Week 6 – First-level fMRI data analysis

L06-07. Residualize (canlab_connectivity_preproc.m)

Hongji Kim
Ph. D. Student in the Cocoan lab

2 April 2021

L06-07. Residualize (canlab_connectivity_preproc.m)

❖ Canlab_connectivity_preproc.m

master | CanlabCore / CanlabCore / @fmri_data / canlab_connectivity_preproc.m | Go to file | ···

wanirepo make the residualize more efficient by reordering the matrix product ✓ | Latest commit 9746178 on 2 Aug 2019 | History

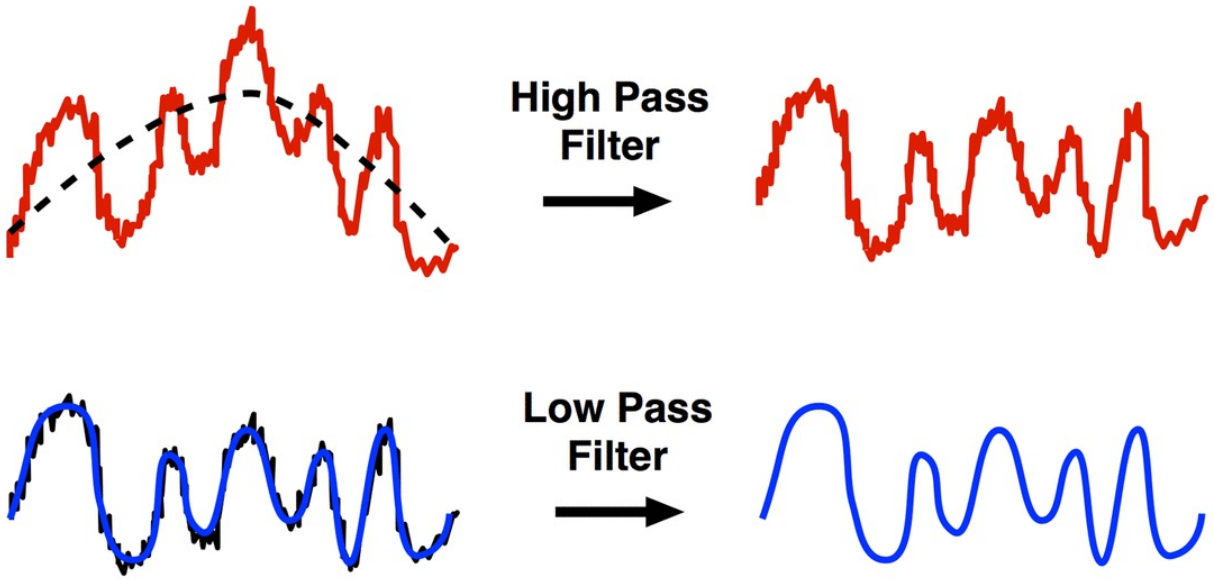
3 contributors

681 lines (542 sloc) | 23.5 KB | Raw | Blame

```
function [dat, roi_val, maskdat, beta_dat, beta_roi_val] = canlab_connectivity_preproc(dat, varargin)

% This function prepares data for connectivity analysis by removing nuisance
% variables and temporal filtering (high, low, or bandpass filter). This also
% can extract values from given masks and return averaged activity or pattern
% expression values.
%
```

- Additional preprocessing
  - ‘**canlab_connectivity_preproc**.m’ function
    - additional_nuisance
    - temporal filtering (e.g., bpf, lpf, hpf)
    - winsorize



L06-07. Residualize (canlab_connectivity_preproc.m)

- Why do you use this instead of spm batch?
  - When trial structure is not clear but you need to residualize using nuisance and apply pass filtering
    - Continuous design (naturalistic design)
    - Connectivity analysis

L06-07. Residualize (canlab_connectivity_preproc.m)

❖ Features

```
roi_masks = which('weights_NSF_grouppred_cvpcr.img');
[preprocessed_dat, roi_val] = canlab_connectivity_preproc(dat, 'vw', 'datdir',
       subject_dir, 'bpf', [.008 .25], TR, 'extract_roi', roi_masks, 'pattern_expression');
```

- can regress out nuisance variables with any additional nuisance matrix
- can remove signal from ventricle and white matter
- can do temporal filtering, including high-pass, low-pass, or bandpass filtering (it uses conn_filter.m from conn toolbox, which is using fast fourier transform)
- can extract data from given ROIs and return averaged value or pattern expression value (dot-product)
- It is possible to extract ROI averages or linear patterns, save them, and run this function afterwards. This will give the same result as filtering each voxel and then calculating the ROI average/pattern response.
- (optional) can run additional GLM model with additional regressors if they are specified. This runs the GLM *additionally*, therefore do not change residualized image data or roi_vals. This GLM uses the same nuisance matrix from above. output: beta_dat

L06-07. Residualize (canlab_connectivity_preproc.m)

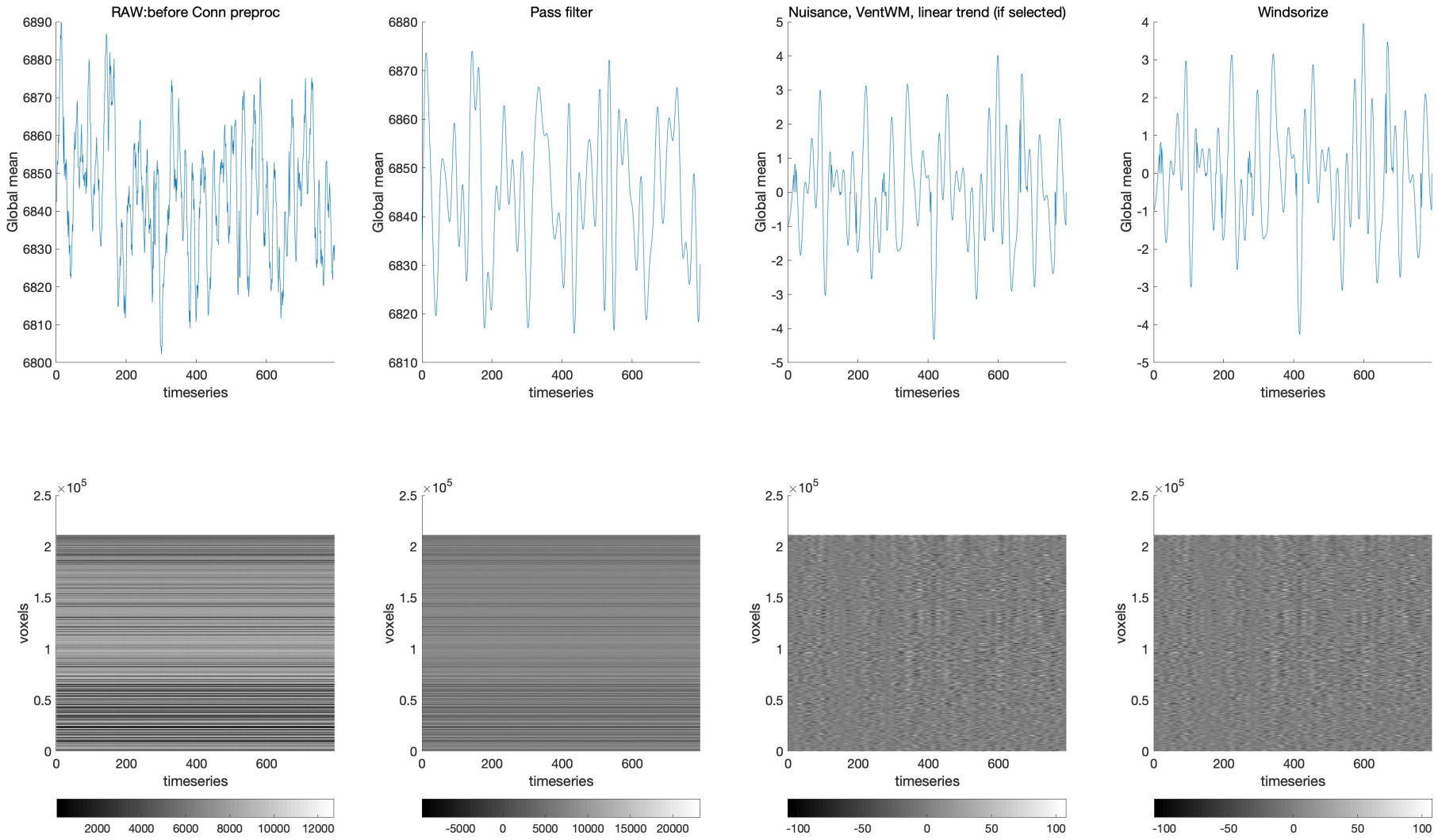
- Steps in order
  1. Remove nuisance covariates (and linear trend if requested)
  2. Remove ventricle and white matter - default: uses canonical images
  3. High/low/bandpass filter on BOTH data and covariates before regression (plot)
  4. Residualization using regression (plot)
  5. (optional) Run additional GLM using the same additional preprocessing -> beta_dat
  6. Winsorize based on distribution of full data matrix (plot)
  7. Extract region-by-region average ROI or pattern expression data

L06-07. Residualize (canlab_connectivity_preproc.m)

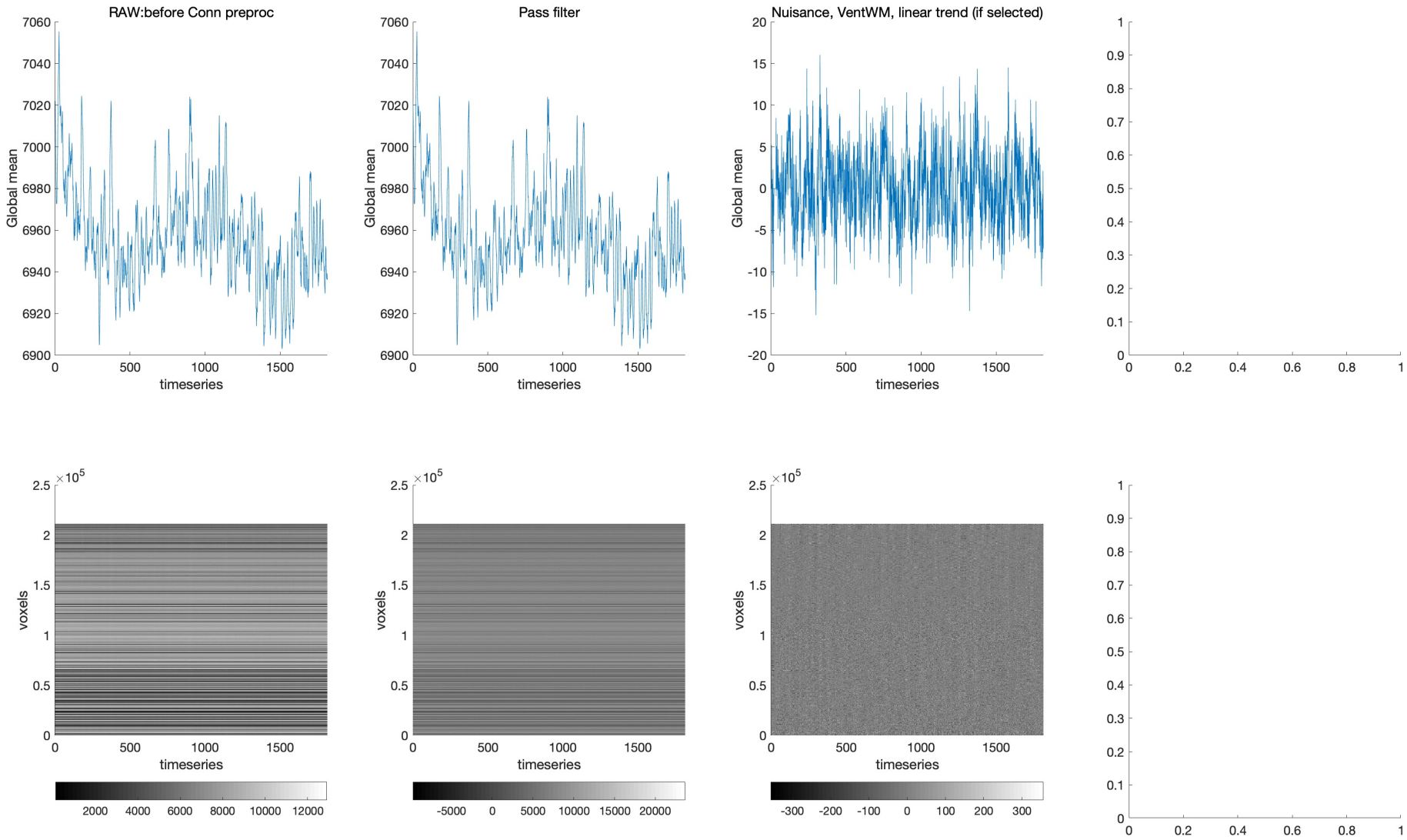
- Codes and plots

```
if run_i == 1 || run_i == 6
    preprocessed_dat = canlab_connectivity_preproc_new(dat, 'bpf', [0.008 0.1], tr, 'windsorize', 5); % no design matrix for resting runs
else
    preprocessed_dat = canlab_connectivity_preproc_new(dat, 'bpf', [0.008 0.1], tr, 'additional_nuisance', additional_reg.X{run_i-1}, 'windsorize', 5);
end
```

L06-07. Residualize (canlab_connectivity_preproc.m)



L06-07. Residualize (canlab_connectivity_preproc.m)



Cocoan 101

https://cocoanlab.github.io