

DYNAMICS

Version 2.0

A Very Brief User's Manual

By David Fushman

Contents

1. Introduction
2. List of included Matlab programs
3. Brief description of some features
4. Installation
5. Running DYNAMICS
6. Demos
7. Copyright, registration and feedback
8. References

1. Introduction

DYNAMICS is a program package for extracting information on protein dynamics from the NMR relaxation data (^{15}N R_1 , R_2 , & $\{^1\text{H}\}^{15}\text{N}$ NOE) [1]. It performs analysis of the ^{15}N relaxation data in terms of the model-free approach [2,3]. Current version of the program allows analysis of the experimental data collected at one or several (practically unlimited number) magnetic field strengths/ spectrometer frequencies and features the monomer-dimer equilibrium model [1], anisotropic overall motion [4], as well as the more conventional isotropic-overall-motion model. The next upgrade (currently under development) will incorporate site-specific ^{15}N CSA-data.

In this Manual, the names of Matlab variables/parameters, as well as the Matlab commands are shown in *italic*; program (*.m) file names are typed in **bold**.

2. List of included Matlab programs

The program package DYNAMICS includes the following *basic* programs:

dynamics.m, **dynini.m**, **dynprep.m**, **dyn_omdc.m**, **dyn_df.m**, **dynr21.m**, **dyn21ind.m**, **dynfzero.m**, **r2r1sol.m**, **dyn_taus.m**, **dyn_ld.m**, **dyn_aas.m**, **dynmsel.m**, **dynfmina.m**, **dynss.m**, **relax.m**, **dyn_res.m**, **dynreprt.m**, **dyndisp.m**, **dynCOVt.m**, **dynMCpar.m**, **dynMCDat.m**, **sortmat.m**, **dynclean.m**, **dyncc.m**.

The following *accompanying* programs (program packages) are currently available:

reldata.m simulates relaxation data for a selected model

- pdb2nh.m** read in a pdb-data set and extracts NH vectors. If no NH vectors are found, builds them using N,CO,CA coordinates. No rotation with regard to inertia tensor frame is performed. Uses a set of programs: **readpdb.m, readasci.m, getNHvect.m, select_at.m, buildNH.m, rotate.m**
- lab2iner.m** reads in pdb-data and prepare a list of the NH vectors (for anisotropic analysis), either in the original pdb-coordinate frame, or in the principal coordinate frame of the inertia tensor. This is a set of programs which also includes: **inertens.m, ordmtrx.m, row2mtrx.m, readpdb.m, readasci.m, atnamlst.m, select.m, get_atwt.m.**

3. Brief description of some features

The program package DYNAMICS is designed to perform the model-free analysis of the ^{15}N relaxation data collected at one or several frequencies. The details of the analysis are described in [1]. Briefly, for the isotropic model of overall motion, a search for optimal overall rotational correlation time, τ_c (the same for all residues), is performed simultaneously with the model selection for the spectral density function (SDF) for each residue. Along with the isotropic model of the overall motion, the program permits data analysis in the case of monomer-dimer equilibrium [1], and anisotropic overall rotation [4]. In the latter case, the overall optimization procedure includes, in addition to τ_c , determination of the ratio, D_z/D_x , of the principal components of the rotational diffusion, as well as orientation of its principal axes.

4. Installation

All the files enclosed in the **dynamics.tar** file fall into two categories: program files, ***.m**, and help files (demo-examples, **demo*.txt, README.txt**, and this file, **dynamics.doc**). The programs and examples are in ascii format. It is recommended that you keep them in one directory, e.g. DYNAMICS, and you either run them from this directory or include it in your MATLAB-path.

Platforms. The programs were tested on both UNIX and DOS/WINDOWS platforms. The modifications required to run the programs under Windows 3.1 are basically related to the file-name limitations: the program modules **dynCOVt.m, dynMCpar.m, dynMCdat.m**, and **NHvect.m** have to be renamed to substitute all upper case letters with the lower case ones; the same changes have to be done in the **dynamics.m** script (error estimation part), and in **lab2iner.m** (**NHvect.m**). See also Section Autosaving Feature.

MATLAB version compatibility. The programs were written for MATLAB v.4.2c and also tested with Matlab 5.1. You should be able to run them on older versions as well as versions 5.0 and higher. Running Dynamics under Matlab 5.x might cause some warning messages to appear. These warnings could be suppressed either by issuing the command *warning off* prior to running dynamics, or by including the corresponding line somewhere in the beginning of **dynamics.m** program.

The maximum number of steps in the optimization procedure (based on the simplex algorithm) is set to 6000. If this number-of-steps limit is reached before the termination tolerance is satisfied, the following warning message which usually can be ignored will be displayed on the screen (I couldn't find a way to suppress this warning in Matlab 4.x):
Maximum number of iterations (6000) has been exceeded
(increase OPTIONS(14)).

MATLAB C-compiler. Using Matlab C-compiler will significantly speed up the program. In fact, many of the program modules were specifically written to take advantage of the C-compiler. If you have a Matlab C-compiler, it is recommended that you compile the program modules either by running the MATLAB-script **dyncc.m** or by issuing the following commands, which are included in **dyncc.m**, line by line:

```
mcc -ri relax.m
mcc -ri dynss.m relax.m
mcc -ri dynfmina.m dynss.m relax.m
mcc -ri dynCOVt.m
mcc -ri dynMCpar.m
mcc -ri dynMCdat.m dynfmina.m dynss.m relax.m
mcc -ri dyn2lind.m dynfzero.m r2r1sol.m
mcc -ri readpdb.m
mcc -ri get_atwt.m
mcc -ri select.m
```

Warning: do not try to compile all the programs included in the package: some of them which are Matlab-scripts rather than Matlab-functions you would not be able to compile; compiling other programs might lead to run-time errors.

If a warning appears during compilation, please refer to your Matlab C-compiler manual.

5. Running DYNAMICS

To start DYNAMICS follow the three steps described below:

- 1) Start Matlab.
- 2) Load all necessary relaxation data. Don't forget to set *freq* and *TAUc*. The variable-naming convention and data sets /variables required to run DYNAMICS are described in INPUT below. For the examples see **demo*.m** files.
- 3) If you are in the directory containing DYNAMICS programs or your Matlab-path includes this directory, type
dynamics
following the Matlab prompt. Follow the instructions/ answer the questions you get on your screen (see RUN-TIME DIALOG below).

INPUT

For the program to work, the following variables must be present in the Matlab memory (their names must be spelled exactly as they appear here):

Absolutely necessary:

freq -- list of ¹H NMR frequencies in MHz, e.g. *freq*=[500.13, 600.13].

TAUc -- starting list of the overall rotational correlation times, in [ns]

r11 -- array of R_1 -values for the first frequency listed in *freq*, in this example, 500.13.
r21 -- array of R_1 -values for the first frequency listed in *freq*, in this example, 500.13.
r31 -- array of NOE-values for the first frequency listed in *freq*, here, 500.13.
r12 -- array of R_1 -values for the second frequency listed in *freq*, here 600.13.
r22 -- array of R_1 -values for the second frequency listed in *freq*, here 600.13.
r32 -- array of NOE-values for the second frequency listed in *freq*, here 600.13.

Naming of the relaxation data arrays, *rLM*, is as follows: L=1, 2, or 3 refers to R_1 , R_2 , or NOE, respectively, and M refers to the corresponding frequency position in the frequency list, *freq*. Some of the relaxation data may be missing (e.g., if *r11* is missing, it will be automatically assigned an empty array; if R_1 value for some residue is missing, it either has to be assigned the NaN number in the second column of *r11* or has to be dropped from the *r11*-list). The minimum number of relaxation data per residue is three. If for some residue it is less than 3, the following message will appear on screen: ‘*residue ... not enough data*’, and the residue will be skipped. The arrays of relaxation data have exactly the same structure as the output from RELAXFIT [1], namely (e.g. for *r11*):

```
r11= [residue1  $R_1$ (residue1) std.error $R_1$ (residue1);
      residue2  $R_1$ (residue2) std.error $R_1$ (residue2);
      .....
      residueN  $R_1$ (residueN) std.error $R_1$ (residueN)].
```

Here ‘residue1’ means the residue number corresponding to the first residue in the R_1 data set.

Note that both the R_1 and R_2 values and their standard errors have to be in [1/msec] (it will be switched to [1/s] in the next version).

The *TAUc* list must contain more than one value, except for the anisotropic case, when one value is sufficient.

Necessary under circumstances:

EXCLUDE a list of residues (residue numbers), which you want to exclude from the analysis, e.g. *EXCLUDE*=[24,36,100:104]. This allows user to perform analysis on a selected set of residues, e.g. protein core.

kovrl = 1 (anisotropic model) or 2 (monomer-dimer equilibrium). If no *kovrl* value is specified, *kovrl* = 0 (isotropic model) will be assumed, and *kovrl* value will be automatically set to 0.

vNH a set of NH-vector coordinates, required only if *kovrl* is set to 1 (anisotropic overall motion), in the following form:

```
[residue1 x y z;
 residue2 x y z;
 .....
 residueN x y z].
```

OUTPUT

DYNAMICS creates the following output arrays containing the results:

RES the results of fit, in the following format (in each row) (resid denotes current residue number):

resid#, τ_c , $S^2(\text{resid})$, $\tau_{\text{loc}}(\text{resid})$, $R_{\text{ex}}(\text{resid})$, $S_{\text{fast}}^2(\text{resid})$, $\tau_{\text{fast}}(\text{resid})$, $\text{ind}(\text{resid})$, $\chi^2(\text{resid})$

for definition of these parameters see [1]. *ind* is the index of a SDF-model selected for the residue: *ind* values from 0 to 6 correspond to SDF-models from A to H in the paper:

<i>ind</i>	SDF-model abbreviation, DYNAMICS	SDF-model, ref.[1]
0	LS_00	A
1	LS_tl	B
2	LS_ex	C
3	LS_tx	D
4	CL_00	E
5	CL_tl	F
6	CL_ex	G
7	CL_tx	H

If experimental data being analyzed were collected at several spectrometer frequencies, the R_{ex} is reported for the first frequency in the *freq* list; the R_{ex} values corresponding to all other frequencies can be recalculated according to $R_{\text{ex}} \propto (\text{freq})^2$; e.g. $R_{\text{ex}}(2) = R_{\text{ex}} * (\text{freq}(2)/\text{freq}(1))^2$. Units of measurement: τ_c , τ_{loc} , and τ_{fast} are in [ns], R_{ex} is in [1/s].

ERR errors, δ , in the evaluated parameters, for all residues listed in RES:

resid#, $\delta\tau_c(\text{resid})$, $\delta S^2(\text{resid})$, $\delta\tau_{\text{loc}}(\text{resid})$, $\delta R_{\text{ex}}(\text{resid})$, $\delta S_{\text{fast}}^2(\text{resid})$, $\delta\tau_{\text{fast}}(\text{resid})$

RESERR a matrix containing both the results of the fit and their errors (a merge of RES and ERR).

TAUCHI a record of all evaluations performed during the current DYNAMICS run (sum over all residues), in the following format (each row):

τ_c , $\chi^2(\text{mod})$, N_{nomod} , $\chi^2(\text{nomod})$, runs, df, $\chi^2(\text{total})$, $\chi^2(\text{total})/\text{df}$, N_{excl}

here *nomod* refers to those residues listed in *NOMOD* (see below) (N_{nomod} is number of such residues, $\chi^2(\text{nomod})$ is their contribution to total χ^2 , $\chi^2(\text{total})$; N_{excl} is the number of residues excluded during model selection (see explanation to *EXCL* below); *df* is the total number of degrees of freedom, estimated as the number of experimental data minus number of fitting parameters; *runs* counts the number of switches between two

‘different’ SDF-models in the adjacent residues (here ‘different’ refers to L&S model [2] and the ‘extended’, Clore’s model [3] (CL) for the spectral density function). *TAUCHI* is displayed in the isotropic and mono-dimer modes (*ANISO* is empty); in the anisotropic mode *ANISO* is displayed (*TAUCHI* is empty), instead.

ANISO a record of all evaluations performed during the current DYNAMICS run in the anisotropic mode:

$$\tau_c, D_z/D_x, \theta, \phi, \chi^2(\text{mod}), N_{\text{nomod}}, \chi^2(\text{nomod}), \text{runs}, \text{df}, \chi^2(\text{total}), \chi^2(\text{total})/\text{df}, N_{\text{excl}}$$

Here $\tau_c = 1/(6 \text{ tr}(\mathbf{D}))$, $D_x (=D_y)$ and D_z are principal components of the rotational diffusion tensor, θ and ϕ are the Euler angles that determine orientation of the overall rotational diffusion tensor in the reference frame of the NH-vectors data set, all other parameters are the same as in *TAUCHI*.

EXCL a list of residues that have been excluded by the program (none of the tested SDF-models was able to provide a physically meaningful set of microdynamic parameters, see [1]).

NOMOD a list of residues for which none of the SDF models passed the goodness-of-fit test (χ^2 too high!), although at least one model provided a physically meaningful set of microdynamic parameters (see [1]).

AUTOSAVING FEATURE

To prevent accidental loss of evaluated data, the results (*RES*, *NOMOD*, *EXCL*, *TAUCHI*, *ANISO*) are automatically saved to a **filename.mat** file after completion (and acceptance) of the model-selection step and then after error evaluation (same parameters as above plus ERR). To make it unique, **filename** contains the current date (in UNIX) and a random number from 0 to 99, e.g. **dy7dec97_21.mat** or **dyres_21.mat**, in UNIX or DOS/WINDOWS, respectively. Current setup is done for UNIX and WIN95. To switch to shorter filenames (DOS/WIN3.x), you have to switch commenting in the lines 17-18 in **dynamics.m**, as shown below:

for UNIX/WIN95:

```
fname=['dyn',lower(strrep(date,'-','')), '_ ',num2str(fix(99*rand))];           %UNIX
%fname=['dynres',num2str(fix(99*rand))];                                     %WIN3.x
```

for DOS/WIN3.x:

```
%fname=['dyn',lower(strrep(date,'-','')), '_ ',num2str(fix(99*rand))];       %UNIX
fname=['dynres',num2str(fix(99*rand))];                                     %WIN3.x
```

RUN-TIME DIALOG

The program runs in a dialog mode, remaining in the parameter-optimization regime or switching to the next step (error evaluation or termination), depending on user’s

response. Questions are marked with the prompt ‘==>’ and usually are self-explanatory. The angle values are in degrees, *TAU* in nanoseconds, *Ct* and *Kd* in mM. As a rule, a negative input value (e.g. for *TAU*, *Dz2Dx*, *THETA*) terminates the current mode and forces the program to go to the next step. For example, input of –1 for *TAU* is considered as a command to accept the results of the very last run (prior to the input of negative *TAU*) and to switch to the error-evaluation regime. Note: if an input prompt contains square brackets, as e.g. in *input the PHI angle [20]==>* , the value in the square brackets (in this case, $\phi = 20^\circ$) will be automatically assumed, if instead of typing a number the user simply hits <ENTER>.

Since the optimization usually goes through several cycles of parameter input (*TAUc* in the iso- or mono-dimer modes, *Dz2Dx*, *THETA*, and *PHI* – in the anisotropic mode), to be sure the final result is calculated with the optimized parameter set, you have to select the optimal parameters and run the calculation with this parameter set before exiting or proceeding to error evaluation. In the isotropic and mono-dimer modes it is done by typing in the final value of *TAU*, when you are asked: *input TAU (TAU<=[0] – break)==>* , letting the program to calculate microdynamic parameters with this value of *TAU*, and then typing in a negative value for *TAU* when you are asked the same question next time. In the anisotropic mode, you have to select ‘2-man.input’ as your response to the ‘satisfied?’-question, then input *Dz2Dx*, *THETA*-, and *PHI* angles as you want them finally to be, let the program run through one calculation cycle, and when the same question pops up next time, select: ‘1-yes(calc.err)’.

The error estimation can be done either by Monte-Carlo simulation of the fitting parameters (using inverse covariance matrix approach) or via Monte Carlo simulation of (synthetic) ‘experimental’ data sets and subsequent fit of these simulated data. The latter approach is usually less accurate and much more time consuming. In those cases when for a given residue the number of experimental data equals the number of fitting parameters, the data-simulation mode will be chosen independent of the user’s selection. The number of accepted points in the Monte Carlo simulation is controlled by parameter *mc_max* in **dynamics.m** (by default set to 500). Performing parameter simulation with *visualization=on* will in addition display confidence regions in the simulated parameters (only if a SDF model with more than one fitting parameter was selected, hence not for LS_00).

Note: the τ_c -error-evaluation option is included for testing purposes only. Therefore it is recommended for the error evaluation of all other microdynamic parameters to avoid *TAU*overall variation, i.e. to answer 0 to the question: ‘should *TAU*overall be varied as well?’.

OPTIMIZATION STRATEGY

When the record of all previous steps is printed on the screen (see *TAUCHI*), a parameter set with the lowest $\chi^2(\text{total})/df$ is indicated by an arrow (<--) on the right. This is done to guide the eye and does not necessarily imply that this is the optimal set the user is looking for. Indeed, when *TAUc* deviates from the actual τ_c towards the lower values, to compensate for this, the number of residues with $R_{ex} \neq 0$ increases, which eventually will lead to a decrease in $\chi^2(\text{total})/df$ and in the number of degrees of freedom, *df*. The

more TAU_c is increased compared to τ_c , on the other hand, the more residues will be assigned to CL-type SDF model (both the slow and fast local motions). This also will lead to a decrease in $\chi^2(\text{total})/df$ and in df . These two extremes, if not accurately analyzed, may lead to erroneous results. The ‘physically meaningful’ optimal parameter set is probably the one with the least $\chi^2(\text{total})/df$ AND the highest df . To preserve continuity in the backbone motional properties for the adjacent residues, and to avoid biasing towards the CL-type motion models, which are usually expected only in the loop regions (in several subsequent residues), this optimum has to be accompanied by a minimum in *runs* (see above).

CLEANING MEMORY

The program generates and keeps most of the necessary intermediate variables in the current (global) memory. They are removed from the memory if the program terminates successfully. If the program run has been terminated by user (e.g. via CTRL/C) or in case of run-time error, these variables will remain in the memory. This might cause an interruption in the normal program execution when you start it next time during the same Matlab session. To assure uninterrupted program flow, it is recommended to clean the Matlab memory from only these remaining intermediate variables by issuing a command: *dynclean* before restarting DYNAMICS.

6. Demos

Three demo scripts, **demo_iso.m**, **demo_ani.m**, and **demomdeq.m** are included, illustrating/covering the basic applications of DYNAMICS to relaxation data analysis in the case of isotropic, anisotropic overall rotation and monomer-dimer equilibrium, respectively. Each of these scripts starts with generating/simulating synthetic experimental data sets, using **reldata.m**, and then performs the data analysis. Additional three text files, **demo_iso.txt**, **demo_ani.txt**, and **demomdeq.txt**, contain a copy of the screen output and the dialog, to illustrate the main steps in data analysis using DYNAMICS.

7. Copyright, registration and feedback

© 1997 Copyright by David Fushman, The Rockefeller University.

DYNAMICS comes with ABSOLUTELY NO WARRANTY. This is a free software, and you are welcome to redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or any later version.

The program package DYNAMICS is provided AS IS. The authors do not have any obligation regarding program support, maintenance, and upgrade of the package. However, if you register yourself, you will be notified of any future program updates/releases.

If you publish any results derived using these programs, please cite our papers

“The main chain dynamics of the dynamin Pleckstrin Homology (PH) domain in solution: analysis of ^{15}N relaxation with monomer/dimer equilibration”, D.Fushman, S.Cahill, & D.Cowburn (1997) *J. Mol. Biol.* **266**, 173-194.

and

“Characterization of the overall and local dynamics of a protein with intermediate rotational anisotropy: Differentiating between conformational exchange and anisotropic diffusion in the B3 domain of protein G”, J.B.Hall, D.Fushman (2003) *J.Biomol.NMR* (2003) 27, 261-275.

In case of questions and for registration please contact:

David Fushman
Department of Chemistry & Biochemistry
Center of Biomolecular Structure & Organization
University of Maryland
College Park, MD 20742-3360, USA
Phone: 301-405-3461
Fax: 301-314-0386
Email: fushman@wam.umd.edu

Please use the above mentioned e-mail address to report any bugs in the program.

8. References

1. Fushman, D., Cahill, S., and Cowburn, D. (1997) *J. Mol. Biol.* **266**, 173-194
2. Lipari, G., and Szabo, A. (1982) *J.Amer.Chem.Soc.* **104**, 4559-4570
3. Clore, G., Szabo, A., Bax, A., Kay, L., Driscoll, P., Wingfield, P., and Gronenborn, A. (1990) *J.Amer.Chem.Soc* **112**, 4989-4936
4. Tjandra, N., Feller, S. E., Pastor, R. W., and Bax, A. (1995) *J. Am. Chem. Soc.* **117**, 12562-12566.
5. J.B.Hall, D.Fushman (2003) *J.Biomol.NMR* (2003) 27, 261-275.