**Pipeline Review 2020 – Issues**

ToDo

* Vince will look at group templates to figure out mapping to MNI
* Make autoseg batch file call .prn file so common input file
* Resolution in header from AtlasViewer – Sobana and Vince will sort with Courtney’s data
* Script to build default input file from a list of subjects + an assumed file structure for data; but also leave flexibility for advanced users; bids file structure. John looks at bids. Maybe stick with csv but use structure from bids.
* Vince will fix current input read script so reads 1 vs multiple subjects
* John: 2mm resolution applied to Adot file; need to also create 2mm headvol there. Note in paper that we’re picking 2mm as a reasonable value but it can be changed here.
* NIRX converter has an issue.
* John: check new .nirs format – maybe have a flag for file format.
* Sobana fix endframe and switch on sign on HRF for HbR
* John figure out how to run ImageRecon on NIH data with multiple sessions.
* John: fix runCt in GLM code
  + Need to run all data for Gates through a script to calc total of runCt so I can scale the betas accordingly – specific to number of regressors with values, so could vary by run. Non-trivial.
  + Fix is easy…just flag ‘doGLM’ and runCt once per loop. But check if this is right – just divide by runs relevant for each regressor? Could that vary?
* John: for both ImageRecon and GLM edit so keeps running if a file is missing and writes problems to a log file.
* John/Vince: Some bits weren’t running first subject – why?
* In paper, will write analysis section generically and provide sample scripts in AFNI

John Data Dump of Issues

* Orientation problem with Courtney and Sara’s data – we have a fix, but why did this occur? AtlasViewer issue?
* Need to document all the front-end decisions about wavelength-specific issues / extinction coeffs and whether extinction coeffs vary by age; also spatial dimension of SD distances (cm vs mm); where does SDgui come in and how precise does that have to be?
* Need imageRecon and RunGLM to run without aborting and write any problems to a log
  + Just need to check existence of each file and write to a log if missing?
* How deal with two session NIH data? Imagerecon for each run at present and then run GLM over multiple image recon files…but two runs should use light model for session 1 and two runs for session 2 light model
  + I think this would work if have two rows for subjects with multiple light models and then put the relevant .nirs runs into separate folders, so light model 1 is linked with folder 1 and light model 2 is linked with folder 2.
  + Could write some code to sort .nirs files by date?
  + I think this is the only step where this is needed? Hmm: would need this duplication for the invert step as well since different ND file for each run and, therefore, different headvols. So best to have one input file for all steps and select ‘unique’ IDs.
* Zscore design matrix for GLM? – see Adam’s email
* Input file is unwieldy – can we do anything there or just leave as is – do we need all those columns?
* Fix Vince’s file read; plays ok with my ‘unique’ code; when apply ‘unique’ code
  + Fix code so single subject run works
* Some bits weren’t running first subject – why?
* Set up GLM for Gates so I can do which analyses?
  + Load
  + First look change v nochange?
  + SLF
  + Others? Easy way to edit design matrix down the road?
* Any way to streamline some of the AFNI code steps, or is that as is…
  + E.g., create group mask, concat, etc.
  + What role will these steps play in pipeline?
  + Put all this example code in an AFNI Example folder in github. Maybe just provide one sample set? I suppose could do multiple—one for each sample project…
* Set up code to extract time series without the average? E.g., from stim1 to stim2? Useful for plots and additional analyses like coherence
* Redo extract code so one file per clust\_order file; files will be smaller and can be added to git
* We used diff files for Y1 and Y2 tform to group space – is that all ok? Needs cleaning?
* Final template issue: can we pull the MNI labels into IndiaFinal and CustomMNI? Or put all Gates and NIH data into MNI as a final step?
  + Vince to do: look at ‘templates’ folder on dropbox to recommend common space
* Create 2mm sampling of headvol in initial steps and copy to relevant folder rather than doing this in registerCommon\_Invert
* Check and fix scaling of betas—fixes in multiple places…just go right to micromolar.
* Fix HRF used for HbR

Hard coded items in pipeline – need to fix?

* ImageRecon:
  + Line 74: /viewer/Subject structure to specify location of Adot files…comes from AtlasViewer? Or transformAnat?
* Threshold ok? Line 82: info.tissue.dim.Good\_Vox=find(sum(A,2)>(aM\*1e-5)); % set threshold here

Is this all ok across systems? Ok to hard code 30mm?

meas=size(SD.MeasList,1);

ch=meas/2;

%%%Can't find in .nirs file...grr. Hardcoding here.

info.pairs=table;

info.pairs.Src=SD.MeasList(:,1);

info.pairs.Det=SD.MeasList(:,2);

info.pairs.WL=SD.MeasList(:,4);

info.pairs.lambda=cat(1,ones(ch,1).\*procInput.SD.Lambda(1), ones(ch,1).\*procInput.SD.Lambda(2));

info.pairs.NN=ones(meas,1);

info.pairs.Mod=repmat({'CW'},[meas,1]);

info.pairs.r2d=ones(meas,1).\*30; %%30MM 2D AND 3D DISTANCE BETWEEN PAIRS

info.pairs.r3d=ones(meas,1).\*30; %%30MM 2D AND 3D DISTANCE BETWEEN PAIRS

Do other systems prune channels in the same way? Is this robust in Homer2?

%%update so pruning channels based on bad channels on either wavelength

for ct=1:ch

if (procResult.SD.MeasListAct(ct) == 0 | procResult.SD.MeasListAct(ct+ch) == 0)

procResult.SD.MeasListAct(ct) = 0;

procResult.SD.MeasListAct(ct+ch) = 0;

end

end

info.MEAS.GI=procResult.SD.MeasListAct;

Questions for Adam/Vince:

* Use age-specific extinction coeffs?
* Are these wavelength specific (e.g., for NIRX)?
* A light thresholding is applied to the sensitivity volumes (0.000001). IS this the only thresholding needed? See script…
* Any other machine-specific values that Adam can think of…
* Zscore design matrix for GLM? – see Adam’s email

Sobana’s questions:

**In the ImageRecon code:**

1.

 params.lambda\_1=0.1;%range between 0.2-0.01--smoothness vs variance

params.lambda\_2=0.1;%range between 0.2-0.01--what is this??

params.gsigma=3;% standard deviation of Gaussian smoothing kernel in mm

Can you clarify where these values come from? I want to make sure that unlike the Extinction coefs, these are not numbers that will need to change for the NIRX system that I use.

2.

       if (endframe > size(procResult.s,1))

            endframe = size(procResult.s,1);

        end

It looks like with using these lines, the last stim marker info is excluded if padding is 0? So, in my case, I specify padding as 0, so my last stim marker gets chopped off. Is there a way to repair this?

3.

hrf=resample\_tts(hrf,infoHRF,newSamplingFreq,1e-3,1);

Do you anticipate the tolerance of 10^-3 to change as per machine?

4.

The script appears to work when I increase the number of %s at the very beginning to accept 15 fields in the .prn fille as the pipeline now expects. If I don’t do this, funky things get done withSubjectList and I get an error downstream. Can you clarify this?

5.

For the IC dataset, I plan to use the same HRF for HbO and HbR – simply because my collaborators and I are using latent change score models, and the interpretation of ‘change’ gets very complicated very quickly if there is more layers of signs one needs to keep in mind. Can you clarify that using the same HRF is fine – as long as we are clear on interpretation as detailed in the email correspondence between Adam, you and I?

6.

            cortex\_HbO = cortex\_HbO.\*1000;

            cortex\_HbR = cortex\_HbR.\*1000;

I notice you multiply this data with 1000, and you have a comment referring to Adam’s email. I was under the impression that this needed to be done at the back-end. Can you clarify what this 1000 does, and then, what the other 1000 is on top of this?

**In the RunGLM code:**

1.

        for j=1:numRegressors

            if ~isempty(info.paradigm.(['Pulse\_',num2str(regressorListND(j))]))

                doGLM = 1;

                runCt = runCt+1;

            end

        end

  b\_HbO = b\_HbO ./ runCt;

    b\_HbR = b\_HbR ./ runCt;

runCt gets calculated and then divides the final beta images to calculate the average beta map as specified by the lines above. However, upon running stuff line by line, I find that runCt is counting regressors instead of runs. For eg. for the subject from the IC data, we have 4 regressors, and 2 runs. When we get to the point where we have to calculate the average betamap (last two lines above), I ended up with runCt = 8. Something is off, I think?

2. Any fixes that are required for different machines?