AFNMR Users' Manual
(Version 1.7.0)

Tong Zhu, Xiao He, Jason Swails and David A. Case

04 October 2024

1 Introduction

The AFNMR package provides a useful method to estimate chemical shifts in biomolecules, density functional calculations on fragments. The program automatically creates the fragments from an input pdb file, prepares a model for environmental effects, then creates input files for various quantum chemistry programs. Once these quantum runs are complete, results can be extracted from their output files and collected into simple database files.

The basic literature references are:

1. X. He, B. Wang, and K.M. Merz, Jr. Protein NMR Chemical Shift Calculations Based on the Automated Fragmentation QM/MM Approach. *J. Phys. Chem. B* **113,** 10380-10388 (2009)

2. T. Zhu, X. He, and J.Z.H. Zhang. Fragment density functional theory calculation of NMR chemical shifts for proteins with implicit solvation. *Phys. Chem. Chem. Phys.* **14,** 7837-7845 (2012)

3. T. Zhu, J.Z.H. Zhang, and X. He. Automated Fragmentation QM/MM Calculation of Amide Proton Chemical Shifts in Proteins with Explicit Solvent Model. *J. Chem. Theory Comput.* **9,** 2104-2114 (2013)

4. S. Tang and D.A. Case. Calculation of chemical shift anisotropy in proteins. *J. Biomol. NMR* **51,** 303-312 (2011).

5. D.A. Case. Chemical shifts in biomolecules. *Curr. Opin. Struct. Biol.* **23,** 172-176 (2013).

6. J. Swails, T. Zhu, X. He and David A. Case. AFNMR: Automated fragmentation quantum mechanical calculation of NMR chemical shifts for biomolecules. *J. Biomol. NMR* **63,** 125-139 (2015).

7. H. Zhang, G. Hou, M. Lu, J. Ahn, I.-J. Byeon, C.J. Langmead, J.R. Perilla, I. Hung, P.L. Gor'kov, Z. Gan, W. Brey, D.A. Case, K. Schulten, A.M. Gronenborn, and T. Polenova. HIV-1 Capsid Function is Regulated by Dynamics: Quantitative Atomic-Resolution Insights by Integrating Magic-Angle-Spinning NMR, QM/MM, and MD. *J. Am. Chem. Soc.* **138,** 14066-14075 (2016).

8. X. Jin, T. Zhu, J.Z.H. Zhang and X. He. A systematic study on RNA NMR chemical shift calculation based on the automated fragmentation QM/MM approach. *RSC Adv.* **6,** 108590-108602 (2016).

9. X. Jin, T. Zhu, J.Z.H. Zhang and X. He. Automated Fragmentation QM/MM Calculation of NMR Chemical Shifts for Protein-Ligand Complexes. *Front. Chem.* **6,** 150 (2018).

10. H. Shi, M.C. Clay, A. Rangadurai, B. Sathyamoorthy, D.A. Case, and H.M. Al-Hashimi. Atomic Structures of Excited State A-T Hoogsteen Base Pairs in Duplex DNA by Combining NMR Relaxation Dispersion, Mutagenesis, and Chemical Shift Calculations. *J. Biomol. NMR* **70,** 229-244 (2018).

11. M. Fritz, C.M. Quinn, M. Wang, G. Hou, X. Lu, L.M.I. Koharudin, J. Struppe, D.A. Case, T. Polenova and A.M. Gronenborn. Accurate determination of backbone chemical shift tensors in microcrystalline proteins by integrated MAS NMR and QM/MM. *Phys. Chem. Chem. Phys.* **20,** 9543-9553 (2018)

12. H. Shi, A. Rangadurai, H.A. Assi, R. Roy, D.A. Case, D. Herschlag, J.D. Yesselman, and H.M. Al-Hashimi. Rapid and accurate determination of atomistic RNA dynamic ensemble models using NMR and structure prediction. *Nature Commun.* **11,** 5531 (2020).

13. D.A. Case. Using quantum chemistry to estimate chemical shifts in biomolecules. *Biophys. Chem.* **267,** 106476 (2020).

Ab initio chemical shifts can be computed using the automated fragment approach implemented in the AFNMR program and described in references [1], [2] and [6]. The explicit solvent formalism described in reference [4] is planned for future versions, but is not included in the present version; however, the current code works well for snapshots taken from MD simulations that include explicit water molecules. Papers [7-12] are recent examples of applications of the afnmr method. Reference [13] is a review.

2 Installation

1. To begin, clone this repository to some directory on your local machine:

```
git clone https://github.com/dacase/afnmr.git
```

2. The path to this new directory should be defined as the environment variable $AFNMRHOME.

```
setenv AFNMRHOME “insert-your-path-here/afnmr” # csh or tcsh
export AFNMRHOME=”insert-your-path-here/afnmr” # sh, bash, ksh, or zsh
```

3. Next, build and install the program , putting executable files into `$AFNMRHOME/bin`:

```
cd $AFNMRHOME
make install
```

   You will probably want to add `$AFNMRHOME/bin to your $PATH.`

4. You can now run some test calculations and report results:

```
make test
```

3 AFNMR

This is a program for carrying out automated fragment NMR (AFNMR) chemical shift prediction. In this approach, the chemical shielding tensors for ^{1}H, ^{13}C, ^{15}N, and ^{31}P are computed using an approached based on density functional theory. Because chemical shielding is a local property, the shielding tensors can be computed by including only a relatively small region around the nuclei of interest. AFNMR automates the procedure of breaking the input structure into fragments small enough to be treated with a high-level quantum mechanical theory (e.g., various density functionals). The shielding tensors are then computed using one of the supported QM packages, *deMon, Orca, Q-Chem,* or *Gaussian*. Users are directed to the

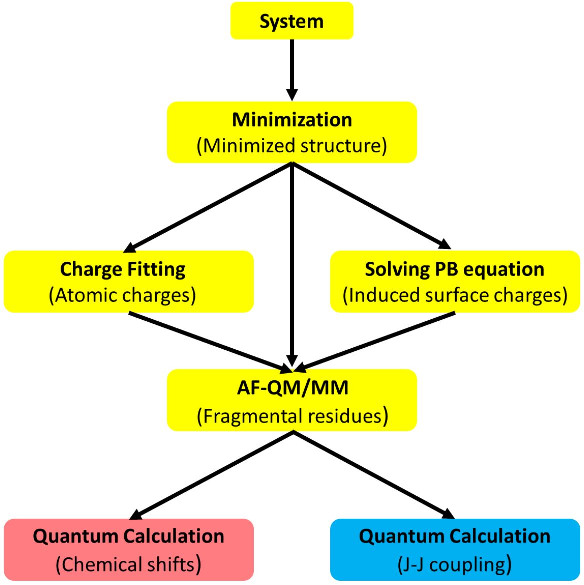

Figure 1: *Workflow followed by AF-NMR to compute chemical shifts*

relevant citations at the beginning of this manual for details regarding the fragmentation scheme for various systems.

Solvent contributions to shielding tensors can be included by modeling the solvent implicitly (e.g., with the Poisson-Boltzmann or 3D-RISM formalisms). A procedure for including nearby water molecules explicitly in the QM calculation has been proposed to improve amide chemical shift predictions and is planned for future versions of AF-NMR.

3.1 General workflow

The first thing that AF-NMR does to the system is perform a local energy minimization using the Amber ff12SB force field. After that, the electrostatic potential (ESP) at the surface of the system is computed and surface point charges are fitted to reproduce this potential. AF-NMR then proceeds to break the system up into manageable fragments and writes input files for the requested QM package. It is then up to the user to run the QM calculations. While these calculations may take several hours to finish, each fragment can be done at the same time. The general workflow for AF-NMR is shown in Figure 1 on page 3.

AF-NMR provides three ways to compute surface charges to model solvent effects. Only the second option is implemented in this release; see the *shifts* program (at `http://casegroup.rutgers.edu/shifts.html`) for information about options 1 and 3.

1. The first uses the PBSA program included with AmberTools to compute the ESP at the surface of the system, setting the dielectric constant of the system interior to 1 and that of the solvent to 80. These surface charges are then used for every fragment while the atoms outside the fragment region are treated as point charges whose values are equal to the partial atomic charges assigned by the Amber force field. The PBSA program from AmberTools must be installed to use this approach.

2. The second approach also solves the PB equation, but instead uses a 3-dielectric model. In this case, the QM region is assigned a dielectric constant of 1, the interior of the system that is *outside* the QM region is assigned a dielectric constant of 4, and the region outside the system is assigned a dielectric constant of 80. With this approach, each fragment is assigned a separate set of surface charges at the surface of the QM region (as defined by the atom radii assigned by the force field). This approach helps to alleviate the overpolarization that occurs from the point charges of the atoms outside the QM region. However, it takes longer to compute the surface charges since a separate solution to the PB equation is required for each fragment; (however, this is still nowhere near the cost of the QM

calculations themselves). This approach uses the *solinprot* program that is part of the *MEAD* package (see `http://stjuderesearch.org/site/lab/bashford`), and is included here as well. This is the default option, and the one that has been most extensively tested.

3. The final approach supported by AF-NMR to compute surface charges uses the 3D-RISM formalism to compute the electrostatic potential. This approach uses a rigorous statistical mechanical approach to compute solvent distributions around solutes with arbitrary charge distributions. Like the first approach we described, it is applied to the entire system and a single set of point charges at the surface of the whole system are used to reproduce the ESP for every fragment.

3.2 Using AFNMR

A shell script, *afnmr*, is provided as the user interface to apply the AF-NMR method to a whole system. It takes a set of flags that can be used to control program flow and specify user-controlled options. If you run *afnmr* without arguments or with the `--help` flag, you receive a brief help and usage statement.

```
Usage: afnmr [flags] basename
```

The file "basename.pdb" must be present. Currently, all protein or nucleic acid residue must be placed first, followed by "general" residues, such as ligands, water molecules, ions, etc. Note that the `basename` must be the last argument. The commonly used flags are described below in more detail.

- *get help:*

  -help

- *residues to analyze:*

  -list list of residues to create fragments for; *List* uses bash syntax and must be quoted, e.g. "{1,2,{4..9},13}", "46", "{5..8}" (Note: unlike earlier versions of *afnmr*, the residue numbers are those in the pdb file, and need not start from 1, nor do they need to be sequential. Residue numbers, however, must be unique: *afnmr* does not use chainID's.) If *-list* is not present, fragments will be made for residues 1 up to the last protein or nucleic acid residue. If the list argument is "0", no fragmentation will be done, and the entire molecule will be treated at the quantum level.

- *basis set and functional selection:*

  -mixedb Uses a mixed basis set in which the central residue whose shifts are being computed are treated with a larger, (*pcS-seg*2) basis set while the surrounding QM residues outside the primary region use a smaller (*pcSseg-1*) basis.

  -seg2 Use *pcS-seg2* for all residues

  -seg1 Use *pcS-seg1* for all residues

  -seg0 Use a smaller basis (*pcS-seg0* by default) for all residues; this is the default if no basis flag is present.

  -o3lyp Use o3lyp in place of olyp; requires -gau, -qchem or -orca (below)

  Note: If you want to use a different DFT functional, or different basis sets, you can edit the input scripts that *afnmr* provides. The question of the "best" way to carry out shift calculations is still an active area of research. Also, the deMon program does not come with the pcSseg-# basis functions installed by default. We have provided a `BASIS` file (in the `basis` folder) that can replace the file of the same name in your deMon installation, and which adds these basis sets. For Gaussian, we have included `.gbs` files (from the Basis Set Exchange web site) for these basis functions, and they are written directly into the Gaussian input files.

- *method to represent atoms outside the fragment:*

  -solinprot Compute the surface charges using the 3-dielectric model implemented by the *solinprot* program from the MEAD package. (default behavior)

-pbsa Compute the surface charges using the PBSA program from AmberTools *(not implemented in this version)*

-rism, -3drism use 3D-RISM to get surface charges *(not implemented in this version)*

- *quantum program to use:*

  -deMon, -demon Set up QM input files to run with the deMon program, version 3

  -demon5 Set up QM input files to run with the deMon program, version 5

  -orca Set up QM input files to run with the Orca program, version 5 or 6 *(default)*

  -gau Set up QM input files to run with the Gaussian program

  -jaguar Set up QM input files to run with the jaguar program from Schrodinger.

  -qchem Set up QM input files to run with the Q-Chem program

- *geometry optimization:*

  -nomin Do not minimize the input structure.

  -xtb Carry out a geometry optimization of each fragment using the *xtb* semi-empirical code; then create input files for one of the other programs (deMon, ORCA, Gaussian, Qchem) to actually compute the shifts. The *xtb* code must be in your PATH.

  -tc Same as the `-xtb` option, but using terachem as the optimizer. (*experimental, probably not working*)

  -qopt do a quantum mechanical geometry optimization in place of a molecular mechanics optimization. (*Work in progress: only fully implemented in Gaussian, Orca and deMon. The number of steps and method of minimization is hard-coded in afnmr.F90.*)

- *miscellaneous flags:*

  -workdir AF-NMR creates many temporary files. By default, they are all created inside the current directory. If this flag is specified, a new directory named `<basename>` will be created and all temporary files will be put there. This is useful, for instance, when running multiple AF-NMR calculations in the same directory.

  -nobuild Skip all steps requiring the Amber force field; assumes that a *basename.pqr* file is present

  -frcmod <file> An additional frcmod file to load into *tleap*

  -offlib <file> An additional unit library file to load into *tleap*

multiafnmr.sh Several structures in the protein data bank (PDB) were solved by refining structures with restraints imposed by NMR experiments. Many of these entries in the PDB contain several conformations. The *multiafnmr.sh* script will split apart the PDB file and run *afnmr* on each of the structures independently. The PDB basename must be the first argument to *multiafnmr.sh*, and the rest of the arguments are the command-line options described above for *afnmr*. The -workdir argument is used by default so that each conformation gets processed in a separate directory.

The processing of structures is parallelized, and setting the environment variable `NCPUS` to an integer will inform *multiafnmr.sh* to process the given number of structures simultaneously. This is primarily useful when using the -solinprot option, which takes longer to compute surface charges with.

3.3 Computing the shielding tensors

After running *afnmr* (or *multiafnmr.sh*), you will have a handful of input files to the QM program you requested—one for each of the fragments (residues) in your system. You can either run the QM program directly at the command-line or you can run the command in a cluster using some type of batch scheduling system like torque, slurm, or Sun grid engine.

We suggest that you pick the QM program you are most familiar with, which will help in debugging problems should they arise. If you've chosen *Gaussian*, the input files will be named `<basename>XXX.com`, where `XXX` ranges from `001` to the total number of fragments. If you've chosen *Orca*, the file name suffix will be `.inp` and a number of other files ending with `.pos` will be written to define the positions of the external point and surface charges. Input files for *deMon* also end with `.inp`, whereas those for *Q-Chem* end with `.in`. You may wish to modify the control headers of the input file to change the amount of requested memory or number of CPUs based on your available hardware.

Sample commands to run with each of the programs is shown below for the first fragment of the `1d3z1` system found in the `test/` directory of the *afnmr* package.

```
g09 < 1d3z1001.com > 1d3z1001.log     # For Gaussian calculations, g03 or g16 also wor
orca 1d3z1001.orcainp > 1d3z1001.out  # For Orca calculations
qchem 1d3z1001.in 1d3z1001.qcout      # For Q-Chem calculations
deMon 1d3z1001.inp                    # For deMon calculations
```

3.4 Extracting chemical shifts

Following the QM calculations on all of the fragment input files, you will have a large number of output files whose chemical shielding tensors need to be extracted. Because we are modeling systems in solution, the molecules are free to tumble in every direction. As a result, we are only interested in the isotropic shielding at each nucleus. If only the shielding tensor is given in the output file, the isotropic shielding can be computed from the trace of the tensor. A reference shift (which is basis set-dependent) need then be subtracted to get the predicted chemical shift. Also, care must be taken to make sure that you extract only those chemical shifts from the nuclei primary QM region and not those in the immediate surroundings—those shifts are computed more rigorously in a different output file.

Included in the SHIFTS package are a set of parsers written in Perl that will extract the relevant chemical shifts from each output file for *deMon*, *Jaguar, ORCA*, *Q-Chem* or *Gaussian* calculations. These parsers use the PQR files that were generated by AF-NMR to determine which nuclei reside in the primary QM region to determine which chemical shifts to extract from each output file. The script responsible for parsing the QM output file is *make-shiftsrdb*.

make-shiftsrdb will try to parse all `<basename>XXX.out` files where `XXX` are any characters (typically numbers from 001 to the number of residues). If you used *Gaussian*, the output file suffix `.log` will be used instead. The output file will be printed in a simple tab-delimited text file that should be self-explanatory; it is also compatible with the RDB relational database format (see the description at `http://compbio.soe.ucsc.edu/rdb/`). Here is an example:

```
# results from afnmr-demon3:
# reference shifts used for H,C,N:  32.00 182.50 242.00 OLYP TZVP 1d3z1
res       atomname  resname   1C44
8N        8         8         10N
1         CA        SER          57.41
1         CB        SER          68.32
1         H1        SER           5.49
1         H2        SER           4.67
1         H3        SER           4.58
1         HA        SER           4.11
1         HB2       SER           4.20
1         HB3       SER           4.59
1         HG        SER           2.87
1         N         SER          27.88
```

Here is more detailed usage information (from `$AFNMRHOME/src/afnmr/make-shiftsrdb`):

```
Usage: make-shiftsrdb qpgm reference <basename> fulltensor
* qpgm must be one of "-demon", "-gau", "-qchem", "-jaguar" or "-orca"
* reference is the name of a reference shielding set The reference sets are defined
     in the getshifts-demon (etc.) scripts, and sometimes change as better sets
     become available. Users can create their own versions.
```

```
* quantum outputs are assumed to be in <basename>nnn.{out,log,qcout} files,
     where "nnn" is a three-digit resiude number (such as "001")
* fulltensor is 1 to get expanded CSA printouts, 0 otherwise
Output goes to <basename>.rdb
```

3.5 Handling ligands or non-standard residues

The *afnmr* code should be able to handle file that follow PDB standards that include proteins, RNA, DNA, water, and monatomic ions. If you have ligands, or non-standard amino acids or nucleotides, the procedure is more involved. Everything is based on the way that Amber handles these situations, so people familiar with Amber should understand what is being said below. (I am hoping to write instructions for more general users in the future.)

3.5.1 Non-covalent ligands

If you have ligands that are not covalently bonded to the macromolecule, use the *antechamber* and *parmchk2* programs in AmberTools to create *off* and *frcmod* files for your ligand.

Follow this tutorial: https://ambermd.org/tutorials/basic/tutorial4b/index.php, but use the *saveOff* command in `tleap` to save an off file. Then use the `-offlib` and `-frcmod` flags in afnmr to get these loaded. You can load files for more than one type of ligand, if needed. One current limitation: you need to put all ligand molecules after all protein or nucleic acid residues. (I plan to eliminate this requirement in future updates.) By default, *afnmr* will only compute shifts up to the last protein or nucleic acid residue, but you can get predicted shifts for ligands by including their residue numbers in the `-list` option.

3.5.2 Modified amino acids or nucleotides

These are currently a bit more compilcated. You can create the mol2 and frcmod files following this tutorial: https://ambermd.org/antechamber/pro4.html. You then use the `-mol2` and `-frcmod` flags to load these into afnmr, as with ligands. But then, you also need to edit the *afnmr.F90* file (in src/afnmr) to add the names of these residues to the `presn` array (for proteins) or the `nresn` array (for RNA and DNA). You will need to update the MAXPRES or MAXNRES variables if you add extra residue names to that list. Again, I am hoping to simplify and automate this process in future versions of the code.