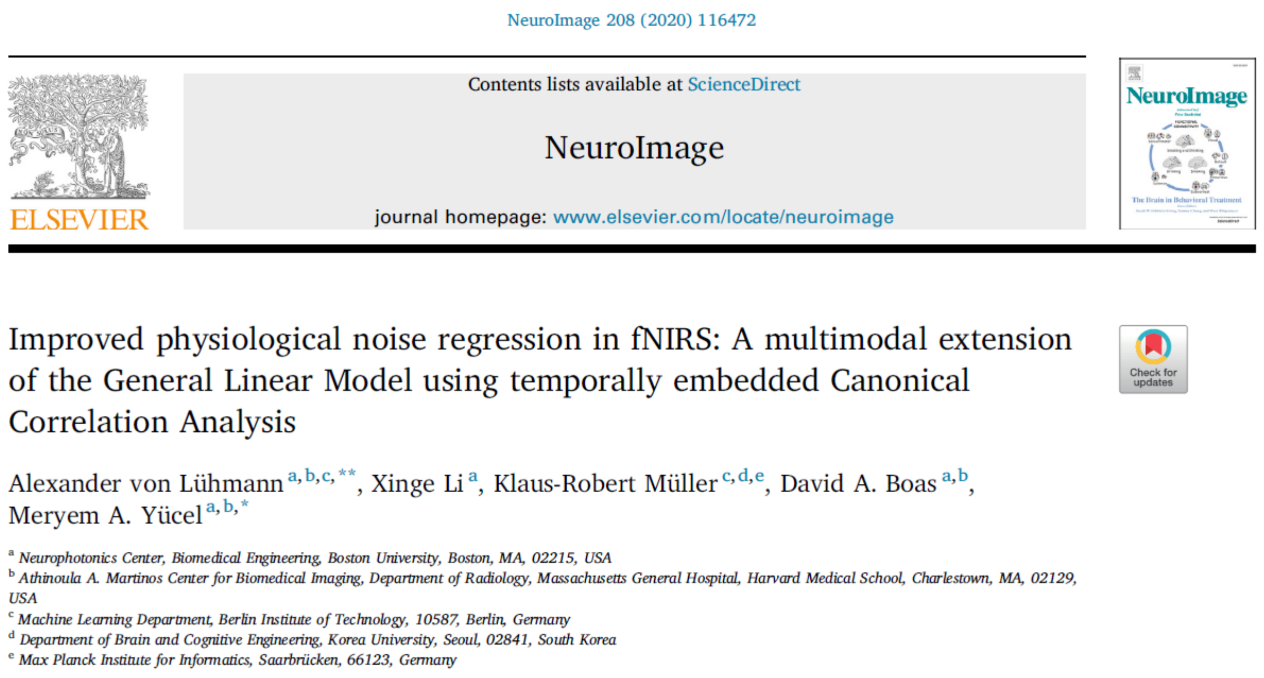
tCCA extension for the General Linear Model in Homer3: A brief introduction

version May 2020, AvL (avolu@bu.edu)

This document provides a brief introduction to the use of temporally embedded Canonical Correlation Analysis (tCCA) to generate more optimal nuisance regressors for the General Linear Model in fNIRS using the Homer3 toolbox. The method and its extensive evaluation and validation were published in NeuroImage 208 (2020), the manuscript can be accessed here (DOI: 10.1016/j.neuroimage.2019.116472):



NeuroImage 208 (2020) 116472

Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/neuroimage

Improved physiological noise regression in fNIRS: A multimodal extension of the General Linear Model using temporally embedded Canonical Correlation Analysis

Alexander von Lühmann[a,b,c,**], Xinge Li[a], Klaus-Robert Müller[c,d,e], David A. Boas[a,b], Meryem A. Yücel[a,b,*]

[a] Neurophotonics Center, Biomedical Engineering, Boston University, Boston, MA, 02215, USA
[b] Athinoula A. Martinos Center for Biomedical Imaging, Department of Radiology, Massachusetts General Hospital, Harvard Medical School, Charlestown, MA, 02129, USA
[c] Machine Learning Department, Berlin Institute of Technology, 10587, Berlin, Germany
[d] Department of Brain and Cognitive Engineering, Korea University, Seoul, 02841, South Korea
[e] Max Planck Institute for Informatics, Saarbrücken, 66123, Germany

What does the method do?

- It reduces nuisance signals in fNIRS, which leads to a more robust estimation of evoked brain activity.
- GLM with tCCA flexibly combines any available auxiliary signals into optimal nuisance regressors. The most important signals to use are short separation fNIRS channels and motion (accelerometer) measured at the head.
- The method significantly improves upon conventional GLM with short separation regression and can be used comparatively easily to extend existing processing pipelines.
- GLM with tCCA improves HRF recovery particularly for low Contrast to Noise Ratios and low number of stimuli/trials.

How does it work?

The method consists of (at least) two different steps/runs:

1. A (machine) **learning run**, in which a resting state measurement (training run) is used to learn tCCA filters that generate the optimal nuisance regressors by linearly mixing time embedded (shifted copies of) auxiliary signals. It is important to use resting state data – not experimental data with stimuli that generate brain responses – to ensure the nuisance regressors fit physiological noise only. The tCCA filter matrix is saved in the group/subject folder as a text file named "tCCAfilter_1_xxxxx.txt", where 'xxxxx' is a tCCA parameter configuration checksum. If you change tCCA parameters, or delete the file, this filter will be newly trained.
2. The **analysis run(s)**, in which the tCCA filters are used to generate nuisance regressors for the subsequent GLM analysis of the fNIRS data that contains stimuli and corresponding brain responses.

In Homer3, this is incorporated by specifying the index of the resting run to use for training, and then running a whole subject's session. The index of the resting run can be determined by checking the order of the resting run as it appears on the "current processing element" list on Homer3 main GUI. The session should at least contain the resting run, and one analysis run. If the resting run is not the first in the processing order, the training will only happen once the resting run is analyzed. All other runs up to the training run are skipped. In this case, after initial execution, you will get a warning pop-up window. Just **re-run** the session to get complete results.

What to watch out for?

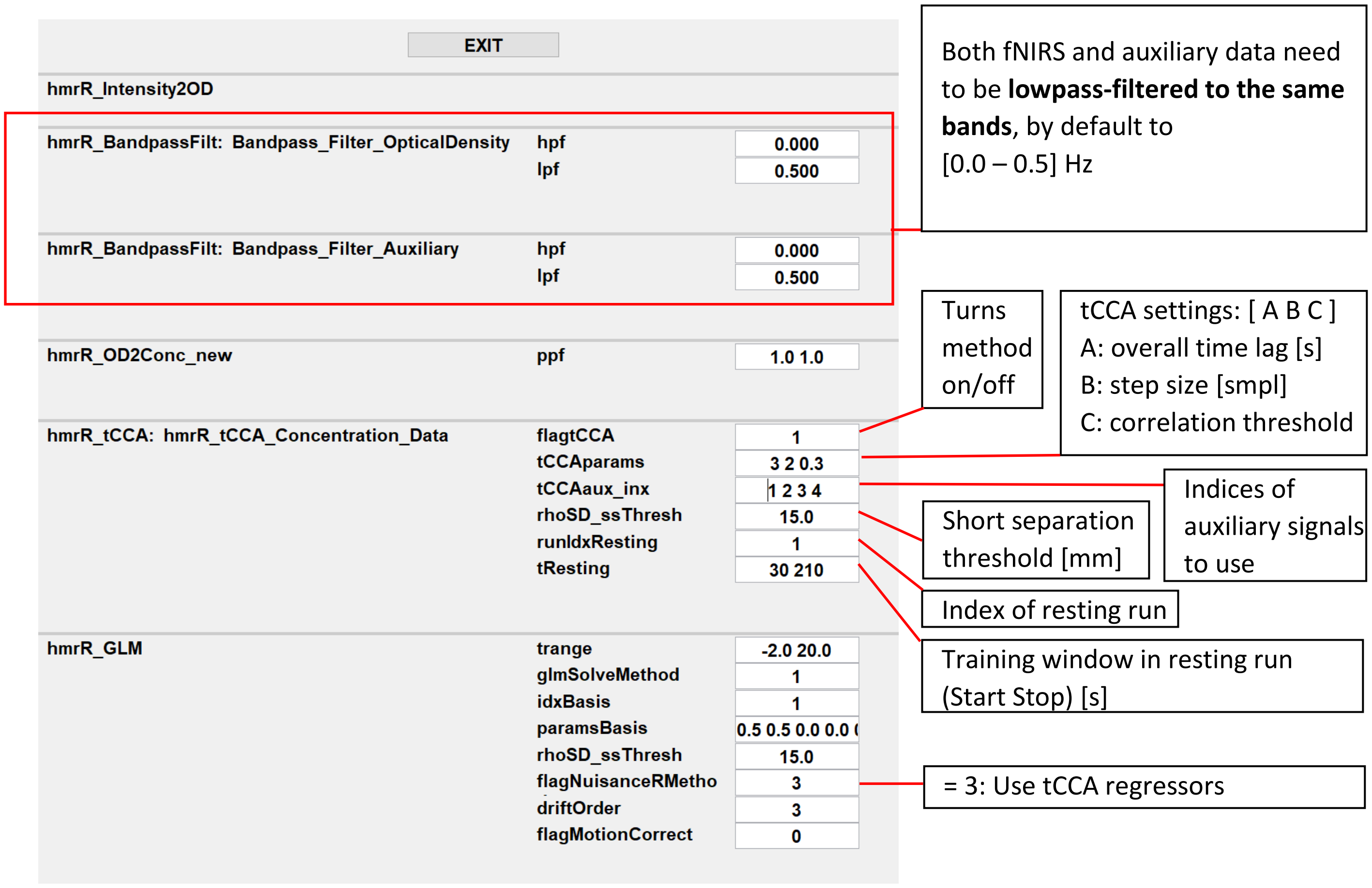
The tCCA approach is a machine learning based method with several hyperparameters. This means that the selection of the parameters can have a big impact on the performance of the method. For an exploration of the parameter space and more details, please see the NeuroImage paper cited above. In general, it is good to keep the following points in mind:

1. <u>Try to avoid overfitting</u>. If the nuisance regressors overfit the fNIRS signals, the HRFs will be underestimated. One indicator for this is a warning pop up that the tCCA function will display when all regressors are above the user set correlation threshold (no tCCA regressors are rejected). Overfitting is more likely to occur, when
    a. The tCCA parameter set yields a *large number of temporal embeddings*: This happens when you use a long absolute time lag and a small step size.
    b. The *correlation threshold is very low*: This will result in a larger number of tCCA regressors being handed over to the GLM.
    c. *Multiple short separation channels are available*: Since short and long separation fNIRS channels are very similar, temporal embedding of multiple SS channels can lead to overfitting. In this case consider setting the parameter ctr > 1 to a fixed number of regressors that you want to use with the GLM.
    d. The *ratio of tCCA nuisance regressors and HRF regressors is bad*: For instance, if you use the GLM with canonical HRF regressor and not gaussian basis functions, you might want to reduce the number of tCCA regressors.
    e. The auxiliary signals have *high frequency components* that were not filtered out well and the number of temporal embeddings is large.

2. <u>Use meaningful resting data</u>:
   a. If the resting data contains many (non-rejected) motion artifacts, the training might be dominated by those spikes and the resulting regressors will be less optimal.
   b. If you want to capture (and regress) slow effects of movement on fNIRS nuisance signals, use resting state data that contains such segments.
   c. Don't use too short (<60s) or very long (>10min) segments for training (these values are experimental and have not been empirically investigated so far).

How to incorporate tCCA into your Homer3 processing stream?

You can simply add the hmrR_tCCA function to your processing stream, usually directly preceding the hmR_GLM function. Please refer to the example processing streams **tCCA_xmpl_procStream_Gauss.cfg**, and **tCCA_xmpl_procStream_Canonical.cfg** that are provided with this documentation under …\Homer3\FuncRegistry\UserFunctions\tcca_glm\ and explained in the following:



hmrR_Intensity2OD		
hmrR_BandpassFilt: Bandpass_Filter_OpticalDensity	hpf	0.000
	lpf	0.500
hmrR_BandpassFilt: Bandpass_Filter_Auxiliary	hpf	0.000
	lpf	0.500
hmrR_OD2Conc_new	ppf	1.0 1.0
hmrR_tCCA: hmrR_tCCA_Concentration_Data	flagtCCA	1
	tCCAparams	3 2 0.3
	tCCAaux_inx	1 2 3 4
	rhoSD_ssThresh	15.0
	runIdxResting	1
	tResting	30 210
hmrR_GLM	trange	-2.0 20.0
	glmSolveMethod	1
	idxBasis	1
	paramsBasis	0.5 0.5 0.0 0.0
	rhoSD_ssThresh	15.0
	flagNuisanceRMetho	3
	driftOrder	3
	flagMotionCorrect	0

Both fNIRS and auxiliary data need to be **lowpass-filtered to the same bands**, by default to [0.0 – 0.5] Hz

Turns method on/off

tCCA settings: [A B C]
A: overall time lag [s]
B: step size [smpl]
C: correlation threshold

Indices of auxiliary signals to use

Short separation threshold [mm]

Index of resting run

Training window in resting run (Start Stop) [s]

= 3: Use tCCA regressors

Please note that by default we do not bandpass filter the data, but only low-pass filter by setting hpf = 0. The low frequencies are rejected by polynomial drift regression in the GLM, and the slow signals in the tCCA regressors. Also, please note that by default we use gaussian basis functions (idxBasis =1) in the GLM. To use the tCCA regressors in the hmrR_GLM function, the flagNuisanceRMethod needs to be set to 3.

Remarks on parameter selection

BandpassFilt**:**	You can choose any typical fNIRS band here. Please note that the tCCA step might perform less good when higher frequency noise is not excluded (lpf high)
flagtCCA:	Turns the tCCA method on (1) or off (0).
tCCAparams:	The main parameters for tCCA analysis, as identified and evaluated in the NeuroImage publication: [timeLag stepSize corrThresh] timeLag: overall timespan in seconds, across which temporal embedding is performed. Default: 3s. stepSize: the step size (Δt) in samples for each time shift of auxiliary signals in the temporal embedding step. NOTE: The effective time shift will depend on your data's sample frequency! Default: 2 samples corrThresh: Correlation threshold. If <=1, this threshold will keep only those tCCA regressors for the GLM nuisance removal that have a canonical correlation with fNIRS signals in the CCA space that is greater than corrThresh. If set to 0, all regressors will be used. Default: 0.3. If corrThresh is set to an integer >1, a fixed number of tCCA regressors set by the user will be used in the GLM, selected by their correlation in descending order. Use this to avoid overfitting or to have better control over the total number and ratio of regressors in the GLM.
tCCAaux_inx:	indices of the auxiliary signals in the .snirf file that are to be used for the tCCA regression. NOTE: to not introduce additional noise, exclude auxiliary channels that are noisy or do not carry meaningful signals.
rhoSD_ssThresh:	Same parameter as in the hmR_GLM function. Threshold to detect short separation channels in the data. All signals measured with a source-detector separation < rhoSD_ssThresh will be used as short separation channels and added to the auxiliary signals to find nuisance regressors in the tCCA step.
runIdxResting:	Index of the run (dataset) that contains the resting data to be used for the tCCA filter training. We recommend to position your resting state data as the first run and set runIdxResting = 1. If your resting run is not the first, the tCCA function will skip all runs up to the training run to generate the tCCA filter and will not output any regressors before the filter is learnt. In that case, simply re-run the whole session after your first execution to generate results for the previous runs as well.
tResting:	Window in seconds [start stop] to be used in the resting state run for tCCA training. We recommend setting the start time to at least a few seconds after the begin of the resting state run to make sure all signals are settled and no major artifacts are present. For better statistical performance, the window should not be shorter than 60s. Default: [30 210] (180s).