README

Distance determination in multimeric membrane proteins by symmetry-constrained analysis of pulsed double electron-electron resonance (DEER) spectroscopy

Constrained 2-distance fit and statistical analysis features were developed and written by:

H. Clark Hyde, Ph.D.
The University of Chicago
Postdoctoral Scholar
hchyde@uchicago.edu
Dec. 14th, 2011

slightly abridged form of the original document – edited by G. Jeschke, Aug. 12th 2013 to reflect actual changes in the main distribution of DeerAnalysis

Summary of changes in DeerAnalysis 2011con

This version of the DeerAnalysis software introduces the use of nonlinear constraints into user model fits. Unlike a linear inequality constraint that enforces limits (bounds) on an individual parameter, a nonlinear inequality constraint allows one to place limits on the interaction between multiple parameters during the fit. If information about parameter interaction is well known in the system under study, application of nonlinear constraint(s) can 1) direct convergence of the fit solution to a set of model parameters that coincides with a physical model of the system, and 2) statistically verify agreement between a fitted model and the physical model.

Here, we have primarily added the ability to perform constrained 2-distance (bimodal) distribution model fits where the ratio of mean distances is constrained within user-defined limits [kmin, kmax] relevant to known symmetry of the protein under investigation. The mean distance ratio is *k = <r2>/<r1>* for a 2-Gaussian model and *k = <nu2>/<nu1>* for a 2-Rice model, where *<r>* and *<nu>* are fitted mean distances. In our implementation, the distance ratio *k* is a nonlinear inequality constraint, although it can effectively be used as a nonlinear equality constraint if lower and upper limits are set equal. This constrained fit option currently only applies to the 2-distance user fit models: “Two_Gaussians.m”, “Two_Gaussians_hom.m”, and “Two_Rice3d.m”. Statistical results are displayed in the Matlab workspace, and are also written to the results output file during a standard “Save” operation, to help the user evaluate statistical significance between fits using different models or different constraints within the same model (e.g., constrained *vs.* unconstrained fit). Note that our use of “constrained” and “unconstrained” only refers to the presence or absence of nonlinear parameter constraints, respectively, regardless of use of conventional linear constraints on individual parameters.

Other minor additions were made to improve the stability of the graphical user interface (GUI), especially prior to loading a dataset, i.e., prevent errors when buttons are pushed with no dataset loaded. Also, minor adjustments were made to the GUI figure (DeerAnalysis.fig) to improve the appearance of objects. The suffix ‘con’ was added to the version year of the software folder name (2011con) to denote that this is the 2011 version with “constrained” fit options. Note that the software implementation presented here is specific to our problem. However, based upon your feedback, we will consider to enhance and expand our current implementation to 1) allow user-defined nonlinear constraints to be applied to any user model simply by writing constraints into the respective model m-file, and 2) remove all hard-coding of fit model names from main program m-files.

How to use constrained 2-distance model fits

We have provided an example dataset “deer_252cl_113scans.DTA” obtained from a homopentameric protein spin-labeled at the same site within each subunit, thus reporting 2 unique distances (adjacent and diagonal). Load the dataset and preprocess it by pressing all of the default (!) buttons within the “Original data” panel. In the “Distance analysis” panel, enable the “Model fit” button and select the “Two_Gaussians” model, which will be used for demonstration. Within the “Model fit” subpanel, notice that in addition to the 5 parameters that describe this model, [*<r1>, s(r1), <r2>, s(r2), p1*], there are 2 additional parameter objects that are enabled: the mean distance ratio constraint limits [kmin, kmax]. Note that the mean distance ratio was defined as *k = <r2>/<r1>*, where *<r1>* and *<r2>* are the mean distances of a bimodal distance distribution.

Different from the effect on model parameters (fixed *vs.* variable), the parameter checkbox located next to each constraint limit toggles its state. The constraint limit is enabled (active) if checked or disabled (inactive) if unchecked. Thus, if both the kmin and kmax constraint checkboxes are disabled, the model fit will proceed as the standard 2-Gaussian model fit (no nonlinear constraints). However, if one or more constraints are enabled, the 2-Gaussian model fit will be constrained such that the mean distance ratio *k = <r2>/<r1>* is maintained within the active constraint limits. Note that *<r1>* and *<r2>* are free to optimize to any values within their respective linear bounds, provided that the *<r2>/<r1>* constraint is simultaneously satisfied.

The theoretical mean distance ratio *k = <r2>/<r1>* is $(1+\sqrt{5})/2 \approx 1.618$ for the example homopentameric protein dataset. First perform an unconstrained model fit: disable (uncheck) both constraint checkboxes and press the model “Fit” button. The unconstrained 2-Gaussian fit results are:

```
Fit: (Two_Gaussians) [p=5,v=130,n=135] <r2>/<r1> = 1.5861
RMS  = 0.0082903
RMSt = 0.0086468 [by F-test for F(v1=5,v2=130,1-0.05)]
AICc = -1282.3680463
```

In the Matlab workspace, 3-4 output lines are displayed with fit results: line 1 reports the fit type/name, statistical parameters (number of fitted parameters *p*, degrees of freedom *v*, number of data observations *n*), and the *<r2>/<r1>* distance ratio (if applicable). Line 2 reports the root mean square (RMS) error of the fit. If a model fit is performed without nonlinear constraints, line 3 reports the RMS threshold (RMSt) value with corresponding *F* distribution parameters: number of degrees of freedom in the numerator *v1*, number of degrees of freedom in the denominator *v2*, and the upper percentile (1-α). We have used the likelihood-ratio F-test to calculate the RMS threshold, which allows direct statistical comparison with this same model fit with nonlinear constraints imposed. The final line (3 or 4) reports the corrected Akaike Information Criterion (AICc) score, which can be used for statistical comparison of multiple fits (nested or non-nested).

Next, set the mean distance ratio constraint limits tightly around the theoretical value: set kmin = 1.60, kmax = 1.64. Now enable (check) both constraint checkboxes (kmin and kmax) and press the model “Fit” button. The constrained fit results are:

```
Fit: (Two_Gaussians +NLcon) [p=5,v=130,n=135] <r2>/<r1> = 1.6000
RMS  = 0.0083306
AICc = -1281.0582743
```

The unconstrained (1st) fit converged to the optimal parameter set with *<r2>/<r1>* = 1.5861. The constrained (2nd) fit approaches this optimal mean distance ratio until it was stopped at the minimum limit: *<r2>/<r1>* = kmin = 1.60. The constrained fit RMS error is higher because the fit was not allowed to reach the optimal parameter set. However, the constrained fit RMS is below the F-test RMS threshold (RMSt), therefore we conclude that the constrained fit parameter set is not significantly different than the unconstrained fit parameter set at the probability level P = 0.05, same as significance level α = 0.05 (see the “Statistical analysis” section). For a given model fit type, the constrained fit RMS error is always higher than (or equal to) the unconstrained fit RMS error, therefore this F-test is always one-tailed.

For comparison, the results of a single Tikhonov fit with regularization parameter $\lambda = 100$ are shown below. Notice that an effective "_eff" suffix has been added to symbols for the number of fitted parameters (*p*) and degrees of freedom (*v*), which is discussed in the "Statistical analysis" section below. The Tikhonov fit results are:

```
Fit: (Tikhonov: Reg.par.=100) [p_eff=57,v_eff=78,n=135]
RMS  = 0.0083372
AICc = -1087.4471769
```

Save the results output file:

The "Save" button, located in the GUI "Data sets" panel, writes several data files for the current model fit, including a formatted text file with model fit results information (e.g., deer_252cl_113scans_res.txt). To display properly, open this file using the WordPad application. The files are saved to the user-selected directory. A new section at the bottom of the file was added: ### Extra fit output ###. It contains the ratio of distances for 2-distance type model fits and other parameters needed for application of the likelihood-ratio F-test. For all model fits and Tikhonov fits, the number of data observations, fitted parameters, and (effective) degrees of freedom are now displayed in the results output file.

Statistical analysis:

We have implemented two statistics for use in DeerAnalysis. The first statistic is the likelihood-ratio test. It is described in the manuscript and its supporting information. It can be used either as an F-test or for construction of confidence regions for nonlinearly constrained model fits.

The second statistic is the corrected Akaike Information Criterion (AICc). At this point, it is not discussed in the manuscript. For model fits, it is a powerful tool for simultaneous comparison of multiple nested or non-nested models. For Tikhonov regularization, its use is questionable for reasons explained below and is only shown here for demonstration purposes and to prompt your feedback.

Comparison of constrained vs. unconstrained fits with same user fit model

Use the likelihood-ratio method as a one-tailed F-test to determine if the parameter set returned by a constrained model fit is within the confidence region for an unconstrained fit of the same model. The equation and test details are discussed in the manuscript and its supporting information. For a model with *p* parameters, the confidence region is a *p*-dimensional surface of constant fit error (e.g., chi-square or RMS) and its size is determined by the number of data observations (*n*), the number of fitted parameters (*p*), and the significance level (α). We refer to this error level that defines the surface as the error threshold (e.g., RMSt). The probability level P (same as significance level α) is user-defined at the top of the "stats_analysis.m" m-file. Typical values are P=0.32 (1-sigma, 68%), P=0.05 (2-sigma, 95%), P=0.01 (99%), and P=0.003 (3-sigma, 99.7%). As a default, we use P=0.05, which is most conventional and also seems to be most appropriate for our DEER datasets.

The F-test is straightforward. First perform an unconstrained model fit (all nonlinear constraints disabled). The fit RMS and also RMS threshold (RMSt) values are reported in the workspace. Without modifying the dataset in any way, next perform a constrained fit (constraints enabled). If RMS(constrained) ≤ RMSt, we accept the null hypothesis that the unconstrained and constrained models have equivalent parameter sets at the probability level P. If RMS(constrained) > RMSt, we accept the alternative hypothesis that the unconstrained and constrained models have significantly different parameter sets at the probability level P. If the unconstrained model fits the dipolar evolution well, but the constrained model reaches significance, this suggests that the given nonlinear constraint is not valid for this dataset, this model, etc. Note that addition of nonlinear constraints does not change the number of fitted parameters. This F-test is only valid for comparison of an unconstrained (best-fit) and constrained fit of the *same model* and *same dataset*.

The probability level P is set in the m-file 'stats_analysis.m' at/near line 8. The default value is: P = 0.05. Probability values should not exceed 0.32. You can manually set other values by changing the code and saving the m-file. All statistical results are saved within the handles structure as: handles.model_stats. Use handles.model_stats to access these results for output file, workspace display, etc.

Comparison of multiple non-nested (or nested) models

Use the corrected Akaike Information Criterion (AICc) for comparison of models that were fit to the same dataset. The AICc is used to compare the relative likelihood of 2 or more models, however it does not accept or reject a model at a statistical significance level and there is no reported P value. The model with the lowest AICc value is most likely to be correct. The corrected AICc is given by:

AICc = N*ln(rms/N) + 2*K + 2*K*(K+1)/(N-K-1)

where N = number of independent data observations, and K = p+1, given p fitted model parameters. The last (right-hand side) term is the correction term.

Given all AICc scores from multiple model fits, it is simple to compute the probability (Akaike weight) that each of the tested models is correct. The AICc comparison is advantageous because there is no requirement for the models to be nested (which allows comparison between a Tikhonov regularization and 2-Gaussian model for example), and more than 2 different models can be compared simultaneously. AICc is more appropriate than AIC when N is small or K is large (approx. N/K < 40), however AICc converges to AIC as N/K increases. Since most fits of dipolar evolution data are below this threshold, we only report the more reliable AICc score. An example of calculating the Akaike weights and evidence ratios for model comparison is given in ‘stats_analysis.m’ function comments.

The application of AICc to analytical model fits is straightforward and correct. However, one must be careful in the interpretation of the Tikhonov regularization fit where the number of degrees of freedom and free parameters is not well-defined. The Tikhonov regularization fit should be first optimized through use of the L-curve series of fits as described in the DeerAnalysis user manual. After the optimal regularization parameter is identified that is most appropriate to fit the data, select or perform a single Tikhonov fit (no L-curve) at the optimal regularization parameter and save the fit result by pressing the “Save” button. For statistical comparison to model fits using AICc, the Tikhonov fit RMS value and number of effective degrees of freedom are displayed in the workspace and can also be obtained from the saved (.txt) fit results output file.

For now, I have interpreted the “effective” number of free parameters p_{eff} as the “NUMBER OF FREE PARAMETERS” output reported in the FTIKREG results log. In this sense, the “effective” degrees of freedom is v_{eff} = N-$p_{\text{eff.}}$ In the AICc equation above, K = p_{eff} +1. Clearly, this approach of using p_{eff} for Tikhonov regularization would only be valid at the optimal λ, determined by the L-curve criterion. The AICc cannot be used to determine optimal λ! We present the AICc calculation for Tikhonov regularization only as a demonstration of the idea and request your critical feedback. Note that further validation of its use as presented here is required.

Per Jeschke, et al. (2006) “DeerAnalysis2006 - A comprehensive software package for analyzing pulsed ELDOR data”, the number of free parameters for Tikhonov regularization is the number of distance *r* values at which *P(r)* is defined. By this definition, the number of free parameters, and thus degrees of freedom, is dependent upon the fitted distance range [Rmin, Rmax], which is not true for analytical model fits. Consider that a small change in the distance range will significantly alter the Tikhonov AICc value! Based upon this critical difference, it seems that the use of free parameters as interpreted above for Tikhonov regularization is highly questionable for statistical model comparison. This raises the question of how to correctly acquire the number of fitted parameters and degrees of freedom for Tikhonov regularization. From the literature on spline theory, namely the generalized cross-validation method, the *effective degrees of freedom* is typically defined as the *trace* of the smoother or hat matrix: *tr*[S(λ)] or *tr*[H(λ)]. However I am unable to perform this computation blind to the inner workings of the FTIKREG

Fortran program. Perhaps the FTIKREG number of free parameters is related to the *effective degrees of freedom* v_{eff} as: p_{eff} = N-v_{eff} = N-tr[H(λ)], with *n* data observations? The known trend is that as the smoothing parameter λ increases, the degrees of freedom decreases, and this trend is observed as implemented here. However, the dependence upon the distance range [Rmin, Rmax] is highly troublesome. Therefore, your feedback in this regard would be warmly appreciated. On a more fundamental level, do you think that the use of the AICc statistic is of any value for comparison of Tikhonov regularization to analytical model fits?

Finding the L-curve corner for Tikhonov regularization (not included in DeerAnalysis2013.2)

I noticed that DeerAnalysis 2011 uses the graphical corner (closest to origin) method. This was the first method proposed on page 1494 in Hansen, P.C. & O'Leary, D.P. (1993), "The use of the L-curve in the regularization of discrete ill-posed problems", SIAM J. Sci. Comput., 14(6), pp. 1487-1503. After reviewing the literature on calculation of the L-curve corner, I decided to implement their second method of maximum curvature (same reference). The result is the new function 'get_l_corner_maxcurve.m' which combines both methods. If the Spline Toolbox is available, the maximum curvature method is used by default, otherwise the graphical corner (closest to origin) method is used. The maximum curvature code was adapted from the 'Regularization Tools v4.1' Matlab software package by Per Christian Hansen. All references and comments are contained in the function header. This new function is now called at line 186 in 'fit_Tikhonov.m'. It also has a plot option at line 185 to generate a plot showing the L-curve with results of both methods as well as a plot of the curvature vs. the residual norm. The plot option is currently turned on so you can see its use, but is easily turned off at line 185. I thought that this combination would be helpful for users to compare and visualize two classic methods.

Software-specific implementation topics

The mean distance constraint limits were added to the subpanel used for model fit parameters to avoid disruption of the GUI layout, although an independent constraint subpanel could be implemented in the future. The mean distance ratio constraint limits are available because they have been added to the header of the fit model m-file (Two_Gaussians.m), modified as follows:

```
% PARAMETERS
% name    symbol default lower bound upper bound
% par(1)  <r1>   2.5     1.5         10          1st mean distance
% par(2)  s(r1)  0.5     0.05        5           std. dev. of 1st distance
% par(3)  <r2>   3.5     1.5         10          2nd mean distance
% par(4)  s(r2)  0.5     0.05        5           std. dev. of 2nd distance
% par(5)  p1     0.5     0           1           fraction of pairs at 1st dist
% par(6)  kmin   1.00    0           Inf         <r2>/<r1> minimum ratio
% par(7)  kmax   2.00    0           Inf         <r2>/<r1> maximum ratio
```

The last two lines of the header, within the PARAMETERS section, are new and introduce the mean distance ratio constraint limits using parameters #6-7: par(6) and par(7) for kmin and kmax, respectively. The default, lower, and upper bound values are set in the same way as for model parameters. If these two header lines are removed from the model fit m-file, the mean distance ratio constraints will not be available in the program. The same modifications discussed here also apply to the homogeneous 2-Gaussian (Two_Gaussians_hom.m) and 2-Rice (Two_Rice3d.m) model m-files.

Within the PARAMETERS section, the fraction of pairs at 1st distance (*p1*) parameter was moved in the 2-Gaussian and 2-Gaussian homogeneous models to be consistent with the 2-Rice model. The parameter *p1* was moved from the par(3) to par(5) position with the 2nd distance parameters correspondingly raised up by one level. The benefit of this change is that the distance mean and standard deviation parameters align vertically for easier reading in the "Model fit" subpanel.

The choice of initial conditions is important to consider when imposing a nonlinear constraint that involves two or more parameters. If initial conditions for distances *<r1>* and *<r2>* are incompatible with the specified mean distance ratio constraint limits, the distances *<r1>* and *<r2>* initial conditions are automatically adjusted (in proportionately equal steps) until first compatible, and then the fit commences. The mean distance ratio constraint limits must be in ascending order (kmin ≤ kmax). If this condition is not met initially, the limits are reversed before proceeding with the fit. Note that a fixed distance ratio (kmin = kmax) is allowed so the user can effectively set a nonlinear equality constraint, which allows one to easily construct an error surface for the nonlinear parameter.

The default function used in DeerAnalysis to fit a model is the built-in Matlab function FMINSEARCH. However, this function does not allow nonlinear constraints, therefore the built-in Matlab function FMINCON was used to perform nonlinearly constrained model fits. Appropriate optimization options are used to promote reliable convergence. During a constrained fit, it is possible to fix the value of one or more model parameters (however fixing both distances *<r1>* and *<r2>* is obviously not allowed and will return an error message in the GUI “Status” box).

For application of the likelihood-ratio F-test, one must know the number of degrees of freedom for both fits under comparison. The number of degrees of freedom is given by $\upsilon = N\text{-}p$, where N is the number of independent data observations in the dipolar evolution, and p is the number of fitted (variable) parameters. For model fits, the number of fitted parameters is readily available; however this is not true for fits by Tikhonov regularization. For this purpose, code was added to extract the number of effective free parameters from the FTIKREG results log that is created after each Tikhonov fit.

Modified Models	Description of changes
Two_Gaussians.m	added mean distance ratio constraint parameters: kmin, kmax; rearranged parameter order to match the format of ‘Two_Rice3d.m’ (the fraction *p1* is now parameter #5, positioned after both distance parameters)
Two_Gaussians_hom.m	added mean distance ratio constraint parameters: kmin, kmax; rearranged parameter order to match the format of ‘Two_Rice3d.m’ (the fraction *p1* is now parameter #5, positioned after both distance parameters)
Two_Rice3d.m	added mean distance ratio constraint parameters: kmin, kmax;

M-files that currently contain “hard-coded” user model names for constrained fits:

fit_user_model.m
update.m
save_result.m
rms_user_model_con.m
stats_analysis.m