

Dynamix

Ben Tatman

July 27th 2020

Dynamix is a program developed to fit dynamics models (see *Models*) to solid state NMR Relaxation data. This guide will go into what features are available, how to use them, how to format data, and also go into any caveats or issues with the model.

Features

As of writing, Dynamix can fit the following models;

- Simple Model Free (SMF) Fits two parameters, S^2 and τ , with spectral density given as;

$$J(\omega) = \frac{(1 - S^2)\tau}{1 + (\omega\tau)^2}$$

- Simple Model Free with Temperature Dependence (SMFT) Fits three parameters, S^2 , τ_0 and Ea , where the correlation time is temperature dependent;

$$\tau(T) = \tau_0 \exp\left(\frac{Ea}{RT}\right)$$

This is then fit as in SMF.

- Extended Model Free (EMF) Fits three parameters, S_{slow}^2 , τ_{slow} , τ_{fast} . The fast order parameter, S_{fast}^2 , is calculated as $S_{\text{dipolar}}^2/S_{\text{slow}}^2$. The spectral density is given as;

$$J(\omega) = \frac{(1 - S_f^2)\tau_f}{1 + (\omega\tau_f)^2} + \frac{S_f^2(1 - S_s^2)\tau_s}{1 + (\omega\tau_s)^2}$$

- Extended Model Free with Temperature Dependence (EMFT) Five parameter model, fitting S_{slow}^2 , $\tau_{0,\text{slow}}$, $\tau_{0,\text{fast}}$, Ea_{slow} , Ea_{fast} . S_{fast}^2 is calculated as in EMF, and the correlation times are temperature dependent as in SMFT. The spectral density is as in EMF.
- Extended Model Free without Dipolar Approximation (DEMF)¹ Four parameter model, S_{slow}^2 , S_{fast}^2 , τ_{slow} , τ_{fast} . Spectral density as in EMF.
- Extended Model Free without Dipolar Approximation, with Temperature Dependence (DEMFT) Six parameter model, fitting S_{slow}^2 , S_{fast}^2 , $\tau_{0,\text{slow}}$, $\tau_{0,\text{fast}}$, Ea_{slow} , Ea_{fast} . Spectral density as in EMF.
- Gaussian Axial Fluctuations (GAF)² Eight parameter model: τ_{slow} , τ_{fast} , $\sigma_{\text{slow}}^\alpha$, $\sigma_{\text{slow}}^\beta$, $\sigma_{\text{slow}}^\gamma$, $\sigma_{\text{fast}}^\alpha$, $\sigma_{\text{fast}}^\beta$, $\sigma_{\text{fast}}^\gamma$. These axial fluctuations are then used to derive order parameters as in Bremi 1997, and then fit to the EMF spectral density function. The relaxation rates take into account dipolar contributions between multiple atom pairs, and anisotropic chemical shift.

¹Generally the DEMF models fit very poorly as the additional overall motional constraint from the dipolar order parameter is used.

²The GAF models have not been fully tested as I do not currently have sufficient data to fit them fully.

- Gaussian Axial Fluctuations with Temperature Dependence (GAFT) Ten parameter model. All of those in GAF, plus fast and slow activation energies. Temperature dependent time constants calculated as in SMFT, EMFT and DEMFT, angles used to calculate S^2 , then fit to EMF spectral density.

These models can be fit to ^{15}N and ^{13}C R_1 and $R_{1\rho}$ values. The models have by and large been tested and verified against MATLAB models for ^{15}N , but not for ^{13}C .

For each of these models, Dynamix will perform a user specified number of optimizations with random starting points using the Nelder-Mead simplex method to find an optimum. Each optimum is output into a `residue_N.dat` file. Once complete, it will perform back calculations for each relaxation data point, outputting these into `backcalc_N.dat` files. If one of the GAF modes is used, it will calculate effective S_{NH}^2 order parameters and output these into `gaf.dat`.

If error mode is enabled, it will perform a further set of optimizations where the starting point is set to the optimized parameters. The relaxation rates are then varied within their experimental error, and optimization performed. The new optimum points for each repeat in the error calculations are then used to determine standard deviations for the optimized values³.

Compilation

To compile the program from source on Karplus, navigate to the directory above `src/` and run

```
gcc src/main.c -lm -pthread -o dynamix -O3
```

This will pull in all other required C files in the `src/` directory, load the `math(s)` and `pthread` libraries, using optimization level 3, and output the program into `dynamix`. Also contained in the `src/` directory is documentation (build using `doxygen` `Doxyfile` if not up to date) and test programs to verify the model against the MATLAB scripts.

Data Formats

There are four data formats used to provide data to Dynamix. In all of these, beginning a line with ‘%’ marks it as a comment.

System File

The system file contains information about the model being fit, and how to fit it. It also tells Dynamix where to find all other data. Generally I use the file extension `.dx` for these, though it is not necessary.

The first portion of the system file consists of key value pairs laid out as

KEY = VALUE

The keys are all upper case, and there must be spaces on either side of the equals.

Key	Value
MODEL	Model being fit - see Features
S2DIP	File containing dipolar order parameters as defined below
CSISON	File containing isotropic chemical shifts for ^{15}N
CSISOC	File containing isotropic chemical shifts for ^{13}C
N_RESIDUES	Number of residues - must be the same as the number of lines in each input file (or bad things may happen)
OUTPUT	Directory (eg <code>output/</code>) to place output files into
N_ITER	Number of iterations for optimization
N_ERROR_ITER	Number of iterations to perform for error calculation

³Note that the output will have the minimum optimized points and two standard deviations for the errors; it will not output the mean of the error calculations (unless you explicitly change the code to do so - at the moment this is line 408 of `main.c`, in which you should change `m.residues[l].parameters[i]` to `m.residues[l].errors_mean[i]`.

Key	Value
IGNORE	Residue to ignore; each residue to ignore should have its own line
OR_NH	N-H orientations
OR_NC	N-C orientations
OR_NCA	N-C α orientations
OR_NCSAxx/yy/zz	Nitrogen chemical shift anisotropy orientations
OR_CCAp	$^{13}\text{C}'$ - $^{13}\text{C}_{i-1}^{\alpha}$ orientation
OR_CCAc	$^{13}\text{C}'$ - $^{13}\text{C}_i^{\alpha}$ orientation
OR_CN	C-N orientation
OR_CNH	C-amide proton orientation
OR_CCSAxx/yy/zz	Carbon chemical shift anisotropy orientations
NTHREADS	Number of threads to run; generally, set to the number of processors you want to run it on

For example

```

MODEL = EMF
S2DIP = system/s2_dipolar.csv
CSISON = system/csisoN.csv
CSISOC = system/csisoC.csv
N_RESIDUES = 56
OUTPUT = output/
N_ITER = 200
IGNORE = 42
N_ERROR_ITER = 200
% Orientations taken from Lienin 1998
OR_NH = system/or_NH.csv
OR_NC = system/or_NC.csv
OR_NCA = system/or_NCA.csv
OR_NCSAxx = system/or_NCSAxx.csv
OR_NCSAyy = system/or_NCSAyy.csv
OR_NCSAzz = system/or_NCSAzz.csv
OR_CCAp = system/or_CCAp.csv
OR_CCAc = system/or_CCAc.csv
OR_CN = system/or_CN.csv
OR_CNH = system/or_CNH.csv
OR_CCSAxx = system/or_CCSAxx.csv
OR_CCSAyy = system/or_CCSAyy.csv
OR_CCSAzz = system/or_CCSAzz.csv
NTHREADS = 4

```

This key value header should be followed by **#RELAXATION**, and then each file containing Relaxation Data should be listed below.

```

#RELAXATION
system/15N_R1_600_50_0_271.csv
system/15N_R1_600_50_0_282.csv
...

```

Relaxation Data

Each piece of relaxation data (eg a ^{15}N R_1 measurement made at 300 K in 600 MHz at 50 kHz) should be placed into an individual file. This file should begin with a header.

Note that there must be a space on either side of the equals sign

```

FIELD = {field in MHz}
WR     = {spinning frequency in Hz}

```

```

W1      = {spin lock frequency in Hz}
TEMP    = {temperature in kelvin}
TYPE    = {15NR1, 15NR1p, 13CR1, 13CR1p}

```

This defines the 'global' parameters for this relaxation data. Then, a line `#DATA` denotes the start of the actual relaxation data, and should be followed by the relaxation data.

```

#DATA
1 {relaxation rate in s-1} {error in s-1 (2 standard deviations)}
2 {relaxation rate in s-1} {error in s-1 (2 standard deviations)}
...

```

The program requires a line for each residue; if you have no relaxation data for one residue, denote the relaxation rate and error with `-1` instead of leaving it blank. This will tell Dynamix that the data does not exist, as opposed to it just being omitted by mistake. The program may run without this, however it will not necessarily inform you that there is insufficient data for fitting which may lead to incorrect fitting and data response.

Orientation Data Formats

For the GAF models the orientation of each interaction vector related to the α , β and γ motional axis is required. In the form of Lienin 1998, this is taken as θ and ϕ angles. For each orientation vector there should be a file containing this data as;

```

1 {theta} {phi}
2 {theta} {phi}
...

```

This data is only needed for GAF fits, but all of it is needed. The orientations required are described below.

Residue Data Formats

Data such as isotropic chemical shifts and dipolar order parameters should be kept in files laid out as;

```

1 {value} {error}
2 {value} {error}
...

```

Running The Model

Once the file is setup, the model may be run as;

```
./dynamix {path to .dx file}
```

This will output the various threads being spawned as the program operates. Passing the `-e` option;

```
./dynamix {path to .dx file} -e
```

Will enable error calculation.

Visualising Results

Back Calculations

Using `gnuplot` the output of Dynamix can be quickly visualised. To view residue specific back calculated data, simply run (after entering the output directory)

```

> gnuplot
...
gnuplot> plot 'backcalc_N.dat' u 2:3 w points pt 7, x lw 3

```

This will produce a graph showing the calculated values (x) against experimental values (y), with a line of $y = x$. In order to view all back calculated points for all residues, run;

```
> cat backcalc_* > backcalc.dat
> gnuplot
...
gnuplot> plot 'backcalc.dat' u 2:3 w points pt 7, x lw 3
```

If the 'VERBOSE' key is enabled in `datatypes.c`, the back calculated files will also contain experimental information. Though beyond the scope of this brief introduction, these can be plot onto the graphs too to show where variation in relaxation rate arises (for example, I find that 700 and 850 MHz data tends to much better fit the $y = x$ line than 600 MHz data. I don't know if this is some systematic experimental difference, or just the model validity breaking down).

Visualising Parameters

In the `utils/` directory there are `gnuplot` scripts to plot data. These are set up assuming that the output directories are named as the model type (eg, an EMF model is in `emf/`). At the moment there are scripts to plot each model, with and without errors, except for the GAF models for which no scripts exist to plot errors as of yet. For non error plots, the colour of each point is related to the chisq value - darker points represent better fits.

Script Name	Description
<code>pp_smf.m</code>	Plots SMF
<code>pp_smf_e.m</code>	Plots SMF with errorbars
<code>pp_emf.m</code>	Plots EMF
<code>pp_emf_e.m</code>	Plots EMF with errorbars
<code>pp_smft.m</code>	Plots SMFT
<code>pp_smft_e.m</code>	Plots SMFT with errorbars
<code>pp_emft.m</code>	Plots EMFT
<code>pp_emft_e.m</code>	Plots EMFT with errorbars
<code>pp_demf.m</code>	Plots DEMF
<code>pp_demf_e.m</code>	Plots DEMF with errorbars
<code>pp_demft.m</code>	Plots DEMFT
<code>pp_demft_e.m</code>	Plots DEMFT with errorbars
<code>pp_gaf.m</code>	Plots GAF
<code>pp_gaft.m</code>	Plots GAFT

These may be run as;

```
> ls
emf/  utils/
> gnuplot utils/pp_emf.m
> ls
emf/  utils/  emf.eps
```

The output is as a `.eps` file. This may be viewed using eg Okular (`okular file.eps`) or converted into a PDF using `epstopdf file.eps`.

Output File Formats

Note: All taus and tauf values are output in nanoseconds (10^{-9} seconds), not seconds. This is done to improve precision.

residues_N.dat

These residue files are generated during the optimization process. Each line refers to a separate optimization process. The first column is the residue number, the second is the minimum value of the chisq function. Plotting the second column generates a nice plot showing how well the optimization went.

The remaining columns depend on which model is in use. In particular;

- SMF: tau (ns), S2

- SMFT: tau (ns), S2, Ea
- EMF: taus (ns), S2s, tauf (ns)
- EMFT: taus (ns), S2s, tauf (ns), Eas, Eaf
- DEMF: taus (ns), S2s, tauf (ns), S2f
- DEMFT: taus (ns), S2s, tauf (ns), S2f, Eas, Eaf
- GAF: taus (ns), tauf (ns), sAs, sBs, sGs, sAf, sBf, sGf
- GAFT: taus (ns), tauf (ns), sAs, sBs, sGs, sAf, sBf, sGf, Eas, Eaf

This may be useful for plotting to see how varied the individual responses are, eg how responsive the model is to one parameter.

final.dat

This contains the final fit parameters. The first column is the residue number, the second is the S2 dipolar value, then the minimum chisq value, followed by the model specific columns as above. The final column is a count of how many repeats went into error calculation (or 0 if this was not done).

errors.dat

This file is only output if error mode is enabled. It is the same as **final.dat**, only after each parameter there is the error (2 standard deviations). eg;

```
{residue number}, {S2dipolar}, {chisq minimum}, {tau}, {tau error}, {S2}, {S2 error}
```

gaf.dat

This file is only produced for GAF models. This is laid out as;

```
{residue number}, {tau s}, {S2 slow}, {tau f}, {S2 fast}
```

Where S_{slow}^2 and S_{fast}^2 are the order parameters for the N-H dipolar interaction.

backcalc_N.dat

These files contain back calculations. They are set out;

```
{relaxation number} {calculated R} {measured R} {error in measured R}
```

Plotting {calculated R} against {measured R} can give a useful indication of goodness of fit. To gain more insight into how different experimental parameters affect the fit, enable the **VERBOSE** flag in **datatypes.c** (eg set it to 1). This will then output a further few columns;

```
{field in MHz} {spinning frequency Hz} {spin lock frequency Hz} {temperature K}
```

Fitting Speed in Dynamix

The amount of time this model will take to run depends on the amount of data being passed to it, the complexity of the modelling, and how “nice” the relaxation data is (some will converge far faster than others). As rough guides, for GB1 ^{15}N data the following times were recorded for the model free models (in seconds)

Model	2000 iterations, 56 threads	50 iterations, 4 threads
SMF	1.38	0.58
SMF + e	2.97	1.17
EMF	3.69	1.83
EMF + e	6.72	3.23
SMFT	4.64	1.89
SMFT + e	11.53	4.33
EMFT	16.51	6.81
EMFT + e	35.04	12.64
DEMF	5.21	2.51
DEMF + e	11.64	4.93
DEMFT	18.37	7.46

Model	2000 iterations, 56 threads	50 iterations, 4 threads
DEMFT + e	38.48	15.11

In lieu of the more complicated calculations required, GAF models take significantly longer than these model free models; still, it should be possible to perform upwards of 500 iterations in a few hours.

In order to verify that the new model was faster than the old model, 15246 S^2 parameters were calculated using GAF (this was done by repeating the calculation for each of the 6 order parameters calculated for ^{15}N for σ_α in the range 0 to 0.1 radians, σ_β from 0 to 0.1 radians, σ_γ from 0 to 0.2 radians using steps of 0.01 rad). Using the MATLAB scripts running single threaded this took 8.20 seconds. With Dynamix single threaded this took 0.33 seconds, representing a possible $25\times$ increase in calculation speed.

To compare the performance of EMF calculations, 10,000,000 R1 and R2 values for random parameters were calculated. In MATLAB (single threaded, Karplus), this took 29.87 seconds. In Dynamix (single threaded, my laptop), this took 2.65 seconds. In MATLAB, R1 and R2 calculation is combined in one and so for each data point both R1 and R2 are calculated regardless of if only R1 is measured. As Dynamix separates these, for calculation of 10,000,000 R1 values MATLAB still takes 29.87 seconds; Dynamix takes 0.99 seconds.

This data suggests that there is a potential $20\times$ speed improvement just on the basis of calculation efficiency from moving from MATLAB to Dynamix, prior to any additional increases in efficiency. For example, an EMFT optimization in MATLAB performed on Karplus prior to the creation of Dynamix took approximately ≈ 6000 seconds to complete 10,000 iterations. The equivalent in Dynamix should only take 82.55 seconds on the same hardware, a 72-fold increase in speed. Similarly, a run of SMF using MATLAB on Karplus took approximately 1800 seconds to complete 10,000 iterations; in Dynamix, this would be expected to take 6.9 seconds, representing a 260-fold increase in speed.