Catalogue of Scripts for Data Analysis and Structure Determination

Commonly Used Scripts (partial listing):

```
aco2cns.s                    parameter.s
addchid.s                    preparia.s
ambig.s                      seq2tbl.s
ambig_nosegid.s              seq2tbl_H.s
calc_dev_shift.s             seq2tbl_dna.s
calc_ttr_shift.s             seq2tbl_nuc.s
compare_shift.s              seq2tbl_rna.s
conect.s                     suppose2pdb.s
correctshift.s               talos2aco.s
genhelhbdist.s               tbl2bmrb.s
getangle.s                   tbl2csi.s
getangle_chid.s              tbl2nv.s
md2me.s                      tbl2seq.s
md2me_old.s                  tbl2talos.s
meadd.s                      unambig.s
merge2xpkfil.s               waterref.s
```

NMR Data Acquisition, Processing and Referencing

`waterref.s` – calculates the chemical shift of water at any given temperature.

Usage: `waterref.s <temp_in_deg_C>`

Example: `waterref.s 25.3`

Comments: This is useful for spectral referencing.

`parameter.s` – reformats NMR data acquisition parameters for processing, record-keeping, and referencing.

Usage: `parameter.s > <output_file>`

Example: `parameter.s > acqpar`

Comments: The command needs to be given from the same directory as the directory that contains the `procpar` file; usually this directory also contains the `fid`, `log` and `text` files. The `acqpar` file contains the reformatted output.

seq2tbl.s – creates an empty chemical shift table given a file containing a protein sequence in one-letter code.

Usage: `seq2tbl.s <seqfile> > <file.tbl>`

Example: `seq2tbl.s pah2.seq > pah2.tbl`

Comments: The table serves as the input for other scripts and programs. Variants of this script are available to generate empty tables for ^{1}H shifts only (`seq2tbl_H.s`) and tables for nucleic acid sequences (`seq2tbl_nuc.s, seq2tbl_dna.s, seq2tbl_rna.s`).

Related: `tbl2bmrb.s, tbl2csi.s, tbl2nv.s, tbl2seq.s, tbl2talos.s`

compare_shift.s – computes the secondary chemical shifts of C^{α}, C' and H^{α} given a chemical shift table.

Usage: `compare_shift.s <co|ca|ha> <ppm_correction> <shift.tbl> > <output.tbl>`

Example: `compare_shift.s ca 1.945 pah2.csi > pah2_ca_scs.tbl`

Comments: Carbon shifts referenced to TSP or any other standard can be corrected *ad hoc*. Secondary shifts are computed relative to DSS. The secondary shifts of only one nucleus can be computed at a time (i.e. C' or C^{α} or H^{α}). The chemical shift table is in a format recognized by the CSI program. Use the `tbl2csi.s` script first to convert the chemical shift table to the CSI format.

Related: `tbl2csi.s`

correctshift.s – applies appropriate corrections (arising from ^{2}H isotope effects) to ^{13}C chemical shifts measured using fully deuterated samples.

Usage: `correctshift.s <shift.tbl> > corr.tbl`

Example: `correctshift.s pah2.tbl > pah2_corr.tbl`

Comments: Carbon shifts referenced to TSP or any other standard can be corrected. Secondary shifts are computed relative to DSS. The secondary shifts of only one nucleus can be computed at a time (i.e. C' or C^{α} or H^{α}).

tbl2csi.s – converts chemical shift table from standard format to a format recognized by the CSI (Chemical Shift Index) program.

Usage: `tbl2csi.s <shift.tbl> > <output_file>`

Example: `tbl2csi.s pah2.tbl > pah2.csi`

Comments: The input file is the chemical shift table in the format generated by `seq2tbl.s`. The output from this script can serve as input for the `compare_shift.s` script for calculating secondary chemical shifts. Variations of the `tbl2csi.s` script exist to convert to the BMRB-STAR (`tbl2bmrb.s`), NMRVIEW (`tbl2nv.s`), and TALOS (`tbl2talos.s`) formats.

Related: `seq2tbl.s, compare_shift.s`

talos2aco.s – generates backbone phi and psi restraint tables for DYANA.

```
Usage: talos2aco.s <input_file> <phi|psi> <first_residue_number>
<reliability_threshold [1-10]> > <acofile>

Example: talos2aco.s pred.tab phi 295 10 > pah2.aco
```

Comments: The input file is the summary file (`pred.tab`) generated by TALOS. By default, the script will only generate restraints for residues that have been categorized as "Good" with reliability scores set by the user.

Related: `aco2cns.s`

aco2cns.s – converts DYANA format torsion angle restraint tables to CNS format.

```
Usage: aco2cns.s <acofile> <segid> > <output_file>

Example: aco2cns.s pah2.aco PAH2 > pah2_dih.tbl
```

Comments: The input file is the output file generated by `talos2aco.s`. Segid (a.k.a segment identifier) needs to have a maximum of four characters.

Related: `talos2aco.s`

calc_dev_shift.s – computes average chemical shift deviations between two states of a protein for each N-H pair based on ^{1}H and ^{15}N shifts.

```
Usage: calc_dev_shift.s <format_type> <bound_xpk_file> <free_xpk_file> |
sort -n > <output_file>

where format_type:    1 == residue name, number and atom SER_213:HN
                      2 == residue name and number (e.g. S213)
                      3 == residue number (e.g. 213)

Example: calc_dev_shift.s 3 ub_bound.xpk ub_free.xpk | sort -n > ub_dev.dat
```

Comments: The 'xpk' files are those exported by Felix in ASCII format. The spectra need to be referenced so the Felix output is in ppm. The order of peaks in the two files need to be identical. The output file can be read into graphing programs such as XMGR.

calc_ttr_shift.s – calculates the chemical shift deviation as a function of molar ratio of an added ligand to a protein from an NMR titration.

Usage: calc_ttr_shift.s <xpk_file> <ttr_points_file> > <output_file>

Example: calc_ttr_shift.s Lys48.xpk ttr_pts.dat > Lys48.dat

Comment: The 'xpk' file is that generated by Felix in ASCII format. The spectra need to be referenced so the Felix output is in ppm. It contains the coordinates of a particular correlation during the course of a titration. The ttr_points_file should contain the molar ratios sampled during the course of the titration, one per line. The resulting file can be read into programs such as XMGR for non-linear regression analysis.

NMR Structure Determination and Analysis

merge2xpkfils.s – merges two Felix-format xpk files into one.

Usage: merge2xpkfil.s <file1> <file2> <start_item_number_in_file2> > <output_file>

Example: merge2xpkfil.s sf1dbd1.xpk sf1dbd2.xpk 1278 > sf1dbd.xpk

Comments: The merged xpk file can be read back into Felix after creating a new database. The script is useful when peaks in the same spectrum picked by multiple personnel needs to be consolidated.

preparia.s – prepares tbl and ppm restraint files for Aria/CNS given Felix xpk and vol files and the standard chemical shift table.

Usage: preparia.s <control_file>

Example: preparia.s ncsdscnohs.inp

Comments: The control file contains the filenames of the xpk, vol, and chemical shift table files along with the referencing parameters for the spectrum. Every 2D/3D NOESY spectrum requires a separate control file.

unambig.s – produces distance restraint tables for Aria/CNS given Felix xpk and vol files, and volume/intensity calibration values.

Usage: unambig.s <felix_xpk_file> <felix_vol_file> <binfile> <stereosfile> <segid|nosegid> <<segid_D1> <segid_D2>>

Example: unambig.s ncsdssfnoeh.xpk ncsdssfnoeh.vol ncsdssfnoeh.bin stereos segid SF1 DNA

Comments: The xpk file should contain assignments in all dimensions. This is typically accomplished using the Suggest feature in Felix. The binfile contains the cut-off intensity/volume and distance for each bin in a separate line starting from the shortest distance. Stereospecific assignments, if available, contains the name of an atom/group (e.g. MGX for valines) in the chemical shift table and the corresponding atom (e.g. HG1*) in the structure. The scripts ambig.s and ambig_nosegid.s can produces similar files except that ambiguous assignments are handled.

Related: `ambig.s, ambig_nosegid.s`

genhelhbdist.s – generates distance restraints for hydrogen bonded atom pairs in α-helices from input residue ranges for input to Aria/CNS.

Usage: `genhelhbdist.s <residue_ranges> <segid> > hbonds.tbl`

Example: `genhelhbdist.s 253-274,305-326 UIM > uim_hbonds.tbl`

Comments: Multiple ranges, each separated by a comma, can be input using the format in the example above. Segid (a.k.a segment identifier) needs to have a maximum of four characters.

getangle.s – calculates inter-helical angles for specified residue ranges in an ensemble of structures.

Usage: `getangle.s <control_file> > <output_file>`

Example: `getangle.s uim.def > helix_angles.out`

Comments: The control file contains the names of the PDB files one per line followed by residue range for each interacting helix pair separated by a comma (e.g. 253-274,305-326).

Related: `getangle_chid.s`

PDB File Manipulations

addchid.s – adds missing chain identifiers in a PDB file.

Usage: `addchid.s <pdb_file> > <new_pdb_file>`

Comments: Useful for PDB files of binary complexes generated by programs such as CNS that do not have the chain identifier field.

conect.s – extracts C^{α} coordinates from an NMR ensemble and adds CONECT records at the end of the new PDB file linking successive CA atoms.

Usage: `conect.s <pdb_file> > <new_pdb_file>`

Comments: Useful for viewing and producing images of chain traces in programs like GRASP.

suppose2pdb.s – converts PDB files produced by SUPPOSE to the standard PDB format.

Usage: `suppose2pdb.s <pdb_file> > <new_pdb_file>`

Comments: Useful for NMR ensembles that need to be superimposed and brought to the same coordinate frame of reference using programs like SUPPOSE before depositing in the PDB.

CD Data Analysis

md2me.s – converts raw CD data from millidegrees to molar ellipticity.

Usage: md2me.s <data_file> -l <path_length_in_cm> -c <conc_in_uM> -n <number_of_residues> > <output_file>

Example: md2me.s pah2.md -l 0.1 -c 50.0 -n 89 > pah2.me

Comments: Currently handles only data generated by the Jasco spectropolarimeter in the Keck facility. The md2me_old.s script handles the old Jasco format.

Related: md2me_old.s, meadd.s

meadd.s – adds two processed CD data sets acquired under identical conditions for added sensitivity.

Usage: meadd.s <file1> <file2> > <output_file>

Example: meadd.s pah2-1.me pah2-2.me > pah2-1+2.me

Comments: The input to this script is the output from the md2me.s script.