



White matter microstructure modelling using a modular and extensible GPU accelerated toolkit

Matteo Bastiani¹, Matteo Bastiani¹, Rainer Goebel¹, Alard Roebroeck¹
¹Maastricht University, Maastricht, Netherlands

INTRODUCTION: Multi-shell high angular resolution diffusion MRI (dMRI) is increasingly used to estimate models of brain white matter that include either multiple fiber directions in each voxel (e.g. Behrens, 2007) or microstructural information such as axonal density, diameters and/or dispersion (Assaf, 2005; Alexander, 2010; Zhang, 2012). Combining the modeling of multiple direction and microstructural information is a computationally expensive operation. In this study we show a new GPU accelerated flexible open source toolbox that can parallelize the microstructure analysis. This abstract shows how this toolbox can be used in the clinical or research pipeline.

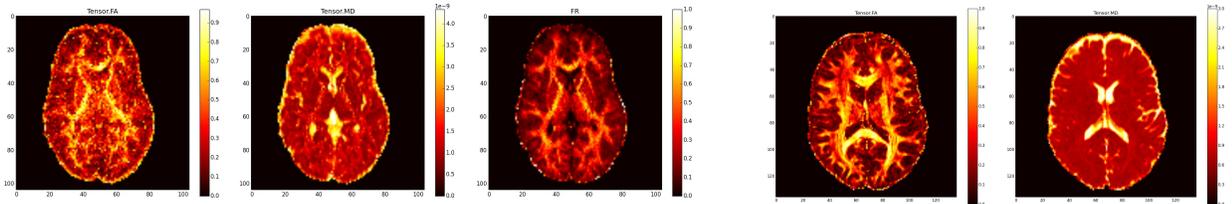
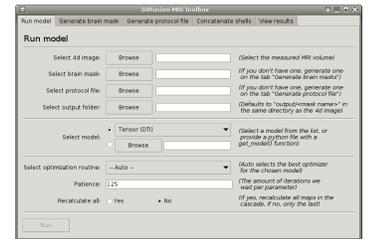
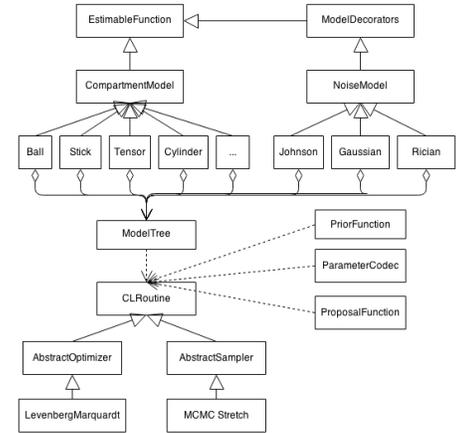
DESIGN: The toolbox is completely written in Python and OpenCL. This allows it to run on Windows, Mac and Linux. The most important functions are packed in a GUI, such that the end-user does not need to program. Model developers can use the Python interface to construct new models. The OO design abstracts microstructure compartment models, noise models and optimization and sampling routines, such that each can be used modularly. Using the toolbox one can build complex models by combining "Estimatable Functions" (e.g. compartment models) in a tree-like language and combine these with a prior, parameter codec and proposal function. The complete model can then be handed to any CLRoutine (e.g. optimizer) to estimate the parameters using different optimization and sampling techniques.

MICROSTRUCTURE MODELLING: In the microstructure modelling language it is possible to generate complex hierarchical multi-compartment models using a tree structure. Compartment models are combined using basic math operators and/or decorated with a "Model Decorator Function" (for example, a noise model). For instance: (Ball(), Stick(), '+'). To fix parameters to specific values one can use the fix command with either a constant or previously calculated results (constants or per voxel). For a complete example of a Ball & Stick model please see figure 2.

RECONSTRUCTION: Given a volume with raw data and a model, the CLRoutines creates an OpenCL kernel that enables the computations on the graphics card (GPU). The data is then passed to the kernel / GPU which fits the model to each voxel. The end result is for each parameter a 3d map with the estimated parameter values. By default these are saved as nifti files in an output directory.

EXAMPLES: The DTI uses a 35dir 1.5 mm dataset.

```
(S0),
((Weight('w_iso'),
Ball().fix('d', 1.7e-9),
'*'),
(Weight('w_ani'),
Stick().fix('d', 1.7e-9),
'*'),
'+'),
'*')
```



AVAILABILITY: For availability mail Robbert Harms (robert.harms@maastrichtuniversity.nl). The toolbox will be released Q1-2014

REFERENCES: [1] Alexander D. (2010), *Neuroimage*, no. 52(4), pp. 1374-1389, [2] Assaf Y. (2005), *Neuroimage*, no. 27(1), pp. 48-58, [3] Behrens A. (2007), *Neuroimage*, no. 34(1), pp. 144-155 [4] De Santis (2014), *Magn. Reson. Med.*, no. 71, pp. 661-671, [5] Panagiotaki E. (2011), *NeuroImage*, no. 59(3), pp. 2241-2254, [6] Sotiropoulos S. (2013), *Proc. Intl. Soc. Mag. Reson. Med.* no. 21, pp. 0052, [7] Zhang H. (2012), *Neuroimaging*, no. 61(4), pp.1000-101.