DOT-HUB Toolbox Manual

Draft 1. RJC, 01/05/2020

Introduction

The DOT-HUB toolbox brings together several years of code development to implement diffuse optical tomography (DOT), and the processes around it. The toolbox is designed primarily to simplify and standardize the reconstruction processes being undertaken within and in collaboration with the UCL DOT-HUB research group.

The toolbox consists of a range of Matlab functions and scripts for the pre-processing, structuring, reconstruction and visualization of DOT data. There is no GUI. Instead, there are a few fundamental functions and filetypes that make the process of going from raw, channel-wise intensity data to 3D images as simple and flexible as possible.

This toolbox is not designed to do everything in the DOT analysis pipeline. Instead it is designed to fill the gap that exists between the extensive channel-wise processing provided by packages like Homer2 and the statistical analysis of image-space data (which are the ultimate outputs of most all neuroimaging experiments).

The guts of the DOT-HUB toolbox are designed to enable pre-processed, channel-wise fNIRS/DOT data to be converted into 3D volume and GM surface images of changes in oxy- and deoxy-haemoglobin concentration.

The fundamental steps of the pipeline are each associated with an output file type, and there is a 'write' function for each file type. This enforces the required organizational structure. There are functions that perform each of the main steps of the pipeline and then write the appropriate files, but in theory you can (using the 'write' functions) create these files yourself and plug in to the pipeline at any step.

Main steps

See DOT-HUB_toolbox_Structure.pptx for flow-charts. To start you need:

- **A .nirs file**, containing your data. This is exactly as per homer2
- **A .SD3D file**. This file contains an SD3D structure, which is exactly as per the homer2 SD structure, but with SD3D.SrcPos and SD3D.DetPos representing the 3D positions of the optodes when recording and SD3D.Landmarks representing the cranial landmarks (Nz, Iz Ar, Al, Cz). These may be subject specific and derived from Polhemus or photogrammetry data, or you may have a single .SD3D file for a group of subjects (e.g. if you are using a single head model for all)

[1] *Pre-processing -> .prepro file*

The pre-processing steps that take raw intensity DOT/fNIRS data are not covered by the pipeline directly, but there is a plug-in for running a Homer2 process stream (via a .cfg file). Pre-processing steps will be things like filtering and motion correction. The output must contains an optical density variable that is the data to be

reconstructed, already re-reference to whatever baseline the user wishes to reconstruct against. The dod variable to be reconstructed can be a pre-averaged HRF, (if your pre-processing pipeline includes e.g. hmrBlockAverage) or it can be the full timecourse of data. By default, the .prepro file will/should take the name of the associated .nirs file.

[2] *Mesh registration -> .rmap file*

Every dataset needs a space for reconstruction to occur in, and source and detector locations registered to that space. We use finite element meshes. The registered mesh and positions file (.rmap) contains the mesh that is to be used for reconstruction for a given subject and the optode locations that match the data acquired in that subject registered to that mesh. The mesh itself can be derived from an atlas or a subject-specific head model in (a .mshs format). By default the .rmap file will/should take the name of the associated .SD3D file.

[3] *Make Jacobian -> .jac file*

Given an rmap file, we can create a forward model. This is currently done with TOAST++. The .jac file contains the Jacobian in the relevant space (full volume mesh, basis) and in the GM surface. By default the .jac file will/should take the name of the associated .rmap file.

[4] *Invert Jacobian -> .invjac file*

The pipeline accommodates various solutions to the inverse problem (standard, multispectral, cortically constrained) and various forms of regularization (Tikhonov, covariance, spatially varying). This step can be run independently of reconstruction if it is appropriate and desirable to save the inverted Jacobian for future use. If this is not needed, step [4] can be skipped, as it is called by step [5] anyway. By default the .invjac file will/should take the name of the associated .jac file.

[5] *Image Reconstruction -> .dotimg file*

This function takes the data from .prepro, and the inverted jacobian (.invjac) (or the .jac and calls step [4]) and creates images of the desired type (haem, mua or both) in the desired space (volume mesh, gm surface mesh or both), and saves the result into a .dotimg file. By default, the .dotimg file will/should take the name of the associated .prepro file.

File types

Each file type in the pipeline is a .mat file with a bespoke extension. The writer function for each filetype outputs a structure (into which each file type is organized, e.g. the structure 'prepro') and the path of the written file (e.g. Users/RCooper/Subject1/Subject1.prepro). When loading any filetype, it is helpful to load it into a structure so it can be parsed to other pipeline functions. E.g. prepro = load('Users/RCooper/Subject1/Subject1.prepro','-mat').

What follows are the definitions of the content of each filetype (or equivalently, structure). Note that each file type contains a logData cell array. This is automatically populated by the main functions of the pipeline, and helps keep track of file structures and the options used for reconstruction.

.prepro file / structure

A file definition for data that has been pre-processed for reconstruction.

prepro.logData

(Optional). logData is a cell array of strings containing useful info. Recommend contents can be found in the comments on the writer function. Parse empty to ignore.

prepro.dod

Change in absorbance (natural log definition). As per Homer2 definition (time x channel) if full timecourse, or time x channel x condition for multi-condition HRF. This is the data that will be reconstructed, and should already have the 'background' subtracted from it. This is intrinsically the case if dod is HRF data from a Homer2-style pipeline, as the HRF is a change relative to the baseline period. For a full timecourse, data is usually reconstructed relative to the mean, which again, is intrinsic to the homer2-style dod variable.

prepro.SD3D

The source-detector structure (Homer2 style) with 3D optode positions in SD3D.SrcPos and SD3D.DetPos, and cranial landmarks in SD.Landmarks (Nz, Iz Ar, Al, Cz)

prepro.tDOD

The time vector in seconds corresponding to the first dimension of dod.

prepro.fileName

the path of the .prepro file to which this prepro structure was saved (useful downstream)

prepro.s

(Optional) Stimulus vector from .nirs file (needed if planning to create HRFs in image space)

prepro.dcAvg

(Optional) Change in concentration HRF data. As per Homer2 definition (time x chromophore x channel) for one condition, or time x channel x condition for multi-condition HRF

prepro.dcAvgStd

(Optional) Std of change in concentration HRF. As per Homer2 definition (time x chromophore x channel) for one condition,or time x channel x condition for multi-condition HRF

prepro.tHRF

(Optional) The time vector in seconds corresponding to the dcAvg HRF data (this can be the same as tDOD if only saving HRF data or can be different if saving both HRF and full time course.

.mshs file / structure

A file type to contain meshes created from a given head model. Could be from an atlas or from an individual. Format is similar to rmap, but this is not for registered meshes and contains no optode information. It is simply a format to keep FEM in.

mshs.headVolumeMesh

The multi-layer volume mesh structure. Contains fields node (nNode x 4, where the first three columns are x,y,z coordinates in mm, and the fourth column is tissue index); face (nFace x 3 for surfaces); elem (nElem x 5 for tetrahedral, plus 5th column for tissue index); and labels (a cell of strings listing the tissue labels corresponding to the indices in the last column of .node. Expected to be a combination of 'ECT','Scalp,'Skull','CSF','GM','WM'.

mshs.gmSurfaceMesh

The gm surface mesh structure, registered to the relevant individual. Contains fields:node, face.

mshs.scalpSurfaceMesh

The scalp surface mesh structure, registered to the relevant individual. Contains fields: node, face.

rmap.vol2gm

The sparse matrix mapping from head volume mesh space to GM surface mesh space

*mshs.*Landmarks

A matrix of cranial landmark positions on this mesh. (5x3: Nz, Iz Ar, Al, Cz)

*mshs.*tenFive

The 10-5 locations for this mesh: .positions (n x 3) 'labels {n}.

.rmap file / structure

A file containing all meshes for a given head model, and a SD2D structure of optode locations registered to that mesh. Contains everything required for undertaking light transport modelling.

rmap.logData

(Optional). logData is a cell array of strings containing useful info. Recommend contents can be found in the comments on the writer function. Parse empty to ignore.

rmap.SD3Dmesh

The source-detector structure (Homer2 style) with 3D optode positions registered to the mesh structures in SD3D.SrcPos and SD3D.DetPos, and cranial landmarks in SD.Landmarks (Nz, Iz Ar, Al, Cz). This may differ from the parsed SD3D file because of the registration process.

rmap.headVolumeMesh

The multi-layer volume mesh structure, registered to the relevant individual. Contains fields node (nNode x 4, where the first three columns are x,y,z coordinates in mm, and the fourth column is tissue index); face (nFace x 3 for surfaces); elem (nElem x 5 for tetrahedral, plus 5th column for tissue index); and labels (a cell of strings listing the tissue labels corresponding to the indices in the last column of .node. Expected to be a combination of 'ECT','Scalp,'Skull','CSF','GM','WM'.

rmap.gmSurfaceMesh

The gm surface mesh structure, registered to the relevant individual. Contains fields:node, face.

rmap.scalpSurfaceMesh

The scalp surface mesh structure, registered to the relevant individual. Contains fields: node, face.

rmap.vol2gm

The sparse matrix mapping from head volume mesh space to GM surface mesh space

rmap.fileName

the path of the .rmap file to which this rmap structure was saved (useful downstream)

.jac file / structure

A file containing the Jacobian or Jacobians, in one of several forms.

jac.logData

(Optional). logData is a cell array of strings containing useful info. Recommend contents can be found in the comments on the writer function. Parse empty to ignore.

jac.J

The Jacobian structure containing J{i}.vol, J{i}.basis and J{i}.gm, where i = wavelength index from 1 to the number of wavelengths. If the Jacobian was calculated using a basis, J{i}.basis will be populated. If the Jacobian was calculated on the full volume mesh, J{i}.vol will be populated (and have dimensions nChan x nVolumeNodes). J{1}.gm is always populated, and is the vol2gm extracted of the full volume Jacobian (useful for visualization, masking etc.).

jac.basis

The dimensions of the basis used. Empty if basis not used.

jac.fileName

the path of the .jac file to which this jac structure was saved (useful downstream)

.invjac file / structure

A file containing the inverted Jacobian(s), allowing fast reconstruction. Note that .invjac file is the only file type in this pipeline that is not actually required, as saving it to disk is not always necessary (or appropriate).

invjac.logData

(Optional). logData is a cell array of strings containing useful info. Recommend contents can be found in the comments on the writer function. Parse empty to ignore.

invjac.invJ

The inverted Jacobian(s) in a cell array: invJ{i} will be the inverted Jacobian for wavelength i (dimensions space x nChan) if inversion was not multispectral. If inversion was multispectral, length(invjac.invJ) == 1, and that single entry is the inverted multispectral Jacobian of dimensions ([2xspace] x [2xnChan]). The space in which the inverted Jacobian exists will depend on the inversion call, but will either be the basis (if J was built in a basis), the full volume, of the GM surface for cortically constrained reconstruction.

invjac.basis

The dimensions of the basis used. Empty if basis not used.

invjac.fileName

the path of the .invjac file to which this invjac structure was saved (useful downstream)

.dotimg file / structure

A file containing the reconstructed data of the requested form.

dotimg.logData

(Optional). logData is a cell array of strings containing useful info. Recommend contents can be found in the comments on the writer function. Parse empty to ignore.

dotimg.hbo

A structure containing image(s) of the change in concentration of oxyhaemoglobin in units of micromolar. Consists of .vol and .gm fields, each of dimensions nFrames x nNodes. Note that both can be empty if haemoglobin images were not requested and .vol can be empty if 'saveVolumeImages' option set to false in DOTHUB_reconstruction.

dotimg.hbr

A structure containing image(s) of the change in concentration of deoxyhaemoglobin in units of micromolar. Consists of .vol and .gm fields, each of dimensions nFrames x nNodes. Note that both can be empty if haemoglobin images were not requested and .vol can be empty if 'saveVolumeImages' option set to false in DOTHUB_reconstruction.

dotimg.mua

A cell of length nWavs of structures .vol and .gm, where change in absorption coefficient images are stored. Either .vol of both .vol and .gm can be empty if not requested.

dotimg.tImg

A time vector associated with the first dimension (nFrames) of the images series.

dotimg.fileName

the path of the .dotimg file to which this dotimg structure was saved (if saved)