COmpressed SEnsing Multiplicative denoising for Magnetic Resonance Spectroscopy and Imaging – Tutorial

# Introduction

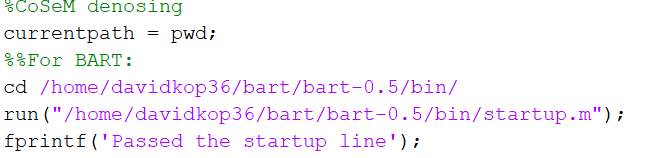
CoSeM is a denoising algorithm that can be used to attenuate multiplicative noise (such as t1-noise, or other random instability-related distortions) in multidimensional MR datasets. The denoising procedure is based on a combination of sub-sampling the dataset in a sparse domain followed by a regularized compressed-sensing (CS) reconstruction; the resulting data is referred to as a “rendition”. CoSeM operates by a selective averaging of many such renditions, chosen in accordance with reducing their “badness” as guided by norm- or entropy-based criteria.

# Prerequisites

* Matlab - tested on version 2019a
* BART toolbox v0.5.00 or newer (can be downloaded [here](https://mrirecon.github.io/bart/))

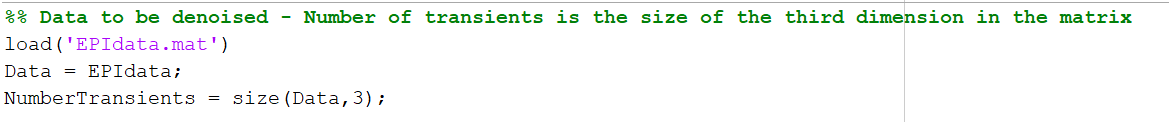
# Operation

Run the startup of BART (located in the base directory) to ensure the toolbox is operating. Both this step and the CoSeM framework can be accessed from CoSeM\_main\_file.m, here the options for the reconstruction can be set:



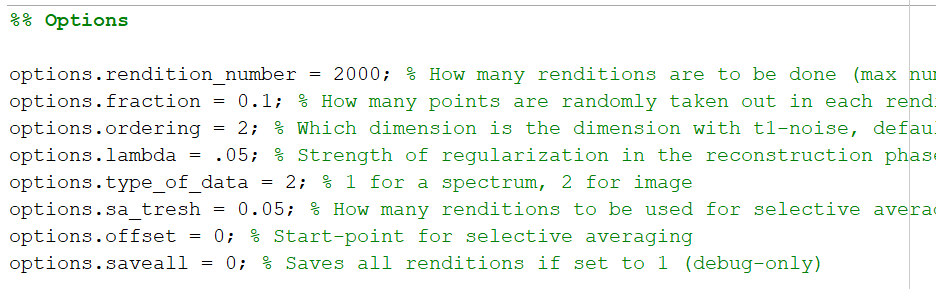
# Input

Function CoSeM\_denoising\_run.m is called with an input that should be assigned to the variable *spektar\_radni* along with a list of pre-specified control parameters via the variable *options*. The input data should be a complex matrix in the image or the spectral domains, with it being a 2D or 3D matrix. It shall be noted that any 3D dimensional denoising is performed as a pseudo-2D. For instance:



# Denoising parameters

The denoising procedure is controlled by a set of parameters that can be tweaked to optimize the outcome, and are present in the structure options:



A quick guide on how to set up these parameters follows below:

* options.rendition\_number – The number of renditions to be conducted. The larger this number is, the better the performance. It determines the run-time of the denoising and memory needed to be allocated. Should be **at least a 1000** for typical 2D NMR datasets.
* options.fraction – Determines the balance between attenuation of the multiplicative noise and the reconstruction error. Should be kept at **around 0.1** for most purposes; it can be larger if there is a large number of transients significantly affected by instabilities.
* options.ordering – Along which dimension is the denoising conducted. The default option is that the noisy *t1* dimension is the **second dimension** of the matrix.
* options.lambda – Regularization strength control parameter. Higher values yield higher denoising, but the value shall not be too high to avoid oversmoothing and/or overegularizing. **Typically in the 0.05 - 10 range.**
* options.type\_of\_data – Set to **1** if the input data is a **spectrum** and to **2** if the input is an **image**.
* options.sa\_tresh – Fraction of renditions to be used for selective averaging. If the rendition number is around 1000 a good choice is **0.05**. Should be decreased if the rendition number is increased (e.g. 0.01 for 10k).
* options.offset – Offset for averaging. Default option is zero and should be kept at **zero** for most cases. It can be set to a higher value if singular behavior of selective averaging is observed.
* options.saveall – Setting this to 1 saves all the renditions individually, as well. Requires allocation of few GB of disk space, should be only done for debugging purposes.

# Output

The output of this function has both a structured variable with the options used in the denoising process, as described aboved and a structured variable of results (entitled FinalResults), which contains:

* FinalResults.selective\_avg - Denoised and selectively averaged via the norm-ratio criterion
* FinalResults.selective\_avg\_ent – Denoised and selectively averaged via the Shannon entropy criterion
* FinalResults.blind\_avg – Denoised and blindly averaged

These are all saved at the end of the code and can be done under whichever name desired.

