Assign_SLP_MD

Introduction

The software package finds the statistically best assignment of experimental measurement to residue using a genetic algorithm. The package uses the experimental noe spectra, rdc's (any number of media, e.g. peg, phage) and chemical shifts, such as from amides of sparse residues. It also uses the calculated noe spectra from an Amber trajectory, order parameters of N-H or C-H spin-pairs, and predicted chemical shifts. The genetic algorithm search is similar to that described for our earlier non-MD version, ASSIGN_SLP (Chalmers G, Glushka JN, Foley BL, Woods RJ, Prestegard JH. Direct NOE simulation from long MD trajectories. Journal of Magnetic Resonance. 2016;265:1-9). Input and output are significantly different.

There are many matlab scripts in the package. The program is written in an object-oriented sense. The primary programs are

- Assign_SLP
- OutputAnalysisZFactors
- Statistics_z_Factors

Assign_SLP uses the input files in a genetic algorithm to create an output Matlab data file with all possible solutions to an assignment with a total fitness less than some value specified by the user.

OutputAnalysisZFactors converts the output Matlab data file into a text file with or without statistical information about the assignments. There are two modes – verbose and non-verbose. The solutions in the output text file are listed from lowest fitness to largest fitness, and are all unique. This program also generates a probability heatmap with a statistical interpretation of the possible solutions in the output text file.

Statistics_z_Factors is a program which generates, for example, 10^6 random possible solutions, and performs statistics. This calculates mean, median, and standard deviations for all the quantities in the output text file. Z-factors are also calculated. These are used to give the user information about the ranked solutions in the output text file.

There are many input parameters in these programs. These will be explained in the program descriptions. The calculated input parameters are –

- Predicted noe vectors found from noe calculations of the trajectory
- Order parameter corrections to the rdc back calculation
- Chemical shift averages of the frames of the trajectory

The experimental inputs are –

- Experimental noe spectra
- Experimental RDC's
- Experimenatl chemical shifts of the nuclear pairs as seen in an HSQC spectrum (amide proton and nitrogen shifts in the example given).

There are bash scripts in the package which demonstrate the use of the software. The input files are described, then the bash scripts.

Assign_SLP_MD improvements from earlier packages

Improvements in the Assign_SLP_MD package are :

- The noe vectors are made from the complete trajectory using the MD2NOE_Protein package
- The back calculation of the rdcs use the average mean coordinates and order parameters from the complete trajectory. The order parameters and average mean coordinates can be calculated in the package MD2NOE_Protein.
- Both the noe calculations and the order parameter calculations are not part of this package, Assign_SLP_MD . These calculations are done in the package MD2NOE_Protein.
- The entire trajectory is used to calculate the order parameters.
- The noe calculations are done in 200ns segments from, for example, a 1000ns trajectory. The average of these 200ns segments for each sparse residue is then used as the input 1-D noe vectors in an excel sheet for this package. The reason for using 200ns segments is found in MD2NOE_Protein.
- The experimental noe peaks are broadened with the program Peak_Widen to mimic predicted peaks. These are broadened to reflect expected errors in chemical shift prediction. It is recommended that a width of .2 ppm be used.
- Shiftx2 and ppmOne are used to calculate the chemical shifts for all frames within the trajectory. The average chemical shifts are used from the full trajectory. This calculation is done using the package MD2NOE_Protein.

- Noe vectors of any length and chemical shifts can be used. The predicted and experimental noe vector files have to be the same size.

- There are prep files in the example directories.

Genetic algorithms

Genetic algorithms are well known to solve or approximate NP-hard problems. Genetic algorithms are a way to optimize problems using evolutionary computation. The algorithms are well known, except the actual implementation can be difficult due to tuning and initializing the population. This evolutionary approach to calculation starts with a population of possible solutions then improves the population. The solution is formulated as a minimum of an objective function.

In our case, each individual of the population is a permutation that matches the peak measurements to the residues. Each of the parameters of the permutation are genes of the individual. If there are N measurements then the chromosome has N genes. There is a fitness given to each of the individuals which involve both the objective function and the constraints. This fitness involves the rdc back calculation, an rmsd of the measured to calculated chemical shifts, and an rmsd of the noe measurements to predicted. The constraints are that there are no two peak measurements assigned to the same residue.

The user must specify when the algorithm will stop by providing a desired accuracy of the solution.

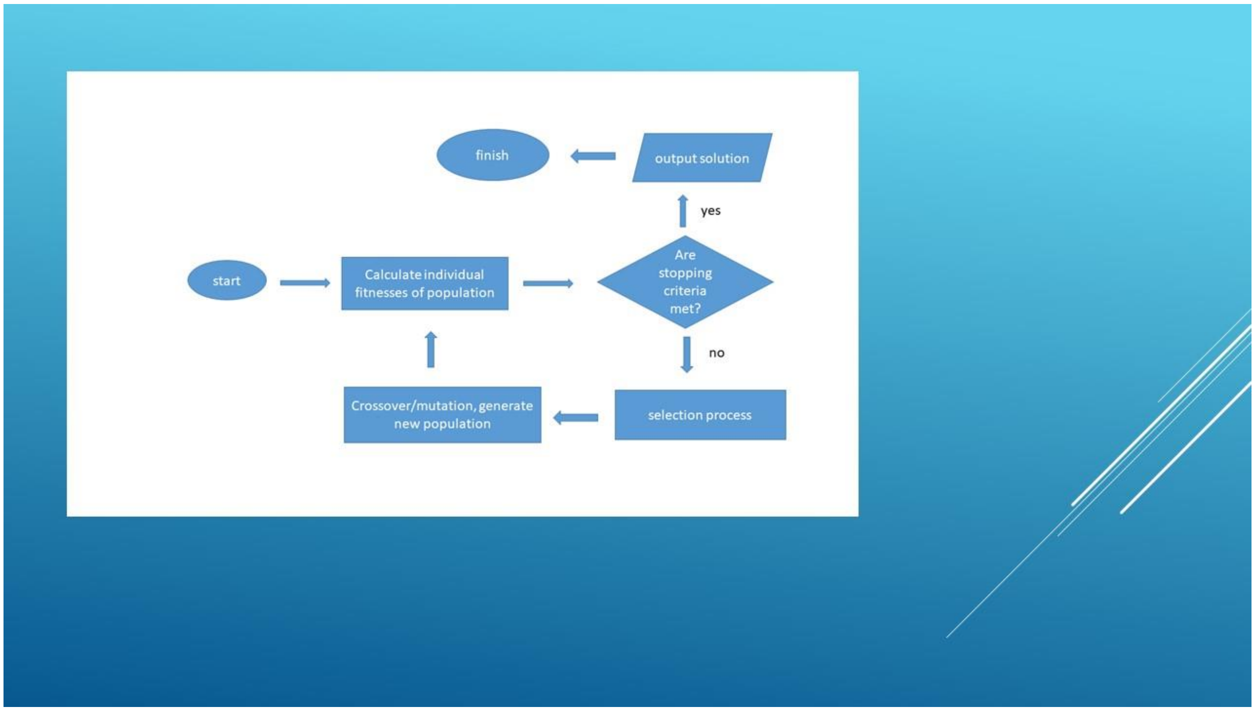
The genetic algorithm peak assignment problem is unusual in that the correct solution of the peak assignment may not be the global minimum of the fitness function. The programs will

- Search the space and save all the information of possible solutions which are within a distance based on expected errrors.
- Scan the crossover and mutation rates. This partially solves the tuning problem, which is part of any genetic algorithm implementation.

The search space that the genetic algorithm must use could be quite large. If there are 20 measurements then the search space is 20!, which is 10^19. If there are 100 residues, then the search space is 10^157. An exhaustive search of all possible solutions is prohibitive beyond 13 measurements and residues.

For problems of this size, the genetic algorithm could be slow. Tuning the parameters carefully can have a dramatic impact on the runtime. The initializing of the population near a possible solution also can increase the efficiency of the genetic algorithm. In our algorithm the population is randomly initialized.

Improving the population requires two operations, crossover and mutation. The selection process and stopping criteria of the algorithm are also necessary. The flow of a genetic algorithm is illustrated in Figure 2. First the initial population is created. Then the algorithm iterates until the stopping criteria is met.


finish
output solution
yes
start
Calculate individual fitnesses of population
Are stopping criteria met?
no
Crossover/mutation, generate new population
selection process

Genetic algorithms in this package

Genetic algorithms are useful to find approximations of NP type of problems. In this work, a genetic algorithm and several programs are given to approximate this problem and examine the output. It should be pointed out that these programs work with

- Different residue types, such as Ala, Lys, Phe,…
- Will work with missing or additional measurements that exceed the number of residues of interest. The number of measurements does not have to be the same as that of residues in the trajectory.
- Will search the possible solution space, not just for the global minimum of the fitness. The global minimum may not be the physical solution to the peak assignment. All possible assignments within the user specified errors of the input files should be possible solutions, not just the global best solution.
- Will work any number of media.

Generally, the genetic algorithm is used to search for the global minimum of the fitness function. A problem with the assignment problem is that the global minimum may not be the correct solution.

The search for the correct physical solution uses a population size and different crossover and mutation parameters. The program is accomplished by using a stopping criteria of 500 iterations, and 16 combinations of crossover and mutation rates. (.2, .4, .6, .8). The default is all 16 combinations,

Assign_SLP

The obective function of the peak assignment is the minimization of the rmsd's between the measured rdc's and calculated rdc's given a possible peak assignment, the rmsd's of the proton and nitrogen (or carbon) chemical shifts from the measurements and the predicted, and the 'rmsd' of the noe between measured and calculated. The rmsd's of the rdc's are found from an order parameter improved non-linear back calculation involving all the residues; the calculation is improved by a correction using the order parameters. The noe uses a Pearson correlation matrix of the calculated and measured 1-D noe vectors.

There could be difficulties in the matching of the residues to the peak measurements in that there could be missing measurements or additional measurements if the number of measurements is not the same as the number of residues. The number of experimental and calculated inputs has to be identical for all measurement types. The program handles this in the input files.

Objective function

The components of the objective function are described for each type of measurement. Each possible solution is a permutation of all measurements associated with a specific residue to another residue. For NOEs, an experimental vector and a calculated vector are checked with the Pearson coefficient *Q(I,j).* This is accomplished in Matlab with corrcoeff – see Mathworks for the algorithm, which is well known as a statistical test for comparing two vectors. This coefficient ranges from -1 to 1 as a correlation coefficient. 1 means total correlation, i.e. the vectors are identical, and -1 means total anti-correlation, the vectors are the negation of each other.

The noe contribution to the objective function is

$$noefactor\ x\ \frac{total}{num\_measurements}$$

Noefactor is the weighting for the noe calculation (statistical information content divided by estimated error). Total is the number of possible matching experimental and calculated noe vectors. Total and num_measurements, in all parts of the objective function take into account any missing measurements. For example, if one noe vector is missing from either and there are 16 residues, then total would be 15. Num_measurements is the number of measurements, i.e. the number of residues in the input files. Q(i,j) is the Pearson coefficient between the i'th experimental noe vector and the j'th calculated noe vector. The summation is over all the matched noe vectors.

The rdc contribution to the objective function is very non-linear. It uses a non-linear least square calculation to match the experimental rdc's to those calculated from the mean average trajectory coordinates and order parameters. These calculations are done for each media.

The mean dipolar interaction vector averaged over the trajectory, in terms of the r1 and r2 vector coordinates from an arbitrary origin, are:

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

and the order parameters are

$$< 3cos^2\theta - 1 > /2$$

where

$$cos\theta = \frac{(r1 - r2) \circ < r1 - r2 >}{|r1 - r2|| < r1 - r2 > |}$$

with unit vectors. These quantities are calculated with the order_parameter program in the MD2NOE_Protein package.

Once the rdc's are back calculated, the contribution to the objective function is

$$\frac{total - 5}{num\_measurements} \sqrt{\frac{\sum((backcalculated(i) - measured\_rdc(j))/weighting(j)))^2}{total}}$$

The back calculation can be found in the programs and is not presented here. The back calculation is taken from: Valafar H, Prestegard JH. REDCAT: a residual dipolar coupling analysis tool. Journal of Magnetic Resonance. 2004;167:228-41. There is a minus 5 in the numerator due to the use of 5 tensor parameters in the back calculation. This calculation is done for the different media and added.

The contribution of the chemical shifts to the objective function is,

$$\frac{total}{num\_measurements}\sqrt{\sum\frac{((calculated(i)-measured(j))/weighting(j))^2}{total}}$$

for both the hydrogen and nitrogen, or hydrogen and carbon.

Total and *num_measurements* do not need to be entered by the user; they are calculated form the experimental and predicted input files, which have a number of entries equal to the number of residues or number of measuremetns, whichever is greater. If some measurements or residues are missing then the input files will contain 999's, as explained in the next section. In the next version, additional types of measurements will be included. If errors are estimated properly and information content is one for each data type, the fitness of an acceptable solution should be equal to the number of types of measurements.

Input/Outputs of program

This section describes the input and output of the different programs**.**

The package uses several files as inputs, which can be calculated using MD2NOE_Protein –

- 2 sets of noe vectors in the form of excel sheets. These are measurements and predictions. The predicted noe vectors could be found from the MD2NOE_Protein package from a trajectory. The noe vectors could also be from a 1/r^6 frame calculation.
- Rdc's, measurements and errors
- Average mean coordinates of the N-H or C-H pairs of atoms. Order parameters of the pairs are in this file, which are used to improve the rdc back calculation.
- Chemical shifts, measured and predicted. The trajectory can be used to find the average chemical shifts from the full trajectory.
- If there are missing measurements then 999 has to be used in the input files. -The calculated and experimental inputs have to have the same number-. For example, if there is an additional measurement not accountable by a residue, then a residue would be included with number 999. The number of measurements is identical to the number of residues. If there is a missing noe vector, then the first value in the vector is 999. If there is a missing measurement, such as rdc or chemical shift than 999 is used for the measurement.

The file formats are clear from the example set of input files. The coordinate file has to be created from the output files of the order_parameter program. In the next version, a Matlab program will be added to parse these files and create the coordinate file.

Weights are the reciprocal of:

$$\frac{error}{information\ content}$$

The are described in the document ‘Weighting assignment scores’ and in the paper, “NMR Resonance Assignment Methodology: Characterizing Large Sparsely Labeled Glycoproteins”, currently submitted to the *Journal of Molecualr Biology*. The errors are taken from estimates of experimental errors or estimates of errors in predictions, whichever is greater. The *information* content has to be calculated. Matlab scripts to calculate *information content* included . For the example work in the paper, the errors and information content were as follows.

Errors:
RDCs individual as before
15N shifts 4
1H shifts 0.4
NOE 0.14

Weights:
RDC peg 0.42
RDC pf1 0.37
15N 0.55
1H 0.35
NOE 0.50

The input files would use 4/.55=7.2727 for the nitrogen chemical shift weight, for example, and .5/.14=3.5714 for the noe_factor.

There is a program statistics_z_factors that will generate a random set of peak assignments. The result is used in the z-factor calculations and also in a statistical addition to the OutputAnalysisZFactor program output. The prep file has an input.

The OutputAnalysis program uses the Matlab data file to generate a text file with all the possible solutions with fitness less than a user specified value. For example, if the user specifies this value to be 1.0, then all possible solutions with fitness less than the minimum fitness in the Matlab file + 1.0 are in the text file. The recommendation is to initially use 1.0 + number of measurement types as the cutoff for this output file.

This program also creates a statistical summation of the possible solutions in this text file. Assignments are summarized in a heatmap in which the fraction of times a particular crosspeak is assigned to a particular residue among the solutions with scores below a final cutoff dictates the intensity. This is found by counting all possible i'th measurement to j'th residue assignments in the output text file from OutputAnalysisZFactors and dividing by the total number. A number_residues x number_crosspeaks matrix is created and then used in a heatmap figure. All values in a row or column add to 1.0.

The next program is the validation program. This program is used to give an estimate of the confidence of a particular assignment of a measurement to a residue. Simulated experimental input files are created from predictions by adding random numbers within expected error limits to the predictions. The data are then run through the program to generate a similar heatmap as from the trajectory and experimental information. The fractional cutoff corresponding to a particular confidence limit can be found by examining the number of correct versus incorrect assignments with fractions above the cutoff. For example, a cutoff that gave 9 correct assignments and one incorrect would correspond to a 90% confidence limit. There are scripts available in the download that explain how to generate the validation input files.

Bash scripts

Bash scripts are now described. These are in the download and can be modified. There are bash scripts for the Assign_SLP program, the OutputAnalysisZFactors program, and the StatisticsZFactors program. These bash scripts were used in the paper, ““““NMR Resonance Assignment Methodology: Characterizing Large Sparsely Labeled Glycoproteins”.

The three bash scripts for use in Assign_SLP_MD are described next. All of the inputs are described in detail.

```
run('Initialization.m')
```

- Initializes global variables used in the programs and functions

```
type='N H';
```

- This spin pair is used in the RDC calculations. 'N H' and 'C H' are possibilities.

```
file_of_coordinate='st6_2ndRun_calculations/coordinate_st6.txt';
```

- This is the coordinate file, which contains also the order parameters and residue numbers.

```
file_of_nitrogen='st6_2ndRun_calculations/N15_shifts_exp_pred_increased_50_percent.txt';
```

- The nitrogen chemical shift file. The file has the measurements, then the predictions, then the weighting factors. These are for the amide nitrogens in the sparse residues.

```
file_of_hydrogen='st6_2ndRun_calculations/H1_shifts_exp_pred_increased_50_percent.txt';
```

- The hydrogen chemical shift file. The file also has the measurements, predictions, and weighting factors. These are for the amide protons of the sparse residues.

```
residues=[132 155 157 171 208 216 240 274 275 290 340 356 357 371 390 398];
```

- These are the residue numbers used for labeling. They don’t have to follow the topology file.

```
max_fitness=2.0;
```

- All possiblie solutions less than this fitness are stored in the Matlab data output. This is a weighted score. The raw score would be 2.0/average information factor, which for st6Gal1 is 2.0/.4=5.0. This would fit the ideal case where the cutoff is equal to number_measurements. In the example errors were increased by 50% in the final run to give solutions below this value.

```
population_size=1000;
```

- Population size of the genetic algorithm. It is recommended to use 1000 and if more are used, there may be a problem with convergence of the population.

```
measurement=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16];
```

- Labels of all the measurements. All measurements, noe’s, rdc’s, and chemical shifts have to be in the same order. This will be used in the output files.

```
max_generations=500;
```

- The algorithm for each combination of crossover and mutation rate will stop at 500 iterations.

gen_limit=200;

- If the lowest fitness chromosome does not change within the tolerances after 200 iterations, the algorithm will stop.

Dmax=24350;

- This is the coefficient used in the rdc back calculation. It is different for ‘N H’ and ‘C H’. This is for ‘N H’.

rdc_type='N-H';

- Again, the rdc-type.

noe_sign=-1;

- The calculated noe vectors from MD2NOE_Protein are negative for a large protein. The experimental noe vectors are usually quoted negative, but could be positive for some reason. These vectors (both predicted and experimental) would be positive for small molecules, such as carbohydrates. This factor changes the calculated noe vector sign to agree with experiment.

noe_factor=2.3810;

- This is the weighting factor. As described in ‘Weighting assignment scores’ it is weight/error for the noe vectors.

number_media=2;

- Number of media in the rdc measurements.

validation="false";

- If this is set to true, then the order parameters are not used in the rdc back calculation.

```
number_exp=512;
```

- This is the number of values in the experimental 1-D vectors. This has to agree with the predicted vectors.

```
residues_total=16;
```

- This is the number of sparse residues.

```
penalty=.0;
```

- If constraints are applied in the genetic algorithm, then this is the penalty. It is recommended to use 100 if there are constraints on which residues can be assigned to which measurements.

```
constraint_assignment=ones(16,16);
```

- This is the constraint matrix. It is a residues_total x residues_total matrix with 1's and 0's. 0 means that the assignment of residue to measurement is penalized and will not occur in the population.

```
file_of_rdcs=char('st6_2ndRun_calculations/rdc_exp_peg_increased_50_percent.txt','st6_2ndRun_calculations/rdc_exp_pf1_increased_50_percent.txt ');
```

- These are the rdc input files. If there is only one media, then 'null' should be used as the file.

```
file_of_Pred_noe=char('st6_2ndRun_calculations/1000ns/calculated_spectra_1000ns_2ndRun.xlsx','null');
```

- This is the predicted noe vector excel file.

```
file_of_Exp_noe=char('st6_2ndRun_calculations/noe_point2_exp_plus_auto.xlsx','null');
```

- Experimental noe vector excel file.

```
file_of_peak='st6_2ndRun_calculations/1000ns/1217_errors_increased_50_percent_max_fit_2point0.txt';
```

- This is the output text file after the OutputAnalysisZFactor program.

```
Peak_Assignment_File=sprintf('st6_2ndRun_calculations/1000ns/1217_errors_increased_50_percent_max_fit_2point0');
```

- This is the output file from the OutputAnalysisZFactors program.

```
FileNameMatlab = [Peak_Assignment_File, '.mat'];
```

- This is the Matlab .mat file with all possible solutions from all iterations. It contains duplicates which will be eliminated after the OutputAnalysisZFactor program.

```
run('Assign_SLP')
```

- This runs the Assign_SLP program.

The OutputAnalysZFactor bash script is next,

```
validation="false";
```

- Specifies if order parameters are to be used in the rdc back calculation.

```
verbose="true";
```

- If verbose is ‘false’ then minimal statistical information is included in the output text file. If ‘true’ then information from the stastics_z_factor program is used and the output text file has statistical information for the possible solutions.

```
noe_sign=-1;
```

- This variable is used to make the sign of the experimental noe vector file agree with the predicted noe vector file.

fitness_cutoff=1.869;

- Maximum weighted fitness of possible solutions in the output text file. The raw fitness is the weighted fitness/average information factor

z_statistics_mean_std='st6_2ndRun_calculations/1000ns/cs_point2_1000000_Z_statistics_errors_1point50.mat';

- This file has the output of the statistical_z_factor program.

probability_figure='st6_2ndRun_calculations/1000ns/1217_errors_increased_1point50_max_fit_1point869';

- The Matlab heatmap figure.

run('OutputAnalysisZFactors.m');

- Program

The statistics program bash script is described next. Most of the inputs are obvious from the previous two bash scripts.

run('Initialization.m');

- Initializes global variables.

z_statistics_mean_std='st6_2ndRun_calculations/1000ns/cs_point2_1000000_Z_statistics_errors_1point50.mat';

- Text file which contains the output of the statistics program.

```
Dmax=24350;
```

- RDC coefficient.

```
rdc_type='N-H';
```

- Type of RDC.

```
number_media=2;
```

- Number of media, e.g. peg, phage, …,

```
noe_sign=-1;
```

- Noe vector sign to be used in comparing experiment with calculation.

```
noe_factor=2.3810;
```

- Weighting factor for noe fitness. It is information_factor/error.

```
number_measurements=16;
```

- Number of measurements, i.e. residues.

```
total_population=1000000;
```

- Number of possible solutions in the statistics calculations.

```
file_of_coordinate='st6_2ndRun_calculations/coordinate_st6.txt';
```

- Coordinate file for rdc back calculation.
-

```
file_of_nitrogen='st6_2ndRun_calculations/N15_shifts_exp_pred_increased_50_percent.txt';
```

- Nitrogen chemical shift file.

```
file_of_hydrogen='st6_2ndRun_calculations/H1_shifts_exp_pred_increased_50_percent.txt';
```

- Hydrogen chemical shift file.

```
file_of_rdcs=char('st6_2ndRun_calculations/rdc_exp_peg_increased_50_percent.txt','st6_2ndRun_calculations/rdc_exp_pf1_increased_50_percent.txt ');
```

- RDC files
  .

```
file_of_Pred_noe=char('st6_2ndRun_calculations/1000ns/calculated_spectra_1000ns_2ndRun.xlsx','null');
```

- Predicted noe vectors.

```
file_of_Exp_noe=char('st6_2ndRun_calculations/noe_point2_exp_plus_auto.xlsx','null');
```

- Experimental noe vectors.

```
validation="false";
```

- If false, the order parameters are used. If true, these are not used.

```
number_exp=512;
```

- Number of values in the experimental noe vectors. This has to agree with the predicted noe vectors.

```
residues_total=16;
```

- Number of sparse residues.

```
run('Statistics_z_Factors.m');
```