RotDif

A computer program for the determination of the complete rotational diffusion tensor of a molecule from ^{15}N relaxation measurements.

Created by Olivier Walker & David Fushman
Department of Chemistry & Biochemistry
University of Maryland, College Park

A brief User's Manual

Things you need before you can start.

1. **Matlab version 6.1 or later** (I haven't tried 6.0, it might work, but 5.3 and earlier will not).
2. You need relaxation data, R_1, R_2, and NOE, organized as matlab matrices (arrays). The format for each data is as following: it is a Nres-by-3 matrix, where Nres is the number of residues that you have data for, the first column contains res. number, the second column contains the actual data (e.g. R2 values) and the third one contains experimental errors for the values in the second column. Here R_1 and R_2 are *relaxation rates* ($R_1 = 1/T_1$; $R_2 = 1/T_2$) and heteronuclear NOE is the ratio of peak intensities in the NOE and NONOE experiments. I usually use the R_1 and R_2 values expressed in 1/ms instead of 1/s, but this should not matter as long as *both R_1 and R_2 are in the same units*.
3. You need a set of NH-vector coordinates (here will be referred to as vNH), in the format of a Nres-by-4 matrix, where the first column is the res. number, and the other three columns contain the x-, y-, and z- coordinates of a *unit* vector in the direction of the NH-bond. The number of residues in vNH can be greater than that in the relaxation data sets – the program will select only those residues that you have relaxation data for. For your convenience, I included a program **pdb2nh.m** that will read in a PDB data set and produce the vNH matrix. It can be called from Matlab shell as follows (assume the pdb-file is 1ubq.pdb) (keep in mind: you have to go to this directory in order to call up this command or you have to include this directory into your matlabpath):

```
vNH = pdb2nh('1ubq.pdb');
```

This is the simplest call for this function. More options could be used as follows:

```
vNH = pdb2nh('1ubq.pdb',reslist, model);
```

here **reslist** is a vector (e.g. [1 2 3 4 5 6 7 8 9]) containing the list of residues you want to take from the pdb-set;
model – the model (or structure) number, when you are reading in a bundle of NMR structures and want to take a particular structure (e.g. type 5 for structure #5). If no model is specified, the first structure is taken by default. When reading a pdb file, the program will consider "TER", "END", or "." At the beginning of a line as stop-signals.

This program recognizes and reads amide protons written as ‘H’ or ‘HN’. If none of the formats was found (e.g. you are dealing with crystal structure), then the program will build amide hydrogens assuming a planar geometry of the peptide plane, according to the rules similar to Insight.

How to run it.

1. Start Matlab.
2. In the Matlab shell, navigate to the directory where the program is located (e.g. ROTDIF directory: **cd ROTDIF**) or include this directory in your current matlab-path, e.g. **path(path,'c:/mymatlab/ROTDIF');**

3. Read in all necessary data. You can either load them separately, e.g.

```
load r2          (will load r2.mat)
load r1
load NOE
load vNH
```

   or (better!) you create a matlab data set containing all these matrices (e.g. relaxdata.mat) and load them at once:

```
load relaxdata
```

4. Now you can run the program by issuing the following command (in one line):

```
RotDif(freq,r1,r2,r3,vNH,reslist,output,pdb_file,alg,mc_max,niter,guessMC_ax,guessMC_full);
```

Here:
freq is the spectrometer frequency (in MHz) = 600.13 (for 600 MHz),
r1,r2, r3 are the R_1, R_2, and NOE data sets (see above)
vNH is the set of NH-vector coordinates (see above);
reslist is the list of residues (e.g. core residues);
output should be substituted with the actual file name (a string variable) where you want to send a detailed report of your calculations (in ascii format), e.g. ’output.txt’;
pdb_file -- specify the filename if you want RotDif to read atom coordinates directly from a PDB-file (must be a string variable, e.g. ‘1ubq.pdb’);
alg specifies the optimization algorithm: enter 1 for Levenberg-Marquard (default), 0 for Newton;
mc_max is the number of points for MC estimation of errors in the parameters (default = 1000);
niter is the number of randomly simulated starting guess sets (default = 50);
guessMC_ax is a vector of the lower and upper boundaries of the fitting params for the axially symmetric diffusion tensor model:
$[(TAUx)_{min}\ (TAUx)_{max}\ (D/D)_{min}\ (D/D)_{max}\ \alpha_{min}\ \alpha_{max}\ \beta_{min}\ \beta_{max}]$;
guessMC_full is a vector of the lower and upper boundaries of the fitting params for the fully anisotropic diffusion tensor model:
$[(Dxx)_{min}\ (Dxx)_{max}\ (Dyy)_{min}\ (Dyy)_{max}\ (Dzz)_{min}\ (Dzz)_{max}\ \alpha_{min}\ \alpha_{max}\ \beta_{min}\ \beta_{max}\ \gamma_{min}\ \gamma_{max}]$).

If you are using default settings, you don't have to specify the last 5-6 parameters. You can use a simpler command:

```
RotDif(freq,r1,r2,r3,vNH,Reslist,Output)
```

The shortest call command possible is

```
RotDif(freq,r1,r2,r3,vNH)
```

In this case the program will assume that you want to take *all* residues into account and will write output to a file **junk.txt.**

If you prefer to let RotDif read in a pdb data set, run it as follows:

```
RotDif(freq,r1,r2,r3,[],reslist,output,pdb_file)
```

where **pdb_file** is the filename (a string variable) of a pdb data set you want to read in, e.g. '1ubq.pdb', and NH was replaced by an empty input (i.e. **[]**). Note that the input parameter vNH has priority: if the input vNH is not empty, the program will take it and ignore the **pdb_file** name.

Further options of running RotDif can be found in the info-header of this program (see also below).

5. During the run you will have to answer questions (yes/no). You can also skip the computer-user dialog if instead of RotDif.m you run an automatic-run program, **RotDif_fly.m**, that has the same call syntax but runs automatically with the computer-user dialog suppressed (I am usually running this one).
6. Watch the program running and enjoy it!

Tips: You can specify a subset of residues (**Reslist**) from the whole data set that you want to be taken for the analysis. For example, I suggest that you leave out residues in flexible loops and those that have significant Rex contributions.
For your convenience, I also included a program **drop_res.m** that allows the user to take out selected residues from a list of residues:

```
subset_list = drop_res(full_list, res_to_exclude)
```

Demo: for your convenience, I also included a demo set that will allow you to run the program and see how it works. This might also be useful if you want to check what data format is required. The demo data are in the file **demo.mat**, to start the demo session, just type

```
rundemo
```

when you are in the RotDif directory. The output parameters are listed in the file **demojunk.txt**, see also the headlines in the file **rundemo.m**

Visualization of the diffusion tensor axes.

A sketch of diffusion tensor axes will be displayed in Fig.1, together with the orientations of the NH vectors. You can also create pseudo-atoms positioned in the center of mass and at the top points of the corresponding axes and add them to the desired PDB file, so that the axes can be displayed using standard protein structure visualization software (e.g. MolMol). This action is performed using program **addaxes.m** that is included in the package. Call it as

addaxes(fname_input, fname_output, rot_angle, rad, reslst)

here **fname_input** is the filename of the input PDB file, **fname_output** is the output PDB filename, **rot_angle** is a vector of the Euler angles [alpha,beta,gamma] (these you obtain from RotDif) in degrees, e.g. [30, 27.3, 156], **rad** is the length (radius) of the axis vector (in Angstroms), and **reslist** is the list of residues to read in and then write out. By default, the pseudo atoms added to the PDB file belong to residue # 999 and have the following meaning:

```
ATOM   9995  N   → positioned at the –z end of the z-axis
ATOM   9996  CA  → positioned at the origin (=center of mass)
ATOM   9997  C   → positioned at the +z end of the z-axis
ATOM   9998 1HA  → positioned at the +x end of the x-axis
ATOM   9999 2HA  → positioned at the +y end of the y-axis
```

A word of caution. Because the initial guesses for the principal values of the diffusion tensor are generated within a certain range, you might run into a problem that the actual parameters are outside this search range. In this case one or more of the final values of D’s (or TAUx) are likely to be exactly or close to the edge(s) of the interval of initial guesses. By default, these ranges are:

Axially symmetric model:

$1.0 < TAUx < 15.0$

$0.5 < D/D < 1.8$

Fully anisotropic model:

$0.5 < Dxx < 12.0$

$0.5 < Dyy < 12.0$

$0.5 < Dzz < 12.0$.

The program prints the actual search ranges on the screen. Make sure that your final results are far from these edges. I would advise that you increase the search interval for the Ds and TAUx and repeat your calculation – this does not take much time.

To change the search range, run the program as

RotDif(freq,r1,r2,r3,vNH,Reslist,Output,1,1000,50,guessMC_ax,guessMC_full),

where for these purposes you want to explicitly enter/modify the following input parameters **guessMC_ax** and **guessMC_full**:

guessMC_ax is a vector of the values of the boundaries of the search range for an axially symmetric model, i.e. $[(TAUx)_{min}\ (TAUx)_{max}\ (D/D)_{min}\ (D/D)_{max}\ \alpha_{min}\ \alpha_{max}\ \beta_{min}\ \beta_{max}]$;

the default settings are [1.0 15.0 1.0 1.8 0 pi 0 pi];

and

guessMC_full is a vector of the default values of the boundaries of the search range for a fully anisotropic model. i.e.**[$(Dxx)_{min}$ $(Dxx)_{max}$ $(Dyy)_{min}$ $(Dyy)_{max}$ $(Dzz)_{min}$ $(Dzz)_{max}$ α_{min} α_{max} β_{min} β_{max} γ_{min} γ_{max}].** The default settings are [0.5 12.0 0.5 12.0 0.5 12.0 0 pi 0.0 pi 0.0 pi].

You might want to increase the upper values of the Ds and D/D and TAUx by a factor of 2 and repeat the calculation. Don't change the ranges for the angles – these are already full ranges needed for the search.
Note that TAUx is in nanoseconds, D/D ($=D_{\|}/D_{\perp}$) is dimensionless, the values of Dx,Dy,Dz are in 10^7 s^{-1}, the Euler angles $\{\alpha,\beta,\gamma\}$ are in radians.

In the case of the axially symmetric model, set $(D/D)_{min} = 1$ and $(D/D)_{max} > 1$ for a prolate model, and $0 < (D/D)_{min} < 1$ and $(D/D)_{max} = 1$ for an oblate model.

Conditions/disclaimer. The program is provided as is. You should include a reference to our paper (*Walker O, Varadan R, Fushman D. 2004. Efficient and accurate determination of the overall rotational diffusion tensor of a molecule from 15N relaxation data using computer program ROTDIF. J. Magn. Reson.* ***168****:336-345*) in any publication that includes/uses data obtained using the program.

References. A detailed description of the algorithm, the underlying theory, and some application examples can be found in
O.Walker, R.Varadan, D.Fushman, "Efficient and accurate determination of the overall rotational diffusion tensor of a molecule from 15N relaxation data using computer program ROTDIF", *J. Magn. Reson.* ***168****, 336-345* (2004);
D.Fushman, *R. Varadan, M. Assfalg, O.Walker* "Determining domain orientation in macromolecules by using spin-relaxation and residual dipolar coupling measurements" *Progress in NMR Spectroscopy* ***44***, 189-214 (2004).