Fabber-Matlab documentation

Release 0.0.1

Martin Craig

Jan 17, 2021

Contents

This module contains a thin wrapper between Matlab and the Fabber Bayesian model fitting tool. It works by calling the command line programs and writing out data sets as temporary Nifti files. The module is available from the Matlab File Exchange

The Matlab toolbox Tools for Nifti and Analyze is required for the wrapper to work.

If you already have Fabber installed as part of FSL 6.0.1+ this interface should work without any further configuration. If you don't have FSL, a packages is available with a pre-built set of Fabber binaries which contain a selection of model libraries. This package can be obtained from:

https://github.com/ibme-qubic/fabber_core/releases

In this case you should set the `FABBERDIR` environment variable to point to the location where you downloaded the package.

Example usage

Simple script to fit a polynomial model to random data - this is included in the `fabber_example.m` script:

```matlab
% Create random 4D dataset and mask
rundata = struct();
rundata.data = rand(10, 10, 10, 5);
rundata.mask = logical(ones(10, 10, 10));

% Infer using polynomial model
rundata.model='poly';
rundata.degree=2;
rundata.method='vb';
rundata.noise='white';
rundata.save_mean=true;
rundata.save_model_fit=true;

output = fabber(rundata);
```

`output` is a structure with a named element for each Fabber output, for example `output.mean_c0` is the parameter map for the `c0` parameter. `output.modelfit` contains the 4D model prediction.

Option specification

Many Fabber options contain – characters (e.g. `save-mean`), which cannot be used as field names in a Matlab structure. This poses a problem, because this is what we use to pass the `rundata` options in the above example.

The solution is to substitute underscore characters _ instead, as we have done for the `save-mean` parameter which is specified as `rundata.save_mean`. The Matlab interface will convert the underscores back into – before passing them to Fabber.

Only a couple of core Fabber options contain a _ normally and these are handled specially. Models should never use an underscore in their options.

Additional voxel data

Additional data required for some models can be specified in the rundata in the expected way, for example:

```
aif = load_untouch_nii('aif.nii.gz')
rundata.suppdata = aif.img
```

Issues/Limitations/ToDo

- Error handling is largely non-existant at present
- There is currently no provision for matrix-data to be passed directly - for example the dataspec in CEST. As currently, this must be written to a file and the filename passed as the appropriate option.

Citation

If you make use of Fabber in your research, you should as a minimum cite[1] . Further references are available on the main Fabber documentation page.

---

[1] *Chappell, M.A., Groves, A.R., Woolrich, M.W., "Variational Bayesian inference for a non-linear forward model", IEEE Trans. Sig. Proc., 2009, 57(1), 223–236.*