MemExp Documentation

Peter J. Steinbach
Center for Molecular Modeling
Center for Information Technology
National Institutes of Health

April 9, 2002

Recovering Distributed Lifetimes and Discrete Exponentials from Kinetics
Documentation for MemExp version 1.0

The program MemExp uses the maximum entropy method (MEM) and nonlinear least squares (NLS) fitting to analyze a general time-dependent signal in terms of distributed and discrete lifetimes. One or two distributions of effective log-lifetimes, $g(log\tau)$ and $h(log\tau)$, plus an optional polynomial baseline (up to a cubic) can be extracted from the data. The h distribution is used to account for signals opposite in sign to those described by the g distribution when analyzing data that rise and fall. Both distributions are obtained numerically from the data and are not restricted to any functional form. Simulataneously, MemExp performs a series of fits by discrete exponentials in which exponentials are added one at a time and are initialized based on the emerging structure in the MEM distribution. The amplitude and log-liftime of each exponential, plus any optional baseline parameters utilized, are varied using nonlinear least-squares (NLS) fitting. MemExp automatically recommends one distributed and one discrete description of the kinetics as optimal. The graphical summary plotted by MemExp permits a thorough evalutaion of the results. Multiple MEM 'priors' are supported, facilitating a comprehensive analysis of the kinetics data.

MemExp was written in FORTRAN-77 and has been built under several operating systems. Graphical output is in PostScript format.

Reference: Please read the following paper before using MemExp and reference it when publishing results obtained using MemExp.
P.J. Steinbach, R. Ionescu, and C.R. Matthews. Analysis of Kinetics using a Hybrid Maximum-Entropy / Nonlinear-Least-Squares Method: Application to Protein Folding. (2002) Biophys. J. 82: 2244-2255.

See the documentation online at http://cmm.cit.nih.gov/memexp/doc for sample input and output.

1 MemExp: Fits by the MEM and by NLS

The MEM

The MemExp program makes synergistic use of both the maximum entropy method (MEM) and nonlinear least squares (NLS) fitting to interpret kinetics data from the distributed and discrete perspectives. Its hybrid algorithm has been described in the literature [1]. Additional algorithmic details and applications of the MEM can be found elsewhere [2, 3, 4].

MemExp fits kinetics using one or two liftetime distributions:

$$\mathcal{F}_i = D_0 \int_{-\infty}^{\infty} dlog\tau (g(log\tau) - h(log\tau)) e^{-t_i/\tau} + \sum_{k=0}^{3} (b_k - c_k)(t_i/t_{max})^k, \tag{1}$$

where $g(log\tau)$ and $h(log\tau)$ are the distributions describing decaying and rising kinetics, respectively, and the polynomial describes the baseline. Baseline coefficients are scaled using the constant t_{max} so they remain comparable in magnitude.

The maximum entropy method (MEM) is used to obtain this fit $\mathcal{F}$ by distributions. A MEM calculation is a 'tug of war' between two or three terms, depending on whether or not the normalization is to be constrained. These terms are summed in the function Q,

$$Q \equiv S - \lambda\chi^2 - \alpha I, \tag{2}$$

which is to be maximized. S is the entropy of the 'image' f,

$$S(\vec{f}, \vec{F}) = \sum_{j=1}^{M} [f_j - F_j - f_j ln(f_j/F_j)]. \tag{3}$$

Here, the image f includes both the g and h lifetime distributions that describe kinetic processes of opposite sign as well as any baseline parameters used. In this documentation, f is alternatively referred to as the *image* or the *distribution* obtained by the MEM. F is the MEM 'prior' distribution used to incorporate prior knowledge into the solution. Unconstrained maximization of S without regard for the fit or normalization ($\lambda = \alpha = 0$) yields $f_j = F_j$.

The quality of the fit $\mathcal{F}$ to the data D produced by f is measured by

$$\chi^2 = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\mathcal{F}_i - D_i}{\sigma_i} \right)^2. \tag{4}$$

The image normalization is given by

$$I = \delta \left(\sum_{j=1}^{M_1} g_j - \sum_{j=1}^{M_2} h_j \right), \tag{5}$$

where δ is the spacing in $log\tau$ and M_1 and M_2 points are used to discretize the g and h distributions, respectively. The competing goals of high entropy, good fit, and appropriate image normalization are controled by λ and α, the Lagrange multipliers used to constrain χ^2 and normalization. A Newton-Raphson optimization of Q is performed in MemExp. The Lagrange multipliers are adjusted automatically.

NLS

As maxima are resolved in the lifetime distributions obtained by the MEM, MemExp performs a series of fits to the kinetics by n_e discrete exponentials using nonlinear least-squares (NLS) fitting:

$$\mathcal{F}_i = D_0 \sum_{j=1}^{n_e} A_j e^{-t_i/\tau_j} + \sum_{k=0}^{3} B_k (t_i/t_{max})^k, \tag{6}$$

where A_j and τ_j are the amplitude and lifetime of the j^{th} exponential, respectively, and the baseline is again approximated as a polynomial.

The value of χ^2 given by equation 4 is calculated for both the continuous and the discrete fits, facilitating a comparison of the distributed and discrete fits. The discrete fits are also characterized by

$$\chi_r^2 = \frac{1}{N - N_p} \sum_{i=1}^{N} \left(\frac{\mathcal{F}_i - D_i}{\sigma_i} \right)^2, \tag{7}$$

where N_p is the number of fit parameters (amplitudes, lifetimes, baseline coefficients) in the NLS fit. Note: The value of χ_r^2 (not χ^2) is used in determining which NLS fit is to be recommended by MemExp as the optimal discrete kinetic description.

Some Definitions

The graphical summaries output by MemExp (*.out.ps* and *.exp.ps* files) are labeled by several quantities, including the values of χ^2 given by equations 4 and 7. The **correlation length of the residuals** τ_c is also reported for both the MEM and NLS fits. It is estimated by summing the absolute value of the autocorrelation function of the residuals over the first three quarters of the autocorrelation function. The normalization of the fits are also given: I (equation 5) for the MEM fits and the sum of exponential amplitudes, ΣA, for the NLS fits.

For the MEM fits, the value of **Test** [5] is also given (denoted as T). Test is zero when the gradients of S and χ^2 are parallel; small values of Test imply that the maximum-entropy solution has indeed been found for the given value of χ^2. **Ratio** is another measure of convergence that is recorded in the *.out* files. Unlike Test, Ratio accounts for the effects of constraining the normalization I. Ratio is the norm of the gradient of Q divided by the norm of a unit vector [6]. If Test and Ratio both approach 1.0, the convergence should be slowed down by reducing GTOLL and GTOLU.

2 Running MemExp

MemExp produces detailed graphical summaries of both the MEM and NLS calculations, but it is not run interactively via a graphical user interface. Instead, input to MemExp is defined in one or more *.def* files that are specified on the command line executed to run the program. This command-line interface provides a convenient way to analyze many kinetics traces in the same way. In addition, the input *.def* files represent a record of the analysis for future reference. Similarly, because of the multiple arguments passed to MemExp on the command line, it is convenient to create a command file, e.g., *memexp.com* (or *memexp.bat* for Windows), in which each line represents a MemExp run. Thus, several data sets can be analyzed with a single command, e.g., *./memexp.com* (or *@memexp* for Windows). Under Windows, the batch file can also be run by right-clicking on it to "open" it.

MemExp can be invoked in one of three modes: *allin1*, *invert*, or *analysis*. The first mode does everything all in one MemExp call, beginning with a MEM inversion of data using uniform error estimates, iterating the error estimation a specified number of times (MXITER), and then inverting the data using the estimated time-dependent standard errors in the mean. **Because all MemExp functionality can be utilized in *allin1* mode, and because standard errors are seldom known precisely, the user will presumably run MemExp in this mode.** The *invert* mode simply inverts a data set once, and the *analysis* mode analyzes a previously obtained MEM distribution and its corresponding fit to the data.

Recommendation: MemExp is best used to interpret raw (unaveraged) kinetics data. Baseline drifts and any other systematic errors in the data can be more difficult to account for quantitatively if multiple data files are averaged together before analyzing using MemExp.

Recommendation: Because MemExp generates several files, it's convenient to place each raw data file in its own subdirectory prior to analyzing it with MemExp in *allin1* mode, as in *memexp.com*.

2.1 allin1 mode

Here is a sample call to MemExp in *allin1* mode:

memexp.exe allin1 2 invert.no.err.def data.001 0.0 analyze.def err1.dat invert.err.def err2.dat err3.dat

At least nine arguments follow the progam name on the command line when MemExp is run in *allin1* mode. The first argument specifies the mode and is set to *allin1*. The second is an integer ≥ 1, MXITER, which specifies the number of preliminary MEM-only iterations to be done to estimate the data's standard errors in the mean. MemExp performs a total of MXITER + 1 inversions, all but the first inversion using time-dependent standard errors estimated from the previous inversion. The third argument, *invert.no.err.def*, is the name of the input file containing parameters to be used in the first inversion of the data using uniform error bars. The fourth, *data.001*, is the name of the data file to be inverted and may include the path of the file (the subdirectory where it is located). The fifth argument is a real number, D0, the data's normalization constant (equations 1 and 6). In the sample command line above, D0 is set to zero, which directs MemExp to estimate the normalization directly from the kinetics data, and the program approximates this normalization constant as the difference between the average of the first two data values and the minimum data value. Consequently, setting *D0* to zero is only appropriate when all kinetics is observed within the time range probed by the experiment. The sixth argument, *analyze.def*, is the name of the input file containing parameters to be used in analyzing each of the MEM distributions automatically chosen by MemExp. This analysis involves writing the MEM fit and residuals to a file and revising the error estimates to be used for the next numerical inversion. The seventh argument, *err1.dat*, is the name of an output file to which the kinetics data and the first estimates of time-dependent errors are written in three-column (t_i, y_i, σ_i) format. The eighth argument, *invert.err.def*, is the name of the input file containing parameters to be used in all inversions of the data using time-dependent error bars. The ninth and any additional parameters each specify the name of a data file to be written with revised error estimates. The number of such parameters depends on MXITER. In the example above, MXITER = 2; so three inversions are done and the data files with errors derived from the second and third inversions are written to files named *err2.dat* and *err3.dat*, respectively.

In *allin1* mode, the data's normalization, D0, is iteratively refined along with the error estimates. The MEM distribution is always normalized before being written to a file. D0 is adjusted for use in the next MEM inversion according to the normalization of the distribution recommended by MemExp in the current inversion.

2.2 invert and analysis modes

To simply invert a data set once (without using MemExp to estimate errors), invoke MemExp with the command:

memexp.exe invert invert.def data.001 D0

where D0 is the non-baseline signal at $t = 0$ (equations 1 and 6). Again, if *D0* is set to zero, the program determines this normalization constant as the difference between the average of the first two data values and the minimum data value. Setting *D0* to zero is only appropriate when all kinetics is observed within the time range probed by the experiment.

To analyze a MEM distribution output during the inversion, invoke MemExp with the command:

memexp.exe analysis analyze.def distribution.dat D0 output.err.dat

The fit derived from the the MEM distribution in the file *distribution.dat* is used to estimate standard errors that are included in the file *output.err.dat.*

3 Inversions of a General Time-Dependent Signal

The MEM inversion is an iterative process that alternates between optimizations of the function Q for given values of the Lagrange multipliers and updates to the multiplier values for the next optimization. Iteration continues up to a maximum number of steps (MXFUNC), until χ^2 reaches a target value (CHI2AM), or until χ^2 stops changing more than a specified amount (CHI2S).

The following paragraphs describe, in order, the line-by-line information needed in an inversion *.def* file. The names of the variables whose values are to be read from the given line are written in bold capital letters. If multiple variables are to be input, do not separate them by commas. A semicolon can be used following the data to precede comments. The information in a few lines specifies a variable number of optional lines that may immediately follow.

The first line specifies **IOFORM**, the format of the data, which is either *RW* or *Fr* (rdwrt or free format). *Fr* format is almost certainly the format you will use, requiring a file without any header information with two or three numbers per line, specifying *t*, *y(t)*, and optionally $\sigma(t)$.

The second line specifies the integer **ITAKE**. If 1, fit to every point in the data files. If 2, every other point, etc.

XSCALE, YSCALE scale the x/y axis of the data (eg., 0.001 would convert from milliseconds to seconds).

MLOG is set to T, true, only for data in transmittance space. For most data, MLOG is F.

If **IGNORE** is set to T, any uncertainties $\sigma(t)$ (error bars) in the data file are ignored and replaced by a constant, as specified by the value of **INSIG** that is read from the next line. If INSIG is negative, it specifies the absolute error common to all data, i.e., $\sigma(t) = -INSIG$. If INSIG is positive, it represents the per-cent error: $\sigma(t) = 0.01 \times INSIG \times y(t)$. If INSIG is zero, the data are assumed free of noise. In this case, the data are actually assigned uniform errors (equal to 0.001 times the second data value), and CHI2AM and DCHI2S are reduced so that the calculation proceeds to very small values of CHI2.

Set **SYNDAT** to F, false. Data are synthesized when SYNDAT is T (not yet supported).

LAGMUL is either *Chi2* or *Chi2,Sumf* specifying that only χ^2 is to be constrained or that the image normalization is also to be constrained (equation 2).

CONST specifies the value of the initial uniform value of $f(log\tau)$. If set to zero, CONST is reset automatically.
Recommendation: Set CONST to 0.001.

CHI2AM is the desired (aimed at) value of χ^2. Iteration terminates if this value is reached.

If convergence is not reached, iteration will be terminated after **MXFUNC** evaluations of Q.

GTOLL and **GTOLU** are the logarithms of ϵ used to limit updates to the Lagrange multipliers λ and α (Cornwell and Evans, 1985). They are lower and upper values of the tolerance allowed on the gradient of Q, the function being optimized. Large values speed up the calculation but do not optimize Q as well. GTOLL is used at low values of χ^2, and GTOLU is used at larger values of χ^2. Thus, by setting GTOLL < GTOLU, the program proceeds quickly to low values of χ^2 before demanding a more precise optimization of Q. If $\chi^2 \geq$ **CHI2U**, then $\epsilon = 10^{GTOLU}$. Otherwise, $log(\epsilon)$ is ramped from GTOLU down to GTOLL as χ^2 drops to **CHI2L**. If $\chi^2 \leq$ CHI2L, then $\epsilon = 10^{GTOLL}$. Clearly, if

GTOLL and GTOLU are equal, the values of CHI2L and CHI2U are irrelevant.
Recommendation for data with reasonable error estimates: GTOLL = -5.0, GTOLU = -0.5, CHI2L= 1.5, CHI2U= 10.0.

LTPTS equally spaced log τ values are used in $g(log\tau)$, ranging from **LTMIN** to **LTMAX**.
Recommendation: Output image files can be kept small without loss of precision by choosing LTPTS, LTMIN, and LTMAX so that the spacing in $log\tau$ is an integer multiple of 0.01 or 0.001. The $log\tau$ values are then written in F6.2 or F7.3 format, respectively. Otherwise, they are written as unformatted output, resulting in larger files. Example: LTPTS = 141; LTMIN = -7.0; LTMAX = 0. The same holds for LTPTS2, LTMIN2, and LTMAX2 (described in the following paragraph).

NDIST, **NBASLN**, **VBASLN**, **TBASLN**: NDIST is an integer, either 1 or 2, specifying the number of distributions to be determined. If NDIST = 1, a single distribution $g(log\tau)$ is used, and the sign of D0 determines whether the data to be fit are rising or decaying (D0 > 0). If NDIST = 2, an additional line is required immediately following the current line. This additional line must specify **LTPTS2**, **LTMIN2**, and **LTMAX2** to be used for the $h(log\tau)$ distribution that describes rising signals. (NOTE: If necessary, MemExp resets LTPTS2 and LTMAX2 so that the spacing in $log\tau$ used to discretize the h distribution is the same as the spacing used for the g distribution.) NBASLN is an integer ranging from 0 (to omit baseline correction) to 4 (to include a cubic baseline). A constant, linear, and parabolic baseline are computed when NBASLN is 1, 2, and 3, respectively. Since the entropy involves a logarithm, only positive variables are possible, and so two variables must be used per polynomial degree of freedeom. The total number of 'image pixels' is therefore LTPTS + 2*NBASLN if NDIST is 1 and LTPTS + LTPTS2 + 2*NBASLN if NDIST is 2. If VBASLN ≤ 0, then the baseline parameters are fixed at the MEM-determined values during NLS fits. If TBASLN < 0, then the baseline is determined by the MEM in a simultaneous fit of distributed kinetics plus baseline drift. If TBASLN > 0, the baseline is prefit at all times greater than TBASLN using linear least squares.
Recommendations: VBASLN > 0 and TBASLN < 0.

CONSTB and **SBIGF** are relevant only if NBASLN > 1, but values must always be specified. Together, they determine the value of BIGF to be used for the baseline coefficients other than the constant term. Setting CONSTB to a very small number (e.g., 1.0E-20) is a way to bias the baseline toward flatness; values of CONSTB as large as 0.001 can result in unwarranted time dependence in the baseline. SBIGF is a multiplicative factor less than or equal to one that progressively suppresses the BIGF value for the jth term of the baseline: BIGF(j) = CONSTB*SBIGF**j (j = 1 for linear, 2 for quadratic, 3 for cubic).

Up to **MAXEXP** discrete exponentials are to be used in the discrete fits. If 0, then no discrete fits are done. The initial values of the discrete $log(\tau)$ are confined to the range **EXPMIN** and **EXPMAX**. **NLS** is the number of preliminary discrete-fit iterations to be performed during which first only the basline and the lifetimes are varied, then only the baseline and the amplitudes are varied. Generally, NLS may be set to 0.

Data to be included in the fit, those that contribute to χ^2, are within times **T1** and **T2** and signal changes **DA1** and **DA2**.

When greater than 1, **IPLOT** and **IPLOT2** reduce the size of the PostScript files created without affecting the calculation. When plotting the data, fit, and baseline (when used) in the second column of the output PostScript file(s), every IPLOTth point is plotted. When plotting the residuals and the autocorrelation function of the residuals, the first three points are plotted and then one out of every IPLOT2 points is plotted. The point plotted for each group of IPLOT2 points is the one with maximum absolute value, ensuring that the worst-fit points are represented. Setting both IPLOT and IPLOT2 to 1 plots all points, producing the most accurate (and largest) graphics output.
Recommendation: Try IPLOT = 10 and IPLOT2 = 1, and adjust the values based on the frequency at which your data are sampled and your need to conserve disk space.

Adjustments to λ and α are at least as often as every **ITMAX** optimization steps.

Values of λ, χ^2, $\mathcal{S}$, Q, etc. are written to the output file every **NPRINT** optimization steps.

The MEM image f is written to a file every **NWRITE** optimization steps, once χ^2 has reached **CHIWRT**. If IBIGF = 1 (below) and CHIWRT < 0, f is not written until after the default BIGF distribution has been created from a blurred image and χ^2 has reached -CHIWRT.
Recommendation: Use ITMAX = NPRINT = NWRITE.
Recommendation: For data with good error estimates, set CHIWRT = 1.2.

BIGF is the default distribution that defines maximum entropy $\mathcal{S}$. When IBIGF = 1 (below), BIGF is created by convoluting F with a gaussian with a full width at half maximum (FWHM) of **BFFWHM** pixels. If BFFWHM < 0 it is interpreted as the percentage of one decade in τ. This conveniently scales the blurring to the resolution of the image. Thus, setting BFFWHM to -20 will blur with a FWHM of one fifth of a decade.

ASPIKE, **SPIKE**, **SPIKE2**, and **PKFWHM** are the parameters that control the *differential blurring* of F into BIGF when IBIGF = 1 (below). IF a MEM peak has area greater than ASPIKE and is well resolved (as determined by SPIKE and SPIKE2), then it is incorporated into BIGF either unchanged (PKFWHM = 1.0) or as a sharpened gaussian having a FWHM scaled from the FWHM of the MEM peak by the multiplicative factor PKFWHM (0 < PKFWHM < 1). To be considered well resolved, the MEM peak must have a maximum-to-minimum ratio exceeding SPIKE on the 'most resolved' side of the peak and SPIKE2 on the other side (i.e., set SPIKE $\geq$ SPIKE2). To preclude differential blurring of any MEM peaks, set ASPIKE to a large number (e.g., 9999). See Steinbach et al. (2002) for a discussion of the pros and cons associated with creating BIGF by differentially blurring F.

When IBIGF = 1 (below), this creation of BIGF from F is performed every time the fractional drop in χ^2 is less than **DCHI2**, or if DCHI2 = 0, only once when CHI2 drops below **CHI2UP**.
Recommendation: For data with good error estimates, set DCHI2 = 0 and CHI2UP = 1.1.

DCHI2S is the fractional drop in χ^2 upon which the the creation of BIGF is performed by blurring F (if IBIGF = 1 below), and DCHI2S is the fractional drop that signals termination of the program.

NFUNC0 is set to the image number (positive integer) to be used whenever a calculation is restarted from a previous run. (This is virtually never done; set to 0.) If > 0, the next line gives the full image name.

IBIGF should be set to: 0 for a constant value of BIGF to be used throughout, 1 if BIGF is to be revised by blurring F, or 2 if the default distribution BIGF is to be read from a file. If IBIGF = 2, the file name is given on the final line.

4 Analysis of the Lifetime Distributions

A second *.def* file controls the analysis of a MEM image stored during a previous inversion. This analysis is one or more of the steps performed when MemExp is run in *allin1* mode. The fit and weighted residuals corresponding to the image are written to files in free format (two columns of numbers without a header). Also, the means and areas of peaks are automatically determined.

The first line, **WRFIT**, should be *T* for true. The fit and weighted residuals will then be written (format-free) to files.

EXTRAP data points are added to the beginning and to the end of the data when evaluating the fit. Use this option to view the kinetics predicted by $f(log\tau)$ that lies outside the experimentally sampled times.

The next several lines specify variables whose meanings are the same as for inversions. See above for descriptions of **IOFORM**, **ITAKE**, **XSCALE** and **YSCALE**, **MLOG**, **IGNORE**, **INSIG**, **t1**, **t2**, **dA1**, **dA2**, and **IBIGF**.

INTGRT is an integer specifying the number of log-tau intervals over which to integrate $f(log\tau)$. The next INTGRT lines give the upper and lower bounds for each interval.

NSMOTH and **ISMOTH** are integers. If NSMOTH > 0, then the deviations between the data and a previous MEM fit are smoothed. If ISMOTH $= 0$ ($\neq 0$), a bell curve (uniform distribution) ranging over NSMOTH points is convoluted with the error estimates. (See PFIT in the next paragraph).

The remaining parameters govern the estimation of the errors from the deviations between the data and a previous MEM fit. At each data point, sigma is estimated as the root mean square deviation of 2 **PFIT** + 1 values. IF NSMOTH ≤ 0, then these estimated sigmas are fit to a piecewise polynomial to obtain error estimates that vary smoothly with time. Otherwise, they are smoothed as described in the previous paragraph.

Next come the fit parameters (needed only if NSMOTH ≤ 0), each followed by a 0 or 1 to indicate the parameter is to be fixed or varied, respectively. The errors are generally fit to a piecewise cubic in log time. A0, A1, A2, and A3, are the constant, linear, quadratic, and cubic coefficients in log time for log t $<$ T. B2 and B3 are the quadratic and cubic coefficients and times log t $>$ T. Note that the constant and linear coefficients at long times are determined by the requirement that the function and derivative of the fit be continuous. If A2 and A3 are set to zero, then the fit is to a piecewise linear function, B2 represents the linear coefficient at long times, and only the function is continuous.

5 Files Output by MemExp

Recommendation: Because MemExp generates several files, it's convenient to place each raw data file in its own subdirectory prior to analyzing it with MemExp.

MemExp produces textual and graphical summaries of each numerical inversion. Let's look at the output generated for the test case that is downloaded with the MemExp software. It results from the following command:

memexp.exe allin1 1 memc_no_errors.def dir1/data.001 0. mem_no_errors.analysis dir1/data.001_e1 memc_errors.def dir1/data.001_e2

Here, the data in file *data.001* stored in subdirectory *dir1* are to be analyzed. All MemExp output will be written to this subdirectory. This *allin1* analysis calls for three calculations in turn: one preliminary MEM inversion, an estimation of the data's standard errors based on the appropriate MEM fit, and a final series of fits to the data by both continuous and discrete kinetic descriptions. **NOTE:** When MemExp is run on different platforms, small differences in round-off may propagate to produce slightly different output. Therefore, your results for the test case may not be exactly the same as the output shown here.

1. Preliminary MEM Fit with Uniform Errors

The preliminary MEM-only run is governed by the parameters read from the file *memc_no_errors.def*. The progress of the calculation is written to the file *data.001.out*. **NOTE:** If MemExp should abort due to incorrect input, the last few lines in the truncated *.out* file will indicate where the input needs to be corrected.

Information is plotted in the PostScript file *data.001.out.ps* every time (up to the first nine) a

MEM distribution is written to disk. The calculation proceeds from top to bottom in these graphical summaries. Each row of plots characterizes the MEM distribution at one point along the calculation. On the left, the data (black), fit (colored), and baseline (dotted, if nonzero) are plotted for the last half of the time interval spanned by the measurement. This column is useful in evaluating the computed baseline. In the middle column, the data, fit, and baseline are plotted over the entire temporal range. More importantly, the residuals, $R_i = (\mathcal{F}_i - D_i)/\sigma_i$ (colored), and the autocorrelation of the residuals (black) are plotted. On the right, the continuous g (colored solid) and h (colored dashed) rate distributions are plotted. If F is derived from f during the MEM calculation by uniform or differential blurring, F is plotted (black solid) (See *data.001_e1.out.ps* below).

In each graphical summary, the fit recommended by MemExp is marked by an asterisk (*) on the plot's right hand side. Because no automated selection criteria will choose optimally in all cases, the user of MemExp should compare these recommended fits to those that immediately precede and follow them in the graphical summary. The MemExp selection criteria performed very well for realistic simulated data [1].

2. Error Estimation

Following this preliminary MEM inversion, the recommended fit is analyzed to determine error estimates according to parmeters read from the file *mem_no_errors.analysis*. These errors are assigned to the kinetics by writing a new file named in the MemExp command *dir1/data.001_e1*. The errors are also plotted in the file *data.001_e1.sigma.ps* Here, the root mean-square deviations (rmsd) of the data and the recommended MEM fit are plotted (squares), along with the smoothed error estimates derived from them (dashed). Also plotted at the top of the page (solid) are the deviations of the smoothed estimates from the unsmoothed rmsd values.

3. MEM / NLS Fits with Time-dependent Errors

Next, *data.001_e1* (original kinetics with MEM-derived standard errors) is numerically inverted. The parameters in the file *memc_errors.def* call for a hybrid MEM / NLS fitting of the data (MAXEXP > 0). The MEM calculation is summarized in *data.001_e1.out* and in *data.001_e1.out.ps*. Note that with the time-dependent error estimates, the residuals are more uniform in magnitude and the values of χ^2 are near 1.0. Here, F was derived from the f plotted in the second row by uniform blurring, and the revised F is plotted in black in the third row. From the third row down, this new F defines maximum entropy according to equation 3.

In addition, the NLS fits by discrete exponentials are summarized in *data.001_e1.exp.ps*. Up to nine NLS fits can be plotted in this PostScript summary. On the right, the discrete exponentials (colored vertical lines, dashed for negative amplitudes) are plotted. Also plotted in the right column of *data.001_e1.exp.ps* is the continuous distribution from which the initial NLS parameter values were derived (black solid).

Once again, the errors estimated from this second MEM inversion are appended to a data file named in the memexp command line, *dir1/data.001_e2*. These errors are plotted in *data.001_e2.sigma.ps*. Convergence of the error estimates can be checked by comparing *dir1/data.001_e1* and *dir1/data.001_e2*.

The .out Files

The *.out* files produced by MemExp begin with the echoing of input parameters. During inversions, these lines begin with 'RDPARS', the name of the subroutine that reads the parameters. (During analyses of images, these lines begin with 'ANALYZ'.) Should a MemExp run terminate prematurely due to incorrect input data, refer to the last few lines of successfully input data and a sample *.def* file to correct the mistake in the input data. If all inversion parameters have been read successfully, the *.out* file contains the line, 'Exiting routine RDPARS.'

The progress of the MEM calculation is recorded every NPRINT steps. Upon writing the MEM distributions to files, peaks in the distribution are characterized. The lines in the *.out* files beginning with 'LT_i:' and 'A_i:' report the mean log lifetime and the area estimated for each peak. The isolation/resolution of each peak is also characterized by two ratios: the intensity of the peak maximum divided by the intensity at the minimum on either side of the maximum. The smaller of these two ratios is reported as 'r_i'. The total area of the distribution is given by 'A_t'. When MAXEXP > 0, NLS fits are performed with parameters initialized based on peaks in the MEM distribution. The NLS fits are summarized by reporting the initial and final parameter values: 'Initial LT_i', 'Initial A_i', 'Final LT_i', etc.

MemExp automatically recommends one discrete and one distributed fit. See the lines that begin 'CHOOSX' and 'CHOOSE', respectively.

Image Files and Others

Distributions obtained by the MEM are written to files with names derived from the data file being inverted. For example, one MEM image stored during the numerical inversion of file *data.001* might be named *data.001.FLT.a.110.* This would be the image obtained after 110 evaluations of the function Q to be optimized by the MEM (see above). The *FLT* indicates that the file was generated by MemExp, i.e, that it contains $f(log\tau)$ values. The a specifies that the optional Lagrange multiplier α was used; an l would indicate that only the Lagrange multiplier λ was used. When data are analyzed with both the g and h distributions (NDIST = 2), all the g parameters are written to the file, then all the h parameters, and finally the baseline parameters. For example, if a linear baseline is used, the final four lines in the output file are the positive constant coefficient, the negative constant coefficient, the positive linear coefficient, and the negative linear coefficient, respectively. To facilitate subsequent plotting of the results when NDIST = 2, the g and h distributions are written separately to *data.001.FLT.a.110_pos* and *data.001.FLT.a.110_neg*, respectively.

The $log\tau$ values and amplitudes obtained in fits by discrete exponentials are written to similar files, named *data.001.FLT.a.N*, where N is the number of exponentials in the fit. The associated baseline parameters are written to files named *data.001.FLT.a.N_bln.*

The rate of MEM convergence is also recorded in the files *data.001.cor* and *data.001_e1.cor*, to which the step number, the estimated correlation length of the residuals τ_c, and χ^2 are written every NPRINT evaluations of Q. These *.cor* files are of less interest.

Recommendation

The residuals and the autocorrelation function of the residuals plotted in the MEM and NLS PostScript summaries should be inspected visually. These plots are very helpful in evaluating the 'optimal' distributed and discrete fits recommended automatically by MemExp. Inspection of the PostScript file helps strike the compromise desired between two goals: a minimally structured kinetic description and residuals that are acceptably uniform in magnitude throughout the entire temporal range of the data with a value of χ^2 of about 1.0 or somewhat less. Simply choosing the final f distribution stored during the inversion will generally result in over-fitting the data; the residuals may look good but at the expense of unwarranted structure in f and an unneccessarily large number of exponentials used in the discrete fit.

6 Error Messages & Troubleshooting

Input/Output Problems MemExp will be aborted if the parameters input from a *.def* file are not as expected. If the number of parameters on a given line or a variable type is incorrect, an I/O error message will be printed to the screen. Check the output file to see which line of the *.def* file was read last, just before the program crashed. This line in the output file will begin with 'RDPARS' if the problem occured during an inversion and with 'ANALYZ' if the problem occured while analyzing a lifetime distribution. Correct this line in the *.def* file by referring to the documentation above and to the *.def* files provided as templates.

Optimization of Q is unnecessarily slow or it's inadequate. See the discussion of GTOLL, GTOLU, CHI2L, and CHI2U above. Adjust these values to speed up or slow down the adjustment of the Lagrange multipliers.

Not enough fits to discrete exponentials. This may well indicate that your data are better described by distributed lifetimes than by discrete exponentials. Try decreasing DCHI2S to force the MEM calculation to proceed to lower values of CHI2.

Residuals not uniform in magnitude. Apparently, the iterative estimation of the standard errors has not yet converged. Increase the number of preliminary MEM inversions (the integer following 'allin1' on the command line).

The first residuals plotted are very large, and convergence is erratic. Presumably, the initial MEM inversion being done with all standard errors set to a constant should be using a larger constant. Increase the absolute value of INSIG.

NEWLAG> BB-4AC ... These messages from subroutine NEWLAG may or may not signify a problem. When updating the Lagrange multipliers during a MEM inversion, a quadratic equation is solved to limit the updates. If the solutions to this quadratic equation are not real, this error message is written to the output file and the Lagrange multipliers are not updated. One of these messages once in a while is probably fine. If a multitude of these messages is printed and convergence appears to be stalled, try reducing GTOLL and/or GTOLU. This slows the multiplier updating and should solve this problem.

References

[1] Steinbach, P.J., R. Ionescu, & C.R. Matthews. (2002) Analysis of kinetics using a hybrid maximum-entropy / nonlinear-least-squares method: Application to Protein Folding. *Biophys. J.* **82**, 2244-2255.

[2] Steinbach, P.J. (1996) Two-dimensional distributions of activation enthalpy and entropy from kinetics by the maximum entropy method. *Biophys. J.* **70**, 1521-1528.

[3] Steinbach, P.J., K. Chu, H. Frauenfelder, J.B. Johnson, D.C. Lamb, G.U. Nienhaus, T.B. Sauke, and R.D. Young. (1992) Determination of rate distributions from kinetic experiments. *Biophys. J.* **61**, 235-245.

[4] http://cmm.info.nih.gov/maxent

[5] Skilling, J. & R.K. Bryan. (1984) Maximum entropy image reconstruction: general algorithm. *Month. Notices Roy. Astronom. Soc.* **211**, 111–124.

[6] Cornwell, T.J. & K.F. Evans. (1985) A simple maximum entropy deconvolution algorithm. *Astron. Astrophys.* **143**, 77–83.