frontiers in Neuroinformatics

ORIGINAL RESEARCH
published: 15 October 2019
doi: 10.3389/fninf.2019.00064

The Dmipy Toolbox: Diffusion MRI Multi-Compartment Modeling and Microstructure Recovery Made Easy

Rutger H. J. Fick[1,2]*, *Demian Wassermann*[3] *and Rachid Deriche*[2]

[1] TheraPanacea, Paris, France, [2] Inria Sophia Antipolis-Méditerranée, Université Côte d'Azur, Sophia Antipolis, France, [3] INRIA, CEA, Université Paris-Saclay, Paris, France

Non-invasive estimation of brain microstructure features using diffusion MRI (dMRI)—known as Microstructure Imaging—has become an increasingly diverse and complicated field over the last decades. Multi-compartment (MC)-models, representing the measured diffusion signal as a linear combination of signal models of distinct tissue types, have been developed in many forms to estimate these features. However, a generalized implementation of MC-modeling as a whole, providing deeper insights in its capabilities, remains missing. To address this fact, we present Diffusion Microstructure Imaging in Python (Dmipy), an open-source toolbox implementing PGSE-based MC-modeling in its most general form. Dmipy allows on-the-fly implementation, signal modeling, and optimization of any user-defined MC-model, for any PGSE acquisition scheme. Dmipy follows a "building block"-based philosophy to Microstructure Imaging, meaning MC-models are modularly constructed to include any number and type of tissue models, allowing simultaneous representation of a tissue's diffusivity, orientation, volume fractions, axon orientation dispersion, and axon diameter distribution. In particular, Dmipy is geared toward facilitating reproducible, reliable MC-modeling pipelines, often allowing the whole process from model construction to parameter map recovery in fewer than 10 lines of code. To demonstrate Dmipy's ease of use and potential, we implement a wide range of well-known MC-models, including IVIM, AxCaliber, NODDI(x), Bingham-NODDI, the spherical mean-based SMT and MC-MDI, and spherical convolution-based single- and multi-tissue CSD. By allowing parameter cascading between MC-models, Dmipy also facilitates implementation of advanced approaches like CSD with voxel-varying kernels and single-shell 3-tissue CSD. By providing a well-tested, user-friendly toolbox that simplifies the interaction with the otherwise complicated field of dMRI-based Microstructure Imaging, Dmipy contributes to more reproducible, high-quality research.

Keywords: diffusion MRI, multi-compartment modeling, microstructure estimation, reproducible research, neuroimaging, optimization, open-source, PGSE

OPEN ACCESS

Edited by:
Ludovico Minati,
Tokyo Institute of Technology, Japan

Reviewed by:
Mara Cercignani,
Brighton and Sussex Medical School,
United Kingdom
Władysław Piotr Weglarz,
Institute of Nuclear Physics (PAN),
Poland

****Correspondence:***
Rutger H. J. Fick
fick.rutger@gmail.com

Received: *21 June 2019*
Accepted: *04 September 2019*
Published: *15 October 2019*

Citation:
Fick RHJ, Wassermann D and Deriche R (2019) The Dmipy Toolbox: Diffusion MRI Multi-Compartment Modeling and Microstructure Recovery Made Easy. Front. Neuroinform. 13:64. doi: 10.3389/fninf.2019.00064

1. INTRODUCTION

For over three decades, multi-compartment (MC) modeling has played a major role in driving diffusion MRI (dMRI)-based microstructure research. It has enabled breakthroughs in our understanding of the orientation of white matter pathways in the brain (e.g., Basser et al., 1994; Behrens et al., 2003; Tournier et al., 2007), axon bundle dispersion (e.g., Kaden et al., 2007; Zhang et al., 2012), axon diameter (e.g., Assaf et al., 2008), extra-axonal diffusivity (e.g., Novikov et al., 2018), and tumor composition (e.g., Le Bihan et al., 1988; Panagiotaki et al., 2014). Importantly, in all these works, the valid interpretation of estimated model parameters with respect to tissue composition always hinges on:

- The appropriateness of the MC-model composition in terms of *biophysical models* for the tissue of interest (Panagiotaki et al., 2012; Ferizi et al., 2016);
- The sensitivity of the dMRI acquisition to the tissue feature of interest (Ning et al., 2015);
- The specificity of the MC-model's parameters to the tissue feature of interest (Jelescu et al., 2016);
- Given an MC-model, the robustness and accuracy of the optimization approach that estimates the model parameters from the measured data (Harms et al., 2017; Canales-Rodríguez et al., 2019).

To leverage and showcase the reproducibility of published works, several platforms have released *application-specific* toolboxes for particular MC-model implementations. Well-known examples are UCL's collected works on MC-based microstructure estimation (Cook et al., 2006)[1], MRtrix's works aimed at fiber tractography (Tournier et al., 2012)[2], and Dipy's wide variety of dMRI-based signal models (Garyfallidis et al., 2014)[3]. While valuable, static implementations of MC-models are limited in that they cannot be easily adjusted for use case specific tissue configurations, data acquisitions, or desired optimization algorithms. On the other hand, implementing your own MC-modeling approach from scratch requires non-trivial knowledge of computer science and optimization theory, often leaving the user no choice but to use what is available.

Dmipy (*Diffusion Microstructure Imaging in Python*) is an open-source software solution based on the idea that MC-modeling—and all that involves—should be transparent, reproducible, and most of all *easy*. To this end, Dmipy facilitates the on-the-fly design, optimization and analysis of custom MC-models for any PGSE-based dMRI acquisition. Dmipy enables having only high-level interaction with model design and parameter recovery, which in turn highly simplifies the implementation of complex state-of-the-art MC-models. In fact, most MC-models in literature can be reproduced in around 10 lines of code, which we demonstrate in this work for IVIM (Le Bihan et al., 1988), ActiveAx (Alexander et al., 2010), AxCaliber (Assaf and Pasternak, 2008), Ball and Stick (Behrens et al., 2003), NODDI(-x) (Zhang et al., 2012), NODDI-Bingham (Tariq et al., 2016), MC-MDI (Kaden et al., 2016), Multi-Tissue CSD (Jeurissen et al., 2014), and Single-Shell Multi-Tissue CSD (Dhollander and Connelly, 2016).

We provide the graphical abstract Dmipy in **Figure 1**. Dmipy's design is based on the observation that one can view different formulations biophysical models as "building blocks," which could be assembled in any combination, and whose meaning can change depending on the application (e.g., Panagiotaki et al., 2012; Fick, 2017). This is despite the fact that the individual tissue models, such as various approximations of restricted geometries like cylinders and spheres (e.g., Balinov et al., 1993; Vangelderen et al., 1994; Callaghan, 1995), can have quite complex mathematical formulations. To some extent, several open-source solutions have already taken advantage of this observation for applications such as modular signal generation with generalized acquisition parameters in MISST (Ianuş et al., 2016), or GPU-accelerated model generation and optimization in MDT (Harms et al., 2017).

In this work, however, we observe that this notion of modularity goes much further than previously explored, and includes alternative dMRI representation frameworks such as the multi-compartment *spherical mean* (MC-SM) models (e.g., Kaden et al., 2015) and multi-compartment *spherical harmonics* (MC-SH) models (e.g., Jeurissen et al., 2014). As a consequence, this also implies the generalization of optimization strategies that have been specifically proposed for fitting the dMRI signal, such as Microstructure Imaging in Crossings (MIX) (Farooq et al., 2016), and multi-tissue CSD (Jeurissen et al., 2014), to any MC-model composition. In particular, by generalizing MC, MC-SM, and MC-SH models to use the same biophysical models, it becomes natural to start exploring *cross-MC-modeling* approaches, where one set of parameters is estimated in one framework and then used to initiate the optimization in another framework or model composition (e.g., Nath et al., 2019; Pizzolato et al., 2018).

In Dmipy, we take this "building-block"-based philosophy on MC-modeling to the next level, and have implemented a "model-agnostic" MC-model-generation and estimation framework that can:

- Assemble any combination of biophysical models into any of these three MC-model frameworks on-the-fly;
- Add parametric and non-parametric parameter distributions to model *axon orientation dispersion* and/or *axon diameter distribution* effects for any appropriate tissue model;
- Impose any predefined or custom parameter constraints or relations;
- Optimize models using generalized, open-source implementations of various optimization strategies;
- Simulate the signal and estimate parameters from any user-defined PGSE-dMRI acquisition, varying over gradient strengths (multi-shell), diffusion times, and echo times.

Dmipy is freely available under open-source MIT license at https://github.com/AthenaEPI/dmipy, where detailed tutorials and implementations of many MC-models in literature are provided in the form of Jupyter Notebooks. As for dependencies, Dmipy's dynamic modeling design is based on numpy (Oliphant, 2006), non-linear optimization on scipy (Jones et al., 2001), convex optimization on cvxpy (Diamond and Boyd, 2016; Agrawal et al., 2018) and for signal-based subroutines (e.g., DTI) and visualization we use dipy (Garyfallidis et al., 2014). As optional dependencies, we speed up code use Numba's just-in-time compilation (Lam et al., 2015), and allow multi-core CPU processing using pathos (McKerns et al., 2012).

[1] http://mig.cs.ucl.ac.uk

[2] http://www.mrtrix.org/

[3] https://nipy.org/dipy/

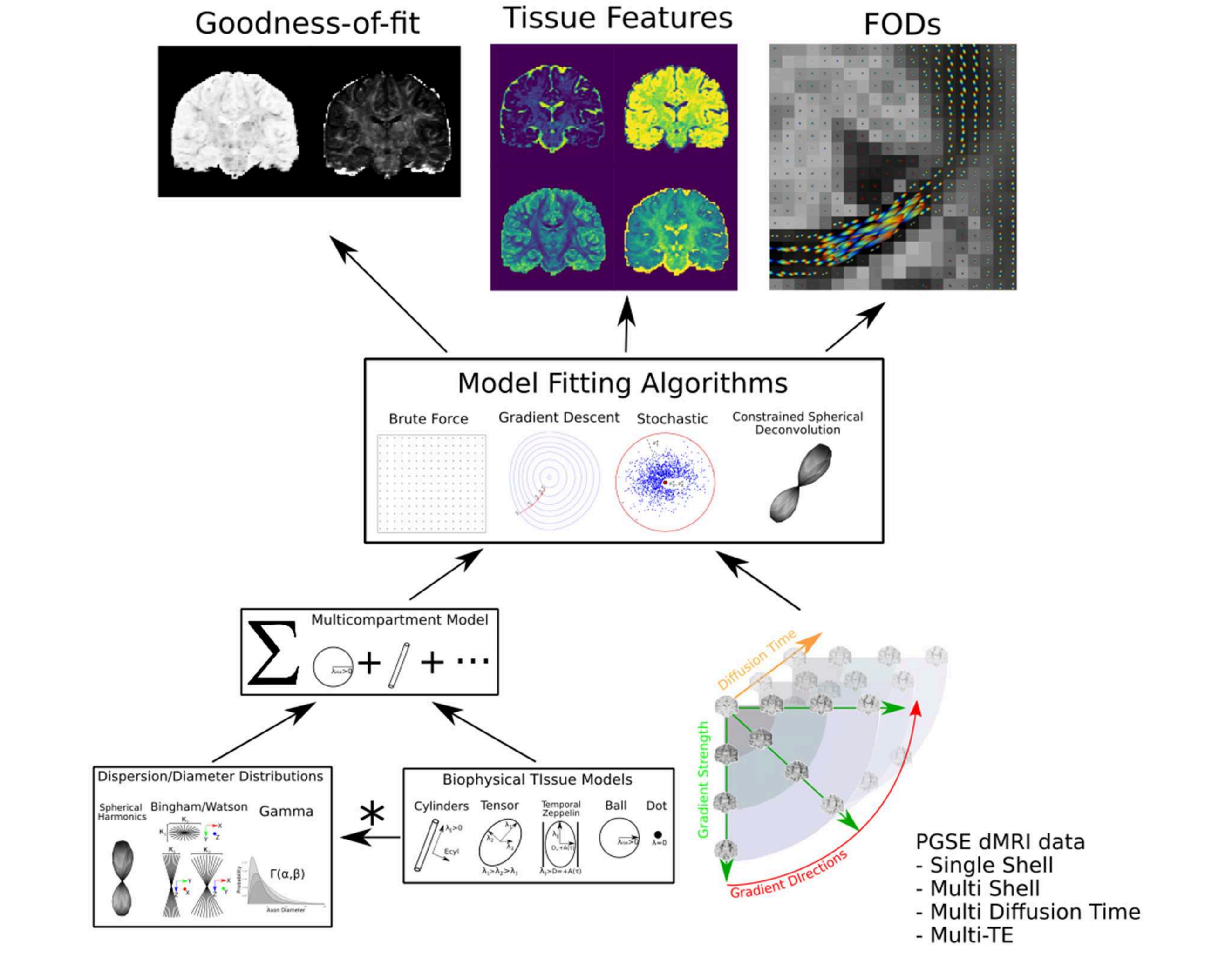


FIGURE 1 | Dmipy workflow: Modular microstructure model setup and parameter estimation. Different biophysical tissue models (see **Figure 3**) are dispersed and/or distributed and combined together in a multi-compartment model, which is then fit to diffusion data using a chosen optimization algorithm to estimate tissue feature parameters, reconstruct FODs, and quantify the quality of the fitting.

The structure of this paper is as follows: in section 2, we explain the theory and basic interaction behind generalizing MC-modeling over the three modeling frameworks, for any PGSE acquisition scheme and biophysical model composition. In particular, we provide Dmipy's technical aspects and workflow, explaining how the generalized signal simulation, parameter linking and optimization can be used. In section 3 we then show Dmipy in action, implementing, and demonstrating various state-of-the-art MC-modeling approaches. Finally, we discuss Dmipy's contribution to dMRI microstructure imaging and further opportunities in section 4, and provide concluding remarks in section 5.

2. THEORY AND IMPLEMENTATION

The goal of the Dmipy framework is to allow for the natural, generalized implementation of MC-based microstructure recovery, based on the PGSE-dMRI sequence. In this section, we explain both the theory of MC-modeling and how it is implemented in Dmipy. Note, however, that we only consider the analytic description of any algorithms we introduce, leaving the numerical implementation for the **Supplementary Material**. As this section describes and implements a considerably large part of the literature, we provide a table of contents in **Table 1**.

2.1. PGSE Diffusion Contrast

In this work, we focus on probing the tissue microstructure using the standard Pulsed Gradient Spin-Echo sequence (PGSE) to obtain diffusion-weighted images (DWIs) (Stejskal, 1965). We provide a schematic representation of a PGSE sequence in **Figure 2A**. To summarize, a diffusion-weighted measurement at position $\mathbf{x} \in \mathbb{R}^3$ is obtained by first applying a 90° radio-frequency pulse, after which two sensitizing diffusion gradients of pulse length δ [s], gradient strength G [T/m], and separated by separation time Δ [s], are applied to the tissue along orientation $\mathbf{n} \in \mathbb{S}^2$. In between the two pulses, at half the Echo Time (TE), a 180° pulse is applied, resulting in a measurable spin echo signal at TE. The PGSE sequence is sensitive to diffusion in that the measured signal attenuates proportionately to the average particle motion along $\mathbf{n}$ in between the two gradient pulses. More precisely, in three dimensions, the signal attenuation for a PGSE sequence is given by $E(b,\mathbf{n}) = \exp\left(-b\mathbf{n}^T\mathbf{D}_{\text{app}}(\tau)\mathbf{n}\right)$, where $\mathbf{D}_{\text{app}}(\tau) \in \mathbb{R}^{3\times3}$ [m^2/s] is the *apparent* diffusion tensor at diffusion time $\tau = \Delta - \delta/3$ [s], and $b = G^2\delta^2\gamma^2(\Delta-\delta/3)$ [s/m^2] is the clinically used b-value with nuclear gyromagnetic ratio γ [s^{-1}T^{-1}] (Le Bihan et al., 1986; Minati and Węglarz, 2007).

TABLE 1 | Table of contents of sections.

Section description	Section
PGSE Diffusion Contrast	2.1
DmipyAcquisitionScheme	2.2
Multi-Compartment Modeling	2.3
Generalized 3D Signal Representations	2.4
Microscopic Compartment Models	2.5
Gaussian Compartment Models	2.5.1
Restricted CompartmentModels	2.5.2
Macroscopic Distibuted Models	2.6
Parameter Distributed Models	2.6.1
Spherical Distributed Models	2.6.2
Tissue Response Models	2.6.3
Multi-Compartment Model Variants	2.7
Multi-Compartment Spherical Mean	2.7.1
Multi-Compartment Spherical Harmonics	2.7.2
Parameter Linking	2.8
Generalized Optimization	2.9
Optimizing MC and MC-SM Models	2.9.1
Optimizing MC-SH Models	2.9.2
Fitting the Signal: Multi-tissue modeling	2.9.3

From these parameters, the q-value is also written as $q = (G\delta\gamma)/(2\pi)$ [m^{-1}] (Callaghan, 1995).

Measuring the PGSE signal at multiple acquisition parameter combinations provides information on different tissue properties; a single-shell acquisitions (only varying $\mathbf{n}$) allows to estimate tissue orientation (Basser et al., 1994; Tuch, 2004; Tournier et al., 2007); multi-shell (varying $\mathbf{n}, G$) allows to delineate the signal contribution of multiple tissue compartments and axon orientation dispersion (Wu and Alexander, 2007; Zhang et al., 2012); multi-diffusion time (varying G, δ, Δ) allows to estimate e.g., axon diameter distributions (Assaf et al., 2008; De Santis et al., 2016); and multi-Echo Time (TE) is needed to recover myelin content in white matter (Whittall and MacKay, 1989). We illustrate the DWI diffusion contrast in a coronal brain slice over varying $\mathbf{n}, G, \Delta$ in **Figure 2B**.

Given the PGSE acquisition parameters, the measured signal S is separable in terms of its non-diffusion weighted signal intensity S_0 when $G = 0$, and the signal *attenuation* E when $G > 0$, such that

$$S(\mathbf{x}, G, \mathbf{n}, \Delta, \delta, TE) = S_0(\mathbf{x}, TE) \cdot E(\mathbf{x}, G, \mathbf{n}, \Delta, \delta). \tag{1}$$

Notice that at any position $\mathbf{x}$ the signal amplitude S_0 is only dependent on TE, and the signal's shape E is dependent on all other parameters except TE.

2.2. Representing Any PGSE Acquisition Using a DmipyAcquisitionScheme

To allow Dmipy to interact with any PGSE acquisition scheme, the DmipyAcquisitionScheme module takes the raw PGSE acquisition scheme parameters and prepares them to be used for signal generation. A DmipyAcquisitionScheme can be instantiated in three ways:

- For clinical data, using b-values [s^2/m] and gradient directions (and optionally δ/Δ/TE [s]);

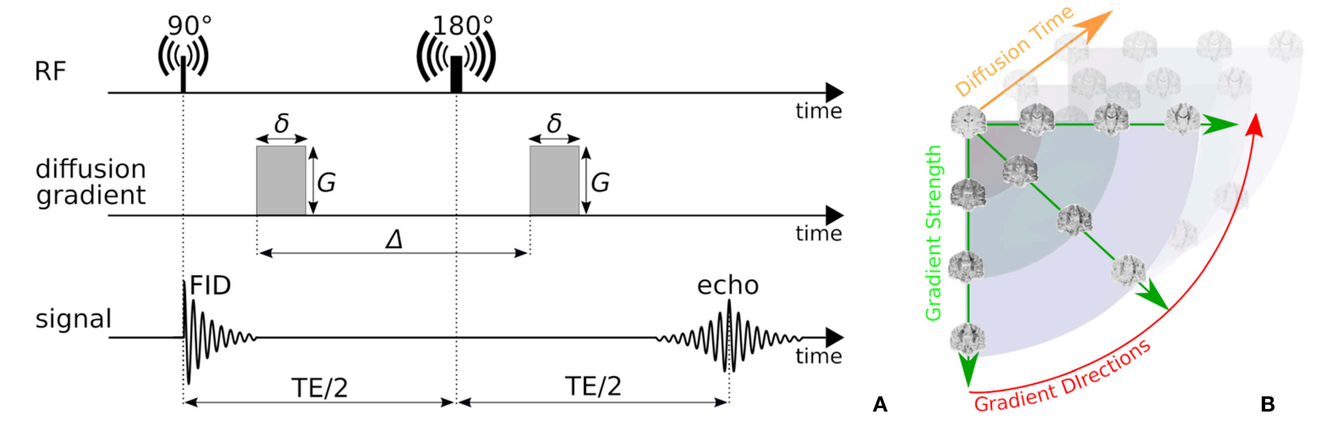


FIGURE 2 | (A) A schematic representation of a Pulsed-Gradient Spin Echo (PGSE) sequence, and **(B)** illustrates the whole-brain diffusion contrasts in a four-dimensional, spatio-temporal PGSE acquisition, varying over gradient strength, gradient direction, and diffusion time. Notice that depending on the tissue configuration, different acquisition parameter combinations result in different image contrasts.

Snippet 1 | Typical DmipyAcquisitionScheme creation using b-values, gradient orientations as 3D unit vectors, and optional single-value or arrays of pulse durations δ, pulse separations Δ and TEs. The DmipyAcquisitionScheme handles all interactions with the fixed acquisition parameters, meaning we only need to focus on CompartmentModel parameters afterwards. Notice that the printed acquisition information is not in SI units to avoid too small or large numbers, but the input must always be given in SI units.

```
from dmipy.core.acquisition_scheme import acquisition_scheme_from_bvalues
dmipy_scheme = acquisition_scheme_from_bvalues(
    bvalues,              # of shape (N_DWI,) in s/m^2
    gradient_directions,  # of shape (N_DWI, 3)
    pulse_duration,       # number or of shape (N_DWI,) in seconds
    pulse_separation,     # number or of shape (N_DWI,) in seconds
    TE)                   # number or of shape (N_DWI,) in seconds

dmipy_scheme.print_acquisition_info  # for WU-MINN HCP acquisition scheme
>> Acquisition scheme summary
>> total number of measurements: 288
>> number of b0 measurements: 18
>> number of DWI shells: 3
>> shell_idx |DWIs |bval [s/mm^2] |grad strength [mT/m] |delta[ms] |Delta[ms] |TE[ms]
>> 0         |18   |0             |0                    |10.6      |43.1      |89.5
>> 1         |90   |1000          |56                   |10.6      |43.1      |89.5
>> 2         |90   |2000          |79                   |10.6      |43.1      |89.5
>> 3         |90   |3000          |97                   |10.6      |43.1      |89.5
```

- Using G $[T/m]$, gradient directions, δ and Δ (and optionally TE);
- Using q $[m^{-1}]$, gradient directions, δ and Δ (and optionally TE).

We show how to create a scheme for a clinical acquisition in python Snippet 1. Notice that pulse duration, pulse separation, and TE can be given as a single number (it does not change throughout the acquisition), or an array the same size as b-values, potentially changing for every single measurement. Once the scheme is created, a summary can be generated indicating the total number of measurements, shells, b0-measurements, and a shell-wise description of the acquisition parameters. Measurements are automatically separated into distinct acquisition shells using a simple linkage clustering algorithm (Müllner, 2011), which clusters measurements that are closer together than a certain distance in b-space. Measurements having different $\delta/\Delta/TE$ are clustered separately, ensuring measurements with similar b-value but different times are never combined into one shell.

To facilitate the notation of separate DWIs per acquisition shell, let us denote the sampled acquisition scheme parameters as $\mathbf{A} = \{[b_{i,s}, G_{i,s}, \mathbf{n}_{i,s}, \delta_{i,s}, \Delta_{i,s}, TE_{i,s}]\}$. Here, we used double linear indexing i, s, such that $s = 1, 2, \ldots, N_{\text{shells}}$ is the acquisition shell index with N_{shells} the number of shells, and $i = 1, 2, \ldots, N_{\text{DWI}}[s]$ is the DWI index with $N_{\text{DWI}}[s]$ the number of DWIs for shell s. We will also use a *shell-wise* acquisition scheme $\mathbf{A}_s = \{[b_s, G_s, \delta_s, \Delta_s, TE_s]\}$ that omits the gradient directions of that shell $\mathbf{n}_s$, and only contains the unique values of the other parameters for each shell.

2.3. Anatomy of dMRI Multi-Compartment Modeling

To recover information on the tissue microstructure, the attenuation of the signal must first be explained in terms of the underlying diffusion process—the Ensemble Average Propagator (EAP). Using the *short gradient pulse* approximation (SGP), i.e., assuming that no diffusion takes place during the gradient pulse ($\delta \rightarrow 0$), the relation between the signal attenuation and the EAP is given by a Fourier transform (Stejskal, 1965). This transforms Equation (1) into

$$E(\mathbf{x}, \mathbf{A}) \overset{\text{SGP}}{=} \int_{\mathbb{R}^3} P(\mathbf{x}, \mathbf{R}; \tau) e^{(-i\gamma\delta\mathbf{G})\cdot\mathbf{R}} d\mathbf{R} \tag{2}$$

where $P(\mathbf{x}, \mathbf{R}; \tau)$ describes the probability that a particle anywhere inside voxel $\mathbf{x}$ traveled a net distance $R = \|\mathbf{R}\|$ [m] *given* a diffusion time τ. Here, $\mathbf{R} = R\mathbf{r}$ where $\mathbf{r} \in \mathbb{S}^2$ and non-negative $R \in \mathbb{R}^+$. Hypothetically, given sufficiently small resolution and dense measurements in $G, \mathbf{n}, \tau$, one can directly recover and estimate microstructure-related properties of the EAP for a single τ using an inverse Fourier transform or Fourier basis (e.g., Wedeen et al., 2005; Özarslan et al., 2013; Fick et al., 2016), or over continuous τ (Fick et al., 2018). However, such approaches still hinge on the appropriateness of the SGP approximation, and can only describe properties of the overall signal and EAP, not of specific tissues in heterogeneous environments.

To estimate specific properties of the tissue microstructure, Multi-Compartment (MC)-models represent the tissue as a linear combination of single compartment models "C," each representing the diffusion signal originating from specific tissue types. Recovering tissue microstructure information is done by finding the model parameters $\mathbf{p}$ that minimize the difference between modeled and the measured signal. The process of MC-modeling any model containing $i \geq 1$ compartments is thus given by

$$\mathbf{p}^*(\mathbf{x}) = \text{argmin}_{\mathbf{p}} \int \left[E(\mathbf{x}, \mathbf{A}) - \hat{E}^{\text{MC}}(\mathbf{A}, \mathbf{p}) \right]^2 d\mathbf{A}, \tag{3}$$

$$\text{with } \hat{E}^{\mathrm{MC}}(\mathbf{A},\mathbf{p}) = \sum_{i}^{N} f_i C_i(\mathbf{A},\mathbf{p}_i) \tag{4}$$

where the parameters of each model $\mathbf{p}_i$ represent its tissue microstructure properties like diffusivity, orientation, axon diameter, dispersion, and others. The non-negative, normalized volume fractions f_i weight the signal contribution of each compartment model. For these parameters to accurately describe the underlying tissue configuration, it is essential that the MC-model composition in terms of compartment models is appropriate for the actual tissue composition. Notice that Equation (3) is general, and its compartments do not necessarily need to be based on the SGP condition in Equation (2). In the next sections, we show which compartment model representations are available in Dmipy, and how they can be combined into an MC-model and be prepared for optimization.

2.4. Dmipy's Generalized Three-Dimensional Signal Representations

In Dmipy, the single models $C(\mathbf{A},\mathbf{p})$ that make up the MC-model in Equation (3) are represented using generalized three-dimensional signal representations,

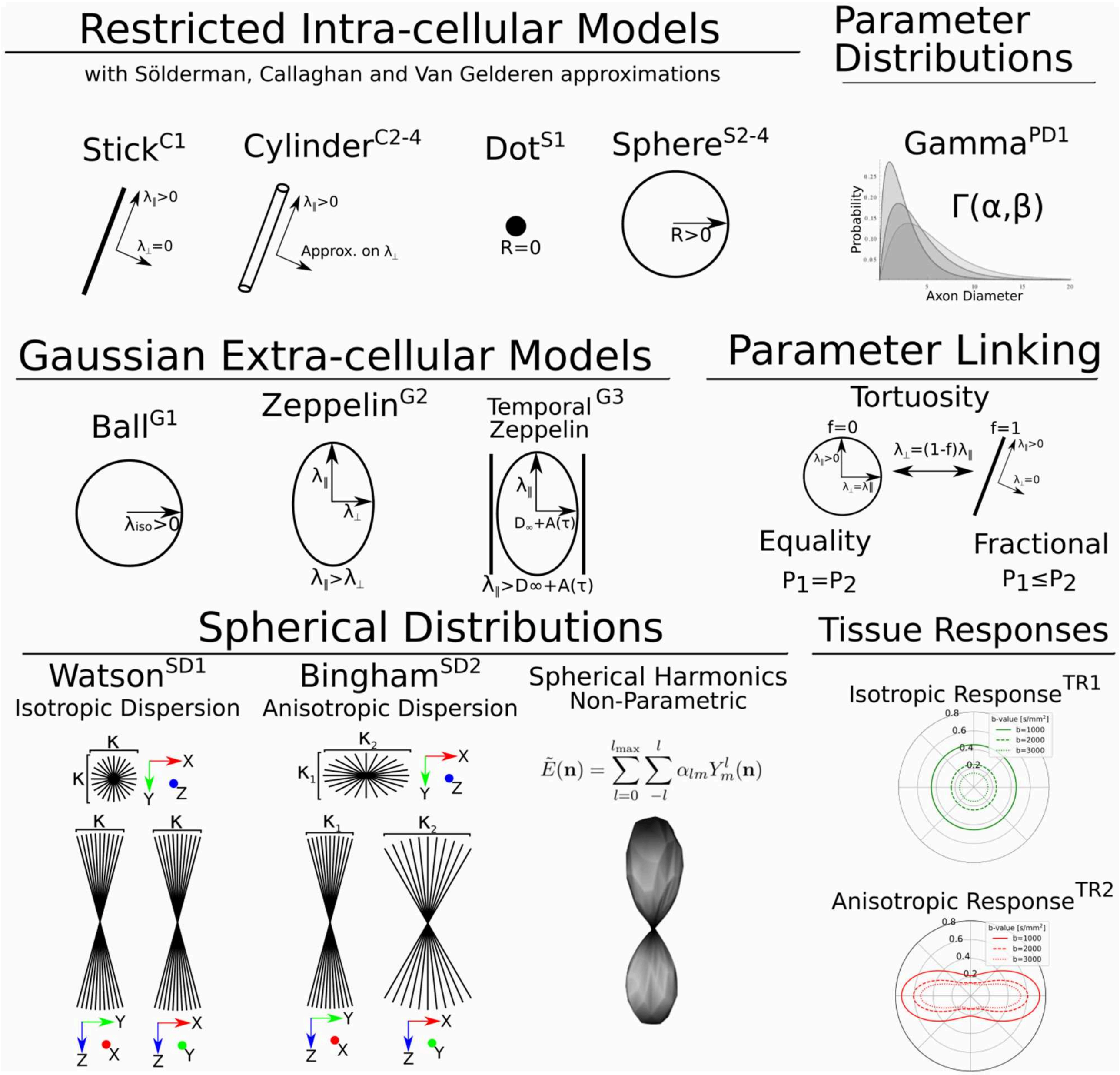


FIGURE 3 | A schematic of most biophysical mode ls that are used in PGSE-based Microstructure Imaging. Using different combinations of these "components," any microstructure model can be assembled using Dmipy.

TABLE 2 | Overview of CompartmentModel parameters with their associated cardinality, SI-units, in which models they occur and description.

Parameter name	Card.	SI unit	Present in models	Description
lambda_iso	1	m^2/s	G1	Isotropic Gaussian diffusivity
lambda_par	1	m^2/s	G2-3, C1-4	Parallel Gaussian diffusivity
lambda_perp	1	m^2/s	G2	Perpendicular diffusivity
lambda_inf	1	m^2/s	G3	Bulk diffusivity
lambda_intra	1	m^2/s	C3-4, S3-4	Intra-cellular diffusivity
A	1	m^{-2}	G3	Characteristic coefficient
mu	2	[rad]	G2-3, C1-4, SD1-2, TR2	Orientation θ/ϕ Euler angles
diameter	1	m	C2-4, S2-4	Cylinder/Sphere diameter
odi	1	[-]	SD1-2	Orientation Dispersion Index
beta	1	[-]	SD2	Secondary Bingham dispersion
alpha, beta	1	[-]	PD1	Shape/Scale of Gamma distribution
sh_coeff	1_{max}	[-]	MC-SH	FOD SH-coefficients
partial_volume_n	1	[-]	MC, MC-SM, MC-SH	volume fraction

called "CompartmentModels" or "DistributedModels." The CompartmentModel is the basic block that contains the signal representation of any single, undistributed biophysical model. CompartmentModels are either isotropic or anisotropic, but axially symmetric. Anisotropic models have an orientation $\boldsymbol{\mu} \in \mathbb{S}^2$ and are described as a separable product of the one- and two-dimensional models, representing their parallel and perpendicular components, respectively (Assaf et al., 2004). DistributedModels consist of other CompartmentModels or DistributedModels and apply parametric distributions on the parameters of its input models. We illustrate Dmipy's available model representations in **Figure 3** with their associated parameters. By combining these models in various combinations, it is possible to construct nearly any MC-model in the literature. In **Table 2** we explicitly write out the unique parameters of these models, provide their cardinality, SI unit, and in which models they are present.

To generate the signal representation for any input model $C(\mathbf{A}, \mathbf{p})$—regardless of the mathematical representation—we only need values for its parameters $\mathbf{p}$ and the acquisition parameters $\mathbf{A}$, represented in a DmipyAcquisitionScheme. Having this input, a CompartmentModel or DistributedModel can return three main signal *attenuation* representations to be used by the higher-level multi-compartment and distributed models:

- A standard representation $\hat{E}(\mathbf{A}) = C(\mathbf{A}, \mathbf{p}) \in \mathbb{R}^{N_{\text{DWI}}}$, evaluating the biophysical model for the given acquisition parameters $\tilde{\mathbf{A}}$, where N_{DWI} is the number of acquired DWIs.
- A spherical mean representation $\hat{E}_{SM}(\mathbf{A}_s) = \frac{1}{4\pi}\int_{\mathbb{S}^2} C(\mathbf{A}_s, \mathbf{n}; \mathbf{p})d\mathbf{n} \in \mathbb{R}^{N_{\text{shells}}}$, returning the shell-wise spherical mean of the signal. If no analytic expression is available, then it is estimated numerically.
- And a multi-shell convolution kernel matrix $\mathbf{M}(\mathbf{A}_s, \mathbf{p}) \in \mathbb{R}^{N_{\text{DWIs}} \times N_{\text{SH-coef}}}$, which enables convolution of the model's rotational harmonics representation with the spherical harmonics expansion of some spherical distributions with number of coefficients $N_{\text{SH-coef}}$. See details in the **Supplementary Material**.

In Snippet 2, we provide the code to instantiate all the available microscopic CompartmentModels that we will present in the next section 2.5. The signal generation for the Distributed Models we present in section 2.6 is analogous.

2.5. Microscopic Compartment Models

Parsimonious models of the tissue microstructure are usually based on what we know the tissue looks like from histological observations. Well-known examples are using cylinders of some diameter to represent axons (Aboitiz et al., 1992; Assaf et al., 2008), or spheres of some diameter to represent tumor cells (Panagiotaki et al., 2014). In this section, we describe the way Dmipy allows modular signal generation for any microscopic model, and which models are available. These include Gaussian models (G) in section 2.5.1, restricted models such as cylinders (C) and spheres (S) in section 2.5.2.

2.5.1. Gaussian Models (G)

In MC-modeling, Gaussian models are used to represent free or hindered diffusion inside tissues that are not explicitly restricting the movement of diffusing particles. Essentially, they say a particle could travel anywhere in $\mathbf{R} \in \mathbb{R}^3$ with non-zero, but exponentially decreasing probability as $R \rightarrow \infty$. They are convenient to represent the Fourier-like relationship between the signal and EAP, as the closed-form Fourier transform of a Gaussian is another Gaussian.

G1: The simplest Gaussian model is the Ball: an isotropic Gaussian compartment whose signal attenuation only depends on isotropic diffusivity $\lambda_{\text{iso}}\,[\text{m}^2/\text{s}]$. Balls usually represent the diffusion signal of free water (Behrens et al., 2003), or tissue types that on average induce an isotropic hindrance on the EAP, like gray matter (Dell'Acqua et al., 2010).

G2: The Zeppelin is an axially symmetric Gaussian distribution aligned along orientation $\boldsymbol{\mu}$, with parallel and perpendicular diffusivity $\lambda_{\parallel} \geq \lambda_{\perp}\,[\text{m}^2/\text{s}]$. It is often used to describe the diffusion signal originating from the oriented, extra-axonal space (Alexander et al., 2010; Zhang et al., 2012).

G3: To represent extra-axonal diffusion, recent works argue that hindered diffusion is actually slower-than-exponential over

Snippet 2 | One-line instantiation of available CompartmentModel representations in Dmipy. For any of the models the parameter cardinality and names can be inspected. Signal representations for any DmipyAcquisitionScheme can be generated by providing valid values for the parameters.

```python
from dmipy.signal_models import gaussian_models, cylinder_models, sphere_models
ball = gaussian_models.G1Ball()
zeppelin = gaussian_models.G2Zeppelin()
temporal_zeppelin = gaussian_models.G3TemporalZeppelin()

stick = cylinder_models.C1Stick()
cylC2 = cylinder_models.C2CylinderStejskalTannerApproximation()
cylC3 = cylinder_models.C3CylinderCallaghanApproximation()
cylC4 = cylinder_models.C4CylinderGaussianPhaseApproximation()

dot = sphere_models.S1Dot()
sphereC2 = cylinder_models.S2SphereStejskalTannerApproximation()
sphereC3 = cylinder_models.S3SphereCallaghanApproximation()
sphereC4 = cylinder_models.S4SphereGaussianPhaseApproximation()

cylC2.parameter_cardinality  # model parameters and their cardinality
>>OrderedDict([('lambda_par', 1),  # parallel diffusivity [s/m^2]
>>             ('mu', 2),          # orientation theta, phi
>>             ('diameter', 1)])   # cylinder diameter [m]
# example signal generation for three modeling frameworks
params = {'acquisition_scheme': dmipy_scheme,  # DmipyAcquisitionScheme
          'lambda_par': 1.7e-9,                # some reasonable WM value
          'mu': [0., 0.],                      # aligned along z-axis
          'diameter': 2e-6}                    # within white matter range [1-3] um
E = cylC2(**params)                                # shape (N_DWI)
E_sm = cylC2.spherical_mean(**params)              # shape (N_shells)
A = cylC2.convolution_matrix(**params, lmax=lmax)  # shape (N_DWI, N_coef)
```

τ due to how the external axon boundaries still restrict diffusing particles (Novikov et al., 2014). To account for this, Burcaw et al. (2015) proposed a modification to the Zeppelin we refer to as the Temporal Zeppelin, where the perpendicular diffusivity is instead given by $\lambda_\perp = \lambda_\infty + \frac{A\ln(\Delta/\delta)+3/2}{\Delta-\delta/3}$, with λ_∞ [m^2/s] the bulk diffusivity for large diffusion times, and A [m^{-2}] the characteristic coefficient for the packing of the axons. The Temporal Zeppelin has been used by De Santis et al. (2016) to characterize extra-axonal hindrance in tandem with axon diameter estimation.

2.5.2. Restricted Cylinder (C) and Sphere (S) Models

Diffusion *restriction* is the phenomenon of particles diffusing during pulse separation time Δ, exploring the geometry and finding they cannot move through boundaries at a certain distance D [m] from their origin point, i.e., $P(\mathbf{R}) > 0, \forall\|\mathbf{R}\| \leq D$. The relationship between the EAP and the diffusion signal inside Cylinders (C) and Spheres (S) has been well studied (Stejskal, 1965; Balinov et al., 1993; Vangelderen et al., 1994; Callaghan, 1995). The most commonly used applications of these models are Cylinders to represent axons (Assaf et al., 2004), and Spheres to represent cell bodies (Panagiotaki et al., 2014), such that the diameter of the cylinder or sphere represents the diameter of axons and cells. Sphere models are isotropic with a diameter D and intra-spherical diffusivity λ_{intra}. Cylinder models are formulated as anisotropic models, having Gaussian diffusivity $\lambda_\|$ along orientation $\boldsymbol{\mu}$, and Cylinder diameter D and intra-cylindrical diffusivity λ_{intra} perpendicular to it.

C1, S1: Both the Cylinder and Sphere models have implementations that assume the geometry diameter is negligible and can be set to zero. This results in the well-known Stick model (C1) (Behrens et al., 2003) and non-diffusing Dot model (S1) (Panagiotaki et al., 2012).

C2-4, S2-4: To relate the EAP to the measured signal for restricted models, several assumptions on the diffusion process are used. Dmipy implements three different approximations of Cylinders and Spheres, which we order by decreasingly stringent assumptions on the diffusion process. The most stringent is the "Stejskal-Tanner" approximation (C2, S2), which simultaneously assumes the previously described SGP, and the long diffusion time limit. The "Callaghan" approximation allows finite diffusion time, only assuming the SGP (C3, S3). Finally, the "Gaussian Phase" approximation only assumes the diffusion is Gaussian during pulse length δ (C4, S4). We note that C4, S4 are examples of models that are not based on the SGP condition in Equation (2). Details can be found in the **Supplementary Material**.

2.6. Macroscopic Distributed Models

DistributedModels include features that are properties of groups of microstructural features, like the distributions of axon diameter (Assaf et al., 2008) and axon orientation (Kaden et al., 2007; Tournier et al., 2007) inside the same white matter bundle. There are two types of DistributedModels in Dmipy. The first is the "ParameterDistributed" (PD) model for one-dimensional distributions over e.g., axon diameter, which we describe in section 2.6.1. The second is the "SphericalDistributed" (SD) model for parametric distributions of models on the sphere for e.g., axon dispersion (Leergaard et al., 2010) in section 2.6.2. We also describe "TissueResponse" models, which are non-parametric representations of the macroscopic measured signal, in section 2.6.3.

2.6.1. Parametric Distributions Over Model Parameters (PD)

A parameter distributed (PD) model can apply parametric distributions over parameters of cardinality one, like diffusivity or cylinder diameter (see **Table 2**). We define a parametric distribution as $P(\xi;\mathbf{\Omega})$ with $\mathbf{\Omega}\in\mathbb{R}$ its parameters and $\xi\in\mathbb{R}^+$ the parameter that is sampled. We can then define any parameter distributed compartment model C_{PD} as

$$C_{PD}(\mathbf{A};\tilde{\mathbf{p}},\mathbf{\Omega})=\frac{\int_{\mathbb{R}^+}\overbrace{P(\xi;\mathbf{\Omega})}^{\text{PDF}}\cdot\overbrace{C(\mathbf{A};\xi,\tilde{\mathbf{p}})}^{\text{(Restricted) Signal Attenuation}}\cdot\overbrace{N(\xi)}^{\text{Volume Correction}}d\xi}{\underbrace{\int_{\mathbb{R}^+}P(\xi;\mathbf{\Omega})\cdot N(\xi)d\xi}_{\text{Normalization}}}\qquad(5)$$

where $\{\xi,\tilde{\mathbf{p}}\}\in\mathbf{p}$ together are all parameters of model C. The distributed parameter ξ is integrated out, and replaced by the distribution parameters $\mathbf{\Omega}$. When distributing diffusivity the normalization is $N(\xi)=1$, but, in the case of distributing the radii of restricted models such as planes, cylinders, or spheres, $N(\xi)$ is the volume function of that geometry. Briefly, $N_{\text{plate}}(\xi)=\xi$, $N_{\text{cylinder}}(\xi)=\pi\xi^2$ and $N_{\text{sphere}}(\xi)=\frac{4}{3}\pi\xi^3$. The reason for this is that it is not the geometries themselves, but the simulated particles diffusing inside these geometries that are contributing to the signal attenuation. In Snippet 3, line 9 it is demonstrated that any model parameter can be distributed only by providing its name upon initialization of the PD model.

PD1: A Gamma distribution with shape parameter α and scale parameter β has been used to model the axon diameter distribution in nervous tissue (Aboitiz et al., 1992; Assaf et al., 2008).

2.6.2. Spherical Distributions Over Model Orientation (SD)

Spherical distributed (SD) models apply parametric distributions of over model orientation. The signal attenuation of a SD model is obtained by means of a spherical convolution of spherical distribution $\text{F}(\mathbf{n};\mathbf{\Omega}):\mathbb{S}^2\to[0,\infty]$ with parameters $\mathbf{\Omega}$, with one or more convolution kernel models $C_i^K(\mathbf{n},\tilde{\mathbf{p}}_i)=C(\mathbf{n},\tilde{\mathbf{p}}_i|\boldsymbol{\mu}_i=[0,0,1])$, meaning $\tilde{\mathbf{p}}_i,\boldsymbol{\mu}_i\in\mathbf{p}_i$ and the orientation $\boldsymbol{\mu}_i$ is parallel to the z-axis. An SD model is thus given by

$$C_{\text{SD}}(\mathbf{A}_s,\mathbf{n};\{\tilde{\mathbf{p}}_i\},\mathbf{\Omega})=\int_{\mathbb{S}^2}\text{F}(\mathbf{n}-\mathbf{u};\mathbf{\Omega})\cdot\sum_{i=1}^{N}f_iC_i^K(\mathbf{A}_s,\mathbf{u};\tilde{\mathbf{p}}_i)d\mathbf{u}\qquad(6)$$

$$=\left[\text{F}(\mathbf{\Omega})*_{\mathbb{S}^2}\sum_{i=1}^{N}f_iC_i^K(\mathbf{A}_s;\tilde{\mathbf{p}}_i)\right](\mathbf{n}),\qquad(7)$$

where f_i are the normalized volume fractions weighting the signal contribution of each convolution kernel, $\mathbf{A}_s$ the shell-wise acquisition parameters and $\mathbf{u}\in\mathbb{S}^2$ the integrated orientation variable. Notice that the spherical convolution is done separately for each shell s. We provide details on the numerical implementation of Equation (6) in the **Supplementary Material**. We will use the second shorthand in Equation (7) for spherical convolution hereafter.

SD-models can be seen as a sub-MC-model, which can, in turn, be combined with other compartment models in Equation (3). In Snippet 3, lines 10 and 11, we illustrate how an SD model is instantiated given one or more input models. Note that, in this way, *multiple* convolution kernels are distributed by *the same* spherical distribution. As in PD models, in SD models the orientation parameters of the input models are removed, and the parameters of the spherical distribution are added. Note that we can stack and apply multiple distributions to parameters of input models. As a concrete example, using lines 15 and 17, we can simulate the signal from a white matter bundle whose axons are simultaneously distributed over diameter and dispersed over orientation.

SD1: The Watson distribution, defined as an isotropic Gaussian distribution on the sphere with orientation $\boldsymbol{\mu}$ and concentration parameter κ. Following Zhang et al. (2012), we use the orientation dispersion index (ODI) as an optimization parameter, defining $ODI=0$ as a spike function along $\boldsymbol{\mu}$, and $ODI=1$ as an isotropically dispersed profile on the sphere.

SD2 The Bingham Distribution, defined as an anisotropic Gaussian distribution on the sphere with orientation $\boldsymbol{\mu}$ and primary and secondary concentration concentrations κ_1,κ_2. Following Tariq et al. (2016), analogously to SD1, we use optimization parameters ODI and β in lieu of the concentrations. The Bingham has an additional ψ "roll" parameter, which rotates the distribution about $\boldsymbol{\mu}$.

2.6.3. Tissue Response Models (TR)

Tissue Response (TR) models are non-parametric representations of the measured signal, which are recovered by averaging the signal in segmented voxels of specific tissue types. These models differ from the previously described parametric models in that they are not related to the "true" underlying EAP of the tissue. In Snippet 4 we provide the code to produce isotropic (TR1) and anisotropic (TR2) tissue response models, given some data segmentation. Formally, let $\mathbf{X}_{\text{iso}}$ and $\mathbf{X}_{\text{aniso}}$ be segmented voxels of isotropic and anisotropic diffusion profiles. Tissue response models are then created by first representing the signal in each voxel as rotational harmonics coefficients $\mathbf{r}$ with order l using a rotational harmonics transform (RHT)—see the **Supplementary Material**—and then averaging those coefficients. In doing so, we can separate the segmented tissue-specific S_0 signal response and its shell-wise shape, allowing for so-called multi-tissue modeling (Jeurissen et al., 2014).

Snippet 3 | One-line instantiation of parameter and spherical distributions and DistributedModels. Notice DistributedModels can be seen as sub-MC-models that apply a distribution to the parameters of base CompartmentModels.

```python
from dmipy.distributions import parameter_distributions, spherical_distributions
# instantiation of the distributions themselves
gamma = parameter_distributions.Gamma()
watson = spherical_distributions.Watson()
bingham = spherical_distributions.Bingham()

# one-line instantiation of distributed models given input models
from dmipy.signal_models import distributed_models
gamma_bundle = distributed_models.PD1GammaDistributed(
    models=[model1], parameter_name=parameter_name)
watson_bundle = distributed_models.SD1WatsonDistributed(models=[model1, model2])
bingham_bundle = distributed_models.SD2BinghamDistributed(models=[model1, model2])

# two equivalent double-distributed models
gamma_watson_bundle = distributed_models.PD1GammaDistributed(
    models=[watson_bundle], parameter_name=parameter_name)
watson_gamma_bundle = distributed_models.SD1WatsonDistributed(models=[gamma_bundle])
```

Snippet 4 | Instantiation of anisotropic and isotropic tissue response models, given some data segmentation. Notice that we can produce an isotropic or anisotropic TR-model regardless of the data, and that we also recover the mean signal amplitude S_0 of the input data, which we can use later when setting up MC-models in Snippet 5.

```python
from dmipy.signal_models import tissue_response_models
# We have some dataset with some binary segmentation mask
data_seg = data[segmentation_mask]  # of shape (N_segmented_voxels, N_DWI)
S0_response, TR1 = tissue_response_models.IsotropicTissueResponseModel(dmipy_scheme, data_seg)
S0_response, TR2 = tissue_response_models.AnisotropicTissueResponseModel(dmipy_scheme, data_seg)
```

TR1: The isotropic tissue response model is given by

$$S_0 C_{\text{TR1}}(\mathbf{A}_s, \mathbf{r}_{s,l=0}) = \frac{1}{N_{\text{vox}}}\sum_{i=1}^{N_{\text{vox}}} \text{RHT}\left[S(\mathbf{x}_i, \mathbf{A}_s)\right] \quad \text{for} \quad \mathbf{x}_i \in \mathbf{X}_{\text{iso}}. \tag{8}$$

Notice that we recover the tissue-specific baseline intensity $S_{0,\text{iso}}$, and that only the $l = 0$ coefficients are needed to represent isotropic profiles.

TR2: The anisotropic tissue response model is given as

$$S_0 C_{\text{TR2}}(\mathbf{A}_s, \mathbf{r}_{s,l}, \boldsymbol{\mu}) = \frac{1}{N_{\text{vox}}}\sum_{i=1}^{N_{\text{vox}}} \text{RHT}\left[S(\mathbf{x}_i, \mathbf{R}_i\mathbf{A}_s)\right] \quad \text{for} \quad \mathbf{x}_i \in \mathbf{X}_{\text{aniso}}. \tag{9}$$

where we note that the TR2 model has an orientation $\boldsymbol{\mu}$ and gradient orientations $\mathbf{n}_s$ are rotated with rotation matrix $\mathbf{R}$ such that the signal orientations (according to a DTI tensor) is pointing along the z-axis.

2.7. Multi-Compartment Modeling Variants

In section 2.3 we introduced the definition of standard MC-modeling. Dmipy, however, also implements alternative MC-modeling frameworks. In this section, we introduce the Multi-Compartment Spherical Mean (MC-SM) variant in section 2.7.1 and the Multi-Compartment Spherical Harmonics (MC-SH) variant in section 2.7.2.

2.7.1. Multi-Compartment Spherical Mean Model (MC-SM)

To estimate properties of the tissue microstructure that are independent of axon orientation dispersion and crossings, Kaden et al. (2015) noticed that the shell-wise spherical mean of the measured signal is invariant to these effects. Taking the spherical convolution in Equation (7), it is straightforward to show that the spherical mean of a spherical convolved convolution kernel is equal to the spherical mean of the kernel itself, by taking advantage of the fact that $\int_{\mathbb{S}^2} F(\mathbf{n};\boldsymbol{\Omega})d\mathbf{n} \equiv 1$. Recalling section 2.4, we can simply use the spherical mean representation of a compartment model in Equation (3) to estimate the parameters for the generalized Multi-Compartment Spherical Mean (MC-SM) model as

$$\tilde{\mathbf{p}}^*(\mathbf{x}) = \text{argmin}_{\tilde{\mathbf{p}}} \int \left[E^{\text{SM}}(\mathbf{x}, \mathbf{A}_s) - \hat{E}^{\text{MC-SM}}(\mathbf{A}_s, \tilde{\mathbf{p}})\right]^2 d\mathbf{A}_s, \tag{10}$$

$$\text{with } \hat{E}^{\text{MC-SM}}(\mathbf{A}_s, \tilde{\mathbf{p}}) = \sum_{i}^{N} f_i C_i^{\text{SM}}(\mathbf{A}_s, \tilde{\mathbf{p}}_i), \tag{11}$$

where $E^{\text{SM}}(\mathbf{x}, \mathbf{A}_s)$ is the spherical mean of the measured signal, defined on shell-wise acquisition scheme $\mathbf{A}_s$. Here $\tilde{\mathbf{p}}$ is again any model's parameters except its orientation $\boldsymbol{\mu}$, which is no longer relevant after taking the spherical mean. Using this MC-SM representation, any "standard" MC-model can be framed as a spherical mean model, enabling the creation of known models such as MC-MDI (Kaden et al., 2016), or

Snippet 5 | One-line instantiation of MultiCompartmentModel variants.

```
from dmipy.core.modeling_framework import(
  MultiCompartmentModel,
  MultiCompartmentSphericalMeanModel,
  MultiCompartmentSphericalHarmonicsModel)
models = [model_1, model_2]  # model_1 is anisotropic and model_2 isotropic
S0_responses = [response_1, response_2]  # optional tissue S0 responses
mc = MultiCompartmentModel(models, responses)
mcsm = MultiCompartmentSphericalMeanModel(models, responses)
mcsh = MultiCompartmentSphericalHarmonicsModel(models, responses, sh_order=8)
mc.parameter_cardinality
>> OrderedDict([('model_1_param_1', 1),   # e.g., diffusivity, diameter
>>              ('model_1_param_2', 2),   # orientation theta, phi
>>              ('model_2_param_1', 1),   # e.g., diffusivity, diameter
>>              ('partial_volume_1', 1),  # linear volume fraction 1
>>              ('partial_volume_2', 1)]) # linear volume fraction 1
mcsm.parameter_cardinality
>> OrderedDict([('model_1_param_1', 1),   # e.g., diffusivity, diameter
>>              ('model_2_param_1', 1),   # e.g., diffusivity, diameter
>>              ('partial_volume_1', 1),  # linear volume fraction 1
>>              ('partial_volume_2', 1)]) # linear volume fraction 1
mcsh.parameter_cardinality
>> OrderedDict([('model_1_param_1', 1),   # e.g., diffusivity, diameter
>>              ('model_2_param_1', 1),   # e.g., diffusivity, diameter
>>              ('partial_volume_1', 1),  # linear volume fraction 1
>>              ('partial_volume_2', 1),  # linear volume fraction 1
>>              ('sh_coeff', 45)])        # sh coefficients of model_1
```

allowing exploration of new effects like an MC-SM version of ActiveAx (Pizzolato et al., 2018).

2.7.2. Multi-Compartment Spherical Harmonics Model (MC-SH)

Using spherical harmonics (SH), it is possible to represent a spherical distribution of models non-parametrically if we know beforehand what are the convolution kernels for the included tissue types. Such an MC-SH model can again be written as a generalization of Equation (3) such that

$$\mathbf{c}^*(\mathbf{x}) = \text{argmin}_{\mathbf{c}} \int \left[E(\mathbf{x}, \mathbf{A}_s) - \hat{E}^{\text{MC-SH}}(\mathbf{A}, \mathbf{c}|\tilde{\mathbf{p}})\right]^2 d\mathbf{A}$$

$$\text{s.t. } \mathrm{F}_i(\mathbf{n}; \mathbf{c}) \geq 0, \qquad (12)$$

$$\text{with } \hat{E}^{\text{MC-SH}}(\mathbf{A}_s, \mathbf{c}|\tilde{\mathbf{p}}) = \left[\sum_{i=1}^{N} \mathrm{F}_i(\mathbf{c}_i) *_{\mathbb{S}^2} C_i^K(\mathbf{A}_s|\tilde{\mathbf{p}}_i)\right](\mathbf{n}), \qquad (13)$$

where $\mathrm{F}(\mathbf{n}; \mathbf{c}) : \mathbb{S}^2 \rightarrow [-\infty, \infty]$ represents a non-parametric spherical distribution given in SH coefficients $\mathbf{c}$, and where $(\cdot|\tilde{\mathbf{p}})$ means that kernel parameters $\tilde{\mathbf{p}}$ are known and fixed. Notice that we must impose $\mathrm{F}(\mathbf{n}; \mathbf{c}) \geq 0$ as the SH distribution is not positive definite by nature. In MC-SH models, the volume fractions f_i are included in $\mathbf{c}_i$, and a unity constraint can be imposed if required. Note, however, that for practical reasons only one anisotropic convolution kernel may be present in Equation (13), unless their volume fractions f_i are fixed (as in Snippet 12).

2.7.3. MC-Model Initialization

We instantiate all three MC-models in Snippet 5. Let model_1 have one orientation and one other parameter with cardinality 1, and model_2 be isotropic with only one parameter. Inspecting the MC-model parameters, notice that they are simply an aggregation of the input parameters with the addition of volume fractions, and the inclusion of S0-responses is handled internally. To avoid duplicate parameter names, each parameter name is prepended with the input model name and its enumeration. The MC-SM model is only different from MC in that the orientation parameter of model_1 is removed, as orientations have no meaning in the spherical mean signal. For the MC-SH model, the orientation is removed as the input models are used as convolution kernels, and the SH-coefficients for the given spherical harmonics order are added as parameters. To simulate the signal for a single voxel the procedure is the same as for the CompartmentModel in Snippet 2. Simulating the signal for multiple voxels is a generalization of the single-voxel simulation, which can be handled by the mc_model.simulate_signal function.

2.8. Parameter Fixing, Linking, and Optimization Parameters

After an MC-model is instantiated, the last step before the potential model fitting is the imposition of linked or fixed parameters or the addition of optimization parameters:

- Parameter linking is used to impose certain mathematical relationships between otherwise independent parameters. Examples are setting parameters equal to each other (e.g.,

Snippet 6 | One-liners to setting fixed or linked parameters to instantiated distributed model or MC-models.

```
# tortuous parameters can be set at distributed model level
distributed_model.set_tortuous_parameter(
    tortuous_diffusivity_parameter_name,
    volume_fraction_name,
    parallel_diffusivity_parameter_name)

# fixing parameters for optimization
mc_model.set_fixed_parameter(parameter_name, value)
mc_model.set_initial_guess_parameter(parameter_name, value)

# other linked parameters set at MC level
mc_model.set_equal_parameter(parameter_name_1, parameter_name_2)
mc_model.set_fractional_parameter(
    fractional_parameter_name,
    target_parameter_name)
```

telling two models to have the same orientation), imposing a tortuosity constraint (e.g., Szafer et al., 1995), or imposing that a parameter must be smaller or equal to another parameter (e.g., $\lambda_\perp \leq \lambda_\parallel$ for a Zeppelin);

- Parameter fixing is used to enable fixing of known global diffusion properties (e.g., CSF diffusivity), or to do parameter cascading, which is to fix voxel-wise varying parameters for a certain dataset, which have been optimized from a previous model;
- Optimization parameters are used to rescale a model parameter to a rescaled version that is more appropriate to the optimization problem, like using the Orientation Dispersion Index (ODI) in lieu of concentration parameter for a Watson distribution (Zhang et al., 2012).

In Snippet 6 we provide an example of how fixed and linked parameters can be included in instantiated distributed models and MC-models. We note that adding a tortuosity constraint can only be done at the distributed model level, and not the MC-modeling level because tortuosity turns the affected volume fraction into a non-linear parameter. At the MC-modeling level, we enforce that all volume fractions remain linear and independent, which is needed for some optimization algorithms like MIX (Farooq et al., 2016) or AMICO (Daducci et al., 2015).

2.9. Generalized Optimization Strategies

Dmipy's generalized modeling approach also extends to implementations of standard optimization algorithms. We describe the fitting of MC and MC-SM models in section 2.9.1, and the fitting of MC-SH models in section 2.9.2. Finally, we discuss the secondary fitting of the signal (not the signal attenuation) for any MC-model variant in section 2.9.3.

Regardless of the MC-model variant, choosing the optimization algorithm is straightforward, as we show in Snippet 7. It suffices to just set the optimization algorithm and potentially adjust the algorithm's parameters before fitting the data. Notice that only at the fit command does the acquisition scheme come into play—before this point the model is data independent. Fitting the data returns a fitted model representation, from which the fitted parameters can be recovered.

2.9.1. Fitting MC and MC-SM Models

From the perspective of optimization, MC and MC-SM models, given in Equations (3) and (10), can be treated equally as non-convex optimization problems. Currently, we implement two optimization algorithms:

Brute2Fine: The standard "brute2fine" optimizer first uses brute force optimization by sampling all optimized parameters **p** on an equispaced grid of *Ns* samples between each parameter's minimum and maximum optimization bounds. Obtaining the best fitting starting position based on Equations (3) or (10), we then use the constrained optimization algorithms L-BFGS-B (Byrd et al., 1995) of the open-source scipy package (Jones et al., 2001) to obtain the locally best fitting parameters $\mathbf{p}^*$. The unity constraint on f_i is enforced by nesting volume fraction following Zhang et al. (2012).

MIX: The recent Microstructure in Crossings (MIX) algorithm (Farooq et al., 2016) is specially designed to return accurate results for highly complicated MC-models. Instead of using brute force, it uses a trick to separate the linear from non-linear parameters and uses differential evolution (Storn and Price, 1997) to find the globally best-fitting parameters. It is accurate at the cost of being slow, so we recommend "brute2fine" for simpler models.

2.9.2. MC-SH Fitting

Fitting MC-SH models is different from MC or MC-SM models in that the problem can be cast in a convex optimization problem. In literature, the spherical harmonics distribution is often estimated using methods like Constrained Spherical Deconvolution (CSD) (Tournier et al., 2007; Jeurissen et al., 2014), although many variations of this approach have been proposed (Canales-Rodríguez et al., 2019). Currently, we have implemented two algorithms:

Tournier07: This is the classical optimization algorithm for CSD following Tournier et al. (2007). It uses the Tikhonov regularization term to penalize negative $F(\mathbf{n};\mathbf{c})$, but not explicitly forbid it. It is a fast implementation

Snippet 7 | Generalized Parameter Optimization.

```python
# same optimization choices for MC and MC-SM
mc_model.set_lbfgsb_optimizer(optimizer_parameters)
mc_model.set_mix_optimizer(optimizer_parameters)

# optimization options for MC-SH
mcsh_model.set_tournier07_optimizer(optimizer_parameters)
mcsh_model.set_cvxpy_optimizer(optimizer_parameters)

# data fitting is then always the same
fitted_model = mc_model.fit(acquisition_scheme, data)
fitted_parameters = fitted_model.fitted_parameters
fods = fitted_model.fods()
peaks = fitted_model.peaks()
mse = fitted_model.mse()
```

only based on Numpy (Oliphant, 2006) but cannot handle multiple kernels, unless the volume fractions are fixed.

CVXPY: This implements the explicit multi-compartment CSD algorithm following Jeurissen et al. (2014) using the CVXPY package (Diamond and Boyd, 2016; Agrawal et al., 2018). It features a hard non-negativity constraint on the FOD, and can optionally enforce a unity constraint on the volume fractions. We note that enforcing the unity constraint is not recommended by Dell'Acqua and Tournier (2018) when tissue S_0 responses are given.

2.9.3. Secondary Multi-Tissue Optimization: Fitting the Signal

So far, we have been considering fitting the signal attenuation only, ignoring the information that is present in the signal amplitude (see Equation 1). Recent works (Jeurissen et al., 2014) show that including knowledge of the tissue-specific S_0 response, and fitting the signal instead of the signal attenuation, can provide a better estimate of the linear volume fractions. We observe that this so-called Multi-Tissue modeling can be seen as a secondary convex optimization to correct the volume fractions after all non-linear parameters have been estimated. The optimization is given

$$\mathbf{f}^*(\mathbf{x}) = \text{argmin}_{\mathbf{f}} \int \left[S(\mathbf{x},\mathbf{A}) - \sum_i^N S_{0,i} f_i C_i(\mathbf{A}|\mathbf{p}_i^*) \right]^2 d\mathbf{A}, \tag{14}$$

$$= \text{argmin}_{\mathbf{f}} \| S(\mathbf{x},\mathbf{A}) - \mathbf{B}\mathbf{f} \|^2 \tag{15}$$

with $\mathbf{B} = [S_{0,i} C_i(\mathbf{A}|\mathbf{p}_i^*)] \in \mathbb{R}^{N_{DWI} \times N_{models}}$ the model-wise evaluated parts of the MC-models at $\mathbf{A}$, multiplied by the tissue-specific S_0 intensity. Signal attenuation-based (shape only) MC-modeling—as we've been doing so far—implicitly assumes equal $S_{0,i} = S_0$. This means that MC-models that model tissues with different S_0 (e.g., CSF, white and gray matter) will always have biased volume fractions unless these effects are taken into account. In Dmipy, any MC-model can be corrected for this, as we showed in Snippet 5.

2.10. Data Set Descriptions

2.10.1. IVIM Data

For our first demonstration, we will use a freely available dataset with an IVIM acquisition, which is downloadable from the dipy examples page (Garyfallidis et al., 2014). It is a small dataset with only 21 DWIs at b-values spread between $0 - 1{,}000\, s/mm^2$, including a single b_0 measurement. Information about diffusion times and TE are unknown for this data, but these will not be needed to fit the 2-compartment Gaussian IVIM model.

2.10.2. Spatio-Temporal Cat Spinal Cord Data

To illustrate MC-models that are aimed at estimating axon diameter, we use a recent dataset where both AxCaliber and multi-shell diffusion MRI acquisitions have been registered to one axial slice of cat spinal cord (Duval et al., 2016). For each voxel, the mean axon diameter, restricted volume fraction and myelin volume fraction are known from histological measurements. The dMRI data is 64 × 64 voxels with resolution $0.16 \times 0.16 \times 0.16\,\text{mm}^3$. One AxCaliber acquisition was acquired (perpendicular to the axon axis) with parameters δ=3/8/8/8 ms, Δ = 7/12/25/40 ms, G = [0... 849] mT/m (199 increments) and TE minimized (36–62 ms). The data was TE-normalized by dividing the data for every TE by the G = 0 signal. The multi-shell acquisition was acquired with parameters δ = 3 ms, Δ = 30 ms, 4 shells with b-value={40, 189, 1,680, 6,720} s/mm^2, TE = 47 ms, with a total of 796 diffusion-weighted images.

2.10.3. Multi-Shell WU-Minn Human Connectome Project Data

To illustrate MC-models that require only multi-shell data, we use the WU-Minn Human Connectome Project data (Feinberg et al., 2010; Moeller et al., 2010; Setsompop et al., 2012; Xu et al., 2012; Glasser et al., 2013; Van Essen et al., 2013). In this dataset, the diffusion directions were obtained such that every subset of the first M directions is still isotropic (Caruyer et al., 2013). The data was sampled on 3 shells with b-values {0, 1,000, 2,000, 3,000} s/mm^2, with {14, 90, 90, 90} directions, respectively. The diffusion time and pulse separation time in this data are

$\delta/\Delta = 10.6/43.1$ ms with $2 \times 2 \times 2$ mm resolution and $TE/TR = 89.5/5, 520$ ms.

3. RESULTS: DMIPY MC-MODEL IMPLEMENTATIONS

In this section, we demonstrate the power of Dmipy's modular MC-modeling design. We start by implementing, fitting and showing the results of many well-known MC-models known from the literature. The models we will illustrate are listed in **Table 3**, where we also indicate each model's composition in terms of biophysical models, the number of lines in Dmipy code it takes to implement, and the primary reference. Note that it is not the point here to deeply study each model's behavior, but to demonstrate how simple it is to create and fit any of them. Deeper, more detailed studies of each model are readily available as examples at the Dmipy Github page at https://github.com/AthenaEPI/dmipy. For each model, we will also denote its schematic, mathematical representation in terms of parameter dependencies, where we will use the "|" sign to separate estimated from fixed parameters during the optimization.

TABLE 3 | Overview and composition of implemented MC-models and the number of lines that is needed to implement them from scratch using Dmipy, along with their primary reference.

Model acronym	Model composition	No. of lines	Primary references
IVIM (Snip. 8)	2×G1	6	Le Bihan et al., 1988
AxCaliber (Snip. 9)	G1+PD1*C4	8	Assaf et al., 2008
Ball and Sticks (Snip. 10)	G1+N×C1	6	Behrens et al., 2003
NODDI(-x) (Snip. 11)	G1+SD3*(C1+G2)	12	Zhang et al., 2012
Bingham-NODDI (Snip. 11)	G1+SD2*(C1+G2)	12	Tariq et al., 2016
MC-MDI-CSD (Snip. 12)	SD1*(C1+G2)	12	Kaden et al., 2016
MT-CSD (Snip. 13)	2×TR1+TR2	6	Jeurissen et al., 2014
SS3T-CSD (Snip. 14)	2×TR1+TR2	19	Dhollander and Connelly, 2016
ActiveAx (**Supplementary Material**)	G1+G2+S1+C4	13	Alexander et al., 2010
VERDICT (**Supplementary Material**)	S4+G1+C1	8	Panagiotaki et al., 2014
SMT (**Supplementary Material**)	G2	5	Kaden et al., 2015
SMT-NODDI (**Supplementary Material**)	G1+SD3*(C1+G2)	21	Cabeen et al., 2019
CSD (**Supplementary Material**)	TR2	6	Tournier et al., 2007

*The MC-models above the midline are presented in the results section of this work, while the implementations of those below the midline are provided in the **Supplementary Material**.*

3.1. IVIM

Intra-voxel incoherent motion (IVIM) is one of the first MC-models used in dMRI (Le Bihan et al., 1988). It uses a 2-compartment model that separates diffusion signal contributions originating from blood flow and Brownian diffusion. The model consists of 2 Ball compartments, each fitting the volume fractions and diffusivities of the blood flow and diffusion, respectively. Changes in e.g., blood volume fraction has been linked to many pathologies such as the vasculature in tumor tissue (Le Bihan, 2017). We represent IVIM in terms of its parameters and the meaning that is assigned to the biophysical models it is composed of, as

$$MC_{\text{IVIM}} = \underbrace{f_{\text{blood}}\overbrace{\text{G1}(\cdot|\lambda_{\text{Blood}})}^{\text{Ball}}}_{\text{Blood}} + \underbrace{f_{\text{Diffusion}}\overbrace{\text{G1}(\lambda_{\text{Diffusion}})}^{\text{Ball}}}_{\text{Diffusion}}. \tag{16}$$

The implementation of Equation (16) is given in Snippet 8. Following recommendations by Gurney-Champion et al. (2016) and Park et al. (2017), we can adjust IVIM's optimization bounds of the diffusion signal diffusivity between $[0.5 - 6] \times 10^{-9}\, m^2/s$, and the blood flow diffusivity between $[6 - 20] \times 10^{-9}\, m^2/s$. Following Gurney-Champion et al. (2018), we can also fix $\lambda_{\text{Blood}} = 7 \times 10^{-9} m^2/s$ to obtain similar results for the remaining parameters. We fit the fixed IVIM model to the IVIM data described in section 2.10.1, and present the parameter maps of an axial brain slice in **Figure 4**. Notice that signal contributions due to blood flow are significant near the ventricles and the sulci. For a voxel where blood flow and diffusion signal contributions were found to be approximately equal, we also demonstrate the fitted signals of the separate Ball compartments. Notice that blood flow contribution is represented by the fast-decaying, green-dashed line, and the diffusion contribution by the slow-decaying, orange-dashed line.

3.2. AxCaliber

The estimation of axon diameter from the dMRI signal has been one of the main focuses of dMRI-based microstructure imaging. To this end, we implement the AxCaliber model (Assaf et al., 2008), which models the axon diameter distribution as a set of cylinders with Gamma-distributed diameters. It is given as a two-compartment model as

$$MC_{\text{AxCaliber}} = \underbrace{(1 - f_{\text{intra}})\overbrace{\text{G1}(\lambda_{\text{h}})}^{\text{Ball}}}_{\text{Extra-Axonal}} + \underbrace{f_{\text{intra}}\overbrace{\text{PD1}(\alpha, \beta|\boldsymbol{\mu}, \lambda_{\parallel})}^{\text{Gamma Distributed Cylinders}}}_{\text{Intra-Axonal}}, \tag{17}$$

where the Ball captures all extra-axonal signal contributions. We note that the cylinder's perpendicular, intra-cylindrical diffusivity is fixed, but it's value is rarely important as for clinically relevant diffusion times the intra-axonal diffusion is restricted anyway (Assaf and Pasternak, 2008). We implement the AxCaliber model in Snippet 9. We use the MIX algorithm to fit AxCaliber to the cat spinal cord data described in section 2.10.2. We chose this data because it has extremely large gradients,

Snippet 8 | Basic Dmipy implementation of classic IVIM model in 6 lines of code. We set optimization ranges according to suggestions by Park et al. (2017) and Gurney-Champion et al. (2018).

```python
from dmipy.core.modeling_framework import MultiCompartmentModel
from dmipy.signal_models import gaussian_models
ball = gaussian_models.G1Ball()
ivim = MultiCompartmentModel(models=[ball, ball])
ivim.set_parameter_optimization_bounds('G1Ball_1_lambda_iso', [0.5e-9, 6e-9])
ivim.set_fixed_parameter('G1Ball_2_lambda_iso', 7e-9)
ivim.parameter_names
>> ['G1Ball_1_lambda_iso',
>>  'G1Ball_2_lambda_iso',
>>  'partial_volume_0',
>>  'partial_volume_1']
```

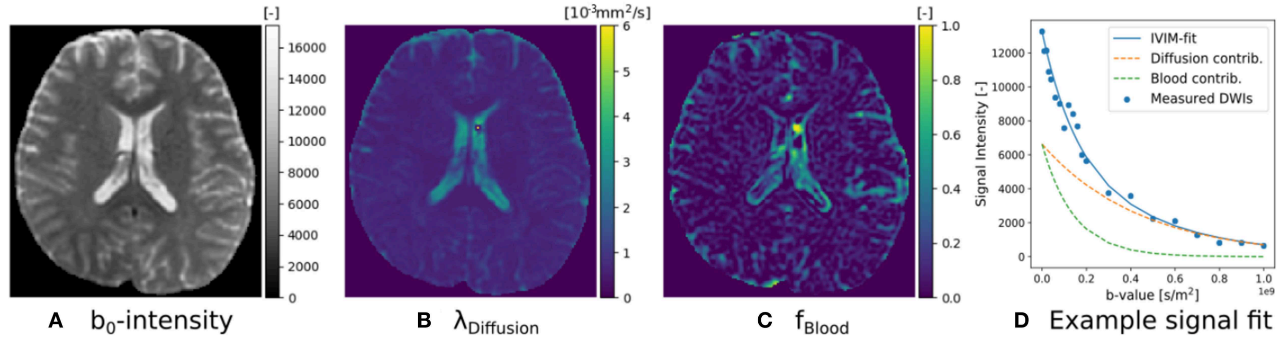


FIGURE 4 | IVIM parameter maps for S_0 **(A)**, $\lambda_{\text{Diffusion}}$ **(B)**, and f_{blood} **(C)**, following Equation (16). Notice that the parameter maps are articulate, and the estimated blood volume fraction is significant near the corpus callosum and the sulci. In **(D)** we show an example IVIM fit in a corpus callosum voxel, where we illustrate that the total fitted signal is just the sum of the diffusion and blood signal contributions.

which are needed to get enough signal contrast for axon diameter estimation (Drobnjak et al., 2016). As we know the axons are pointing in-plane for the spinal cord, we fixed the orientation of the cylinders to the z-axis, and the parallel diffusivity $\lambda_\| = 1.7 \times 10^{-9}[m^2/s]$. We show the resulting parameter maps in **Figure 5A**. As the results show, the mean axon diameter, estimated as $2\alpha\beta$ of the Gamma distribution, falls within a slightly larger range 3–9 μm than what is known from histology 1–5 μm. This is similar to what Duval et al. (2016) previously found when studying this data with AxCaliber. Even so, even though the mean axon diameter may be similar, we do obtain different diameter distributions as illustrate in **Figure 5B**.

3.3. Ball and Stick and NODDI Variants

Another main focus of the field has been the estimation of axon orientation dispersion, which is described as the concentration with which the orientation of individual axons is centered around the main bundle axis. In some respects, this tissue property is easier to estimate from the data because it affects the signal at near-clinically feasible gradient strengths (Zhang et al., 2012). The main hypothesis of axon dispersion-based MC-models is that at lower b-values the axon diameter has no effect on the signal, so is assumed to be negligible. The Cylinder model we used in section 3.2 is therefore replaced with a Stick model.

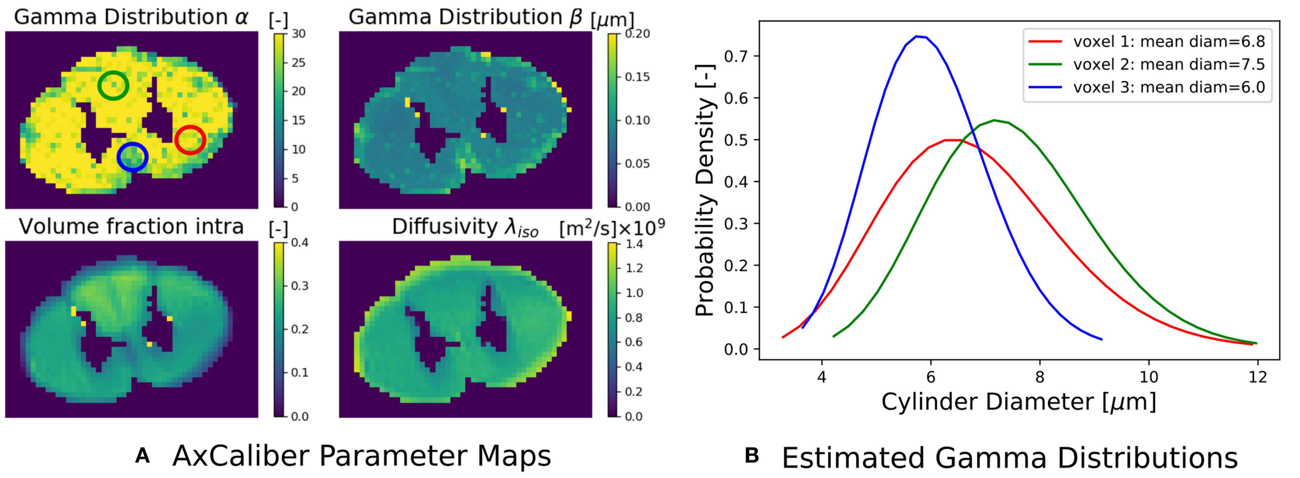
Representing the intra-axonal signal as a Stick, and anything else as Ball, the iconic Ball and Stick (BAS) MC-model was one of the first to estimate the white matter orientation and signal contribution from the dMRI signal (Behrens et al., 2003). Axon dispersion was then modeled by adding a parametric distribution on the orientation of the Stick (Kaden et al., 2007; Sotiropoulos et al., 2013). In particular, using a Watson distribution, in combination with a tortuous Zeppelin to capture the extra-axonal signal, is called the NODDI model (Zhang et al., 2012). Adding multiple dispersed Stick and Zeppelin bundles to represent crossing tissue configurations is called NODDI in crossings (NODDIx) (Farooq et al., 2016), and replacing the Watson with a Bingham distribution is called Bingham-NODDI (Tariq et al., 2016). These MC-model variants are represented as

$$MC^{\text{Ball and}}_{\text{Sticks}} = \underbrace{f_h\,\overbrace{G1(\lambda_{\text{iso}})}^{\text{Ball}}}_{\text{Extra-Axonal}} + \sum_{i=1}^{N}\underbrace{f_{i,r}\,\overbrace{C1(\boldsymbol{\mu}_i|\lambda_\|)}^{\text{Stick}}}_{\text{Intra-Axonal}}, \tag{18}$$

$$MC^{\text{NODDIx}}_{\text{Watson}} = \underbrace{f_{\text{CSF}}\,\overbrace{G1(\cdot|\lambda_{\text{CSF}})}^{\text{Ball}}}_{\text{CSF}} + \sum_{i}^{N}\overbrace{SD1(\kappa_i,\boldsymbol{\mu}_i)}^{\text{Watson}} *_{\mathbb{S}^2}$$

Snippet 9 | Dmipy implementation of AxCaliber model in 8 lines of code.

```python
from dmipy.core.modeling_framework import MultiCompartmentModel
from dmipy.signal_models import gaussian_models, cylinder_models, distributed_models
ball = gaussian_models.G1Ball()
cylinder = cylinder_models.C4CylinderGaussianPhaseApproximation()
gamma_cylinder = distribute_models.PD1GammaDistributed(models=[cylinder])
axcaliber = MultiCompartmentModel(models=[ball, gamma_cylinder])
axcaliber_gamma.set_fixed_parameter(
    'PD1GammaDistributed_1_C4CylinderGaussianPhaseApproximation_1_lambda_par', 1.7e-9)
axcaliber_gamma.set_fixed_parameter(
    'PD1GammaDistributed_1_C4CylinderGaussianPhaseApproximation_1_mu', [0, 0])
```



A AxCaliber Parameter Maps

B Estimated Gamma Distributions

FIGURE 5 | (A) The estimated AxCaliber parameter maps. In the map for α, we circled three areas with somewhat different values for α, β, which we use to generate the estimated axon diameter distribution in **(B)**. Notice that while the mean axon diameter of these distributions is similar, the distributions themselves may be quite different.

$$\left[\underbrace{f_{h,i}\,\overbrace{G2(\cdot|\lambda_\perp^{\text{tort}},\lambda_\|)}^{\text{Zeppelin}}}_{\text{Hindered Extra-Axonal}} + \underbrace{f_{r,i}\,\overbrace{C1(\cdot|\lambda_\|)}^{\text{Stick}}}_{\text{Intra-Axonal}}\right], \tag{19}$$

$$MC^{\text{NODDI}}_{\text{Bingham}} = \underbrace{f_{\text{CSF}}\,\overbrace{G1(\cdot|\lambda_{\text{CSF}})}^{\text{Ball}}}_{\text{CSF}} + \overbrace{SD2(\kappa_1,\kappa_2,\boldsymbol{\mu},\psi)}^{\text{Bingham}} *_{\mathbb{S}^2}$$

$$\left[\underbrace{f_h\,\overbrace{G2(\cdot|\lambda_\perp^{\text{tort}},\lambda_\|)}^{\text{Zeppelin}}}_{\text{Hindered Extra-Axonal}} + \underbrace{f_r\,\overbrace{C1(\cdot|\lambda_\|)}^{\text{Stick}}}_{\text{Intra-Axonal}}\right]. \tag{20}$$

It is clear that in terms of MC-model configuration the models in Equations (18–20) are minor variations of each other. We implement BAS in Snippet 10 and the NODDI variants in Snippet 11, and fit them to a subsection of a coronal slice of the HCP data we described in section 2.10.3. We show the main orientation or parametric Fiber Orientation Distributions (FODs) for all models in **Figure 6**. In all cases, the total signal contribution of the non-Ball compartments is used as background.

3.4. Spherical Mean-Based Constrained Spherical Deconvolution

As we explained in section 2.7.1, spherical mean-based MC-models can estimate dispersion-free tissue parameters by fitting the spherical mean of the measured signal with the spherical mean of the MC-model. To delineate features of the intra- and extra-axonal in the tissue micro-environment, Kaden et al. (2016) proposed the Multi-Compartment Microscopic Diffusion Imaging (MC-MDI) model, composed of the spherical mean of the classic Stick and Zeppelin model with tortuosity constraint. As a cross-modeling framework example, we can then use the voxel-wise estimated MC-MDI parameters to define a voxel-wise varying convolution kernel for an MC-SH model of the same model composition. We describe these two models as

$$\text{Step 1}: \quad MC_{\text{MC-MDI}} = \underbrace{f_r\,\overbrace{C1^{SM}(\lambda_\|)}^{\text{Stick}}}_{\text{Intra-Axonal}} + \underbrace{(1-f_r)\,\overbrace{G2^{SM}(\lambda_\||\lambda_\perp^{\text{tort}})}^{\text{Zeppelin}}}_{\text{Hindered Extra-Axonal}}, \tag{21}$$

Snippet 10 | Dmipy implementation of Ball and Stick in 6 lines of code.

```
from dmipy.core.modeling_framework import MultiCompartmentModel
from dmipy.signal_models import gaussian_models, cylinder_models
ball = gaussian_models.G1Ball()
stick = cylinder_models.C1Stick()
ball_and_stick = MultiCompartmentModel(models=[ball, stick])
ball_and_stick.set_fixed_parameter('C1Stick_1_lambda_par', 1.7e-9)
```

Snippet 11 | Dmipy implementation of NODDI or NODDIx with 2 bundles in 12 lines of code. The implementation of Bingham-NODDI is equivalent to Watson-NODDI, only replacing SD1WatsonDistributed with SD2BinghamDistributed.

```
from dmipy.core.modeling_framework import MultiCompartmentModel
from dmipy.signal_models import gaussian_models, cylinder_models, distributed_models
ball = gaussian_models.G1Ball()
zeppelin = gaussian_models.G2Zeppelin()
stick = cylinder_models.C1Stick()
watson_bundle = distributed_models.SD1WatsonDistributed(models=[stick, zeppelin])
watson_bundle.set_tortuous_parameter(
    'G2Zeppelin_1_lambda_perp','C1Stick_1_lambda_par','partial_volume_0')
watson_bundle.set_equal_parameter('G2Zeppelin_1_lambda_par', 'C1Stick_1_lambda_par')
watson_bundle.set_fixed_parameter('G2Zeppelin_1_lambda_par', 1.7e-9)
NODDI_watson = MultiCompartmentModel(models=[ball, watson_bundle])
NODDI_watson.set_fixed_parameter('G1Ball_1_lambda_iso', 3e-9)
NODDI_2x = MultiCompartmentModel(models=[ball, watson_bundle, watson_bundle])
NODDI_2x.set_fixed_parameter('G1Ball_1_lambda_iso', 3e-9)
```

$$\text{Step 2}: \quad MC_{\text{CSD}}^{\text{vox-varying}} = \text{FOD}(\mathbf{c}) *_{\mathbb{S}^2} \left[\underbrace{\overbrace{f_r^* \, C1^K(\cdot|\lambda_\|^*)}^{\text{Stick}}}_{\text{Intra-Axonal}} + \underbrace{(1-f_r^*)\,\overbrace{G2^K(\cdot|\lambda_\|^*,\lambda_\perp^{\text{tort}})}^{\text{Zeppelin}}}_{\text{Hindered Extra-Axonal}} \right].$$

Notice that we use the superscript "*" in step 2 to indicate the fitted parameters of step 1. In the second step, only the spherical harmonics coefficients **c** are estimated, while all the parameters in the convolution kernel are fixed. In fact, using a voxel-varying convolution kernel for CSD-based FOD estimation was the winning approach for the ISMRM 2017 TraCED challenge for reproducible tractography https://my.vanderbilt.edu/ismrmtraced2017/. We fit the 2 steps of our model design to the same coronal slice of the HCP data, and show the FOD field in **Figure 7**. Notice that on the left we show the $f_r, \lambda_\|$ parameters of step 1, in the middle we show the estimated FOD field of step 2, and on the right we show cross-sections of the estimated convolution kernel for three voxels containing different tissue configurations.

3.5. Multi- and Single-Shell, Multi-Tissue Constrained Spherical Deconvolution

Dmipy defines MC-SH models as those estimating the non-parametric fiber distribution of some fixed convolution kernel. Unlike the MC-models in the previous sections, the first implementations of MC-SH models were implemented using tissue response models, which are directly estimated from the data, being representative of specific tissue types like white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF) (Tournier et al., 2004, 2007; Jeurissen et al., 2014). In section 3.5.1 we demonstrate how we recover tissue response models directly from the dMRI data. Then, in section 3.5.2, we use these tissue responses to implement and compare Single-Shell 3-tissue CSD with standard Multi-Shell, 3-tissue CSD.

3.5.1. Unsupervised 3-Tissue Response Model Estimation

Recently, Dhollander et al. (2016) proposed an unsupervised method to estimate the GM, WM, and CSF tissue response function directly from the dMRI data. This method is implemented in Dmipy, and we illustrate the results, obtained from the same coronal slice as before, in **Figure 8**. On the left, we show the gray matter (green), white matter (red), and CSF (blue) segmented voxels, which were found to be the best candidates to represent the three tissue types. In the middle, we show the spherical mean of the three recovered tissue responses *including* the tissue-specific S_0 intensities. On the right, we show the recovered kernel cross-sections without the S_0 intensities. Notice that the CSF S_0 intensity makes the CSF signal contribution much larger than the others at b_0, but it quickly becomes negligible at higher b-values.

3.5.2. Comparing Single and Multi-Shell, Multi-Tissue CSD

Having recovered the tissue response functions, we can now define the standard Multi-Tissue Constrained Spherical

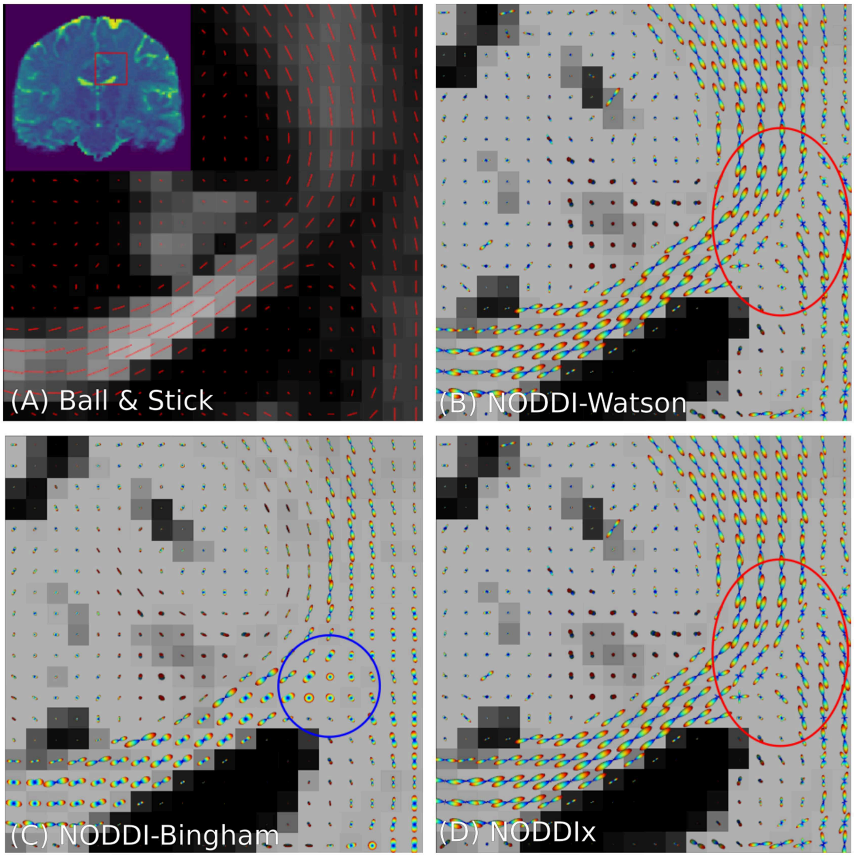


FIGURE 6 | Peak and FOD fields of Ball and Stick **(A)**, NODDI-Watson **(B)**, NODDI-Bingham **(C)**, and NODDIx with two Watson-distributed bundles **(D)** in an area near the centrum semiovale. We expect to find the corpus callosum entering on the left and joining the corona radiata on the right, where we usually find crossing bundle geometries. Models **(A–C)** all appear to correctly reconstruct the orientation of the corpus callosum, but due to their single-bundle design cannot reconstruct crossing structures, instead finding some "average" orientation. It is interesting that NODDI-Bingham, in fact, finds "pancake" FODs to cope with modeling crossing structures (blue circle). NODDIx **(D)**, having two bundles, also accurately reconstructs the corpus callosum despite being over-parameterized for single bundles, and reconstructs some crossings in the centrum semiovale. However, some crossing structures still appear poorly reconstructed (red circle).

Deconvolution (MT-CSD) model (Jeurissen et al., 2014).

$$
\begin{aligned}
S_{\text{MT-CSD}} = {} & \underbrace{S_{0,\text{CSF}} f_{\text{CSF}} \overbrace{\text{TR1}(\cdot)}^{\text{Iso Response}}}_{\text{CSF}} + \underbrace{S_{0,\text{GM}} f_{\text{GM}} \overbrace{\text{TR1}(\cdot)}^{\text{Iso Response}}}_{\text{Gray Matter}} \\
& + \underbrace{S_{0,\text{WM}} f_{\text{WM}} \overbrace{\text{FOD}(\mathbf{c})}^{\text{Fiber Dist.}} *_{\mathbb{S}^2} \overbrace{\text{TR2}_{\text{WM}}(\cdot)}^{\text{Aniso Response}}}_{\text{White Matter}}
\end{aligned}
\tag{22}
$$

Notice that in contrast to the previous models, MT-CSD is *signal-based*, including the tissue-specific S_0 intensities. By not normalizing the b_0 intensity of each model to unity, we can include the signal information in the b_0 image as another "shell" to estimate information from. This is exactly what Dhollander and Connelly (2016) takes advantage of when they proposed so-called Single-Shell 3-Tissue (SS3T) CSD.

In SS3T, it is possible to obtain multi-shell-like parameter maps from "single"-shell data, that resemble those of multi-shell MT-CSD. This is counter-intuitive, as it seems impossible to fit three compartments from only two shells. As we will show, the trick of the SS3T algorithm lies in that it uses an iterative, partial optimization scheme based on parameter cascading, starting from a very specific initialization of the MT-CSD. First, they fix $f_{\text{WM}} = 0$ and optimize only f_{GM} and f_{CSF}. Then, they fix f_{CSF} to the previous solution, and only optimize f_{WM} and f_{GM}. The process is repeated by then fixing f_{WM} to the previous solution and so on, for a *fixed* amount of iterations, where Dhollander and Connelly (2016) used 4. Dmipy implements this meta-algorithm in Snippet 14.

Applying SS3T then to the whole coronal slice of the HCP data, we can visualize the volume fractions as RGB values in **Figure 9**. Notice that as the single-shell data has a higher b-value, the SS3T results more closely approximate the multi-shell MT-CSD reference. On the bottom, we visualize the estimated FODs with the RGB volume fraction values as background. Notice that higher b-value SS3T results seem to resolve the crossing slightly better, but that the biggest difference can be seen in the lower GM volume fractions for lower b-values.

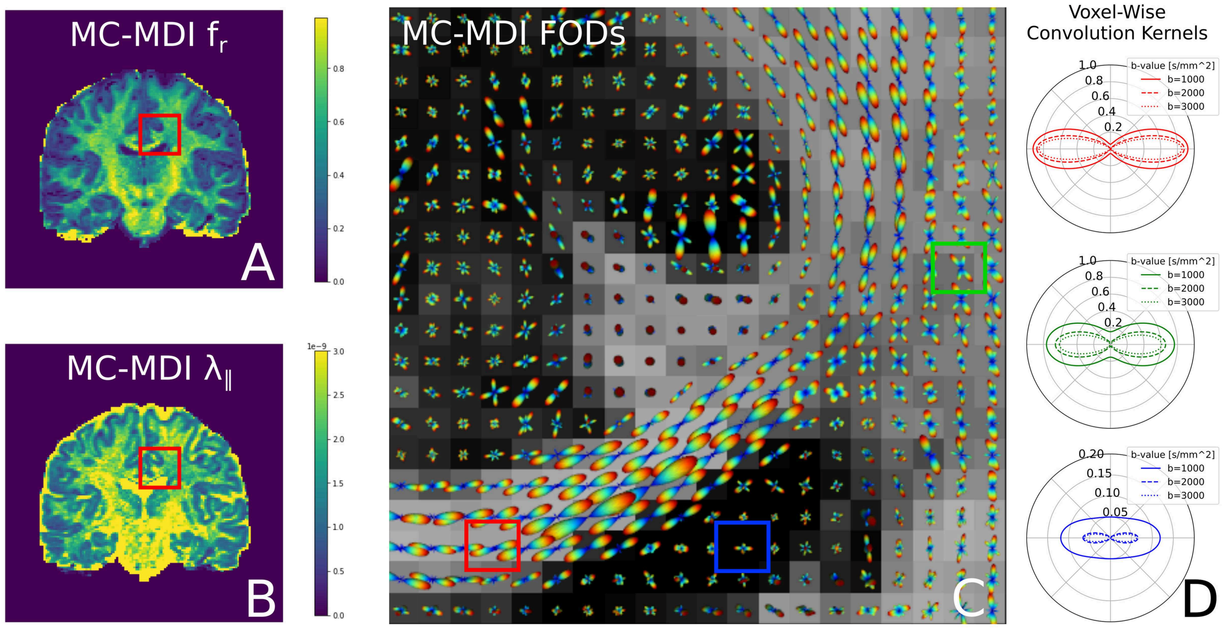


FIGURE 7 | FODs estimated with CSD using varying convolution kernels, estimated voxel-wise using the MC-MDI spherical mean model, see the MC-MDI parameter maps for restricted volume fraction **(A)** and parallel diffusivity **(B)**. In **(C)**, the colored squares highlight three voxels with different tissue compositions; red for a coherent white matter bundle, green for a crossing geometry, and blue for CSF. In **(D)**, we show the corresponding MC-MDI estimated convolution kernels in these voxels for each b-value of the HCP acquisition scheme. Notice that the white matter kernel is more anisotropic than the crossing one, and the CSF kernel is most isotropic and much smaller than the other two. While CSF areas appear more noisy with this voxel-varying kernel approach, it does appear quite effective at resolving crossing structures.

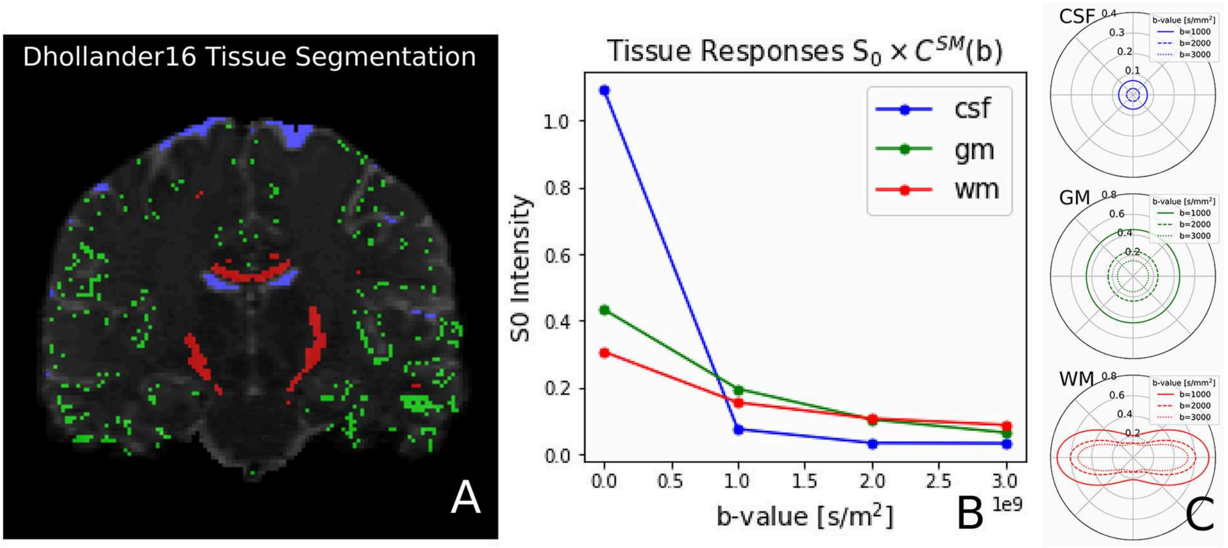


FIGURE 8 | (A) Unsupervised three-tissue segmentation algorithm according to Dhollander et al. (2016); **(B)** The spherical mean of the three recovered tissue responses *including* the tissue-specific S_0. notice that the $S_{[0,CSF]}$ is three times higher than that of GM and WM. **(C)** The convolution kernels (without S_0) of the three tissues. Notice that WM is the only anisotropic one, and the CSF tissue response is much smaller than the other two.

4. DISCUSSION

This work introduces Dmipy, an open-source software solution for generalized dMRI signal modeling and Microstructure Imaging using custom-designed multi-compartment models. The toolbox is freely available at https://github.com/AthenaEPI/dmipy. In this section, we discuss Dmipy's contributions to multi-compartment-based microstructure research.

Snippet 12 | Dmipy implementation of 2-step MC-MDI to voxel-varying CSD in 12 lines of code.

```
from dmipy.core import modeling_framework
from dmipy.signal_models import cylinder_models, gaussian_models
from dmipy.distributions.distribute_models import BundleModel
bundle = BundleModel([stick, zeppelin])
bundle.set_tortuous_parameter('G2Zeppelin_1_lambda_perp',
    'C1Stick_1_lambda_par','partial_volume_0')
bundle.set_equal_parameter('G2Zeppelin_1_lambda_par', 'C1Stick_1_lambda_par')
mcdmi_mod = modeling_framework.MultiCompartmentSphericalMeanModel(models=[bundle])
mcdmi_fit = mcdmi_mod.fit(scheme, data, mask=data[..., 0]>0)
mcmdi_csd_mod = modeling_framework.MultiCompartmentSphericalHarmonicsModel(
    models=[bundle])
for name, value in mcdmi_fit_hcp.fitted_parameters.items():
    mcmdi_csd_mod.set_fixed_parameter(name, value)
mcmdi_csd_fit = mcmdi_csd_mod.fit(scheme, data, mask=data[..., 0]>0)
```

Snippet 13 | Dmipy implementation of unsupervised tissue response estimation and Multi-Tissue CSD in 6 lines of code.

```
from dmipy.tissue_response.three_tissue_response import three_tissue_response_dhollander16
from dmipy.core.modeling_framework import MultiCompartmentSphericalHarmonicsModel
S0_tissue_responses, tissue_response_models, selection_map = three_tissue_response_dhollander16(
  scheme_hcp, data_hcp, wm_algorithm='tournier13',
  wm_N_candidate_voxels=150, gm_perc=0.2, csf_perc=0.4)
mt_csd_mod = MultiCompartmentSphericalHarmonicsModel(tissue_response_models, S0_tissue_responses)
```

4.1. Generalized, On-the-Fly Multi-Compartment Modeling and Signal Generation

Dmipy's key contribution is the way it generalizes dMRI signal modeling regardless of the MC-model composition or acquisition scheme design. In this way, Dmipy can be better seen as a "model generator," i.e., it does not hard-code any model combinations, but can generate any one of them on the fly. By standardizing the interaction between the PGSE acquisition scheme representation and the mathematical representations of any biophysical model, all signal representations become the same from a user point of view. An MC-model can then be seen as nothing more than an aggregated set of the input model parameters, including volume fractions, which is neatly represented in Equation (3). In this way, Dmipy enables the user to have only high-level interaction in choosing which model is appropriate the chosen application and never has to worry about the underlying mathematical implementation.

Throughout this work we have provided the basic commands in Snippets that allows easy interaction with all aspects of MC-modeling. In particular, we can load the acquisition parameters for any PGSE sequence (Snippet 1), generate the dMRI signal for any individual compartment model (Snippet 2), design any MC-model variant and set S_0 responses (Snippet 5), define parameter links (Snippet 6), and fit an MC-model to data (Snippet 7), from which the fitted model parameters and fitting error can be obtained. Using these elementary commands, we can implement and fit many well-known MC-models from literature in around 10 lines of code (see **Table 3**). In section 3, we explicitly provided the Dmipy code and described the estimated model parameters of IVIM (Le Bihan et al., 1988), AxCaliber (Assaf et al., 2008), Ball and Stick (Behrens et al., 2003), various forms of NODDI (Zhang et al., 2012; Farooq et al., 2016; Tariq et al., 2016), spherical-mean-based MT-CSD (Jeurissen et al., 2014) and spherical-harmonics-based MC-MDI (Kaden et al., 2016). For completeness, in the **Supplementary Material**, we also provide the implementations of ActiveAx (Alexander et al., 2010), VERDICT (Panagiotaki et al., 2014), SMT (Kaden et al., 2015), SMT-NODDI (Cabeen et al., 2019), and CSD (Tournier et al., 2007). Extended explorations of these models can also be found at the Dmipy website.

4.2. Multi-Compartment Spherical Mean (MC-SM) Modeling

Dmipy's generalization of MC-modeling also extends to alternative modeling frameworks, including spherical mean modeling (MC-SM) in section 2.7.1. With respect to MC-SM, its key interest is that by taking the shell-wise spherical mean of the signal and the compartment model, the effects of axon dispersion or crossing bundle geometries on the signal is obviated. Kaden et al. (2016) showed using such MC-SM models one could recover a voxel-varying, parametric representation of the undispersed tissue parameters. Dmipy generalizes the spherical mean representation to all three-dimensional compartment models—including those without closed-form like Cylinders and Spheres (section 2.5.2). In previous work, we explored this in the setting of spherical mean-based axon diameter estimation (Pizzolato et al., 2018). See the **Supplementary Material** for the implementation details.

Snippet 14 | Dmipy implementation of Single-Shell Three-Tissue (SS3T) Meta-Algorithm in 19 lines (ignoring commenting lines).

```python
from dmipy.tissue_response.three_tissue_response import three_tissue_response_dhollander16
from dmipy.core.modeling_framework import MultiCompartmentSphericalHarmonicsModel
S0_tissue_responses, tissue_response_models, selection_map = three_tissue_response_dhollander16(
  scheme_hcp, data_hcp, wm_algorithm='tournier13')
N_iterations = 4  # following Dhollander16
fit_args = {'acquisition_scheme': acquisition_scheme,
            'data': data, 'mask': data[..., 0] > 0}  # where b0 has positive value
for it in range(N_iterations):
    # step one: fix WM and fit GM + CSF
    mt_csd_mod = MultiCompartmentSphericalHarmonicsModel(tissue_response_models, S0_tissue_responses)
    if it == 0:
        mt_csd_mod.set_fixed_parameter('partial_volume_0', np.zeros(data.shape[:-1]))
    else:
        mt_csd_mod.set_fixed_parameter(
            'partial_volume_0', mt_csd_fit.fitted_parameters['partial_volume_0'])
    mt_csd_fit = mt_csd_mod.fit(**fit_args)

    # step two: fix CSF and fit WM + GM
    mt_csd_mod = MultiCompartmentSphericalHarmonicsModel(tissue_response_models, S0_tissue_responses)
    mt_csd_mod.set_fixed_parameter(
        'partial_volume_2', mt_csd_fit.fitted_parameters['partial_volume_2'])
    mt_csd_fit = mt_csd_mod.fit(**fit_args)
```

4.3. Multi-Compartment Spherical Harmonics (MC-SH) Modeling

In section 2.7.2, we showed that Dmipy's MC-modeling generalization also extends to models based on spherical convolution by means of spherical harmonics (SH). The key interest in such MC-SH models is that the Fiber Orientation Distribution (FOD) of any fixed convolution kernel can be estimated in a model-free way (Tournier et al., 2004). Dmipy generalizes the interface by which the type and number of convolution kernels can be defined in MC-SH models, allowing for mixes of both parametric and non-parametric compartment models. In this way, it can be seen as a generalized implementation the Multi-Tissue Constrained Spherical Deconvolution (MT-CSD) framework by Jeurissen et al. (2014).

4.4. Generalized Implementation of Tissue Response Models

Tissue Response models are have been used in literature to get an approximation of the shape and amplitude of the signal in specific tissue segmentations—typically white matter, gray matter, and CSD (Tournier et al., 2007; Jeurissen et al., 2014). Different algorithms have been proposed to recover the most appropriate white matter response, either by looking at DTI's FA map (Tournier et al., 2007) or by iterative means (Tournier et al., 2013; Tax et al., 2014). Dmipy implements both these approaches, including the recovery of all three tissue types directly from dMRI data using the unsupervised algorithm by Dhollander et al. (2016) (see **Figure 8**). But, more importantly, Dmipy provides a generalized theory to create anisotropic or isotropic tissue responses, just by providing the data from segmented voxels, in section 2.6.3. In this way, both the tissue response and the tissue-specific S_0 intensity can be recovered for any user-defined segmentation, opening the paths to multi-tissue modeling beyond the three classic WM, GM, and CSF tissues. Moreover, tissue responses can be given to all MC-modeling variants, opening the door to Multi-Tissue modeling for MC-models other than MT-CSD (Jeurissen et al., 2014).

4.5. Modular, Generalized Optimization Algorithms

By unifying the theory of MC, MC-SM, and MC-SH modeling, by extension Dmipy allows generalized implementations of standard or dMRI-specific data fitting algorithms, based on Equations (3, 10, 12). Calling different optimization algorithms can be done using a single line of code (see Snippet 7). For MC and MC-SM models, Dmipy uses Scipy (Jones et al., 2001) to implement the "brute2fine" optimizer, using standard brute-force to find a good starting point, and then using L-BFGS-B (Byrd et al., 1995) to refine the solution to a local minimum. Enforcing the unity constraint on the volume fraction is done by nesting the volume fractions based on works by Zhang et al. (2012). We also implement the recent MIX algorithm by Farooq et al. (2016), taking advantage of Scipy's differential evolution algorithm (Storn and Price, 1997). For MC-SH models, Dmipy uses Numpy (Oliphant, 2006) and CVXPY (Diamond and Boyd, 2016; Agrawal et al., 2018) to implement classic Tikhonov-regularized CSD (Tournier et al., 2007) and Multi-Tissue CSD (Jeurissen et al., 2014). However, it should be noted that Dmipy's optimizers are modular, and thus alternative optimization algorithms can be easily plugged in. We provide details on the implementation of the optimization algorithms in the **Supplementary Material**.

4.6. Enabling Iterative, Cross-Framework Modeling Through Uniform Parameter Definitions

By generalizing the meaning of parameters over MC-modeling frameworks, it follows we can initialize the parameters to be

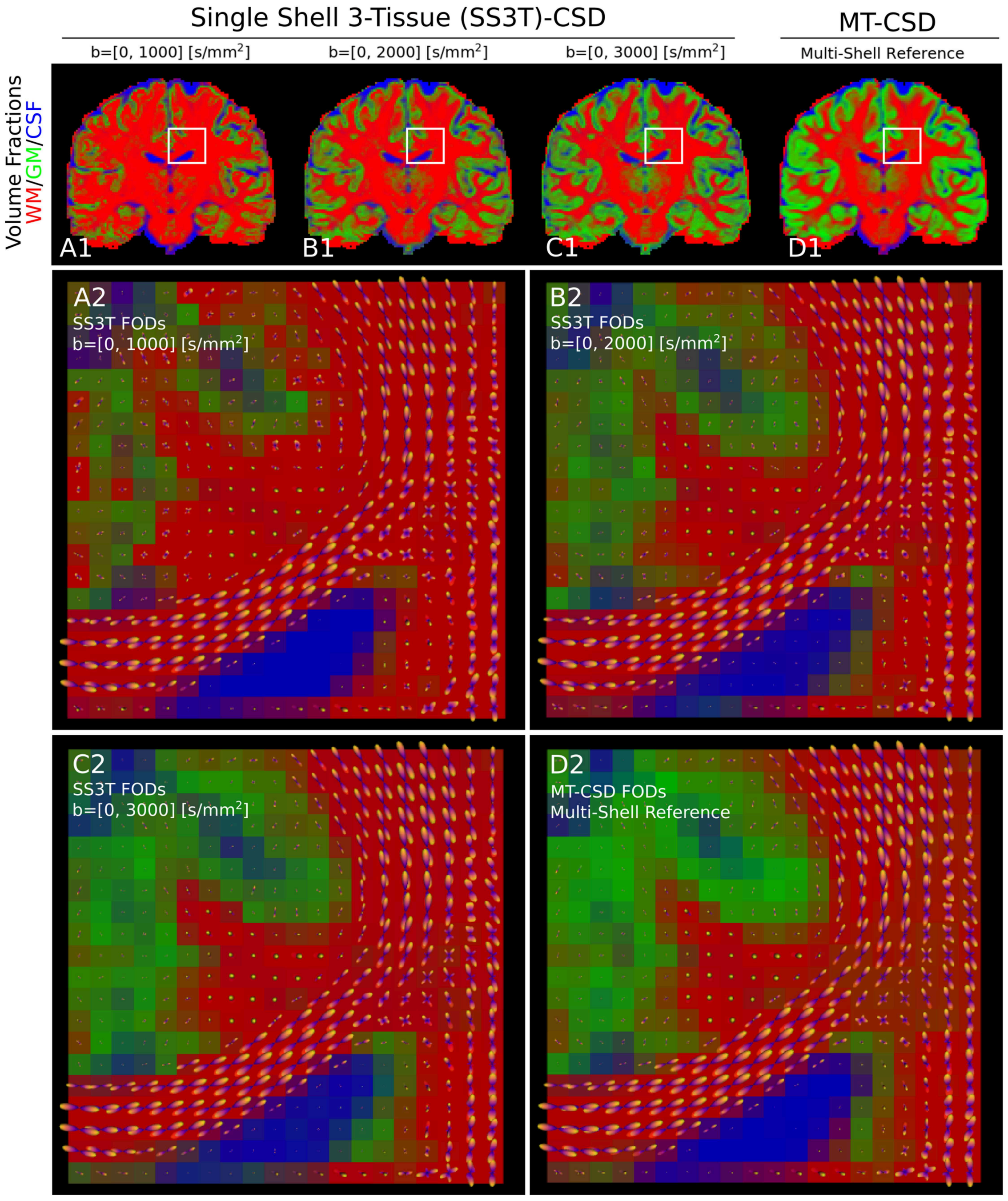


FIGURE 9 | (A1–D1) Single-shell three-tissue (SS3T) volume fraction comparison with the standard multi-shell CSD (MT-CSD) implementation. From left to right, the SS3T algorithm is applied to higher b-shell data. Notice that as higher b-value data is used, the gray matter fraction becomes more prominent, coming close to resembling the MT-CSD results for the b=[0,3,000] [s/mm^2] data. In **(A2–D2)** we illustrate the corresponding FODs for each SS3T and MT-CSD optimization. Notice that the crossing appears better resolved at higher b-values (as expected), but the gray matter fraction is significantly suppressed at lower b-values.

optimized in one MC-model with the parameters obtained from another MC-model. Harms et al. (2017) used this concept to initialize more complicated MC-models with the obtained values of simpler MC-models. In Dmipy, we take this a step further, in that we can take advantage of the unique properties of different MC-modeling frameworks. In section 3.4, we use this concept to first estimate the parameters of spherical mean-based MC-DMI model (Kaden et al., 2016). Then, we use optimized parameters of MC-MDI to define a spherical harmonics-based Multi-Compartment CSD model that uses a voxel-varying convolution kernel to estimate the FOD for each voxel. We illustrate the results in **Figure 7**, where we can notice how well crossings are resolved. In fact, this was also the winning approach for the ISMRM 2017 TraCED challenge for reproducible tractography https://my.vanderbilt.edu/ismrmtraced2017/, which can now be easily implemented and further explored using Dmipy.

This process can also be exploited within the same MC-modeling framework, even within the same MC-model. In section 3.5, we implement the state-of-the-art Single-Shell Three-Tissue (SS3T) algorithm in Snippet 14. As we illustrate in **Figure 9**, the SS3T algorithm is able to obtain "multi-shell-like" MT-CSD results by only using one DWI shell (plus a b_0). It is based on premature termination of an iterative optimization process, whose parameters *briefly* look like the multi-shell solution by virtue of the chosen initialization point. Though, it appears this algorithm works better on higher b-shell data.

4.7. Limitations of the Dmipy Toolbox

At this moment, Dmipy only implements biophysical models designed for the PGSE-dMRI pulse sequence. The main reason for initially targeting the PGSE sequence is because it is currently still the most used sequence, and high-quality PGSE-dMRI data is widely available from excellent sources like the Human Connectome Project. However, Dmipy's modular philosophy may also be applied to MC-models based other sequences, like double-diffusion encoding (Coelho et al., 2019), free waveforms (Ianuş et al., 2016) or multi-dimensional MRI (Nilsson et al., 2018), if only the appropriate acquisition scheme and biophysical model formulations were implemented. All subsequent steps, i.e., combining, fitting, and analyzing the fitted parameters of an MC-model, are already dMRI-sequence agnostic in Dmipy.

4.8. Improving Research Reproducibility Through High Coding Standards

Dmipy provides a concrete means by which dMRI-based microstructure researchers can easily construct and fit the models that are appropriate for their application. Furthermore, Dmipy's user-friendly, building-block-based coding style makes it easy to reliably construct and apply the same MC-model and optimization algorithm, removing the need for researchers to spend time implementing and testing the underlying algorithms. To ensure high-quality coding standards, we currently have over 83% testing coverage and follow Google's style guide for consistent readability and maintenance. In this way, Dmipy contributes to open-source, highly reproducible microstructure research.

5. CONCLUSION

The open-source Dmipy framework represents the unification and implementation of dMRI-based MC-modeling over the last decades. By adopting a "building-block" based philosophy in both the theory and the coding framework, Dmipy is highly modular and is able to construct on-the-fly MC-models in around 10 lines of code. Furthermore, "standard" MC-modeling is generalized with spherical mean-based and spherical convolution-based MC-modeling, allowing for new, cross-MC-modeling approaches. Importantly, Dmipy's key innovation is not that it proposes new methods, but that it generalizes and combines the advantages of already existing works[4]. By providing a well-tested, user-friendly toolbox that simplifies the interaction with the otherwise complicated field of dMRI-based Microstructure Imaging, Dmipy contributes to more reproducible, high-quality research.

DATA AVAILABILITY STATEMENT

The code and all results in this work can be completely reproduced using the tools available at the Dmipy repository at https://github.com/AthenaEPI/dmipy. The Human Connectome Project (HCP) data used to illustrate several of the algorithms is available at https://db.humanconnectome.org/. The spinal cord data is available at the White Matter Microscopy Database at https://dipy.org. The IVIM dataset was downloaded through the Dipy example at http://nipy.org/dipy/examples_built/reconst_ivim.html.

ETHICS STATEMENT

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

RF devised the project, developed the Dmipy software to its mature form, and wrote the manuscript. DW designed the seminal code for the Dmipy framework and restricted signal models. RD supported this project and co-proposed and co-supervised it with DW. RF, DW, and RD discussed the framework and results, provided critical feedback, helped shape the research, analysis, and final manuscript.

FUNDING

This work was partly supported by ANR MOSIFAH under ANR-13-MONU-0009-01, the ERC under the European Union's Horizon 2020 research and innovation program (ERC Advanced Grant agreement No. 694665:CoBCoM).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fninf.2019.00064/full#supplementary-material

[4] Embodying Newton's famous saying "If I have seen further it is by standing on the shoulders of Giants."

REFERENCES

Aboitiz, F., Scheibel, A. B., Fisher, R. S., and Zaidel, E. (1992). Fiber composition of the human corpus callosum. *Brain Res.* 598, 143–153. doi: 10.1016/0006-8993(92)90178-C

Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. (2018). A rewriting system for convex optimization problems. *J. Control Decis.* 5, 42–60. doi: 10.1080/23307706.2017.1397554

Alexander, D. C., Hubbard, P. L., Hall, M. G., Moore, E. A., Ptito, M., Parker, G. J., et al. (2010). Orientationally invariant indices of axon diameter and density from diffusion MRI. *Neuroimage* 52, 1374–1389. doi: 10.1016/j.neuroimage.2010.05.043

Assaf, Y., Blumenfeld-Katzir, T., Yovel, Y., and Basser, P. J. (2008). Axcaliber: a method for measuring axon diameter distribution from diffusion MRI. *Magn. Reson. Med.* 59, 1347–1354. doi: 10.1002/mrm.21577

Assaf, Y., Freidlin, R. Z., Rohde, G. K., and Basser, P. J. (2004). New modeling and experimental framework to characterize hindered and restricted water diffusion in brain white matter. *Magn. Reson. Med.* 52, 965–978. doi: 10.1002/mrm.20274

Assaf, Y., and Pasternak, O. (2008). Diffusion tensor imaging (DTI)-based white matter mapping in brain research: a review. *J. Mol. Neurosci.* 34, 51–61. doi: 10.1007/s12031-007-0029-0

Balinov, B., Jonsson, B., Linse, P., and Soderman, O. (1993). The NMR self-diffusion method applied to restricted diffusion. simulation of echo attenuation from molecules in spheres and between planes. *J. Magn. Reson. Ser. A* 104, 17–25. doi: 10.1006/jmra.1993.1184

Basser, P. J., Mattiello, J., and LeBihan, D. (1994). MR diffusion tensor spectroscopy and imaging. *Biophys. J.* 66:259. doi: 10.1016/S0006-3495(94)80775-1

Behrens, T., Woolrich, M., Jenkinson, M., Johansen-Berg, H., Nunes, R., Clare, S., et al. (2003). Characterization and propagation of uncertainty in diffusion-weighted MR imaging. *Magn. Reson. Med.* 50, 1077–1088. doi: 10.1002/mrm.10609

Burcaw, L. M., Fieremans, E., and Novikov, D. S. (2015). Mesoscopic structure of neuronal tracts from time-dependent diffusion. *NeuroImage* 114, 18–37. doi: 10.1016/j.neuroimage.2015.03.061

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* 16, 1190–1208. doi: 10.1137/0916069

Cabeen, R., Sepehrband, F., and Toga, A. (2019). "Rapid and accurate noddi parameter estimation with the spherical mean technique," in *ISMRM 2019*.

Callaghan, P. T. (1995). Pulsed-gradient spin-echo NMR for planar, cylindrical, and spherical pores under conditions of wall relaxation. *J. Magn. Reson. Ser. A* 113, 53–59. doi: 10.1006/jmra.1995.1055

Canales-Rodríguez, E. J., Legarreta, J. H., Pizzolato, M., Rensonnet, G., Girard, G., Rafael-Patino, J., et al. (2019). Sparse wars: a survey and comparative study of spherical deconvolution algorithms for diffusion MRI. *NeuroImage* 184, 140–160. doi: 10.1016/j.neuroimage.2018.08.071

Caruyer, E., Lenglet, C., Sapiro, G., and Deriche, R. (2013). Design of multishell sampling schemes with uniform coverage in diffusion MRI. *Magn. Reson. Med.* 69, 1534–1540. doi: 10.1002/mrm.24736

Coelho, S., Pozo, J. M., Jespersen, S. N., Jones, D. K., and Frangi, A. F. (2019). Resolving degeneracy in diffusion MRI biophysical model parameter estimation using double diffusion encoding. *Magn. Reson. Med.* 82, 395–410. doi: 10.1002/mrm.27714

Cook, P., Bai, Y., Nedjati-Gilani, S., Seunarine, K., Hall, M., Parker, G., et al. (2006). "Camino: open-source diffusion-MRI reconstruction and processing," in *14th Scientific Meeting of the International Society for Magnetic Resonance in Medicine, Vol. 2759* (Seattle, WA).

Daducci, A., Canales-Rodríguez, E. J., Zhang, H., Dyrby, T. B., Alexander, D. C., and Thiran, J.-P. (2015). Accelerated microstructure imaging via convex optimization (amico) from diffusion MRI data. *NeuroImage* 105, 32–44. doi: 10.1016/j.neuroimage.2014.10.026

De Santis, S., Jones, D. K., and Roebroeck, A. (2016). Including diffusion time dependence in the extra-axonal space improves *in vivo* estimates of axonal diameter and density in human white matter. *NeuroImage* 130, 91–103. doi: 10.1016/j.neuroimage.2016.01.047

Dell'Acqua, F., Scifo, P., Rizzo, G., Catani, M., Simmons, A., Scotti, G., et al. (2010). A modified damped richardson–lucy algorithm to reduce isotropic background effects in spherical deconvolution. *Neuroimage* 49, 1446–1458. doi: 10.1016/j.neuroimage.2009.09.033

Dell'Acqua, F., and Tournier, J.-D. (2018). Modelling white matter with spherical deconvolution: how and why? *NMR Biomed.* 32:e3945. doi: 10.1002/nbm.3945

Dhollander, T., and Connelly, A. (2016). "A novel iterative approach to reap the benefits of multi-tissue CSD from just single-shell (+ b= 0) diffusion MRI data," in *24th International Society of Magnetic Resonance in Medicine* (Singapore).

Dhollander, T., Raffelt, D., and Connelly, A. (2016). "Unsupervised 3-tissue response function estimation from single-shell or multi-shell diffusion MR data without a co-registered t1 image," in *ISMRM Workshop on Breaking the Barriers of Diffusion MRI, Vol. 5* (Lisbon).

Diamond, S., and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *J. Mach. Learn. Res.* 17, 2909–2913.

Drobnjak, I., Zhang, H., Ianuş, A., Kaden, E., and Alexander, D. C. (2016). PGSE, OGSE, and sensitivity to axon diameter in diffusion MRI: insight from a simulation study. *Magn. Reson. Med.* 75, 688–700. doi: 10.1002/mrm.25631

Duval, T., Perraud, B., Vuong, M., Rios, N., and Stikov, N. Cohen-Adad, J. (2016). "Validation of quantitative MRI metrics using full slice histology with automatic axon segmentation," in *ISMRM* (Singapore).

Farooq, H., Xu, J., Nam, J. W., Keefe, D. F., Yacoub, E., Georgiou, T., et al. (2016). Microstructure imaging of crossing (mix) white matter fibers from diffusion MRI. *Sci. Rep.* 6:38927. doi: 10.1038/srep38927

Feinberg, D. A., Moeller, S., Smith, S. M., Auerbach, E., Ramanna, S., Gunther, M., et al. (2010). Multiplexed echo planar imaging for sub-second whole brain fMRI and fast diffusion imaging. *PLoS ONE* 5:e15710. doi: 10.1371/journal.pone.0015710

Ferizi, U., Scherrer, B., Schneider, T., Alipoor, M., Eufracio, O., Fick, R. H., et al. (2016). Diffusion MRI microstructure models with *in vivo* human brain connectom data: results from a multi-group comparison. *arXiv preprint arXiv:1604.07287*. doi: 10.1002/nbm.3734

Fick, R. (2017). *Advanced dMRI signal modeling for tissue microstructure characterization* (Ph.D. thesis). Nice, France: Université Côte d'Azur.

Fick, R. H., Petiet, A., Santin, M., Philippe, A.-C., Lehericy, S., Deriche, R., et al. (2018). Non-parametric graphnet-regularized representation of DMRI in space and time. *Med. Image Analysis* 43, 37–53. doi: 10.1016/j.media.2017.09.002

Fick, R. H., Wassermann, D., Caruyer, E., and Deriche, R. (2016). MAPL: Tissue microstructure estimation using laplacian-regularized MAP-MRI and its application to HCP data. *NeuroImage* 134, 365–385. doi: 10.1016/j.neuroimage.2016.03.046

Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., Van Der Walt, S., Descoteaux, M., et al. (2014). Dipy, a library for the analysis of diffusion MRI data. *Front. Neuroinformatics* 8:8. doi: 10.3389/fninf.2014.00008

Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., et al. (2013). The minimal preprocessing pipelines for the human connectome project. *Neuroimage* 80, 105–124. doi: 10.1016/j.neuroimage.2013.04.127

Gurney-Champion, O. J., Froeling, M., Klaassen, R., Runge, J. H., Bel, A., van Laarhoven, H. W., et al. (2016). Minimizing the acquisition time for intravoxel incoherent motion magnetic resonance imaging acquisitions in the liver and pancreas. *Invest. Radiol.* 51, 211–220. doi: 10.1097/RLI.0000000000000225

Gurney-Champion, O. J., Klaassen, R., Froeling, M., Barbieri, S., Stoker, J., Engelbrecht, M. R., et al. (2018). Comparison of six fit algorithms for the intra-voxel incoherent motion model of diffusion-weighted magnetic resonance imaging data of pancreatic cancer patients. *PLoS ONE* 13:e0194590. doi: 10.1371/journal.pone.0194590

Harms, R., Fritz, F., Tobisch, A., Goebel, R., and Roebroeck, A. (2017). Robust and fast nonlinear optimization of diffusion MRI microstructure models. *Neuroimage* 155, 82–96. doi: 10.1016/j.neuroimage.2017.04.064

Ianuş, A., Alexander, D. C., and Drobnjak, I. (2016). "Microstructure imaging sequence simulation toolbox," in *International Workshop on Simulation and Synthesis in Medical Imaging* (Athens: Springer), 34–44.

Jelescu, I. O., Veraart, J., Fieremans, E., and Novikov, D. S. (2016). Degeneracy in model parameter estimation for multi-compartmental diffusion in neuronal tissue. *NMR Biomed.* 29, 33–47. doi: 10.1002/nbm.3450

Jeurissen, B., Tournier, J.-D., Dhollander, T., Connelly, A., and Sijbers, J. (2014). Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data. *NeuroImage* 103, 411–426. doi: 10.1016/j.neuroimage.2014.07.061

Jones, E., Oliphant, T., and Peterson, P. (2001). *SciPy: Open Source Scientific Tools for Python.*

Kaden, E., Kelm, N. D., Carson, R. P., Does, M. D., and Alexander, D. C. (2016). Multi-compartment microscopic diffusion imaging. *NeuroImage* 139, 346–359. doi: 10.1016/j.neuroimage.2016.06.002

Kaden, E., Knösche, T. R., and Anwander, A. (2007). Parametric spherical deconvolution: inferring anatomical connectivity using diffusion MR imaging. *NeuroImage* 37, 474–488. doi: 10.1016/j.neuroimage.2007.05.012

Kaden, E., Kruggel, F., and Alexander, D. C. (2015). Quantitative mapping of the per-axon diffusion coefficients in brain white matter. *Magn. Reson. Med.* 75, 1752–1763. doi: 10.1002/mrm.25734

Lam, S. K., Pitrou, A., and Seibert, S. (2015). "Numba: A LLVM-based python JIT compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC* (Austin, TX: ACM).

Le Bihan, D. (2017). What can we see with IVIM MRI? *NeuroImage* 187, 56–67. doi: 10.1016/j.neuroimage.2017.12.062

Le Bihan, D., Breton, E., Lallemand, D., Aubin, M., Vignaud, J., and Laval-Jeantet, M. (1988). Separation of diffusion and perfusion in intravoxel incoherent motion MR imaging. *Radiology* 168, 497–505. doi: 10.1148/radiology.168.2.3393671

Le Bihan, D., Breton, E., Lallemand, D., Grenier, P., Cabanis, E., and Laval-Jeantet, M. (1986). MR imaging of intravoxel incoherent motions: application to diffusion and perfusion in neurologic disorders. *Radiology* 161, 401–407. doi: 10.1148/radiology.161.2.3763909

Leergaard, T. B., White, N. S., De Crespigny, A., Bolstad, I., D'Arceuil, H., Bjaalie, J. G., et al. (2010). Quantitative histological validation of diffusion MRI fiber orientation distributions in the rat brain. *PLoS ONE* 5:e8595. doi: 10.1371/journal.pone.0008595

McKerns, M. M., Strand, L., Sullivan, T., Fang, A., and Aivazis, M. A. (2012). Building a framework for predictive science. *arXiv preprint arXiv:1202.1056.*

Minati, L., and Węglarz, W. P. (2007). Physical foundations, models, and methods of diffusion magnetic resonance imaging of the brain: a review. *Concepts Magn. Reson. Part A Educ. J.* 30, 278–307. doi: 10.1002/cmr.a.20094

Moeller, S., Yacoub, E., Olman, C. A., Auerbach, E., Strupp, J., Harel, N., et al. (2010). Multiband multislice GE-EPI at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fMRI. *Magn. Reson. Med.* 63, 1144–1153. doi: 10.1002/mrm.22361

Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378.*

Nath, V., Schilling, K. G., Parvathaneni, P., Huo, Y., Blaber, J. A., Hainline, A. E., et al. (2019). Tractography reproducibility challenge with empirical data (TraCED): the 2017 ISMRM diffusion study group challenge. *J. Magn. Reson. Imaging*. doi: 10.1002/jmri.26794. [Epub ahead of print].

Nilsson, M., Szczepankiewicz, F., Lampinen, B., Ahlgren, A., De Almeida Martins, J. P., Lasic, S., et al. (2018). "An open-source framework for analysis of multidimensional diffusion MRI data implemented in matlab," in *Proc Intl Soc Mag Reson Med, Vol. 26* (Paris).

Ning, L., Laun, F., Gur, Y., DiBella, E. V., Deslauriers-Gauthier, S., Megherbi, T., et al. (2015). Sparse reconstruction challenge for diffusion MRI: Validation on a physical phantom to determine which acquisition scheme and analysis method to use? *Med. Image Analys.* 26, 316–331. doi: 10.1016/j.media.2015.10.012

Novikov, D. S., Jensen, J. H., Helpern, J. A., and Fieremans, E. (2014). Revealing mesoscopic structural universality with diffusion. *Proc. Natl. Acad. Sci. U.S.A.* 111, 5088–5093. doi: 10.1073/pnas.1316944111

Novikov, D. S., Veraart, J., Jelescu, I. O., and Fieremans, E. (2018). Rotationally-invariant mapping of scalar and orientational metrics of neuronal microstructure with diffusion MRI. *NeuroImage* 174, 518–538. doi: 10.1016/j.neuroimage.2018.03.006

Oliphant, T. (2006). *NumPy: A Guide to NumPy.* Trelgol Publishing.

Özarslan, E., Koay, C. G., Shepherd, T. M., Komlosh, M. E., İrfanoğlu, M. O., Pierpaoli, C., et al. (2013). Mean apparent propagator (map) MRI: a novel diffusion imaging method for mapping tissue microstructure. *NeuroImage* 78, 16–32. doi: 10.1016/j.neuroimage.2013.04.016

Panagiotaki, E., Schneider, T., Siow, B., Hall, M. G., Lythgoe, M. F., and Alexander, D. C. (2012). Compartment models of the diffusion mr signal in brain white matter: a taxonomy and comparison. *Neuroimage* 59, 2241–2254. doi: 10.1016/j.neuroimage.2011.09.081

Panagiotaki, E., Walker-Samuel, S., Siow, B., Johnson, S. P., Rajkumar, V., Pedley, R. B., et al. (2014). Noninvasive quantification of solid tumor microstructure using verdict MRI. *Cancer Res* 74, 1902–1912. doi: 10.1158/0008-5472.CAN-13-2511

Park, H. J., Sung, Y. S., Lee, S. S., Lee, Y., Cheong, H., Kim, Y. J., et al. (2017). Intravoxel incoherent motion diffusion-weighted mri of the abdomen: the effect of fitting algorithms on the accuracy and reliability of the parameters. *J. Magn. Reson. Imaging* 45, 1637–1647. doi: 10.1002/jmri.25535

Pizzolato, M., Wassermann, D., Deriche, R., Thiran, J.-P., and Fick, R. (2018). "Orientation-dispersed apparent axon diameter via multi-stage spherical mean optimization," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer), 91–101.

Setsompop, K., Gagoski, B. A., Polimeni, J. R., Witzel, T., Wedeen, V. J., and Wald, L. L. (2012). Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced G-factor penalty. *Magn. Reson. Med.* 67, 1210–1224. doi: 10.1002/mrm.23097

Sotiropoulos, S. N., Jbabdi, S., Xu, J., Andersson, J. L., Moeller, S., Auerbach, E. J., et al. (2013). Advances in diffusion mri acquisition and processing in the human connectome project. *Neuroimage* 80, 125–143. doi: 10.1016/j.neuroimage.2013.05.057

Stejskal, E. (1965). Use of spin echoes in a pulsed magnetic-field gradient to study anisotropic, restricted diffusion and flow. *J. Chem. Phys.* 43, 3597–3603. doi: 10.1063/1.1696526

Storn, R., and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optimiz.* 11, 341–359. doi: 10.1023/A:1008202821328

Szafer, A., Zhong, J., and Gore, J. C. (1995). Theoretical model for water diffusion in tissues. *Magn. Reson. Med.* 33, 697–712. doi: 10.1002/mrm.1910330516

Tariq, M., Schneider, T., Alexander, D. C., Wheeler-Kingshott, C. A. G., and Zhang, H. (2016). Bingham–noddi: Mapping anisotropic orientation dispersion of neurites using diffusion MRI. *NeuroImage* 133, 207–223. doi: 10.1016/j.neuroimage.2016.01.046

Tax, C. M., Jeurissen, B., Vos, S. B., Viergever, M. A., and Leemans, A. (2014). Recursive calibration of the fiber response function for spherical deconvolution of diffusion MRI data. *Neuroimage* 86, 67–80. doi: 10.1016/j.neuroimage.2013.07.067

Tournier, J.-D., Calamante, F., and Connelly, A. (2007). Robust determination of the fibre orientation distribution in diffusion MRI: non-negativity constrained super-resolved spherical deconvolution. *NeuroImage* 35, 1459–1472. doi: 10.1016/j.neuroimage.2007.02.016

Tournier, J.-D., Calamante, F., and Connelly, A. (2012). Mrtrix: diffusion tractography in crossing fiber regions. *Int. J. Imaging Syst. Technol.* 22, 53–66. doi: 10.1002/ima.22005

Tournier, J.-D., Calamante, F., and Connelly, A. (2013). Determination of the appropriate B value and number of gradient directions for high-angular-resolution diffusion-weighted imaging. *NMR Biomed.* 26, 1775–1786. doi: 10.1002/nbm.3017

Tournier, J.-D., Calamante, F., Gadian, D. G., and Connelly, A. (2004). Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution. *NeuroImage* 23, 1176–1185. doi: 10.1016/j.neuroimage.2004.07.037

Tuch, D. S. (2004). Q-ball imaging. *Magn. Reson. Med.* 52, 1358–1372. doi: 10.1002/mrm.20279

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., et al. (2013). The WU-MINN human connectome project: an overview. *Neuroimage* 80, 62–79. doi: 10.1016/j.neuroimage.2013.05.041

Vangelderen, P., DesPres, D., Vanzijl, P., and Moonen, C. (1994). Evaluation of restricted diffusion in cylinders. Phosphocreatine in rabbit leg muscle. *J. Magn. Reson. Ser. B* 103, 255–260. doi: 10.1006/jmrb.1994.1038

Wedeen, V. J., Hagmann, P., Tseng, W.-Y. I., Reese, T. G., and Weisskoff, R. M. (2005). Mapping complex tissue architecture with diffusion spectrum magnetic resonance imaging. *Magn. Reson. Med.* 54, 1377–1386. doi: 10.1002/mrm.20642

Whittall, K. P., and MacKay, A. L. (1989). Quantitative interpretation of NMR relaxation data. *J. Magn. Reson.* 84, 134–152. doi: 10.1016/0022-2364(89)90011-5

Wu, Y.-C., and Alexander, A. L. (2007). Hybrid diffusion imaging. *NeuroImage* 36, 617–629. doi: 10.1016/j.neuroimage.2007.02.050

Xu, J., Moeller, S., Strupp, J., Auerbach, E., Chen, L., Feinberg, D., et al. (2012). "Highly accelerated whole brain imaging using aligned-blipped-controlled-aliasing multiband EPI," in *Proceedings of the 20th Annual Meeting of ISMRM* (Melbourne).

Zhang, H., Schneider, T., Wheeler-Kingshott, C. A., and Alexander, D. C. (2012). Noddi: practical *in vivo* neurite orientation dispersion and density imaging of the human brain. *Neuroimage* 61, 1000–1016. doi: 10.1016/j.neuroimage.2012.03.072

Conflict of Interest: RF was employed by company TheraPanacea.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.