Table of Contents

[Pre-Processing 2](#_Toc197264194)

[By Jayson 2](#_Toc197264195)

[Parcellation 4](#_Toc197264196)

[HMM Mar 5](#_Toc197264197)

[FormatHmmmarInputs 5](#_Toc197264198)

[AnalyseResults 6](#_Toc197264199)

# Pre-Processing

## By Jayson

Specifically, the data has been 'cleaned', but not smoothed yet. We may want to smooth later (6mm fwhm, this can be done in nilearn)

Steps:

SPACE space-MNI152NLin6Asym – *fMRIPrep: spatial normalisation to the MNI152NLin6Asym standard space*

ICA\_AROMA True – *ICA-AROMA: head-motions related movement artefacts removed*

Remove first 1 volumes (non-steady-state) and replace with first steady-state volume – *fMRIPrep: unstable signals are replaced with the first steady-state volume*

Set regions outside mask to zero – *fMRIPrep: zeroes out all voxel values that fall outside a defined brain mask*

detrend False – *Linear trending not applied*

remove\_confounds True – *Confounds below removed*

bandpass 0.01-0.15 – *Band pass filtering done here (lower = drift, higher = noise)*

add\_orig\_mean\_img False - *After confound regression and filtering, do not re-add the original mean image back to the data. Some workflows subtract the mean during processing and then restore it afterward. Setting this to False leaves the mean-removed (demeaned) data as-is.*

smooth False – *spatial smoothing not applied*

fwhm 0 – *as above (Full-width at half-maximum (FWHM) of smoothing kernel = 0 m)*

save\_confounds False – *confounds not saved, good for space saving, bad for reproducibility*

CONFOUND\_LIST ['csf', 'white\_matter', 'rot\_x', 'rot\_x\_power2', 'rot\_x\_derivative1', 'rot\_x\_derivative1\_power2', 'rot\_y', 'rot\_y\_power2', 'rot\_y\_derivative1', 'rot\_y\_derivative1\_power2', 'rot\_z', 'rot\_z\_power2', 'rot\_z\_derivative1', 'rot\_z\_derivative1\_power2', 'trans\_x', 'trans\_x\_power2', 'trans\_x\_derivative1', 'trans\_x\_derivative1\_power2', 'trans\_y', 'trans\_y\_power2', 'trans\_y\_derivative1', 'trans\_y\_derivative1\_power2', 'trans\_z', 'trans\_z\_power2', 'trans\_z\_derivative1', 'trans\_z\_derivative1\_power2', 'physio1', 'physio2', 'physio3', 'physio4', 'physio5', 'physio6', 'physio7', 'physio8', 'physio9', 'physio10', 'physio11', 'physio12', 'physio13', 'physio14', 'physio15', 'physio16', 'physio17', 'physio18']

*This is the list of confound regressors to use during the denoising step.*

*Examples of confounds included:*

*Anatomical noise:*

*'csf', 'white\_matter' – Mean signals from cerebrospinal fluid and white matter (often reflect non-neural noise).*

*Motion parameters:*

*'rot\_x', 'trans\_y', etc. – Rotational and translational motion estimates from realignment.*

*Their derivatives (\_derivative1) and squared terms (\_power2) are included to better model motion artifacts.*

*Physiological regressors:*

*'physio1' through 'physio18' – Principal components of physiological noise (likely extracted via aCompCor or similar).*

*Why it matters:*

*This detailed set of regressors aims to comprehensively capture various sources of non-neural variability to improve the signal quality.*

# Parcellation

|  |  |
| --- | --- |
| Choice | Explanation |
| Using the Yeo 17 thick atlas |  |
| standardize='zscore\_sample', #"zscore\_sample", #Z scores the voxels to make mean = 0 |  |
| memory="nilearn\_cache", |  |
| verbose=5, |  |
| #Temporal filter settings  high\_pass=0.01, # High pass frequency in Hz  low\_pass=0.15, # Low pass frequency in Hz  t\_r=0.8 # Repetition time in seconds |  |
|  |  |
|  |  |
|  |  |

# HMM Mar

## FormatHmmmarInputs

Options.\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Setting | Set | Definition | Explanation |
| K | 6 | Number of hidden states | Less computationally expensive and easier to interpret? |
| Id\_mixture | 1 | if false, the model will be a mixture of distributions instead of an HMM, ignoring the temporal structure of the data (default to false). | Want HMM rather than mix of distributions |
| order | ?? | Maximum order of the MAR model; if zero, an HMM with Gaussian observations is trained (mandatory, with no default).  The order is how many past time points go into the MAR model. | @@@ needs to be non zero |
| Zeromean | 0 | if 1, the mean of the time series will not be used to drive the states (default to 1 if order is higher than 0, and 0 otherwise). | will default to 1 as order is >=1 |
| covtype | full | choice of the covariance matrix of the noise; "full" to have a full covariance matrix for each state (with off-diagonal elements different from zero), "sharedfull" to have one full covariance matrix for all states, "diag" to have a diagonal full covariance matrix for each state, and "shareddiag" to have one diagonal covariance matrix for all states (default to "full"). | @@@ |
| standardise | 0 | whether or not to standardise each subject/trial such that each channel has mean equal to zero and standard deviation equal to one (default 1). | Standardised in pre-parcellation |
| standardise\_pc | X | whether or not to standardise each subject/trial such that each principal component (if PCA has been applied by setting options.pca) has mean equal to zero and standard deviation equal to one (default 0). |  |
| verbose | 1 |  |  |
| Fs | 1/0.8 | Sampling frequency Fs is 1/0.8 (frequency is how many pictures per sec) |  |
| DirichletDiag | X | Makes states more sticky. Value of the diagonal of the prior of the transition probability matrix; the higher, the more persistent the states will be (default to 10). Note that this value is relative; the prior competes with the data, in such a way that if we have very long time series, DirichletDiag will have an irrelevant effect unless is set to a very big value. |  |
| cyc | ? | maximum number of variational inference cycles. The algorithm with stop earlier if tol is reached. (Default to 1000). | @@ |
| initrep | ? | number of repetitions of the initialisation algorithm, out of which the best will be used as a starting point for the variational inference (default to 5). |  |
| initcyc | ? | maximum number of optimisation cycles in the initialisation algorithm, per repetition (default is 25). |  |
| pca | X | Dimensionality reduction form initial number to number specified |  |
| timelag | X | the lapse between lags; for example, timelag==2 skips one sample for each sample that is taken, time\_lag=3 skips 2, etc (default to 1). |  |
| exptimelag | X | base for the exponential spacing of regressor samples. Samples are spaced by round(exptimelags^n) with n=0,1,2, never going further in the past than indicated by order. To check which past samples will be used, use the function formorders(); for finding out which value of exptimelag is needed to cover until certain frequency using a given number of lags, use higherorder(). If a value for exptimelag higher than 1 is specified, then timelag is ignored. |  |
| orderoffset | X | offset to set the starting lag. For example, assuming order==4 and timelag==1, we will use lags {1,2,3,4} for orderoffset=0, or lags {3,4} for orderoffset==2. This parameter becomes particularly useful in situations of strong autocorrelations, as for example in MEG (default to 0). |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## AnalyseResults

### Transition Probabilities

The function getTransProbs(hmm) returns the transition probabilities from any state to any other state, without considering the persistence probabilities (i.e. the probability to remain in the same state). The transition probability matrix including the persistence probabilities is contained in hmm.P.

### Fractional Occupancy

using getFractionalOccupancy(Gamma,T,dim). This can refer to either (i) how much time the HMM spends on each state at each time point on average (across trials), or (ii) how much time each subject/trial/session spends in each state (i.e. the average state probability across time, per session or subject). The former, useful for task, is computed when dim=1; the latter, useful to investigate differences in occupancies between subjects, is computed when dim=2.

Very similar to Gamma or a soft state time course.

Fractional Occupancy where dim = 1 is identical to Gamma for one subject pipeline.

|  |  |  |  |
| --- | --- | --- | --- |
| Setting | Set | Definition | Explanation |
| dim | 1 | 1 = how much time the HMM spends on each state at each time point on average (across trials). Useful for task.  2 = how much time each subject/trial/session spends in each state (i.e. the average state probability across time, per session or subject). Useful for differences in occupancies between subjects. | Currently only one subject, so 1 and 2 should give same output, will need to revisit when running whole dataset through, probably still 1 |

### Viterbi Path

i.e. hard state time course