NMR calculations in ORCA

The most updated version of this handout can always be found at:

https://github.com/duartegroup/sbmcc/blob/master/nmr/sbmnmr.pdf

In this project, we will look at the three candidate structures for cordycepol A, proposed by Reddy and Kutateladze in 2016.[1] You should calculate chemical shifts, and key coupling constants, for all three candidates. From these, you can determine which is most likely to be the correct structure by comparison against the experimental shifts.

Warning

We **strongly** recommend that you create a separate directory for each of the following sections, or it can get very messy!

1 Conformer generation

NMR properties, such as chemical shifts and coupling constants, are calculated as averages over multiple low-energy conformers. In order to calculate the NMR properties of a molecule, the first step is to generate a sufficiently large ensemble of conformers. This is most commonly done with methods based on molecular dynamics; here we will utilise the *Conformer–Rotamer Ensemble Sampling Tool* (CREST), a programme developed by Stefan Grimme.[2,3] First, download and install the CREST programme using:

```
source <(wget -O - https://raw.github.com/duartegroup/sbmcc/master/nmr/getcrest.sh)
```

To generate conformers for a molecule, first create an appropriate `.xyz` file (in Avogadro: use **File** > **Save As...** after you finish drawing the molecule). Upload it onto the cluster and submit the CREST job using

```
qcrest <molecule>.xyz
```

When the job is complete, you can access the output at `<molecule>.out`. The bottom of this file will tell you the number of conformers found, as well as their energies (in kcal/mol). The conformer geometries can be found in `crest_conformers.xyz`. The other `.xyz` files are just collections of unoptimised or unfiltered conformers, and can be safely ignored.

2 Geometry optimisation

The conformers that were previously found are optimised using the semi-empirical model GFN2-xTB.[2] To obtain more accurate geometries, it is typical to perform a second optimisation using a higher level of theory, and we will use ORCA for this.

We need to create an input file for each conformer generated: this means that `crest_conformers.xyz` needs to be split up into many files. For a small number of conformers, this can be done by hand, but for a larger number it is necessary to write a script to automate the task, such as the one which we will now use. Copy `crest_conformers.xyz` to a different folder (bearing in mind our earlier warning), `cd` to that folder, and run

```
crest2opt
```

The script will prompt you to enter the keyword line for the ORCA optimisation. This will allow you to choose the level of theory. We suggest using

```
! PBEh-3c CPCM(Chloroform) Opt
```

Here, `PBEh-3c` is a reasonably low-cost method which provides decent geometries for organic molecules. The original NMR data[4] were acquired in $CDCl_3$, and the `CPCM(Chloroform)` keyword requests an implicit solvation

model to account for that. Feel free to choose a different level of theory, but do not be over-ambitious, especially if you find that you have many conformers to optimise! In particular, it's not usually worth using anything more than a double-zeta basis set.

You can submit *all* the input files at once using `qorca *.inp`. The asterisk acts as a wildcard. When your optimisations are complete (check using `squeue -u <username>`), move on to the next section.

3 Single-point energies

The *geometries* obtained with `PBEh-3c` (or whatever you used) are probably sufficiently accurate, but the *energies* can (and should) be re-calculated using an even higher level of theory, since single-point calculations are cheap. To do this, copy all the final geometries to yet another new folder. You can do this from the command line (after making the directory) with

```
cp *.opt.xyz <target_directory>
```

Searching for `*.opt.xyz`, and not `*.xyz`, conveniently ignores all the `.opt_trj.xyz` files which ORCA generates as part of the optimisation output (these are *trajectory* files which illustrate how the optimisation proceeds, but we are not particularly interested in them here).

As before, we have a helper script which converts the optimised geometries into ORCA input files. `cd` to this new directory which contains all the `.opt.xyz` files and run

```
opt2sp
```

You can choose a suitable level of theory here: just make sure that everybody in the group uses the same choice, or the results will not be comparable. A sensible choice would be a hybrid functional combined with a triple-zeta basis set. Try to include dispersion corrections (using the keyword `D3`, only available with some functionals: see `https://www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3/functionalsbj`) and solvation (`CPCM(Chloroform)`, as before).

Once these are all done, run

```
grep "FINAL SINGLE POINT ENERGY" *.out
```

This will print all the calculated energies (in hartrees). You should now perform some kind of filtering to remove high-energy conformers which do not contribute significantly to the Boltzmann-averaged properties (but as always, make sure this is done in a consistent manner). Ideally, you should be left with a number of conformers for which the remaining steps can be reasonably done "by hand". As a reminder, the population p_i of conformer i with energy ε_i is given by:

$$p_i = \frac{\exp(-\varepsilon_i / k_\mathrm{B}T)}{\sum_j \exp(-\varepsilon_j / k_\mathrm{B}T)}$$

where the sum in the denominator (the partition function) runs over all conformers, including conformer i itself.

4 Chemical shifts

Copy the `.opt.xyz` files which survived the previous filtering step to yet another new directory. You should now manually construct the ORCA input file for calculations of chemical shifts using the following example (see also §8.15.6 of the ORCA 4.2.1 manual):

sample_shifts.inp

```
! FUNC BASIS AUXBASIS D3 CPCM(Chloroform)

%pal nprocs 16 end

*xyz 0 1
... (coordinates go here) ...
*

%eprnmr
  Ori GIAO
  Nuclei = all C { shift, ist = 13 };
  Nuclei = all H { shift, ist = 1 };
end
```

`GIAO` stands for *gauge-independent atomic orbitals* (or *including* or *invariant*; chemists can't seem to make up their minds). This is the most widely-used method for calculating NMR chemical shifts (more information on this can be found in the recommended books). You should be comfortable with `FUNC` and `BASIS`, but `AUXBASIS` is probably new to you. This refers to an auxiliary basis set, which is used by ORCA when approximating certain integrals. For now, it suffices to follow this table when choosing `AUXBASIS`:

`BASIS`	Appropriate `AUXBASIS`
`def2` family	`def2/JK`
`cc-pVnZ` family	`cc-pVnZ/JK`
Anything else	`AutoAux`

When the calculation finishes, there will be a section at the bottom of the output file containing a `CHEMICAL SHIELDING SUMMARY`. The *isotropic* shielding is the number that we are after (because molecules in solution undergo rapid tumbling and experience all possible orientations). You need to perform a Boltzmann average:

$$\sigma_{\text{avg}} = \sum_i \sigma_i p_i = \frac{\sum_i \sigma_i \exp\left(-\varepsilon_i / k_{\text{B}} T\right)}{\sum_i \exp\left(-\varepsilon_i / k_{\text{B}} T\right)}$$

and that's it... or is it? In fact, you will find that the *chemical shielding* is markedly different from the *chemical shift*, the parameter conventionally reported in journals. To convert the shieldings to shifts, we need a benchmark molecule, and a convenient molecule to use is tetramethylsilane, since its chemical shifts (both proton and carbon) should be identically zero.

So, you will need to optimise the geometry of TMS and calculate its chemical shieldings. Make sure to use the same levels of theory as before. Since there should only be one conformer, there's no need to perform conformer generation or the single-points. After averaging over all the symmetry-equivalent nuclei (four carbons or twelve protons), you should obtain the relevant shielding, which we denote σ_{TMS}. Then, for the original molecule, we have:

$$\delta = \sigma_{\text{TMS}} - \sigma$$

where δ is the chemical shift in units of ppm (notice that ORCA reports shieldings in units of ppm, so no other conversion is required).

5 Coupling constants

To calculate the coupling constants, the general format for the input file is:

sample_couplings.inp

```
! FUNC BASIS D3 CPCM(Chloroform)

%pal nprocs 16 end

*xyz 0 1
... (coordinates go here) ...
*

%eprnmr
  Nuclei = all H { ssfc, ist = 1 };
  SpinSpinRThresh 3.6
end
```

First, note that `AUXBASIS` and `Ori GIAO` are no longer needed. Another difference is that the keyword `shift` has been changed to `ssfc`: this stands for *spin–spin* [coupling constant], *Fermi contact* term. In general there are four terms that contribute to the coupling constant, but the Fermi contact term is by far the most important, at least for proton–proton couplings. Calculating only this term saves a lot of time.

There is a new addition inside the `%eprnmr` block, namely the `SpinSpinRThresh` keyword. This instructs ORCA to only calculate couplings between atoms that are within 3.6 Å; this is a safe value to use if we are only interested in ^2J and ^3J.

Unfortunately, in ORCA, there is no way to specify *exactly* which pairs of atoms we want to calculate the couplings for. Instead, the syntax of the `%eprnmr` block specifies a *list* of atoms for which we are interested; ORCA then calculates all couplings between each of those atoms and nearby atoms within `SpinSpinRThresh`.

In the example above, this list of atoms that we are interested in simply consists of `all H` atoms. There is no point including `all C` as well for C–H couplings, because as long as a carbon is within `SpinSpinRThresh` of a proton, the corresponding coupling will be calculated.

If we want to pare down this list further (e.g. restrict it to only certain protons of interest), then we just need to replace `all H` with a comma-separated list of atom numbers. For example,

```
%eprnmr
  Nuclei = 2, 3, 4 { ssfc, ist = 1 }
  SpinSpinRThresh 3.6
end
```

will calculate all couplings between proton number 2 and neighbours within 3.6 Å of it; then the same for protons 3 and 4.

Warning

Within the `%eprnmr` block, atom labels are counted **beginning from 1, not 0**.

So, the atom number that you use here will correspond **exactly** to those that are shown in Avogadro. Or in other words, they will be one greater than usual, since the usual atom numbers are those in Avogadro minus one.

Confusingly, in the *output*, the atom numbering returns to "normal" and is counted from 0 again. As with the previous section, the quantity we are looking for is the isotropic coupling, labelled with `iso`. These are in units of Hz and do not require any further scaling, so are ready to use right away.

References

1. Reddy, D. S.; Kutateladze, A. G. *Org. Lett.* **2016**, *18*, 4860–4863.
2. Bannwarth, C.; Ehlert, S.; Grimme, S. *J. Chem. Theory Comput.* **2019**, *15*, 1652–1671.
3. Grimme, S. *J. Chem. Theory Comput.* **2019**, *15*, 2847–2862.
4. Sun, Y.; Zhao, Z.; Feng, Q.; Xu, Q.; Lü, L.; Liu, J.-K.; Zhang, L.; Wu, B.; Li, Y.-Q. *Helv. Chim. Acta* **2013**, *96*, 76–84.