

Mesoscale Individualized NeuroDynamic (MINDy) Modeling

Matthew F. Singh*, Todd S. Braver, ShiNung Ching
Washington University in St. Louis
*f.singh@wustl.edu

Abstract

This document provides instructions for performing MINDy model-fitting in MATLAB. Note that the provided code generally makes heavy use of binary implicit expansion (R2016b or later).

1 Data and Formats

For the highest-level MINDy functions (e.g. “MINDy_Simple”) only one argument needs to be supplied: the data time series (see below). For lower-level functions, there are three main types of inputs (excluding options): data cells/matrices (two arguments), a structure containing model-fitting hyper-parameters (“ParStr”) and a structure containing information for performing the deconvolution (“Pre”).

1.1 Data

All data is assumed to be formatted as **region** \times **time**. Data can be input as a single matrix for a single continuous recording or as a cell containing multiple matrices (e.g. multiple recording segments). In general, we assume that data for each channel has already been z-scored—otherwise the hyperparameters may need to be adjusted to compensate for the new data range. For the highest-level MINDy functions (e.g. MINDy_Simple) data is input from the original time series as a single argument (one matrix or one cell) prior to hemodynamic deconvolution. The function performs hemodynamic deconvolution and derivative estimation automatically.

These functions then feed this information into lower-level functions (e.g. MINDy_Base) which use the deconvolved and differentiated data to perform model-fitting. Alternatively these lower-level functions can be accessed directly with data that has already been appropriately processed. In this case there are two input data arguments: a deconvolved time series and the

temporal derivative of the deconvolved time series. Both of these time-series should have the same length. Accessing these functions directly allows the user to customize the method of differentiation/deconvolution as opposed to the default method of Weiner-deconvolution with finite-difference derivative estimation.

For a single scan-session data can be formatted as a matrix or a cell containing a single matrix. For multiple scan sessions, data should be input as a cell containing one matrix per recording segment. For instance, a set of two scanning sessions for one participant would be input as: $\{X_{n \times t_1}^{(1)}, X_{n \times t_2}^{(2)}\}$. with $X^{(1)}$ and $X^{(2)}$ denoting the data for the first and second scan, respectively. Both scans contain n brain region, but can differ in lengths (t_1 and t_2).

2 Functions

The functions included with MINDy include the functions for fitting models, processing/deconvolving data, simulating fit models, and analyzing the quality of fits.

2.1 High-Level Fitting Functions (Where to start)

MINDy contains four main high-level-functions: MINDy_Simple, MINDy_Tinker, and MINDy_FULL. These functions differ as follows:

1. **MINDy_Simple**: The most basic function.
Inputs: Data, TR, do-smoothing
2. **MINDy_HRFbold_Simple**: Also estimates HRFs.
Inputs: Data, TR, do-smoothing
3. **MINDy_Tinker**: Optionally specify (important) hyperparameters.
Inputs: Data, TR, do-smoothing, do-preprocessing, hyperparameters (optional), connectivity mask (optional), record estimation process (optional).
4. **MINDy_FULL**: Fully customizable via “Pre” and “ParStr”.
Inputs: Data, “Pre” (pre-processing structure), “ParStr” (parameter structure), do-preprocessing, do-global readjustment, do-smoothing,

MINDy_Simple only needs a single input argument: The parcellated time-series data formatted as region \times time (see above). MINDy_Simple also allows an optional input to specify whether preprocessing (deconvolution/filtering) should be performed with admissible values ‘y’ (default) or ‘n’. MINDy_HRFbold_Simple functions analogously but also estimates hemodynamic response functions (HRFs) for each brain region.

MINDy_Tinker provides an easy means to further tune important hyperparameters. New values can be supplied to retune the designated hyperparameters or empty input ("[]") can be provided to use default values. These include regularization parameters (lam1 (sparsity), lam2 (low-rank structuring), and SpDiag (penalize recurrent connections) which may need to be increased when less data is available. MaxRank denotes the maximal rank of the low-dimensional component. MINDy_Simple and MINDy_Simple_HRFbold automatically rescale this value based upon the number of brain regions supplied. If you do not change MaxRank, it will use the original value of (150) instead of readjusting as is done by MINDy_Simple.

The size of each optimization sample (BatchSz) and the number of iterations (NBatch) can also be specified. The degree of filtering in the Wiener deconvolution is denoted "ConvNSR". Connectivity patterns can be constrained by supplying a binary $n \times n$ matrix "Wmask". Connections will then be constrained to cases with Wmask not equal to zero. Lastly, "doRecord" specifies whether to record the series of parameter (e.g. connectivity) estimates made during the MINDy optimization process. This variable accepts either "y" to record all variables or combinations of which variables are desired (e.g. " 'DA' " for Decay and Curvature). We note that recording the connectivity estimates can use a large amount of memory.

The function MINDy_FULL allows greater customization. Its first three input arguments correspond to the data (same format as MINDy_Simple) and the ParStr, Pre fields (which can be generated for the published parameters using "ChosenParSTR" and modified from there). The remaining arguments in MINDy_FULL correspond to additional pre/post-processing options. These arguments and their admissible values are as follows:

| Argument | Values | Effect |
|-----------|--------|---|
| doSplit | 'n' | Define $dx_t = x_{t+1} - x_t$ (advised for large TR/low noise) |
| | 'y' | Define $dx_t = (x_{t+2} - x_t)/2$ (advised for small TR/high noise) |
| doPreProc | 'n' | Don't perform deconvolution/censoring (you've already deconvolved) |
| | 'y' | Do perform deconvolution/censoring (otherwise choose) |
| doInflate | 'n' | Don't readjust for regularization bias |
| | 'y' | Do readjust (advised) |
| doSmooth | 'n' | Don't smooth timeseries |
| | 'y' | Convolve time-series with [1 1] (after preprocessing) |

2.2 MINDy Outputs

All fitting functions produce a single structure for output with similar formatting. In all cases, this structure contains the field "Param" which contains parameter values in the cell ordering: $\{W_S, \alpha, b, c, W, D\}$. Here, W_S denotes the sparse components of the weights and the low-rank component can be reconstructed as $W_L = W - W_S$. The c element refers to a constant-

drive term which is currently fixed at zero. The b term contains two elements: the slope variable denoted b in the paper (left column) and a shift-term for the transfer function (right column) which is currently fixed at zero.

In terms of their phenomenological significance, the parameters are as follows:

Weights (W): Param{5}

Curvature (α):Param{2}

Decay (D): Param{6}

All MINDy fitting functions also output the field: BadDerivs which gives the time points of each frame censored based upon derivative amplitudes and the field: GoodFrames which contains the indices of the remaining (uncensored) time points. BadDerivs gives indices relative to the concatenated time-series in its first cell, whereas GoodFrames has separate cells of indices for each recording session (e.g. scanning run). The field “oLength” gives the original length of each time-series before censoring. The field “Mat” contains the cell: $\{W_S, W_1, W_2^T\}$ in which W_S is the sparse component of the weights: $W = W_S + W_1 W_2^T$. The field “GLMweights” gives the global scaling factor for the W and D terms for the regularization-compensating regression. These factors will be already factored into the Param field, but not the Mat field. Values before global scaling are contained in the fields “oldDecay” and “oldW1”.

Don’t Worry! Hyperparameters only need to be passed when using Lower-Level MINDy functions (e.g. for customization). Even then, you won’t need to change most of them.

2.3 Hyperparameters: ParStr

The hyperparameters to the model-fitting are passed to all MINDy functions as a structure (“ParStr”). This structure can be generated using the script: “ChosenParSTR” which will also populate the fields with the default values used for fitting HCP models as in the original publication. There are two exceptions: “ChosenParSTR” does not automatically include the values for the size of each training batch (“BatchSz”) or the number of batches (“NBatch”). Within the high-level MINDy functions (e.g. MINDy_Simple) these are automatically set as in the paper, whereas users will need to set them manually to use lower-level functions—this omission is intended as a reminder to set the training length appropriately as the parameter most likely to need adjustment between use-cases (data quality, amount of data, etc.).

The relation between ParStr fields and variables in the original paper is as follows:

| Field | Symbol in Paper | Meaning |
|-----------|---|---|
| SpScale | λ_1 | Sparse Regularization |
| SpDiag | λ_2 | Recurrent Regularization |
| SpPls1 | λ_3 | Low-Rank Regularization (Pre) |
| SpPls2 | λ_3 | Low-Rank Regularization (Post) |
| L2SpPlsEN | λ_4 | L2-Regularization of Low-Rank |
| Dmin | Dmin | Minimal value of D |
| Bmin | b^{-1} | Fixed b-parameter of transfer fct. |
| wPC | k | Maximum rank of W_L |
| Dec | μ | NADAM AR-1 |
| Dec2 | ν | NADAM AR-2 |
| Reg | $\varepsilon(W)$ | NADAM Regularization for W (all W 's) |
| AReg | $\varepsilon(\alpha)$ | NADAM Regularization for α |
| DRegScale | $\varepsilon(D_2)/\varepsilon(W)$ | NADAM Reg. for D_2 rel. to W |
| Rate | $\text{Rate}(W_S)$ | NADAM update rate for W |
| ARate | $\text{Rate}(\alpha)$ | NADAM update rate for α |
| wk1Rate | $\text{Rate}(W_1)/\text{Rate}(W_S)$ | Relative Rate for W_1 |
| wk2Rate | $\text{Rate}(W_2)/\text{Rate}(W_S)$ | Relative Rate for W_2 |
| DRate | $\text{Rate}(D_2)/(\text{Rate}(W_S) \times \text{DRegScale})$ | Relative Rate for D_2 |
| Dstart | $\min(D_2\text{seed})$ | Minimal initial seeding for D_2 |
| RandScale | $\text{SD}(W\text{seed})$ | STD of initial seeding for all W 's |

2.4 Hyperparameters: Pre

The “Pre” structure, like “ParStr” is generated by evaluating the script “ChosenParSTR”. Pre contains information used for performing the deconvolution step which is automatically performed by “MINDy_PreProc”

| Field | Meaning |
|-----------|---|
| FiltAmp | Cut-off threshold in SD's for censoring spikes (linear interpolation) |
| TR | Temporal Resolution in seconds |
| ConvLevel | SNR filtering parameter for Wiener-deconvolution |