fitted parameters than the more common 67% limits.

confidence intervals are within 0.1 percent of the mean plus or minus one or two standard deviations, as is expected for normally distributed parameter estimates. Large differences between the standard deviation and confidence intervals may reflect fitting pathologies such as one-sided parameter bounds, and multiple least-squares minima in parameter space (Johnson, 1983), which should be apparent in bootstrap parameter histograms. Identifying such pathologies is important, since it almost certainly indicates that the model is not adequately constrained by the constructs being fitted, and that either the model or the data set should be modified.

As described above, parameter correlation can be a significant source of uncertainties in fitted parameters. Even for the simplest types of fits, strong correlations are common, such as the fitted slope and intercept parameters in linear regression against a single independent variable (Johnson, 1983), and the $\Delta G°$ and m-values describing simple two-state folding (see Appendix 1). The bootstrap analysis allows correlations between pairs of parameters to be clearly visualized in a scatter plot in which each point represents a pair of parameter values from the same bootstrap iteration (Figure 4). Correlations between bootstrap parameter pairs can also be seen in the correlation coefficient matrix that is generated by *ising_fitter.py* (as *bootstrap_corr_coef.csv*, Table 3).

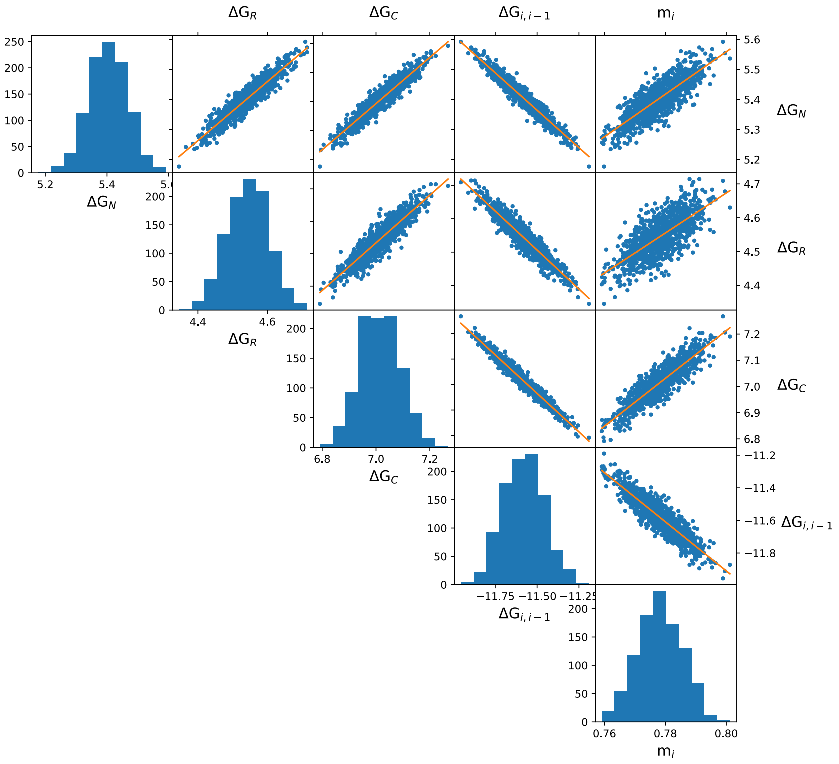


Figure 4. Bootstrap histograms and correlation plots of fitted thermodynamic parameters for capped homopolymeric ankyrin arrays. Capped consensus ankyrin homopolymer data from Figure 3 was used along with best-fitted parameters from Model 1 to generate 1000 bootstrap fits. The lower diagonal panels show histograms for bootstrapped thermodynamic parameters. The upper triangular panels show correlations among pairs of thermodynamic parameters, along with a best-fitted line to bootstrapped pairs.

For the capped homopolymer fit (Figure 3), the strongest correlations are between the interfacial free energy ($\Delta G_{R-1,R}$) and the three intrinsic free energies (ΔG_N, ΔG_R, and ΔG_C; Table 3); these correlations are negative (Figure 4).

This negative correlation between intrinsic and interfacial free energies is expected—an increase in $\Delta G_{R-1,R}$ (i.e., a decrease in the interfacial free energy), which would decrease midpoints of unfolding transitions, can be compensated by decreasing ΔG_N, ΔG_C, and especially ΔG_R (since this term contributes three times as much, on average, to the stabilities of the constructs in Figure 1 as ΔG_N and ΔG_C).

In addition, there appear to be significant positive correlations among ΔG_N, ΔG_R, and ΔG_C (Table 3, Figure 4). These correlations are unexpected—if one intrinsic free energy parameter increases, the others would be expected to decrease to maintain the midpoint of the folding transition. Thus, it seems likely that these apparent correlations are indirect—since each intrinsic free energy is negatively correlated with $\Delta G_{R-1,R}$, each may end up positively correlated with the other two. To test for indirect correlation, we calculated partial correlation coefficients among the thermodynamic bootstrap parameter values, which reveal the correlation between two parameters when correlations to the other parameters are factored out (Table 3, lower triangle). Based on partial coefficients, the negative correlation between $\Delta G_{R-1,R}$ and ΔG_R remains high ($\rho_{\Delta G_R,\, \Delta G_{i,i-1} \bullet \{\Delta G_R,\, \Delta G_R,\, m_R\}}$ is actually more negative than the total correlation coefficient between these two parameters; Table 3), but the correlations between $\Delta G_{R-1,R}$ and ΔG_N and ΔG_C are decreased. Moreover, the correlations among the intrinsic free energies (ΔG_N, ΔG_R, and ΔG_C) are also decreased, and the signs of the partial correlation coefficients are negative, matching expectations.

Partial correlation analysis reveals a similar indirect correlation between the bootstrapped m_R-values and the three bootstrapped intrinsic free energy parameters. Though the total correlation coefficients of all four free energy bootstrapped parameters with m_R are large (Figure 4), only $\Delta G_{R-1,R}$ shows the expected negative correlation (negative correlation is expected since an increase in m_R should increase the stability in the absence of denaturant, decreasing ΔG values; see Appendix I). The three intrinsic free energy parameters show unexpected positive total correlations with m_R, with slightly larger correlation coefficients for ΔG_N and ΔG_C than for ΔG_R. The partial correlation coefficients for these three parameter pairs ($\rho_{m_R,\, \Delta G_i \bullet \{\Delta G_j,\, \Delta G_k,\, \Delta G_{R-1,R}\}}$) have reversed sign, with the strongest negative correlation between m_R and ΔG_R, suggesting the positive total correlation is indirect. In contrast, the correlation between m_R and $\Delta G_{R,R-1}$ remains large and negative.

Overall, the picture that emerges is that there is a strong negative direct correlation between ΔG_R and $\Delta G_{R-1,R}$, and also between those two parameters and m_R. ΔG_N and ΔG_C have weaker direct negative correlations to these three parameters, and weaker correlations to each other. Again, it is worth pointing out that despite these correlations, ΔG_R and $\Delta G_{R-1,R}$ are determined within 0.2 kcal mol^{-1}. As we have previously described, fitted values of ΔG_R and $\Delta G_{R-1,R}$ are consistent with a strong positive cooperativity in which intrinsically unstable repeats are driven to fold by strongly stabilizing interfaces with folded neighbor repeats (Aksel and Barrick, 2014; Aksel et al., 2011).

6. A TEN-PARAMETER (MODEL 2) FIT OF A CAPPED HETEROPOLYMER SERIES

To illustrate how the Ising fitting programs perform with a more complex heteropolymeric model and to examine the effects of a mutation on intrinsic and interfacial stabilities, we performed Ising fitting and bootstrap analysis on folding transitions of a series of capped consensus ankyrin repeats in which a conserved threonine from one or two R repeats is substituted with a valine at position 7 in the ankyrin TPLH motif (referred to here as T7V). The T7V substitution has previously been shown to destabilize a four-repeat NXRC construct (where repeat X is an R-type repeat that harbors the point substitution) by 2.6 kcal mol^{-1} using a simple two-state model (Preimesberger et al., 2015). By combining the X-substitution series (Figure 1B) with the homopolymeric capped constructs analyzed with model 1 above,[11] we were able to fit a model for heteropolymeric capped repeats containing ten thermodynamic parameters (model 2).

To accurately determine the intrinsic and interfacial free energies, it is essential that the constructs being analyzed provide adequate constraints on these quantities during the fit. For capped homopolymer series, having the sufficient constructs to constrain ΔG_N, ΔG_R, ΔG_C, and $\Delta G_{R-1,R}$ is fairly intuitive[12], but for capped heteropolymeric series, the complexity of the model and the number of thermodynamic parameters can obscure this sufficiency. To check whether or not the intrinsic and interfacial free energy parameters of the model are adequately determined by the constructs in the data set, the rank of the coefficient matrix defined by the series of constructs is evaluated by *generate_fitting_equations.py*. If the coefficient matrix has full column rank, the

[11] For the combined data set, we included three melts for each of the six heteropolymer constructs in Figure 1B, and two melts for each of the nine homopolymer constructs in Figure 1A. With this combination there are an equal number of homopolymer and heteropolymer melts in the fit, giving equal weight to each.

[12] Each cap must be removed one at a time, and the length must be varied (see Figure 1A).

constructs adequately define the thermodynamic parameters (Appendix 2). Indeed, the combined constructs from Figure 1A and B pass the rank test for the parameters in model 2 (Table 1), permitting us to determine four intrinsic free energies (ΔG_N, ΔG_R, ΔG_X, and ΔG_C) and four interfacial free energies ($\Delta G_{R-1,R}$, $\Delta G_{X-1,X}$,$\Delta G_{R-1,X}$, and $\Delta G_{X-1,R}$). As a whole, the data set comprises 36 melts with a total of 1057 observations. In addition to the eight global thermodynamic parameters above, there are two m-values that give the denaturant dependences of ΔG_R and ΔG_X (m_R and m_X; equation 6), and 144 baseline parameters. Thus, there are 1057–10–144 = 903 degrees of freedom. The fit took about 10 seconds of wall-time, about twice that for model 1, which involves half the number of melts.

Overall, the heteropolymeric capped repeat data set is reasonably well-fitted by model 2 (Figure S1). Fitted values for thermodynamic parameters shared between models 1 and 2 are within about 0.5 kcal mol^{-1} for intrinsic folding energies and 1 kcal mol^{-1} for the interfacial energy. The reduced sum of squares of the residuals is 0.00038, corresponding to an average residual of about 2 percent per point. Uncertainties in fitted thermodynamic parameters for the model 2 fit are quite similar to those from model 1, both for parameters that are shared by the two models and for the parameters that are unique to model 2 (Table 2). An exception is m_X, which has about three times the uncertainty as m_R. This is not unexpected since there are significantly fewer repeats in the fitted data set that have an m_X-denaturant dependence (the X repeats) than repeats that have an m_R dependences (N, R, and C repeats).

Although model 2 contains more fitted parameters than model 1, correlations among fitted bootstrap parameters seem to be somewhat reduced. There are modest decreases in both total and partial correlation coefficients for most of the parameter pairs that are shared by both models (compare Tables S1 and 3). This difference is borne out visually in a comparison of the two sets of correlation plots (Figures S2, 4). The weakest correlations involve the two parameters in model 2 that are least well determined: m_x and in particular, ΔG_X. This relationship between parameter correlation and uncertainty underscores the point made above that strong parameter correlation is not a direct proxy for high parameter uncertainty. Rather, at least in the examples shown here, high parameter uncertainty appears to mask correlation, whereas low parameter uncertainty reveals correlation.

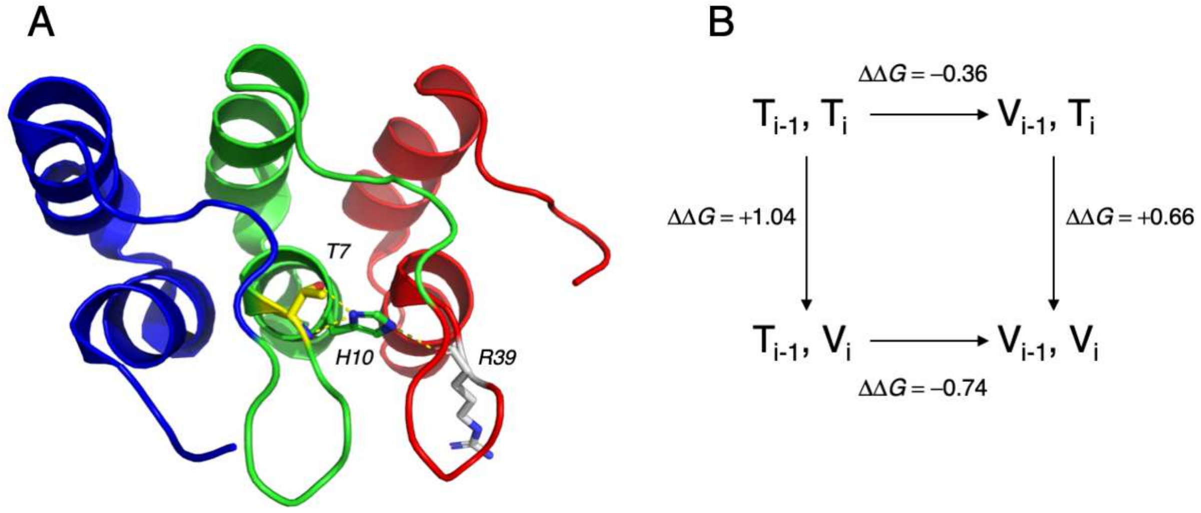


Figure 5. Threonine 7, histidine 10 and their hydrogen bonding patterns within and between ankyrin repeats. (A) Hydrogen bonding pattern in a consensus ankyrin array from Mosavi et al. (1N0R.pdb, Mosavi et al., 2002); note that the histidine ring has been flipped by 180° around χ_2 to match the bonding pattern determined by Preimesberger et al., 2015). Threonine at position 7 (yellow) forms a bifurcated hydrogen bond to the Nδ_1 of histidine 10 via its main-chain NH and side-chain OH within the same repeat. In turn, histidine 10 hydrogen bonds to a main-chain CO in the C-terminal neighboring repeat. (B) Thermodynamic cycle involving changes to interfacial interaction energies ($\Delta G_{i-1,i}$) in response to T7V substitution in adjacent repeats. For each arrow, the ΔΔG value is the difference between the interaction energies in the substituted (arrowhead) and unsubstituted interfaces (arrowtail). For example, for the top arrow, $\Delta\Delta G = \Delta G_{X-1,R} - \Delta G_{R-1,R}$. The difference between ΔΔG values on the top and bottom (and equivalently, left and right) arrows gives the interaction energy between the two substitutions, here -0.4 kcal mol^{-1}.

The additional parameters in model 2 provide a quantitative measure of how the point-substitution affects intrinsic and interfacial stability. Comparing ΔG_X with ΔG_R reveals that intrinsic stability is decreased by 1.6 kcal mol^{-1}. This is consistent with the loss of an internal hydrogen bond between the threonine (O$_\gamma$)H and the *i+3* histidine side-chain N$_\delta$1 (Preimesberger et al., 2015). This hydrogen bond is part of a hydrogen bond network that includes a second, bifurcated hydrogen bond from the *i+3* histidine N$_\delta$1 to the threonine main-chain (N)H, and from the *i+3* histidine N$_\varepsilon$2 (N)H to the main-chain C=O from the residue preceding the threonine in the next repeat (asparagine *i+32*; Figure 5A).

Comparing $\Delta G_{X-1,X}$ with $\Delta G_{R-1,R}$ provides one measure of how interfacial stability is affected by point substitution. This comparison indicates that the interface may be weakly destabilized by valine substitution in adjacent repeats by 0.3 kcal mol^{-1}, although this value is only slightly beyond the 95% confidence intervals (0.12 kcal mol^{-1} for each free energy). However, by combining substitutions to R-repeats (X) with unsubstituted N, R, and C repeats as in Figure 1B, we can also determine interfacial free energies between an N-terminal R repeat and an adjacent X repeat ($\Delta G_{R-1,X}$), and an N-terminal X repeat and an adjacent R repeat ($\Delta G_{X-1,R}$). In other words, we can compare the energetic effects of single point substitution in the N- (*i, i-1*) and C-terminal (*i, i+1*) directions. Comparison of $\Delta G_{R-1,X}$ and $\Delta G_{X-1,R}$ with $\Delta G_{R-1,R}$ shows that the N-terminal interface is destabilized by valine substitution by +1.0 kcal mol^{-1}, but that this stabilization is partly compensated by a stabilization of the C-terminal interface by -0.4 kcal mol^{-1} (Figure 5A). The modest stabilization of the C-terminal interface may reflect a re-orientation of the *i+3* histidine side-chain in the substituted repeat to optimize the two remaining hydrogen bonds (Preimesberger et al., 2015).

Finally, the fact that the two interfacial free energy perturbations do not sum to the perturbation resulting from a tandem substitution identifies a thermodynamic coupling between adjacent repeats. This coupling can be represented in a thermodynamic cycle (Figure 5B). Although the magnitude of the coupling is moderate (~0.4 kcal mol^{-1}), the sign of the coupling favors interfaces with the same residue at repeat position 7 of adjacent repeats. Pairs of threonine residues at repeat position seven result in the most stable interface; although pairs of valines at position seven result in a destabilized interface, a valine pair is more stable than would be expected based on the interfacial energy perturbations from single-repeat substitutions. In other words, pairs of valines at position seven are mutually stabilizing. This type of long-range sequence coupling should serve to reinforce the repetitive nature of ankyrin sequence motif. The approach described here provides a means to identify other positions in the 33-residue ankyrin motif that are similarly coupled.

7. SUMMARY

Understanding cooperativity in protein folding, which is determined by the distribution of local and long-range stabilities, is likely to have important functional consequences in terms of avoiding misfolding and in allosteric functional transitions. The analysis of tandem repeat protein arrays with a 1D Ising model provides a unique approach to quantifying cooperativity in terms of local stabilities and long-range coupling free energies. The collection of programs presented here, which are available on github, provide a robust platform for data processing, fitting, plotting, and analysis of parameter uncertainties and correlations. Given the complexity of the models, careful inspection of fitted parameter uncertainties is an essential part of the analysis. Models can easily be adapted to evaluate the effects of point-substitutions in intrinsic and interfacial energies, providing insights into the structural basis of cooperativity.

Acknowledgments

KG, JM, SK, and MP were supported by NIH training grant 2T32 GM008403. KS was supported by NIH training grant 5T32 GM007231. This work was supported by NIH grant R01 GM068462 to DB. The authors declare no conflict of interest.

REFERENCES

Aksel, T., and Barrick, D. (2009). Analysis of repeat-protein folding using nearest-neighbor statistical mechanical models. Methods Enzymol. *455*, 95–125.

Aksel, T., and Barrick, D. (2014). Direct observation of parallel folding pathways revealed using a symmetric repeat protein system. Biophys. J. *107*, 220–232.

Aksel, T., Majumdar, A., and Barrick, D. (2011). The contribution of entropy, enthalpy, and hydrophobic desolvation to cooperativity in repeat-protein folding. Struct. Lond. Engl. 1993 *19*, 349–360.

Alber, T., and Matthews, B.W. (1987). [27] Structure and thermal stability of phage T4 lysozyme. In Methods in Enzymology, (Academic Press), pp. 511–533.

Cortajarena, A.L., Mochrie, S.G.J., and Regan, L. (2011). Modulating repeat protein stability: The effect of individual helix stability on the collective behavior of the ensemble. Protein Sci. *20*, 1042–1047.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. Ann. Stat. *7*, 1–26.

Fersht, A.R., Matouschek, A., and Serrano, L. (1992). The folding of an enzyme: I. Theory of protein engineering analysis of stability and pathway of protein folding. J. Mol. Biol. *224*, 771–782.

Geiger-Schuller, K., and Barrick, D. (2016). Broken TALEs: Transcription Activator-like Effectors Populate Partly Folded States. Biophys. J. *111*, 2395–2403.

Greene, R.F., and Pace, C.N. (1974). Urea and guanidine hydrochloride denaturation of ribonuclease, lysozyme, alpha-chymotrypsin, and beta-lactoglobulin. J. Biol. Chem. *249*, 5388–5393.

Johnson, M.L. (1983). Evaluation and propagation of confidence intervals in nonlinear, asymmetrical variance spaces. Analysis of ligand-binding data. Biophys. J. *44*, 101–106.

Johnson, M.L. (2008). Nonlinear Least-Squares Fitting Methods. In Methods in Cell Biology, (Academic Press), pp. 781–805.

Kajander, T., Cortajarena, A.L., Main, E.R.G., Mochrie, S.G.J., and Regan, L. (2005). A New Folding Paradigm for Repeat Proteins. J. Am. Chem. Soc. *127*, 10188–10190.

Lowe, A.R., Perez-Riba, A., Itzhaki, L.S., and Main, E.R.G. (2018). PyFolding: Open-Source Graphing, Simulation, and Analysis of the Biophysical Properties of Proteins. Biophys. J. *114*, 516–521.

Marold, J.D., Kavran, J.M., Bowman, G.D., and Barrick, D. (2015). A Naturally Occurring Repeat Protein with High Internal Sequence Identity Defines a New Class of TPR-like Proteins. Struct. Lond. Engl. 1993 *23*, 2055–2065.

Matt Newville, Renee Otten, Andrew Nelson, Antonino Ingargiola, Till Stensitzki, Dan Allan, Austin Fox, Faustin Carter, Michał, Dima Pustakhod, et al. (2019). lmfit/lmfit-py 0.9.13 (Zenodo).

Matthews, C.R., and Hurle, M.R. (1987). Mutant sequences as probes of protein folding mechanisms. BioEssays *6*, 254–257.

Mello, C.C., and Barrick, D. (2004). An experimentally determined protein folding energy landscape. Proc Natl Acad Sci U A *101*, 14102–14107.

Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., Kumar, Am., Ivanov, S., Moore, J.K., Singh, S., et al. (2017). SymPy: symbolic computing in Python. PeerJ Comput. Sci. *3*, e103.

Mosavi, L.K., Minor, D.L., and Peng, Z.Y. (2002). Consensus-derived structural determinants of the ankyrin repeat motif. Proc Natl Acad Sci U A *99*, 16029–16034.

Perez, F., and Granger, B.E. (2007). IPython: A System for Interactive Scientific Computing. Comput. Sci. Eng. *9*, 21–29.

Preimesberger, M.R., Majumdar, A., Aksel, T., Sforza, K., Lectka, T., Barrick, D., and Lecomte, J.T.J. (2015). Direct NMR Detection of Bifurcated Hydrogen Bonding in the α-Helix N-Caps of Ankyrin Repeat Proteins. J. Am. Chem. Soc. *137*, 1008–1011.

Randles, B.M., Pasquetto, I.V., Golshan, M.S., and Borgman, C.L. (2017). Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study. In 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), pp. 1–2.

Vallat, R. (2018). Pingouin: statistics in Python. J. Open Source Softw. *3*, 1026.

Wetzel, S.K., Settanni, G., Kenig, M., Binz, H.K., and Plückthun, A. (2008). Folding and unfolding mechanism of highly stable full-consensus ankyrin repeat proteins. J. Mol. Biol. *376*, 241–257.

(2016). Anaconda Software Distribution.

Table 1. Adjustable parameters for capped homopolymer and heteropolymer models

Global parameters

Model 1: capped homopolymeric repeat protein

Parameter	Description	Number
$\Delta G_{N,H2O}$	Intrinsic folding energy, repeat N	1
$\Delta G_{R,H2O}$	Intrinsic folding energy, repeat R	1
$\Delta G_{C,H2O}$	Intrinsic folding energy, repeat C	1
$\Delta G_{R-1,R}$	Interfacial coupling energy	1
m_R	Intrinsic denaturant dependence	1

Model 2: capped heteropolymeric repeat protein

Parameter	Description	Number
$\Delta G_{N,H2O}$	Intrinsic folding energy, repeat N	1
$\Delta G_{R,H2O}$	Intrinsic folding energy, repeat R	1
$\Delta G_{X,H2O}$	Intrinsic folding energy, repeat X	1
$\Delta G_{C,H2O}$	Intrinsic folding energy, repeat C	1
$\Delta G_{R-1,R}$	Interfacial coupling energy (R, R)	1
$\Delta G_{X-1,X}$	Interfacial coupling energy (X, X)	1
$\Delta G_{R-1,X}$	Interfacial coupling energy (R, X)	1
$\Delta G_{X-1,R}$	Interfacial coupling energy (X, R)	1
m_R	Intrinsic denaturant dependence (R)	1
m_X	Intrinsic denaturant dependence (X)	1

Local parameters (both models)

$a_{n,q}$	Native baseline intercept	q
$b_{n,q}$	Native baseline slope	q
$a_{d,q}$	Denatured baseline intercept	q
$b_{d,q}$	Denatured baseline slope	q
	Model 1 total parameters	5+4M
	Model 2 total parameters	10+4M

Table 2. Fitted global thermodynamic parameters and bootstrap statistics

		Bootstrap statistics [b]					
Parameter[a]	Best fit value	Mean	Standard deviation	lower 95% CI [c]	lower 67% CI [c]	upper 67% CI [c]	upper 95% CI [c]
Model 1							
ΔG_N	5.47	5.41	0.062	5.28	5.34	5.46	5.52
ΔG_R	4.57	4.54	0.060	4.43	4.48	4.60	4.66
ΔG_C	7.06	7.01	0.075	6.87	6.94	7.09	7.16
$\Delta G_{R-1,R}$	-11.63	-11.58	0.12	-11.81	-11.70	-11.46	-11.34
m_R	-0.78	-0.78	0.0073	-0.764	-0.773	-0.785	-0.791
Model 2							
ΔG_N	4.89	4.89	0.064	4.77	4.82	4.95	5.01
ΔG_R	4.08	4.08	0.065	3.96	4.02	4.14	4.22
ΔG_X	5.64	5.64	0.091	5.46	5.55	5.73	5.82
ΔG_C	6.34	6.34	0.073	6.20	6.27	6.41	6.49
$\Delta G_{R-1,R}$	-10.46	-10.46	0.12	-10.71	-10.58	-10.36	-10.25
$\Delta G_{X-1,X}$	-10.16	-10.17	0.12	-10.39	-10.29	-10.05	-9.94
$\Delta G_{R-1,X}$	-9.42	-9.42	0.11	-9.64	-9.52	-9.33	-9.22
$\Delta G_{X-1,R}$	-10.82	-10.83	0.12	-11.07	-10.95	-10.72	-10.59
m_R	0.71	0.708	0.0071	0.69	0.70	0.71	0.72
m_X	0.97	0.98	0.021	0.93	0.95	0.10	1.02

[a] ΔG values in kcal mol^{-1}; *m*-values in kcal mol^{-1} M denaturant^{-1}. [b] Values from 1000 bootstrap iterations. [c] CI, confidence intervals.

Table 3. Bootstrap parameter correlation coefficients for Model 1 [a]

Parameters	ΔG_N	ΔG_R	ΔG_C	$\Delta G_{R-1,R}$	m_i
ΔG_N	1	0.941	0.949	-0.967	0.826
ΔG_R	-0.710	1	0.917	-0.947	0.721
ΔG_C	-0.485	-0.699	1	-0.981	0.891
$\Delta G_{R-1,R}$	-0.782	-0.986	-0.766	1	-0.909
m_R	-0.753	-0.988	-0.689	-0.989	1

[a] Values from 1000 bootstrap iterations. Values above the diagonal are Pearson correlation coefficients between pairs of variables. Values below the diagonal (in red) are Pearson partial correlation coefficients between pairs of variables, where the effects of the other variables in the table are factored out (see text).

Appendix I. Correlation between fitted parameters in simple two-state unfolding analysis

The Ising analysis of repeat protein folding transitions above shows that there are significant correlations between fitted parameters, based on bootstrap analysis. For example, in the homopolymer fit using model 1, there are strong direct correlations between ΔG_R and $\Delta G_{R-1,R}$ (Table 3). Here we will use the Ising programs outlined in Figure 1 along with melts from a single construct to demonstrate that this type of correlation is not unique to multiparameter Ising fits, but is also endemic (but perhaps often overlooked) in fitting of simple two-state unfolding transitions.

To directly investigate correlation using a two-state model, the Ising fitting procedure outlined in Figure 1 was performed using two melts from just one construct, NRC. To fit with a two-state model, the Aviv .dat files were renamed R_1.dat and R_2.dat. As a result, *generate_fitting_equations.py* builds a single partition function of the form

$$q = 1 + \kappa_R = 1 + e^{-(\Delta G_R + m_R[x])/RT} \tag{A.1}$$

Differentiating with respect to κ_R and converting the result to an expression for Y_{calc} (equations 5 and 7) gives a simple two-state model that can be fitted to the renamed NRC melts and subjected to two-parameter (ΔG_R and m_R) bootstrap analysis (Figure A.1).

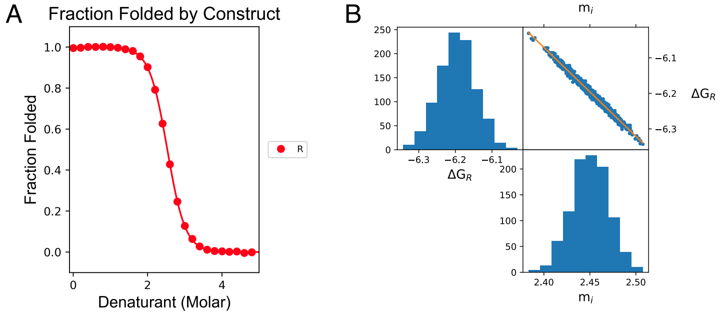


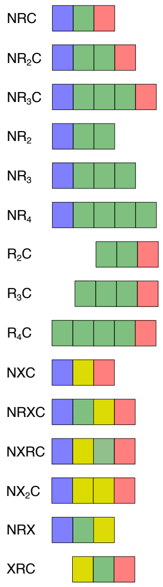
Figure A.1. Parameter correlation in a simple two-state fit. NRC (Figure 1A) was fitted with a two-state model (deriving from equation A.1) using the Ising programs described here. (A) The NRC folding transition is well-fitted by the two-state model. (B) Bootstrap analysis (1000 iterations) shows that the two fitted thermodynamic parameters are determined with narrow confidence intervals, but they are tightly correlated.

The fits are excellent (Figure A.1A), and bootstrapped ΔG_R and m_R-values are determined with narrow confidence intervals (within 1 to 2 percent). However, these bootstrapped parameters are very strongly correlated (Figure A.1B): the correlation coefficient between ΔG_R and m_R is -0.993. Note that because there are just two thermodynamic parameters in the correlation analysis, partial and total correlation coefficients are equivalent.

Appendix II. Rank analysis of the thermodynamic parameters in model 2

To accurately determine intrinsic and interfacial free energies from Ising analysis of repeat protein folding, the constructs being analyzed must present enough structural variation to accurately determine the model's parameters. For example, model 2 has eight free energies (four intrinsic folding energies and four interfacial energies). To accurately determine these parameters, the set of constructs analyzed (the combined constructs in Figure 1A and B for the analysis described here) must provide uncorrelated variations in structural features that determine each parameter. For complex models and large data sets, it is not always obvious that a given data set is adequate to determine the parameters in a given model.

Here we present a simple method to check whether a data set provides enough structural variation to adequately constrain the parameters for a specific model. Since the folding free energy for each construct is expressed as a sum of Ising parameters, the compatibility of a thermodynamic model can be tested by evaluating the set of linear free energy equations defined by the model for each of the constructs being analyzed (Aksel and Barrick, 2009). The constructs analyzed using model 2 define 15 linear free energy equations (Figure A.2). If the system of equations has an empty null space, then the system of equations has a unique solution. An easy way to test whether the null space of a system of equations is empty is to compute the matrix rank. If the rank of a coefficient matrix is equal to the number of columns in the matrix, the null space is empty and a unique solution to the system of equations exists. The *generate_fitting_equations* program performs this rank analysis, and reports the rank (and its sufficiency/insufficiency) during run time. In Figure A.2, which gives the free energy equations for the constructs used for Ising analysis with model 2 in matrix form, the coefficient matrix has a full column rank of 8. Thus, this set of constructs adequately defines the unknown parameters in model 2.



$$\begin{bmatrix} 1 & 1 & 0 & 1 & 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 3 & 0 & 0 & 0 \\ 1 & 3 & 0 & 1 & 4 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 3 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 & 3 & 0 & 0 & 0 \\ 0 & 4 & 0 & 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta G_{N} \\ \Delta G_{R} \\ \Delta G_{X} \\ \Delta G_{C} \\ \Delta G_{R-1,R} \\ \Delta G_{X-1,X} \\ \Delta G_{R-1,X} \\ \Delta G_{X-1,R} \end{bmatrix} = \begin{bmatrix} \Delta G_{NRC} \\ \Delta G_{NR_2C} \\ \Delta G_{NR_3C} \\ \Delta G_{NR_2} \\ \Delta G_{NR_3} \\ \Delta G_{NR_4} \\ \Delta G_{R_2C} \\ \Delta G_{R_3C} \\ \Delta G_{R_4C} \\ \Delta G_{NXC} \\ \Delta G_{NRXC} \\ \Delta G_{NXRC} \\ \Delta G_{NX_2C} \\ \Delta G_{NRX} \\ \Delta G_{XRC} \end{bmatrix}$$

Figure A.2. A matrix representation of the equations for repeat protein stability with the Ising model. The constructs that are used to fit the parameters from model 2 are shown on the left. These constructs define a set of 15 linear equations in which the free energies of the fully folded state relative to the unfolded state (the vector on the right-hand side) can be represented as a sum of intrinsic and interfacial free energies (the 8 by 1 vector on the left-hand side), weighted by the number of times each term is represented in a given construct (the 8 by 15 coefficient matrix). To have a unique solution, each column in the coefficient matrix must be independent of the others, that is, the matrix must have full column rank (eight in the example here).

In contrast, a model that treats interfaces between X-repeats and capping repeats as unique (with terms ΔG_{NX} and ΔG_{XC}) cannot be uniquely fitted by the data set in Figure A.2. The corresponding coefficient matrix has ten columns, but the rank remains eight. Although the Ising fitter converges on a least-squares solution for the ten-repeat model, the fitted parameter values differ significantly from values from simpler models, and the confidence intervals on many of the parameters are extremely broad. Such cases require either simplification of the model (to model 2, for example), or in favorable cases, inclusion of additional constructs that provide the missing structural variation.