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-based Filtering/Prediction in MATLAB. Note that the provided code generally makes heavy use of binary implicit expansion (R2016b or later). The coded provided within this folder is self-contained. We assume that all data has already been put in a normalized scale (baseline mean=0, SD=1) and parcellated (this code is not meant for vertex/voxel level analysis).

1 Framework

The high-level MINDy functions consist of MINDy_RAW_CV and MINDy_HRF_CV. Both of these functions use training data to produce a MINDy model ([1],[2]) and (optionally) apply it to predict any testing data provided (for predictions of training data, just include it as one of the testing sets). Both of these functions parameterize models as a difference equation ($\Delta x = x(t+1) - x(t) = f(x_t)$), but the predictions are made for $x(t+1)$ as is used in MINDy-based Filtering with hemodynamics.

1.1 MINDy-based Filtering and Prediction

The provided code aims to predict the temporal evolution of one data set (the "testing data") using MINDy models fit to another data set (the "training data"). One use of these functions is MINDy-based filtering in which the influence of endogenous dynamics is filtered (subtracted) from task timeseries ([3]). Hence, we seek to estimate the task influences (I_t) from the model $BOLD_{t+1}^{(task)} \approx f_t^{(rest)}(BOLD) + I_{t+1}$ with $f^{(rest)}$ denoting the resting-state model. To perform, MINDy-based Filtering, resting-state data is used in training and task data is used for testing. The estimates of I_{t+1} are given by the residuals i.e. the call to perform MINDy-based Filtering is

[model,~filtDat]=MINDy_HRF_CV(rest,task,TR) (see below). The filtered timeseries can then be substituted into any conventional analysis pipeline. Note that by performing MINDy-based filtering, the first timepoint of each scan is lost.

1.2 Hemodynamic Modeling

Hemodynamic modeling in MINDy leverages surrogate deconvolution ([2]). This approach enables rapid analytic estimates of how changing the shape of a convolution kernel (i.e., the Hemodynamic Response Function/HRF) changes the shape of the deconvolved (i.e., neural) timeseries. This enables MINDy to quickly solve the neural and hemodynamic modeling simultaneously. We model HRFs using the canonical double-gamma mode ([4]):

$$h_i(\alpha,\beta,t) = \frac{t^{\alpha_i-1}e^{-\beta_i t}\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} - \frac{t^{15}e^{-t}}{6(16!)} \tag{1}$$

The parameters α, β are fit simultaneously with MINDy models. These parameters primarily reflect the positive portions of the HRF (increases in BOLD following neural activity) rather than the undershoot, as we found the latter component (far left side of the equation) was less impactful in estimating deconvolved neural activity. Typical values are $\alpha \approx 6$ and $\beta \approx 1$. The values for $\{\alpha, \beta\}$ are stored in the output field "HRF" and the kernel timeseries are stored in the field "HRF_kern".

Two region-specific scaling constants are also associated with each value of the HRF parameters: m_{pre} and m_{post}. These constants keep the scaling (variance) of estimated neural activity constant despite changes in the HRF parameters. We use this scaling to prevent regional differences in HRF shapes from biasing estimates of neural model parameters. Thus, m_{pre}^{-1} is equally to the standard-deviation of deconvolved BOLD for a given parcel (before scaling) and the estimated neural activity is $x = m_{pre}(H\hat{*}^{-1}BOLD)$. The second constant performs the opposite operation–ensuring that the reconvolution of estimated neural activity also has unit variance. Thus, m_{post}^{-1} is equal to the standard deviation of $H * (m_{pre}(H\hat{*}^{-1}BOLD))$. The notation $\hat{*}^{-1}$ indicates the use of a numerical deconvolution method (i.e., Wiener deconvolution; [**?**]), hence $H * (H\hat{*}^{-1}) \neq \mathbf{1}$. The values of m_{post} for each parcel are stored in the field "NormScale" and the values for m_{pre} are stored in the output field "NormScalePre".

1.3 Predicting BOLD timeseries

MINDy_HRF_CV produces combined neural-hemodynamic models by calling the function MINDy_HRFbold_D2D. This function initially calls MINDy_HRFnorm_DinX to create models for the error-function:

$$BOLD(t+1) = H * [(W\psi(x(t)) + (1 - D_0)x(t))] \quad (2)$$

with the region-specific HRF convolution denoted"$H*$" and $x(t)$ the estimate of latent neural activity produced by deconvolving the BOLD signal by H–the region-specific hemodynamic response function. Since the linear univariate terms $(1 - D_0)x(t)$ commute with an exact convolution-deconvolution sequence, we convert this model into a form containing these terms on the outside to reduce bias due to inexact deconvolution. Hence the new model is of the form:

$$BOLD(t+1) = H * [W\psi(x(t))] + (1 - D)BOLD(t) \quad (3)$$

Predictions are made via the function MINDy_Predict_NoDeriv_PrePost_Dout using Fourier transform $\mathcal{F}$ in the convolution and deconvolution steps:

$$X_\tau = m_{Pre}\hat{H*}^{-1}BOLD = m_{Pre}\mathcal{F}^{-1}\left[\frac{\mathcal{F}[H]^*\mathcal{F}[BOLD]}{\|\mathcal{F}[H]\|^2 + \eta}\right] \quad (4)$$

$$BOLD(t+1) = (1 - D)BOLD(t) + m_{Post}\mathcal{F}^{-1}\left[\mathcal{F}(H)\mathcal{F}\left[W\psi(X_\tau)\right]\right] \quad (5)$$

with η denoting relative NSR (this value is stored in the structure "Pre" under the field "ConvLevel"). Larger values of η produce more spectral filtering in the deconvolution (we recommend keeping the default value of .002).

1.4 Predicting Neural timeseries

MINDy_Raw_CV is provided for data that does not use further hemodynamic modeling (e.g., is already deconvolved). The function produces a MINDy model by calling MINDy_Base and the parameters are combined into a function handle ["FastFun"] that produces the predicted difference. The 1-step predictions are computed as $x(t+1) = x(t) + FastFun(x(t)) = x(t) + [W\psi(x(t)) - Dx(t)]$.

2 Data and Formats

The high-level MINDy functions consist of MINDy_RAW_CV and MINDy_HRF_CV. For both of these functions, at least one time series needs to be supplied: the training time series (see below) and, optionally a testing time series. If prediction to a new time-series is not desired, the second input (testing data) can be left empty as "[ ]". MINDy_RAW does not invoke any hemodynamic analysis. We therefore recommend MINDy_HRF for all BOLD data. Unlike, MINDy_HRF also requires the sampling TR (in seconds) to be supplied. Both functions also accept the "Pre" and "ParStr" structures (below) as optional arguments to further customize regularization etc.

2.1 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:
$\{X^{(1)}_{n\times t_1}, X^{(2)}_{n\times t_2}\}$. 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).

2.2 MINDy Outputs

The MINDy_RAW_CV and MINDy_HRF_CV produce up to four outputs. The first output consists of the fit MINDy model (described in more detail below). The second output consists of 1-step predictions for the testing-data (formatted the same as data inputs), while the third consists of residuals (i.e. the error in these predictions). The fourth output consists of the goodness-of-fit (R^2) of these predictions for each brain region, pooled across all testing sessions.

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}

MINDy functions without HRF estimation (e.g., MINDy_Base) censor spikes in the model-fitting data (which is assumed to have already been deconvolved). No modifications/cleaning are ever made to a testing time series as used by higher-level prediction functions (e.g., MINDy_RAW_CV). They 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".

2.3 Automatic Hyperparameter Rescaling

The most important MINDy parameters are the regularization terms whose default value is based upon using 400 parcels. In the high-level MINDy functions, these values are automatically adjusted to the provided number of parcels using the function "MINDy_Naive_Hyper_Scale" based upon the assumption that the typical weight coefficient scales inversely with parcel count (e.g., if you divide a region in half, the total output from each subregion is half of the original output). This function accepts the number of parcels and (optionally) a ParStr structure as input (see below). If you pass a ParStr structure through the rescaling function more than once, it will divide/multiply the hyperparameters time (don't do this!). So if you are working with multiple parcellation sizes, it is better to use a new ParStr for each case. This will be done automatically when you only provide the parcel size as input.

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.

2.4 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 publications ([1],[2],[3]). 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 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	Rate(W_S)	NADAM update rate for W
ARate	Rate(α)	NADAM update rate for α
wk1Rate	Rate(W_1)/Rate(W_S)	Relative Rate for W_1
wk2Rate	Rate(W_2)/Rate(W_S)	Relative Rate for W_2
DRate	Rate(D_2)/(Rate(W_S)×DRegScale)	Relative Rate for D_2
Dstart	min($D_2 seed$)	Minimal initial seeding for D_2
RandScale	SD($W seed$)	STD of initial seeding for all $W's$

2.5 Hyperparameters: Pre

The "Pre" structure, like "ParStr" is generated by evaluating the script "ChosenParSTR". Pre contains information used for performing the decon-

volution 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

References

[1] M. F. Singh, T. S. Braver, M. W. Cole, and S. Ching, Estimation and validation of individualized dynamic brain models with resting state fmri,NeuroImage, 221, 117046, 2020.

[2] M. F. Singh, A. Wang, T. S. Braver, and S. Ching, Scalable surrogate deconvolution for identification of partially-observable systems and brain modeling, Journal of Neural Engineering, 2020.

[3] M. F. Singh, A. Wang, M. W. Cole, S. Ching, and T. S. Braver "Enhancing Task fMRI Preprocessing via Individualized Model-Based Filtering of Intrinsic Activity Dynamic," bioRxiv, doi:10.1101/2020.12.10.420273, 2020.

[4] K. J. Friston, P. Fletcher, O. Josephs, A. Holmes, M. D. Rugg and R. Turner, "Event-Related fMRI: Characterizing Differential Responses," NeuroImage, vol. 7, pp. 30-40, 1998.

[5] N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series, vol. 2. Cambridge, MA, USA: MIT Press, 1949