Documentation for the MD2NOE_Protein package

Theory of calculation.

The Bruschweiller formulation of noe spectral intensities is used to calculate the noe itensities. This means that a complete relaxation rate matrix is used in calculating the noe build-up curves and intensities. The isolated spin pair approximation is also used. This complete relaxation rate matrix is well known, and provides a better approximation to the noe's than a 1/r^6 approximation. It has to be used in a differential equation to find the magnetization transfer from a perturbed set of protons back to equilibrium.

The calculations are based on MD trajectories, without approximations such as curve fitting exponentials to the correlation function. There are many Matlab programs in this package. The package is developed to calculate these spectral intensities for molecules which have a large tumbling time, such as proteins. The programs can also be used to calculate the noe spectral intensities for small molecules, such as carbohydrates. The difference in these calculations from a typical complete relaxation rate calculation is that for large molecules the package uses an approximation of the correlation function from the MD trajectory. This effectively increases the sampling of the motion of the molecule. Large molecules require a very long trajectory for accurate sampling of the molecular orientations. A 'sphere approximation' is used, in which the tumbling is artificially added to an aligned trajectory. This is added to the global motion to model the overall tumbling. The sphere approximation improves sampling for trajectories which don't sample all the orientations.

There are two relevant correlation functions. The first uses an unaligned trajectory, with the orientation of the B-field. This is the Bruschweiler correlation function. The overall correlation function between two protons is (1)

$$C(t) = < \frac{P_2(\cos(\chi_1))P_2(\cos(\chi_2))}{r(t)^3 r(t+\tau)^3} >$$

The angles χ_1, χ_2 are defined by the inter-distance vector of the H-H at times $t$and $t+\tau$ with the direction of the B-field. P_2 is the 2nd order Legendre polynomial. $r(t), r(t+\tau)$ are the inter-distances between the H-H spin-pair at these two times. This correlation is well-suited for molecules and trajectories in which the orientation between the two protons of the molecule is sampled often enough to reproduce a distribution which is even in all directions. The correlation function can be represented by a sum of exponentials, for each of the processes.

The problem with this correlation function for a large molecule, such as a protein, is that the sampling within the trajectory is not frequent enough to model the ergodic motion. These correlation functions, if well-defined, have to decay to zero and if the trajectory is not long enough then sampling of these orientations is not sufficient for a well-defined calculation. A protein of the size of St6Gal1, with an approximate M_r of 36 kDa, can not sample adequately enough the orientations in a 200 ns trajectory.

If the overall tumbling is uncorrelated with the internal motion, and the tumbling is isotropic, then the correlation function should take the form (2)

$$C(t) = C_{internal}(t)\, C_{tumbling}(t) \qquad C_{tumbling}(t) = e^{-t/\tau}$$

where τ is the overall bulk tumbling time. The internal motion correlation function from the trajectory for a H-H spin-pair is defined as in (2), which is also the same as averaging the first correlation function over all directions of the B-field (4),

$$C(t) = < \frac{P_2(\cos(\chi_{t,t+\tau}))}{r(t)^3 r(t+\tau)^3} >$$

where the angle between distance vectors at times t and t+τ is $\chi_{t,t+\tau}$. P_2 is the 2nd order Legendre

polynomial. $r(t), r(t+\tau)$ are the inter-distances between the H and H at these two times. This "internal" correlation function is multiplied by $e^{-t/artificial\_tumbling}$ to include the overall tumbling, using the fact that the correlation times of the internal motion is much different than the overall correlation time (2).

The correlation function is then numerically Fourier transformed to find the spectral density function, J. This function is used to calculate the complete relaxation rate matrix. The advantage of a numerical Fourier transform is that no approximation is made in curve fitting the correlation function to a sum of exponentials.

Next, the complete relaxation rate matrix is found from the approximated correlation function.

Here, $J_{i,j}$ is the Fourier transform of the correlation function (i.e. spectral density function) for a pair of protons (i,j), in our case including both distance and angle variations. It is sampled at the relevant frequencies, 0 and 2ω, where ω is the proton Larmor precession frequency in radians/s. Hence, σ can be evaluated for any proton pair using the correlation functions described in (2,3).

$$\sigma_{i,j} = \frac{(dd)^2}{4}\{-J_{ij}(0) + 6J_{ij}(2\omega)\} \quad (5)$$

Experimentally, extraction of an initial slope from an NOE build-up curve is challenging because of the presence of indirect transfers and processes causing relaxation back to equilibrium. The rate constant for the latter relaxation process, ρ, is given in Equation 3 where the summation is over all spins having a significant interaction with spin i (typically within a distance of 4 or 5 Å).

$$\rho_i = \frac{(dd)^2}{4}\sum_{j\neq i}\{J_{ij}(0) + 3J_{ij}(\omega) + 6J_{ij}(2\omega)\} \quad (6)$$

There is a normalization pre-factor of in the evaluation of the spectral density functions.

$$dd^2 = \left(\frac{\mu_0}{4\pi}\gamma_H\gamma_H\frac{h}{2\pi}\right)^2 \quad (factor) = 1.897 * 100000 * 1.897 * 100000$$

This normalization factor includes the fact that the correlation function is calculated in angstroms. γ is the gyromagnetic ratio.

These two variables $\rho_i, \sigma_{i,j}$ are the diagonal and off-diagonal elements of the complete relaxation rate matrix.

Our implementation of the sphere approximation has one more layer of complication. The variables i and j range over all protons in the molecule in principle. For a protein this is too many protons. We use a radius cutoff of a user input, i.e. 7 Angstroms, about the amide protons and only include spin-pairs in the matrix which are < 5 Angstroms in the matrix. The formalism, and the reasons for it, are described in detail in the MD2NOE_Protein presentation.

In order to find the spectral intensities, i.e. the 1-D noe curves, the complete relaxation rate matrix is exponentiated in a linear differential equation,

d/dt X = - R X

where R is the complete relaxation rate matrix. Boundary conditions on X at t=0 to t=\infinity specify the type of NMR experiment done. These boundary conditions model the initial and final spin states of the protons. This is documented also in the MD2NOE_Protein presentation.

Aspects of the calculation

The primary improvement in MD2NOE_Protein from previous packages is that the trajectory is used in all parts of the calculation. The output of this package is used in the companion package Assign_SLP_MD. This latter package uses a genetic algorithm approach to find the statistically most relevant assignment of residue to spectral intensity in the collected NMR spectral data. The package

MD2NOE_Protein is also a stand-alone package that can be used without Assign_SLP_MD to simulate the output of a NMR spectrometer for any type of experiment. There is an analogous version of MD2NOE_Protein which uses a single frame of a trajectory, i.e a PDB file. The MD2NOE_Protein package uses pthreads and takes hours to generate many 1-D curves for a sparsely labeled protein calculation. The single frame simplification operates quickly, within minutes, uses the 1/r^6 approximation, but is not accurate for large proteins which have many protons, where spin diffusion and proton motion is important.

The NOE 1-D curves, for which we are interested in the case of sparsely labeling, use the trajectory in the correlation functions. The complete relaxation rate matrix is used to deal with spin diffusion and also frames of the trajectory are anonymously closer than 5 Angstroms. The MD2NOE_Protein gives several examples in which the trajectory is required to avoid miscalculation. The 1-D noe curves also require chemical shifts of the protons within each 1-D curve. These are found from an average of chemical shift predictions of protons from both ppmOne and shiftx2, using the entire trajectory or a sampling of the entire trajectory. The noe 1-D curves are much closer to the experimental result than using a 1/r^6 approximation, and generally the spectral intensity 1-D curves contain >50 contact protons in each for a protein the size of st6Gal1.

The package Assign_SLP_MD package uses the calculated 1-D curves, the improved RDC back calculation, and the calculation of predicted chemical shifts of the H and N or H and C. The MD2NOE_Protein has a program which calculates order parameters from the average over frames in the trajectory. These order parameters improve the RDC back calculation in the Assign_SLP_MD package.

The drawback of the current version of MD2NOE_Protein is that it could potentially generate >200 GB of files. Options exist for the user to limit the output of figures, etc.., but the memory space required can be quite large. There will be an improvement in the next version.

It is recommended that a 1000ns trajectory is used for a large protein such as st6Gal1 for reasons of sampling. The trajectory has to be aligned first, such as with all backbone C atoms in the protein's backbone. The example calculation used a trajectory separated into 5 segments, each 200ns. This, however, is not necessary, and an identical result for the average of 5 segments can be found by using a single 1000ns trajectory. The correlation functions are calculated using a 'sliding window'. Then the programs should be used to calculate the 1-D spectral curves for each segment, and then averaged over all segments.

The entire 1000ns trajectory should be used to calculate the order parameters for each sparse residue. Not every frame has to be used, and in the work associated with this package every 100^{th} frame was used, i.e. sample_frequency=100.

Documentation and Instructions

This software package is primarily designed to calculates noe's for large molecules from MD trajectories in which there is not adequate sampling. If the isotropic correlation time for a molecule is greater than a few nanoseconds, then a trajectory of a microsecond is still not enough to completely sample all orientations. The software is also designed to be of use for general NMR observables, not just noe's due to the fact that the correlation functions and spectral density functions (SDF's) are calculated. These can be evaluated at different frequencies for other NMR observables; an example are the spin-spin and spin-lattice relaxation rates (e.g. there is a package at glycam.org, particular_relaxation_rate, which uses the same SDF's except is evaluated at different frequencies).

The software is designed to be very user friendly. The package is designed so that not a single output file has to be opened. The end result will be an excel sheet with the 1-D noe curves. The intermediate files

are created in an organized set of directories, one for each residue in the user input of sparse labeling. There are 3 directories for each residue - the correlation functions, the fourier transforms and complete relaxation rate matrices, and the noe build-up curves.

The software is broken into 3 segments

- Internal motion correlation function calculation
- Spectral density function calculation and complete relaxation rate matrix calculation, i.e. Fourier transform of the correlation function and the complete relaxation rate matrix at a particular magnetic field strength.
- Noe calculation using initialization/finalization of spin system.

The reason the software is broken into these 3 segments is due to non-repetition of calculation and user friendliness.

There are several peripheral programs.

- The order parameter calculation program.
- -The noe 1-D curve calculation program, which parses the output of the 2nd program and generates an excel sheet with the columns being the 1-D noe curves. This can be interpreted as a 2-D or 3-D noesy experiment. Instructions for calculating the noe are in this document.
- The chemical shift calculation program. This program calculates the average chemical shifts over the trajectory and could also be used to average the output from both the use of ppmOne and shiftX2. These chemical shift prediction programs do miss chemical shift predictions due to a lack of information in their databases. In averaging, a more complete output file of chemical shift predictions is obtained.

The correlation functions are calculated once, and this does not depend upon the parameters of the experiment such as B-field or type of experiment, i.e. initial and final spin states of the protons. The type of experiment could be non-selective proton spin inversion, selective proton spin inversion such as amide spins, band selective spin inversions, … This first program is the most time consuming and calculationally intense program. The next 2 programs will calculate in a few minutes, for a protein the size of st6Gal1 with 16 phenylalanines.

Next the spectral density functions and complete relaxation rate matrices are numerically calculated without curve fitting with sums of exponentials at some magnetic field strength. The magnetic field strength is required, which specifies the spectrometer. The complete relaxation rate matrix could be re-sscalculated again in a matter of minutes for a different field strength, in case a different spectrometer is required.

The last segment of the package uses the complete relaxation rate matrix from the 2nd program to find the noe curves. This step of the calculation requires a specification of the experiment, which means specifying the initial and final conditions of the spin system of protons. If the experiment has to be changed, for example, in spin inversion of some protons, then the 1-D noe build-up curves could be re-calculated in minutes.

The next program is the noe 1-D curve calculation program. It parses the output of the 3rd program and creates a set of 1-D spectral noe curves. The chemical shift program has to be used first to specify the location of the noe intensities in chemical shifts for each spectral curve.

At each step of the calculation, the previous calculations do not have to be reproduced.

Programs

NOTE – The number of threads is currently hardcoded in the internal correlation function program. Only this program is pthreaded. The default is 2 threads, in the variable NUM_THREADS. If you would like a different number of threads, change NUM_THREADS and recompile the program.

There are bash scripts for the st6Gal1 protein in the download. This protein has 16 PHE amino acids which are used in the sparse labeling. These bash scripts can be changed for any protein. In later versions of the package a GUI will be provided.

There is a cpptraj script, cpptraj_align_rST6Gal1, that aligns a 1000ns trajectory and then breaks it into 5 segments. This script, for st6Gal1, can be modified for different proteins.

It is recommended that absolute paths are used in the bash scripts.

internal_correlation_function.cc

This program calculates the internal correlation function for all spin pairs within the sphere and whose distance is less than the maximum spin pair distance. There is a pre-calculation of the average distance of the spin-pair to shorten the calculation time of the program.

// argv[1] the topology file of the trajectory

// argv[2] the aligned trajectory

// argv[3] the total time of the simulation in ns

// argv[4] the display time in correlation function output – this is used in the output in the correlation file figure and also in what amount of the trajectory is used in the correlation calculation. For a molecule that tumbles in 20 ns, all 200 ns of a trajectory and 100 should be used. There are 2 variables in the correlation function, t and τ

// argv[5] the sample_frequency – the calculation time can be reduced by a factor of 10, for example, if this parameter is set to 10.

// argv[6] maximum distance of spin pair to be in calculation

// argv[7] jpeg,eps,pdf,epslatex – figures – anything else no figure

// argv[8] all/notall – all spin pairs

// argv[9] output directory

// argv[10] residue name – center of sphere

// argv[11] residue number – center of sphere

// argv[12] atom name – center of sphere

// argv[13] centerRadius – radius of sphere – 7 Angstroms is recommended for a large protein

// argv[14] test fraction – percentage of frames in precalculation. There is a test to make the calculation more efficient by checking if a spin-pair could be within the user input of spin-pair distance. For example, 2 protons 10 Angstroms away from each other certainly will not contribute and there is no reason to calculate the full correlation function.

```
mkdir internal_1_200_noe_7.0_29

./internal_correlation_function rST6Gal1.hLoop.reduce.nowat.nobox.prmtop
trajectory_1_200_no_tumbling.crd 200 200 1 5.0 jpeg all internal_1_200_noe_7.0_29 PHE 29 H 7.0 .05
&

sleep 10
```

This program will create a text file with the information of the correlation function for each spin pair. It has 2 columns. The first is the correlation time in nanoseconds. The second is the value of the correlation function. If figures are chosen then a figure of the correlation function is also created. There are potentially hundreds of files of correlation functions and hundreds for figures. The figures can be made in jpeg, eps, epslatex, or pdf. If any other input is used, then no figure is generated.

There is a file file_distance.txt with the distance information of all the spin pairs.

There is a head file internal_correlation_function_internal_801_1000_noe_7.0_295.txt with all information of the correlation functions of that residue. It includes the information used to create the information in the directory for reproducibility, what is in the 7.0 angstrom sphere, and the information such as atom numbers, atom names, residues, …, of all the spin pairs. This file is used in the 2nd program, the complete relaxation rate program for parsing purposes, as it tells where all the internal correlation function files are..

complete_relaxation_rate_program.cc

This program calculates the complete relaxation rate matrix from the output of the first program. In the process the Fourier transforms of all the spin pair correlation functions are calculated.

// argv[1] input directory of internal correlation function output

// argv[2] magnetic field in tesla

// argv[3] maximum frequency of output Fourier transform in MHz. This depends on the magnetic field strength and 500 is recommended for a 21 T spectrometer.

// argv[4] jpeg,eps,pdf,epslatex – anything else no figure

// argv[5] output directory

// argv[6] artificial tumbling – in nanoseconds

```
mkdir crrm_1_200_29

./complete_relaxation_rate_program internal_1_200_noe_7.0_29 14 500 jpeg crrm_1_200_29 21 &

sleep 10
```

This program will create text files of all the correlation functions, with the tumbling time included. These are not internal correlation functions. Figures will be generated if jpeg, eps, epslatex, pdf, is in the input. If these aren't in the input, then there will be no figures.

Fourier transforms are created and stored in a 2 column text file, for each spin pair. The first column is the frequency in MHz. The second column is the Fourier transform. Figures could also be created.

The complete relaxation rate matrix is in the file complete_relaxation_rate_matrix_14_21.txt. It has the inputs that created the directory and the inputs of the internal correlation function calculation. It also the information of all the spin pairs in the complete relaxation rate matrix. This file is used in the 3rd program to direct all of its input.

noe_curve.cc

This program calculates the noe curves of all the spin pairs from the protons in the complete relaxation rate matrix output. The spin system has to be initialized/finalized. There are 2 shortcuts for this input.

The user can type selective or non-selective for these types of proton initialization. The first will invert all spins except the amide and the second will invert all proton spins.

// argv[1] input file – output of complete relaxation rate program

// argv[2] maximum mixing time of noe curve. For a large protein such as st6Gal1, 200 milliseconds is recommended. For carbohydrates, 1000 milliseconds is. Most noe curves will max out at less than 100 ms for a large protein, but there are anomalous noe's.

// argv[3] inter-ms of mixing time – step size of noe corve in ms, e,g, 1000 ms ad step size of 10 will generate files with 100 numbers in each column.

// argv[4] jpeg,eps,pdf,epslatex – anything else no figure

// argv[5] output directory

// argv[6] in spins – initialization/finalization of proton spins, i.e. selective/non-selective inversion, …

mkdir noe_1_200_noe_7.0_29

./noe_curve crrm_1_200_29/complete_relaxation_rate_matrix_14_21.txt 3000 1 jpeg noe_1_200_noe_7.0_29 0 0 &

sleep 10

The program creates text files with the noe curves in 2 column format. The first column is the mixing time in milliseconds. The second column is the noe intensity. Figures could also be created.

There is also a header file, noe_information.csv, in each noe directory which has a list of all the calculated noe build-up curves and which protons are in each spin pair. This file is used by the next program, which creates the excel sheet with all of the 1-D spectral noe vectors.

spectral 1-D curve creation –

This program uses the merged output list of all of the -noe curve header files-, i.e. in the different noe curve directories. There is an example of a merged noe list file in the example directory. This file, merged_noe_401_600 has all of the files in noe's were made from the 16 residues of st6Gal1. Merging

is straightforward for the 16 residues of st6. For each of the segments, the noe_information.csv files are merged,

cat noe_1_200_noe_7.0_29/noe_information.csv noe_1_200_noe_7.0_52/noe_information.csv noe_1_200_noe_7.0_54/noe_information.csv noe_1_200_noe_7.0_68/noe_information.csv noe_1_200_noe_7.0_105/noe_information.csv noe_1_200_noe_7.0_113/noe_information.csv noe_1_200_noe_7.0_137/noe_information.csv noe_1_200_noe_7.0_171/noe_information.csv noe_1_200_noe_7.0_172/noe_information.csv noe_1_200_noe_7.0_187/noe_information.csv noe_1_200_noe_7.0_237/noe_information.csv noe_1_200_noe_7.0_253/noe_information.csv noe_1_200_noe_7.0_254/noe_information.csv noe_1_200_noe_7.0_268/noe_information.csv noe_1_200_noe_7.0_287/noe_information.csv noe_1_200_noe_7.0_295/noe_information.csv > merged_noe_1_200.csv

Each of these can be used for the 200ns segments. If the full peak list from all segments is wanted, then all of these are merged, for the entire 1000ns trajectory,

cat merged_noe_1_200.csv merged_noe_201_400.csv merged_noe_401_600.csv merged_noe_601_800.csv merged_noe_801_1000.csv > merged_noe_1_1000.csv

These merged lists create a direction for the spectral curve scripts to include all of the amide proton spin any proton if in the merged list.

The output list is of all noe curves calculated in all different residues of the sparsely labeling. For example, in st6Gal1 PHE was used and the merged list contains all the different noe build-up curve information for 16 residues.

The user has to make the merged csv file. The mixing time has to be specified. The predicted chemical shift files also have to be specified.

Chemical shift calculations

The use of ppmOne and shiftX2 is now explained. There is a problem between the output of the chemical shift programs and Amber. Several residues in the chemical shift output files have to be renamed.

First, the chemical shift predictions are created for ppmOne and shiftX2. ppmOne does not use disk in its calculations and is much more efficient. shiftX2 is much slower because it uses disk space. Note also that the output format of shiftX2 is different for a trajectory than a single frame.

Both of these packages have to be installed. These are not available online for use in trajectories. The online programs only use a single pdb file.

- **Download instructions**

The use of shiftX2 and ppmOne is straightforward. For shiftX2 a 'model' pdb has to be created using cpptraj. There are instructions at many cpptraj sites. This is an example. This will create a large file with frames of the trajectory turned into pdb files separated by ENDMDL.

parm topology.prmtop

trajin md2.mdcrd 1 19000 5 [start stop offset]

trajout output.pdb pdb append

%% residue names are corrected to agree with amber - such as NLN (N-linked) to ASN

The PDB file has to be corrected to make the names of residues agree with Amber,

cat noe_output_rST6Gal1-3cyx-2ndRun.pdb | sed -e 's/HIP/HIS/' -e 's/HIE/HIS/' -e 's/HID/HIS/' -e 's/CYX/CYS/' -e 's/NLN/ASN/' | grep -v " 0YB " > noe_output_rST6Gal1-3cyx-2ndRun_corrected.pdb

The output file will have these names corrected to agree with Amber.

The shiftX2 function is,

shiftx2.py -i noe_output_rST6Gal1-3cyx-2ndRun_corrected.pdb -p 6.5 -t 298 -m -z /var/tmp/

The directory /var/tmp/ is used as a temporary directory where shiftX2 saves its calculations. shiftX2 has one file in its output.

ppmOne uses the same model pdb file and is called by,

ppm_linux -pdb noe_output_rST6Gal1-3cyx-2ndRun_corrected.pdb

The ppmOne program will output the options if the inputs are incorrect.

ppmOne has several files in its output – proton_predict.dat and bb_proton.dat for sidechain and backbone protons are used in the parsing program. There is also a single file with these 2 .dat files, but this is not used. These proton_predict and bb_proton files potentially contain -999 as chemical shifts. The user has to replace all -999's with 999's in proton_predict.dat and bb_proton.dat for the files to be parsed correctly. This must include a space, so replace "-999" with " 999".

Next the chemical shift files are used with the noe calculations to create the peak lists. The merged list is used to scan for all amide to proton spin pairs that were calculated, and in the merged list. This means that any set of residues in which the noe curves were calculated could be limited if the merged list is limited. This program than parses the noe_curve output files and finds the noe intensity at the input mixing time. The chemical shift file is then parsed, and if the proton is not there, the peak list will contain the noe but the chemical shift will be '999'.

%% shiftX2

input_chemical_shiftX2_file='proton_chemical_shiftX2.dat';

%% noe information file

noe_input_file='renamed_merged_noe_1_1000.csv';

input_residues=[29 52 54 68 105 113 137 171 172 187 237 253 254 268 287 295];

mixing_milliseconds=60;

```
run('/home3/chalmers/MD2NOE_Protein/shiftX2_noe_vectors_trajectory.m
);


%% ppmOne
input_chemical_ppm_one_file='proton_predict.dat';
input_chemical_ppm_one_second_file='bb_predict.dat';
%% noe information file
noe_input_file='renamed_merged_noe_1_1000.csv';
input_residues=[29 52 54 68 105 113 137 171 172 187 237 253 254 268 287 295];
mixing_milliseconds=60;
run('/home3/chalmers/Desktop/SingleFrameNMR/noe_vectors/ppm_one_noe_vectors_trajectory.m');
```

A peak list is generated with information of residues and protons contributing to the 1-D noe spectral curve. An example is in the st6_example directory. Next are two Matlab scripts that can be used to make the 1-D spectral curves. The first averages the 2 peak lists, from ppmOne and shiftX2, if wanted. The second uses the averaged peak list to create the 1-D noe spectral curves.

The script that averages the peak lists from ppmOne and shiftX2 is called 'ppm_one_shiftX2_peak_list_average'. There are 2 input peak lists. These have to be loaded by hand at the moment.

The script that creates the noe spectra from the peak list is called 'noe_vectors_from_peak_list'. Chemical_shift_list is an array with all measured chemical shifts, e.g. 11 to -2. The st6_example has an excel sheet called 'chem_shift.xlsx' with all of these; this is chemical_shift_list. The width of the gaussian widened peaks in the peak list is a user input, e.g. .2 . Peak list is that which generates the spectra. It come from anywhere but it has to be in the format representative in the example directory called peak_list_1_1000_2ndRun.xlsx. All of the scripts given in the src directory will do that.

The program 'noe_vectors_from_peak_list' adds an autopeak to normalize all the 1-D curves with the experimental curves, which should also contain an autopeak. The autopeak has the intensity of the average of the minimum peak (negative noes) from all the noe vectors. It is placed with a width of .2 ppm at the smallest value of chemical shifts, -2.135 in the example.

The spectra will be generated. At the moment the user has to manually save this to an excel sheet. The st6_example directory has the calculated spectra from a 1000ns trajectory.

order_parameter.cc

This program calculates the different order parameters which are used in the improved RDC back calculation. These order parameters provide information about the structure of the protein.

```
//      argv[1] topology file
//      argv[2] trajectory file
//      argv[3] the samplefrequency – 1 in this number of frames
//      argv[4] out directory
//   atom pair
//      argv[5] Residue_name_1
//      argv[6] residue1
//      argv[7] atom1
//      argv[8] Residue_name_2
//      argv[9] residue2
//      argv[10] atom2
```

./order_parameter rST6Gal1.hLoop.reduce.nowat.nobox.prmtop trajectory_no_tumbling.crd 100 order_parameters PHE 29 H PHE 29 N &

The program creates a text file with the order parameters of the atom pair. The text file also has average coordinates of the two atoms over the trajectory and the 'mean normalized' average coordinates. These are,

$$< r1 - r2 > * |r1 - r2|/(|r1| - |r2|)$$

The latter coordinates should be used in the RDC back calculation. The order parameter which is used

$$< 3cos^2\text{-}1>/2.$$

There is additional information in the output file of this program. The program has to be ran on each residue in the sparse labeling. The mean coordinates and order parameters are used in the next package Assign_SLP_MD.