pcarpet Documentation

Release 0.2

Nikoloz Sirmpilatze

Aug 09, 2022

CONTENTS

pcarpet is a small python package that creates a **carpet plot** from fMRI data and decomposes it with **PCA**.

Author: Nikoloz Sirmpilatze (German Primate Center)

Citation: Sirmpilatze et al., 2022

For an overview of the project, please refer to the README file in the Github repository.

Contents:

CHAPTER

ONE

RATIONALE

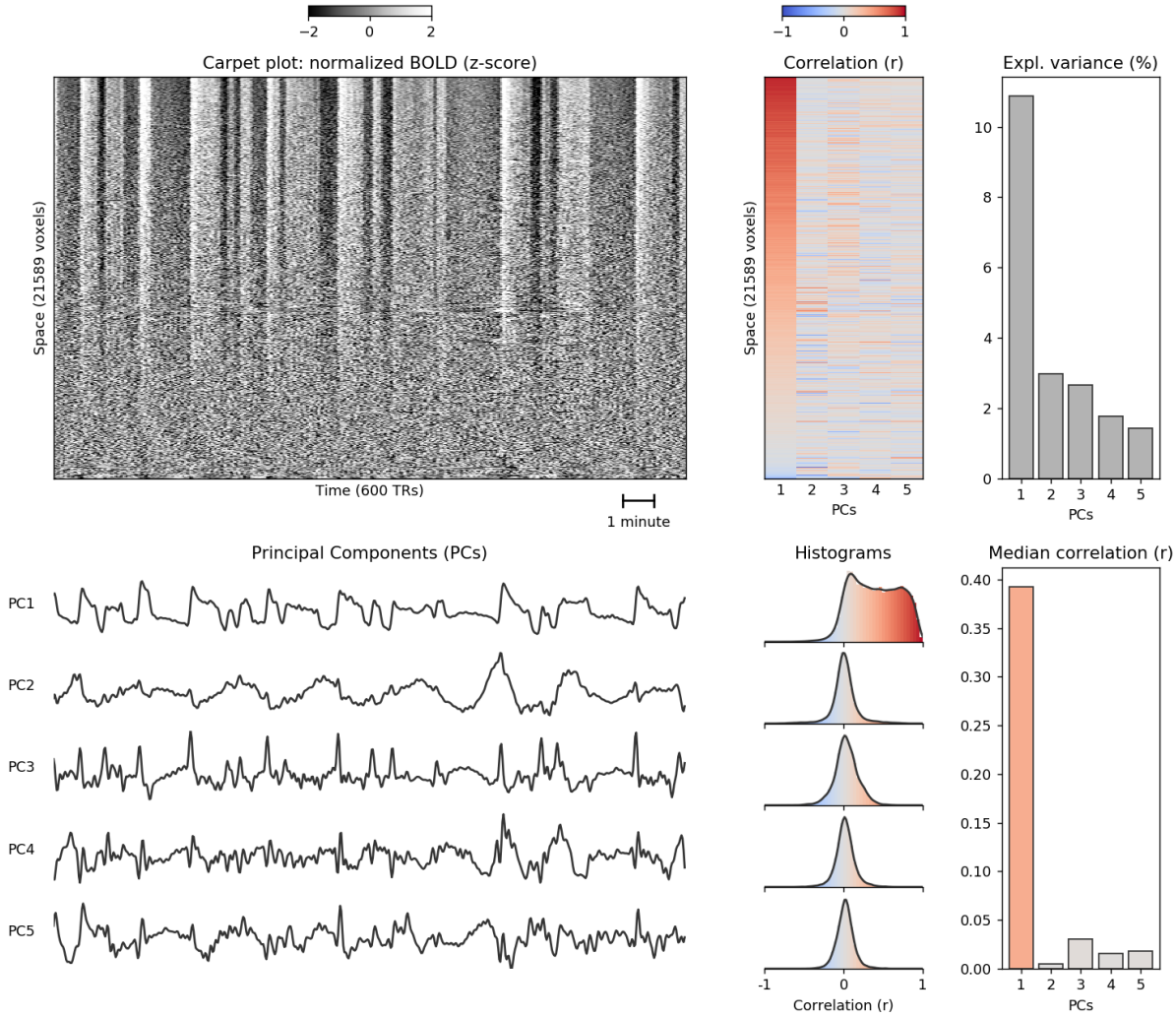
A 'carpet plot' is a 2d representation of fMRI data (voxels x time), very similar to 'The Plot' described by Jonathan D Power (Power 2017). This visual representation of fMRI data is suited for identifying wide-spread signal fluctutations (Aquino et al., 2020), which often come from non-neural sources (e.g. head motion).

That said, the carpet plot can also reveal 'real' neural activity, especially when the activity is slow and synchronous, as is the case for **anesthesia-induced burst-suppression** (Sirmpiltze et al., 2022). The `pcarpet` package implements the analytical pipeline used in the Sirmpiltze et al., 2022 paper to identify instances of burst-suppression in anesthetized humans, nonhuman primates, and rats.

1.1 How it works

The pipeline consists of the following steps:

1. First tha necessary data is imported, consisting of a preprocessed fMRI scan (4d NIFTI file) and a mask (3d NIFTI file) defining a single region-of-interest.
2. A carpet plot is generated from within the mask. To make wide-spread fluctuations more visually prominent, the voxel time-series (carpet rows) are normalized (z-score) and re-ordered according to their correlation with the mean time-series.
3. Principal Component Analysis (PCA) is applied to the carpet matrix (using the `scikit-learn` implementation) and a given number (`ncomp`, default is 5) of first Principal Components - hereafter referred to as 'fPCs' - is extracted. The fPCs (e.g. PC1 - PC5) represent the temporal patterns of activity with the highest explained variance ratios.
4. The fPCs are correlated with all voxel time-series within the carpet to get a distribution of Pearson's correlation coefficients (r) per fPC.
5. The fPCs are also correlated with the entire fMRI scan, including areas outside the mask, to get the brain-wide spatial distribution of each fPC.
6. A visual summary of results from steps 1-4 is plotted (example below).



The above image corresponds to an instance of burst-suppression in a female long-tailed macaque (*Macaca fascicularis*) anesthetized with isoflurane. The carpet plot (using a cortical mask) shows a wide-spread, slow, quasi-periodic signal fluctuation, which is well captured by PC1. PC1 is positively correlated with most cortical voxel timeseries, resulting in a heavily asymmetric distribution of correlation coefficients (r), while PCs 2-4 show symmetric r histograms centered on zero. This property can be quantified by taking the median of carpet-wide r values (bottom right). According to the terminology introduced in Sirmpiltze et al., 2022, PC1 is an 'asymmetric PC`. Under the right circumstances, the presence of an asymmetric PC in a cortical carpet plot can be an fMRI signature of burst-suppression, with the brain-wide distribution of the asymmetric PC representing a map of burst-suppression (see manuscript for details).

CHAPTER

TWO

INSTALLATION

2.1 a. pip

You can install the latest release from PyPI via

```
pip install pcarpet
```

Pip will try to ensure that the following requirements are satisfied:

1. Python 3.6 or higher
2. numpy
3. scipy
4. matplotlib
5. pandas
6. scikit-learn
7. nibabel
8. ipython

2.2 b. Anaconda

If you are having issues with resolving package dependencies, you can create a virtual environment using Anaconda:

1. Install an Anaconda distribution of python 3, choosing your operating system.
2. Download the `environment.yml` file from this repository. You can clone the repository or copy-paste the file contents into a text document on your local computer.
3. Open a terminal/anaconda prompt with conda for python 3 in the path.
4. Navigate to the directory where the `environment.yml` is stored and run `conda env create -f environment.yml`
5. Activate the environment with `conda activate pcarpet-env` (Note: you will always have to activate `pcarpet-env` before using `pcarpet`)

CHAPTER

THREE

RUNNING PCARPET

Here we will demonstrate how to run the pcarpet pipeline data on some example preprocessed fMRI data.

3.1 Inputs and initialization

`pcarpet` needs 3 paths to be specified:

1. **fMRI file**: a 4d nifti file containing the preprocessed fMRI data
2. **Mask file**: a 3d nifti file, containing a binary mask that defines a single region-of-interest (e.g. cortex)
3. **Output folder**: for storing the outputs generated by pcarpet

The fMRI and Mask files need to be in the same space, with identical x-y-z dimensions. For example, they can both be in the native anatomical space of the subject, or in a template space to which the data have been resampled.

[1]:
```
import os
import pcarpet
```

[2]:
```
# Folder containing example data
example_folder = '/home/niko/MRI/pcarpet_example/macaque'

# 1. Path to preprocessed fMRI file
func = os.path.join(example_folder, 'func_preproc.nii.gz')
# 2. Path to a cortical mask
cortex_mask = os.path.join(example_folder, 'cortex_mask.nii.gz')
# 3. Path to a folder for storing the outputs
output_folder = os.path.join(example_folder, 'outputs')
```

To initialize pcarpet, we first create `Dataset` object, using the three paths from above

[3]:
```
MyData = pcarpet.Dataset(func, cortex_mask, output_folder)
```

```
Initialized Dataset object:
        fMRI file: /home/niko/MRI/pcarpet_example/macaque/func_preproc.nii.gz
        Mask file: /home/niko/MRI/pcarpet_example/macaque/cortex_mask.nii.gz
        Output directory: /home/niko/MRI/pcarpet_example/macaque/outputs
```

Once the `Dataset` object is created, we have two ways of running the `pcarpet` pipeline on it:

1. Running the entire pipeline at once
2. Running the pipeline step-by-step

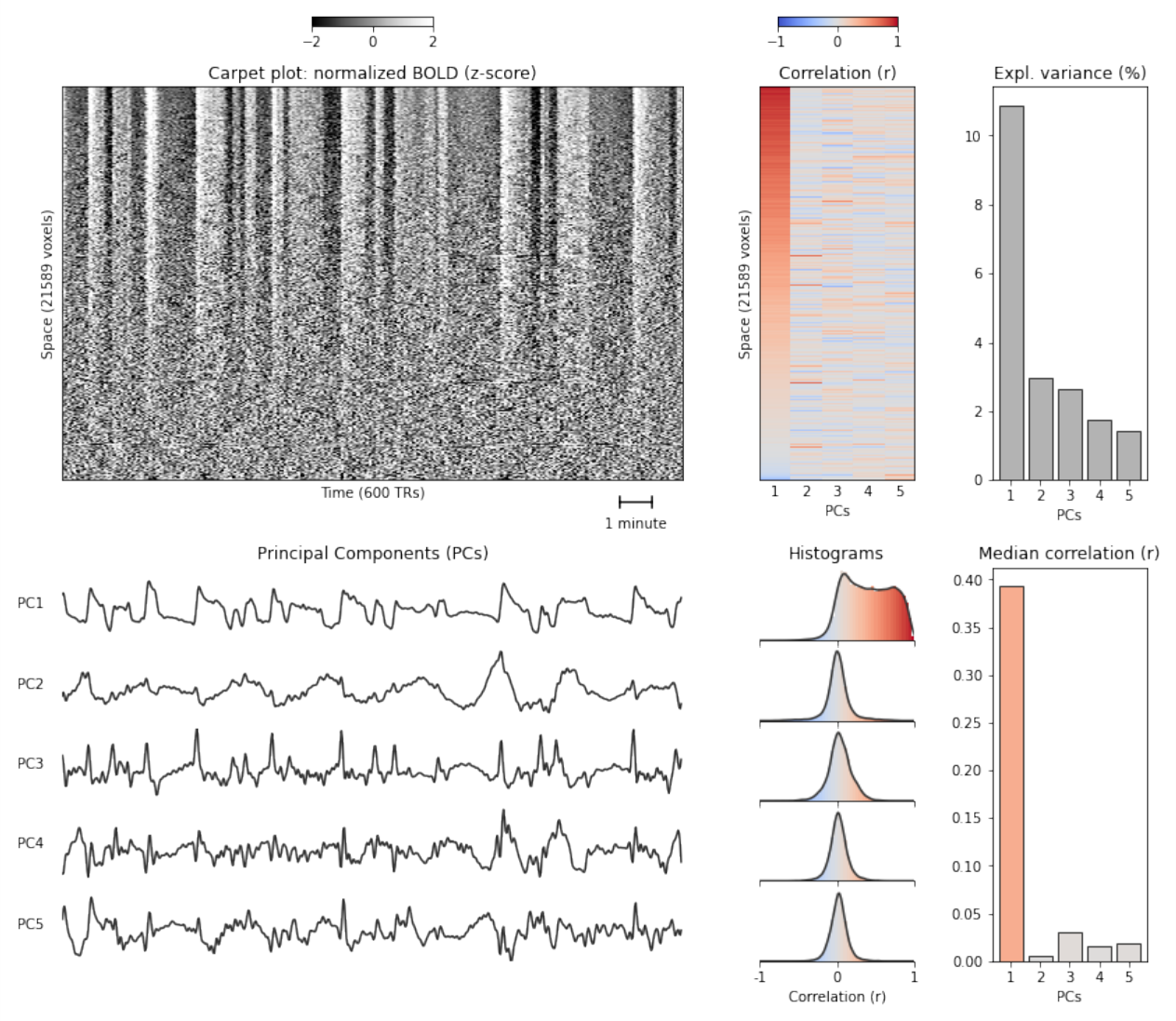
Below, we will first demonstrate the convenient first option. We will then go through the step-by-step option, which may be useful for debugging and for understanding the vairous inputs and outputs of the pipeline.

3.2 Option 1: Running the entire pcarpet pipeline at once

We can directly call the `run_pcarpet` method of the `Dataset` object, which will execute the pipeline with default options.

[4]:
```
MyData.run_pcarpet()
```

```
Reading data...
        fMRI data read: dimensions (80, 33, 80, 600)
        Mask read: dimensions (80, 33, 80)
fMRI data reshaped to voxels x time (211200, 600).
21589 voxels retained after masking.
Carpet matrix created with shape (21589, 600).
Carpet normalized to zero-mean unit-variance.
Carpet reordered.
PCA fit to carpet and results saved.
First 5 PCs correlated with carpet.
Out of these, 3 sign-flipped.
First 5 PCs correlated with fMRI data.
TR of 2.000 seconds read from fMRI header
Visual report generated and saved as fPCs_carpet_corr_report.
```



We can see what the default options were, since they are stored in a dictionary called `used_options`.

[5]:
```
ops = MyData.used_options
for key, val in ops.items():
    print(f'{key}: {val}')
```

```
tSNR_thresh: 15.0
reorder_carpet: True
save_carpet: False
save_pca_scores: False
ncomp: 5
flip_sign: True
TR: auto
```

Any of the above default options can be overriden by explicitly passing them as arguments to the `run_pcarpet` call. For example:

```
MyData.run_pcarpet(tSNR_thesh=10, reorder_carpet=False)
```

The meaning of each of these options will become clear in the step-by-step guide that follows.

3.3 Option 2: Running pcarpet step-by-step

3.3.1 Step 1: Importing the data

The first function, `import_data`, uses the `nibabel` package to import the fMRI data and the Mask into python as numpy arrays, and stores them as object attributes `data` and `mask`, respectiveyl. It also stores other attributes, like the x-y-z-t dimensions of fMRI data, the nifti header and the affine matrix.

[6]:
```
MyData.import_data()
```

```
Reading data...
        fMRI data read: dimensions (80, 33, 80, 600)
        Mask read: dimensions (80, 33, 80)
```

3.3.2 Step 2: Getting the carpet

The second function, `get_carpet`, generates a 'carpet' from the fMRI data and the mask. A carpet is a 2d matrix, shaped voxels x time, which contains the normalized (z-score) BOLD-fMRI signal from within the mask.

The fMRI data is first reshaped to 2d. Voxels that are outside the mask as well as voxels below a specified temporal signal-to-noise ratio threshold `tSNR_thresh` (default = 15) are discarded. The retained voxels are tranformed into 2d and normalized through z-scoring (subtract mean and divide by standard deviation along the time dimension).

By default, the rows of the carpet matrix (voxels) are ordered according to their (decreasing) correlation with the global (mean across voxels) signal. The reordering helps to highlight widespread signal fluctuations. It can be turned off by setting the `reorder_carpet` argument to `False`.

The carpet matrix is stored as a `carpet` attribute of the `Dataset` object. Optionally, it can be written to the output folder as a `carpet.npy` file (can be large), by setting the `save_carpet` argument to `True`.

[7]:
```
MyData.get_carpet(tSNR_thresh=15.0,
                  reorder_carpet=True, save_carpet=True)
```

```
fMRI data reshaped to voxels x time (211200, 600).
21589 voxels retained after masking.
Carpet matrix created with shape (21589, 600).
Carpet normalized to zero-mean unit-variance.
Carpet reordered.
Carpet saved as 'carpet.npy'.
```

3.3.3 Step 3: Fit PCA to carpet

The third function, `fit_pca2carpet`, fits PCA to the `carpet` matrix and saves the principal componens (PCs), the explained variance ratios, and optionally the PCA scores (PCA-tranformed carpet).

The PCs are stored as a `pca_comps` attribute of the `Dataset` object and are also written to the output folder as a `PCs.npy` file.

The explained variance ratios per PC are stored as a `expl_var` attribute of the `Dataset` object and are also written to the output folder as a `PCA_expl_var.npy` file.

If the `save_pca_scores` option is set to `True`, the PCA-transformed data will be written to the output folder as a `PCA_scores.npy` file (will be large).

```
[8]: MyData.fit_pca2carpet(save_pca_scores=True)
```

```
PCA fit to carpet and results saved.
```

3.3.4 Step 4: Correlate PCs with carpet

The fourth function, `correlate_with_carpet`, Correlates the first `ncomp` (defaults to 5) principal components (first PCs = fPCs) with all carpet voxel time-series. The fPCs are written to the output folder as `fPCs.csv`. The correlation matrix is written to the output folder as `fPCs_carpet_corr.npy`.

If the `flip_sign` option is `True` (enabled by default), an fPC (and its correlation values) will be sign-flipped when the median of its original correlation with carpet voxels is negative. This enforces the sign of the fPC to match the sign of the BOLD signal activity for most voxels. The sign-flipped fPCs are only used for downstream analysis and visualization. If any flips occurr, new versions of the two above output files are written: `fPCs_flipped.csv` and `fPCs_carpet_corr_flipped.npy`.

A report table of the above analyis is written to the output folder as `fPCs_carpet_corr_report.csv`. For each fPC, the table reports the explained variance ratio, the original median value of the correlation across carpet voxels, and whether sign-flipping was done for downstream analysis.

```
[9]: MyData.correlate_with_carpet(ncomp=5, flip_sign=True)
```

```
First 5 PCs correlated with carpet.
Out of these, 3 sign-flipped.
```

3.3.5 Step 5: Correlate PCs with fMRI

The fifth function, `correlate_with_fmri`, correlates the retained (and possibly sign-flipped) first `ncomp` PCs (fPCs) with the original 4d fMRI dataset. The resulting Pearson's correlation maps are written to the output folder as a 4d NIFTI file (3d space + `ncomp`) named `fPCs_fMRI_corr.nii.gz`. This file contains `ncomp` 3d correation maps, which represent the spatial distribution of each fPC across the brain.

```
[10]: MyData.correlate_with_fmri()
```

```
First 5 PCs correlated with fMRI data.
```

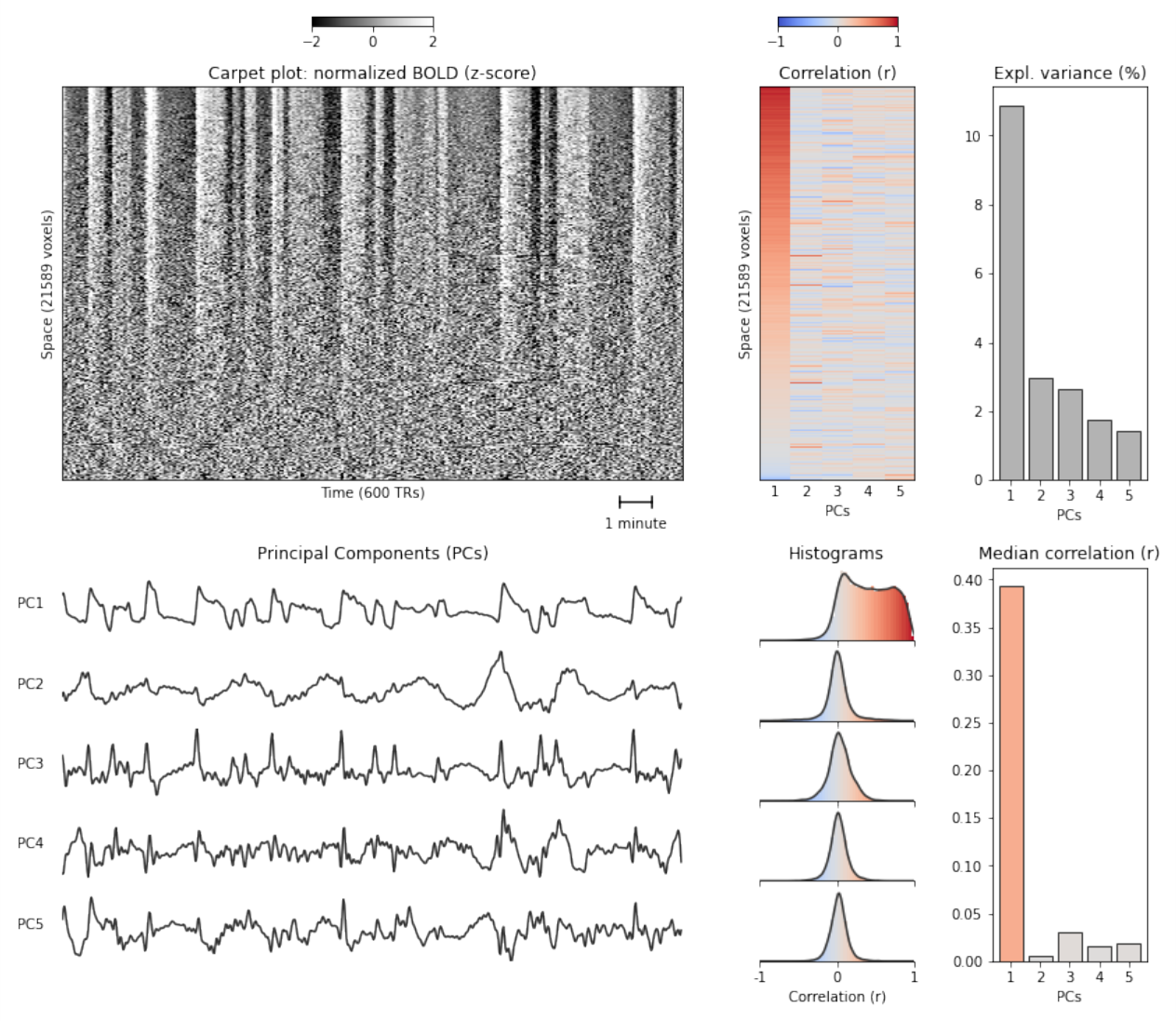
3.3.6 Step 6: Visualize results

The sixth and final function, `plot_report`, generates a visual report of the results, including the carpet plot, the first `ncomp` PCs (fPCs), their correlation with the carpet, and their explained variance ratios. The plot image, named `fPCs_carpet_corr_report` is written to the output folder in both '.png' (raster) and '.svg' (vector) formats.

To correctly display the time-scale, the function attempts to read the repetition time `TR` from the nifti header (`TR='auto'`). This can be overriden by explicitly passing a `TR` value (in seconds) as an argument

```
[11]: MyData.plot_report(TR='auto')
```

```
TR of 2.000 seconds read from fMRI header
Visual report generated and saved as fPCs_carpet_corr_report.
```



3.4 Outputs

Here we will list the outputs of the `pcarpet` pipeline that are written to the pre-specified `output_folder`.

3.4.1 Default outputs

The following outputs are always generated:

1. `PCs.npy`: 2d numpy array containing all Principal Components (PCs)
2. `PCA_expl_var.npy`: 1d numpy array with explained variance ratios per PC
3. `fPCS.csv`: comma-separated table containing the first `ncomp` PCs (fPCs) as columns
4. `fPCs_carpet_corr.npy`: 2d numpy array containing the correlation (r) values between fPCs and carpet voxel time-series
5. `fPCs_carpet_corr_report.csv`: comma separated table containing the following columns:

- PC: PC1, PC2, ...
- expl_var: explained variance ratio
- carpet_r_median: median *r* value across carpet voxels
- sign flipped: boolean, `True` if `flip_sign = True` and if `carpet_r_median < 0`

6. `fPCs_carpet_corr_report.png`: visual report in raster format
7. `fPCs_carpet_corr_report.svg`: visual report in vector format
8. `fPCs_fMRI_corr.nii.gz`: 4d nifti file, correlation maps between each (potentially sign-flipped) fPC and the entire fMRI scan. The file contains `ncomp` 3d maps.

3.4.2 Conditional outputs

If `flip_sign = True` and if at least one sign flip occurrs, new versions of outputs 3 and 4 are generated:

9. `fPCS_flipped.csv`
10. `fPCs_carpet_corr_flipped.npy`

3.4.3 Optional outputs

The following outputs are generated only when explicitly specified. Their file size can be large, depending on the fMRI and mask file sizes.

11. `carpet.npy` (saved if `save_carpet = True`): 2d numpy array containing the carpet matrix
12. `PCA_scores.npy` (saved if `save_pca_scores = True`): 2d numpy array containing PCA scores, i.e. the PCA-transformed carpet data

3.4.4 Importing outputs into python

Importing .npy files with `numpy`

[12]:
```
# importing .npy files with numpy
import numpy as np

carpet = np.load(os.path.join(output_folder, 'carpet.npy'))
print(f'Carpet, shape = {carpet.shape}')

PCs = np.load(os.path.join(output_folder, 'PCs.npy'))
print(f'PCs, shape = {PCs.shape}')

PCA_scores = np.load(os.path.join(output_folder, 'PCA_scores.npy'))
print(f'PCA scores, shape = {PCA_scores.shape}')

expl_var = np.load(os.path.join(output_folder, 'PCA_expl_var.npy'))
print(f'Explained variance ratio, shape = {expl_var.shape}')

fPCs_carpet_corr = np.load(os.path.join(output_folder, 'fPCs_carpet_corr.npy'))
print(f'fPCs_carpet_corr, shape = {fPCs_carpet_corr.shape}')
```

```
Carpet, shape = (21589, 600)
PCs, shape = (600, 600)
PCA scores, shape = (21589, 600)
Explained variance ratio, shape = (600,)
fPCs_carpet_corr, shape = (21589, 5)
```

Importing .csv files with pandas

[13]:
```
import pandas as pd

fPCs = pd.read_csv(os.path.join(output_folder, 'fPCs.csv'))
fPCs.head()
```

[13]:

	PC1	PC2	PC3	PC4	PC5
0	-0.068611	0.020185	-0.062595	-0.091971	0.049780
1	-0.071512	0.005875	-0.038379	-0.091214	0.092050
2	-0.055034	0.019877	0.017327	-0.053867	0.101125
3	-0.026231	0.039495	0.061318	-0.019174	0.076513
4	-0.000228	0.039751	0.059199	-0.009998	0.040623

[14]:
```
report = pd.read_csv(os.path.join(output_folder, 'fPCs_carpet_corr_report.csv'))
report.head()
```

[14]:

	PC	expl_var	carpet_r_median	sign_flipped
0	PC1	0.108726	-0.392726	True
1	PC2	0.029700	-0.005034	True
2	PC3	0.026533	-0.030266	True
3	PC4	0.017705	0.015283	False
4	PC5	0.014326	0.018219	False

.

CHAPTER

FOUR

API

4.1 Classes

Dataset(fmri_file, mask_file, output_dir)	Class for generating carpet plot from fMRI data and fitting PCA to it

4.1.1 pcarpet.Dataset

class pcarpet.**Dataset**(*fmri_file*, *mask_file*, *output_dir*)

Class for generating carpet plot from fMRI data and fitting PCA to it

correlate_with_carpet(*ncomp=5*, *flip_sign=True*)

Correlates the first ncomp principal components (PCs) with all carpet voxel time-series. Saves the correlation matrix.

Parameters

ncomp [int] Number of PCA components to retain. These first PCs (fPCs) are correlated with all carpet voxels. Default: 5

flip_sign [boolean] If True, an fPC (and its correlation values) will be sign-flipped when the median of its original correlation with carpet voxels is negative. This enforces the sign of the fPC to match the sign of the BOLD signal activity for most voxels. The sign-flipped fPCs are only used for downstream analysis and visualization (the saved PCA components and scores retain the original sign). Default: True

correlate_with_fmri()

Correlates the retained (and possibly sign-flipped) first ncomp PCs (fPCs) with the original 4d fMRI dataset and saves the resulting correlation maps as a 4d NIFTI file (3d space + ncomp).

fit_pca2carpet(*save_pca_scores=False*)

Fits PCA to carpet matrix and saves the principal componens (PCs), the explained variance ratios, and optionally the PCA scores (PCA-tranformed carpet)

Parameters

save_pca_scores [boolean] Whether to save the PCA scores (transformed carpet) in the output directory. The file might be large (possibly > 100MB depending on fMRI data and mask size). Default: False

get_carpet(*tSNR_thresh=15.0*, *reorder_carpet=True*, *save_carpet=False*)

Makes a carpet matrix from fMRI data. A carpet is a 2d matrix shaped voxels x time which contains the normalized (z-score) BOLD-fMRI signal from within a mask

Parameters

tSNR_thresh [float or None] Voxels with tSNR values below this threshold will be excluded. To deactivate set to None. Default: 15.0

reorder_carpet [boolean] Whether to reorder carpet voxels according to their (decreasing) correlation with the global (mean across voxels) signal Default: True

save_carpet [boolean] Whether to save the carpet matrix in the output directory. The file might be large (possibly > 100MB depending on fMRI data and mask size). Default: False

`import_data()`
Loads fMRI and mask data using nibabel.

`plot_report(`*TR='auto'*`)`
Plots a report of the results, including the carpet plot, the first ncomp PCs (fPCs), their correlation with the carpet, and their explained variance ratios. The plot image is saved in '.png' (raster) and '.svg' (vector) formats.

Parameters

TR ['auto' or float] fMRI repetition time in seconds. If 'auto', the program attempts to read TR from the fMRI header. This can be bypassed by explicitly passing TR as a float. Default: 'auto'

`run_pcarpet(`***kwargs*`)`
Runs the entire pcarpet pipeline using the default options for each function. The defaults can be overriden by passing the following optional keywords arguments:

Parameters

tSNR_thresh [float or None] Voxels with tSNR values below this threshold will be excluded from the carpet. To deactivate set to None. Default: 15.0

reorder_carpet [boolean] Whether to reorder carpet voxels according to their (decreasing) correlation with the global (mean across voxels) signal Default: True

save_carpet [boolean] Whether to save the carpet matrix in the output directory. The file might be large (possibly > 100MB depending on fMRI data and mask size). Default: False

save_pca_scores [boolean] Whether to save the PCA scores (transformed carpet) in the output directory. The file might be large (possibly > 100MB depending on fMRI data and mask size). Default: False

ncomp [int] Number of PCA components to retain. These first PCs (fPCs) are correlated with all carpet voxels and subsequently also with the entire fMRI dataset. Default: 5

flip_sign [boolean] If True, an fPC (and its correlation values) will be sign-flipped when the median of its original correlation with carpet voxels is negative. This enforces the sign of the fPC to match the sign of the BOLD signal activity for most voxels. The sign-flipped fPCs are only used for downstream analysis and visualization (the saved PCA components and scores retain the original sign). Default: True

TR ['auto' or float] fMRI repetition time in seconds. If 'auto', the program attempts to read the TR from the fMRI header. This can be bypassed by explicitly passing TR as a float. Default: 'auto'

4.2 Functions

pearsonr_2d(A, B)	Calculates row-wise Pearson's correlation between 2 2d-arrays
get_axis_coords(fig, ax)	Gets various coordinates of an axis within the figure space.

4.2.1 pcarpet.pearsonr_2d

pcarpet.**pearsonr_2d**(*A*, *B*)

Calculates row-wise Pearson's correlation between 2 2d-arrays

Parameters

A [2d-array] shape N x T

B [2d-array] shape M x T

Returns

R [2d-array] N x M shaped correlation matrix between all row combinations of A and B

4.2.2 pcarpet.get_axis_coords

pcarpet.**get_axis_coords**(*fig*, *ax*)

Gets various coordinates of an axis within the figure space.

Parameters

fig [matplotlib figure object]

ax [matplotlib axis object]

Returns

coords [dictionary] Contains the various coordinates: xmin, xmax, ymin, ymax, W (width), H (height), xcen (x center), ycen (ycenter)

INDEX