

NMR Pro Quant 2024

Fitting

Fitting the standard (colloquially known as "red") to a sample ("blue") is done at many stages: getting and using a gradient. Fitting is a 3-step process:

1. Find the position of best fit of red (pro std) on blue (a sample). The entire fitting process is designed to fail if the score of the best fit (measured by *normalized cross-correlation score*, or `normxcorr`) is still too low (less than 0.8).
2. Get Pearson correlation of certain manually pre-selected regions of fitted red (from 1) to blue. This is to mitigate the presence of contaminants that persistently appear at known regions.
3. Use OLS to compute a multiplier that scales red, and additive constant that shifts red up or down.

Calculating Gradient (Training Step)

- Training data: 33 AF IDs with to bootstrap a gradient with. Final output is a distro of gradients.
- We don't use a traditional cal curve, instead we use the difference in area between a spiked sample and the unspiked version of the same sample. Spiked amount is 1115 mg/L. There are 33 such samples.
- Use bootstrapping on AWS to generate a distribution of coefficient x , where $\text{conc} = x * \text{auc}$, and finally compute `np.average(x)` (bootstrapping = resampling a bootstrap sample, $n = 33$, with replacement. Do this 5000x to get a distribution of x) This need only be done once, or very infrequently.
- Error: $\text{grad} * (\text{AUC}(\text{spiked}) - \text{AUC}(\text{original})) - 1115$ should be 0

Testing Step

For each test sample:

1. Shape matching: `normxcorr`-based template-matching,
2. Based on manually-pre-specified regions of high or low reliability, do linear regression to calculate scaling factor and additive constant to scale template (red) to signal (blue). Calculate the area of red.
3. Predict the conc of red based on AUC of red from the previous step.

Notes

- Notably missing: various imported .py files, in particular `nmr_fitting_funcs.py` and `nmr_targeted_utils.py`. Get these from the `/bin-of-tools` repo for DRY.
- The `/manual_integration...` dir is a fairly self-contained folder containing everything you need for manual integration for when auto-integration fails, but you'll have to regenerate `blue_m1_dict.pkl` everytime as input data (that is, dict of M1 multiplets where the ref peak at 0 has been adjusted to be at 0ppm). The manual integration workbook also needs the latest working versions of `nmr_fitting_funcs.py` and `nmr_targeted_utils.py`.
- `bootstrap_results.csv` as at 12Sep2023: `s3://awri-ma-archives/bootstrap_results_12sep2023.csv`
- an archival folder of old code, `/archive_old_workflows_pre2024`, is not backed up on GH.

