2009

Documentation for Ordinal Trend Analysis and Generalized Covariance Analysis

in MATLAB/SPM5

Christian Habeck
James Moeller
Tony Ng

Support: NIBIB 5R01EB006204

Columbia University Medical Center
5/27/2009

1 Quick Worksheet for the Impatient

Type "PCA_suite_f" after all software modules have been installed. A welcome screen with 16 options of selectable actions will appear.


```
OPTIONS:
  Extract/Upload Data
    1) Clear current workspace and upload variables/results from a previous analysis
    2) Clear current workspace and upload new images
  Analysis
    3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
    4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
    5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables
  Statistical Inference
    6) Execute a single-principal-component (PC) bootstrap
    7) Execute an OrT-fit bootstrap
    8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
    9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
   10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
   11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit
  Miscellaneous
   12) List what variables/results are required for running an analysis or a statistical inference
   13) List attributes of critical variables/results in the current workspace
   14) Direct manipulation of workspace
  Storage and Exit
   15) Save all presumably critical variables/results in the workspace to a file
   16) Exit module (WITHOUT saving any variables or results)
 ------------------------------------------------------------------------------------------------------------
Please make your selection:
```

You can jump in headlong. We hope this screen is reasonably self-explanatory. You should pick at least one action from each of the three categories in the following order – "Extract/Upload Data", "Analysis", and "Statistical Interference". At any stage in this chain of steps, you can employ Options #12 to #15 to save and manipulate all workspace data during the intermittent steps. This enables you to interrupt and resume your analysis later in the most flexible way.

Here are a few sequences of steps of the most commonly performed analyses:

- **Ordinal Trend Analysis sequence:** 2 → 3 → 4 → 7 → 10
- **PCA with subsequent fit of a behavioral variable:** 2 → 3 → 5 → 8 → 11
- **PLS with design featuring behavioral/anatomical information:** 2 → 3 → 6 → 9

2 Introduction to Ordinal Trend (OrT) Analysis

2.1 Tutorial on Basic Concepts in OrT

The processing chain of OrT and all other PCA-based approaches can be broken down into the following modules:

1. File input: loading of scans to be analyzed
2. Specification of design matrix
3. Principal component analysis, with optional storage of Eigen images and scores
4. Tailoring of a covariance pattern on the basis of a fit to behavioral or experimental variables, with optional storage
5. Inferential tests:
    a. Permutation test to check significance of findings
    b. Bootstrap test of robustness of weights in covariance pattern

The steps outlined before are really a nested structure: the user will execute steps 1-4 once to arrive at a *point estimate*, i.e. a covariance pattern with subject expression values and brain-behavioral correlations for the sample of participants under considerations. Such a point estimate of a quantity *s* in the usual statistics literature is usually denoted as: š. The inferential strategy usually proceeds in the following way: for selected test-statistics T(s) or F(s) a null-hypothesis distribution is theoretically known beforehand and a p-level can be estimated by computing analytically the part of the distribution that is more extreme than T(š) or F(š). Confidence levels for the most probable value of *s* around š itself go along with this.

Conceptually, the non-parametric approach is thus really easy. Computationally it requires the point estimation computation several hundred or thousands times on resampled data. This is no big deal any more either in the age of powerful Linux machines.

Concretely, this means steps 3-4 in our list will be executed many more times on resampled data. We have two kinds of resampling schemes: (a) a permutation test to approximate a p-level. Here all data are used, but the subject-condition assignment is broken and null-hypothesis conditions are created with brute force. Computing our test statistic on such a null-hypothesis sample many times we can generate the distribution for the test statistic from the data. The p-level is literally the relative frequency of the bogus statistic values exceeding the value for our point estimate. (b) The second test we use is a bootstrap test. This test is not for rejecting a null-hypothesis; it rather estimates how robust the loadings are in our covariance pattern. This time the data are resampled *with replacement*, but *without* breaking the subject-condition assignment. The data are meaningful, but the test tries to estimate the natural variability under the real distribution that the sample was drawn from. We execute the algorithm that produced the loadings in the point estimate covariance pattern on resampled data a few hundred times (100-500) and compute a normally distributed Z-statistic for every voxel. This Z-statistic gives an estimate of reliability of the voxel weight in the covariance pattern.

The fact that the Z-statistic computed from the data in this manner is itself a semi-parametric assumption and one shouldn't derive p-values with 0.00001 precision from this leap. We usually follow this rule of thumb: values of $|z| > 3$ are good, values of $|z| < 2$ are not so good. Keep in mind that $|z|$ only tries to localize the covariance pattern as a whole to a few key regions – a very voxel-wise SPM-inspired way of thinking, i.e. the traditional "colored-blobs" school of neuroimaging research. We might find a covariance pattern that explains the data well and gives good brain-behavioral correlation, with lousy $|z|$ values. This doesn't mean analytic failure, but only implies that the pattern cannot be boiled down to just a few key regions: every voxel participates in it to a small degree.

3 OrT Analysis Package Access and Software Requirements

3.1 Basic Description of OrT Analysis Package

The package consists of multiple modules. The functions of upload, analysis, and statistical inference are compartmentalized. In particular, since, depending on the size of the data and the kind of analysis, statistical inference could be a time-consuming and/or computationally intensive process, the user might want to perform the inferential procedure during a time when the allocation of system resources are less stringent.

3.2 Instructions for Download

Upload and unzip a file named *GCVA_package.zip* in the *Files* section of the GCVA Google Groups page.

3.3 MATLAB 7.3

3.4 SPM5

4 Functionalities of OrT Analysis Package

4.1 Package Execution

To begin running the package, type in “PCA_suite_f”.

4.2 Data Upload (Options #1 and #2)

There are two methods for uploading data into the workspace for subsequent analysis. One can either extract saved results and variables from a previous analysis, or upload a new set of image data.

During the former operation, the program downloads a MATLAB structure from a user-specified *.mat* file, which contains working variables and results from a prior session. Each saved structure also retains a time stamp that signifies the last time variables in the structure was written to a file.

If a user opts to start afresh with a new image dataset, they would first need to create a set of text (*.txt*) files, one for each task condition, that list paths to each of the images in rows. The order of the rows in each text file has to be consistent across the set of text files. After selecting the option to upload a new set of image data from the main menu, the user would then be prompted to specify the number of task levels, exclusive of baseline condition. Upon successful input of all the paths given the aforementioned user-specified text files, the images are uploaded into the workspace as a single target image array. This is also when the user needs to provide a grey matter mask that is of the same dimensions as each of the images.

All critical variables and results are stored in a MATLAB structure called *vars*. During data upload, an instance of *vars* is either extracted from an existing external file or created by the user indirectly during the image upload process.

4.3 Data Analysis

After performing a Data Upload routine and extracting/creating the prerequisite fields in *vars*, the program is ready to analyze the data.

4.3.1 General Principal Component Analysis (PCA) and Design Matrix Specification (Option #3)

This module performs a PCA on a target image array, with optional application of design matrix and background correction to remove task-independent effects from all images. Output arguments such as Eigen values and their respective Eigen images, subject scaling factors, etc. are copied into the workspace structure *vars* for subsequent analyses and statistical inferences.

4.3.2 Dyadic or Triadic Contrast Group Fit Using OrT (Option #4)

Given a set of OrT Eigen images that reside in the workspace *vars* structure, the program performs a linear regression using linear contrasts of subject scores/subject scaling factors as independent variables and group membership as a dependent variable. To perform the fit, the user can choose from either a selected subset of Eigen images based on Akaike Information Criteria information provided by the program or specification of which Eigen images to apply by themselves.

4.3.3 Behavioral Fit Using OrT (Option #5)

Given a set of PCA Eigen images that reside in the workspace *vars* structure, the program performs a linear regression using a best-fit linear combination of subject scores/subject scaling factors as independent variables and a linear combination of behavioral data as a dependent variable. To perform the fit, the user can choose from either a selected subset of Eigen images based on the Akaike Information Criterion (AIC) provided by the program or specification of which Eigen images to apply by themselves.

4.4 Statistical Inference

Upon completing an analysis, the user can perform various statistical inferences using the routines that follow.

4.4.1 Voxel Weight Reliability of Pattern Expression (Options #6, #7, and #8)

For each voxel in a pattern expression, there is a corresponding z-score that measures the reliability of its weight in the pattern. A complete collection of these z-scores across the entire brain constitutes a map of inverse coefficient of variation (ICV), or z-map. For each module in the Data Analysis segment, there is a respective module that computes the z-map for that particular analysis. These modules apply bootstrap resampling to test the significance of each voxel weight in a given image, be it a single Eigen image, an OrT-fit composite image, or a behavioral-fit composite image. Results from bootstrapping are saved to the external file periodically to prevent complete loss of data due to system failure or other interruptions.

4.4.2 Analysis-Specific Statistics (Options #9, #10, and #11)

For each of the three analyses in Data Analysis, the user might find it useful to test some of the default statistics for significance. The program does this by repeatedly permuting subjects across tasks, creating a resampling data sample, and performing the original analysis on the resampled data. The resulting sampling distribution is written to an external file. In the case of OrT fit and behavioral fit, a corresponding p-value for the statistic is passed into the workspace and written to a field in the *vars* structure (mentioned in the “Data Upload” section). Results are saved to the external file periodically to prevent complete loss of data due to system failure or other interruptions.

The default statistics of a general PCA are Eigen values from the PCA itself.

The default statistic of an OrT fit is the minimum number of exceptions to a complete separation of fitted subject group values derived from a linear combination of contrast scores that are associated with a pre-selected set of principal components.

The default statistic of a behavioral fit is the R-square between sets of actual and fitted linear combinations of behavioral data across task conditions. The fit is performed using an across-task linear combination, same as the linear combination applied to the behavioral data, of subject scaling factors associated with a pre-selected set of principal components as regressors.

4.5 Miscellaneous (Options #12, #13, and #14)

The user can choose to output metadata information on the workspace *vars* structure to screen.

The program also provides an option to display the general flow and progression of data/results from one module to the next. This is helpful in understanding what the prerequisite steps are for performing a specific routine.

Direct manipulation of the workspace allows a user access to variables used in the program and to customize the package to their own usage. For example, if the user is interested in viewing the content of the field *vars.ssf*, they can enter "vars.ssf" at command prompt and a list of subject scaling factors will be displayed. In general, to view the content of a specific field in the *vars* structure, issue the command "vars.*<name of field>*". Note that if a user opens up an image field such as *vars.eigenimages_noZeroes*, MATLAB will not display a brain image. Instead, there will be a giant array of numbers (representing voxel values in the image) filling up the display window screen after screen. Pressing Ctrl+C will break the command and return the user to a command prompt.

Another example would be when a user desires to save a field in *.txt* format. This could be done by issuing the command "save *<directory name/filename>*.txt -struct vars *<name of field1 name of field 2 … name of field N; separated by space>*".

4.6 Storage of Results and Data (Options #15 and #16)

The latest *vars* structure, with its fields of critical variables/results and a last-saved time stamp, is written to an external *.mat* file with a suffix that translates to the current date in *yyyymmdd* format. To save a particular field through the workspace, see previous section ("Miscellaneous").

Finally, the user can choose to exit the package. When selected, the program will *not* write the latest *vars* structure to an external *.mat* file prior to aborting. Storage of results and data has to be performed by executing the aforementioned write-to-file option.

5 Additional Notes

5.1 How should PCs be selected for further analysis in an unbiased way that avoids data-dredging?

Anybody who has worked with PCA will eventually encounter this question. People field a plethora of different solutions. Some look at principal components one at a time, some employ various rotation schemes (ProMAX, VariMAX, etc), some take linear combinations coming out of a regression analysis. We prefer the latter approach.

Picking one PC at a time, even after the application of some theoretically plausible rotation scheme, might satisfy the desire of arriving at a principled "blind" mode of analysis, i.e. one that is least plagued by arbitrary decisions on the part of the analyst. While this is psychologically comforting, it doesn't make a whole lot of neuroscientific sense to us: it's a big leap to assume that cognitive and background activity in the brain neatly breaks down with perfect alignment to the PC-ordering obtained from any particular rotation scheme. After all, the orthogonal nature of the PCs follows strictly from the mathematics of Eigen decompositions of covariance matrices – *not* in any way from the neurobiology of the brain. The safest default assumption is that cognitive processes give rise to activity whose capture probably necessitates *more* than one PC and that it will be impossible to rotate the PC set in such a way that most *individual* PCs encapsulate meaningful activity in the data.

This immediately causes a new problem and introduces some more anxiety: if more than one PC is needed to adequately capture the information of interest, for instance in a linear regression that predicts a meaningful outcome measure, how can one decide in a principled way *which* PCs should be selected? Unfortunately there is really no best answer. We have our own rules of thumb, but there might be others too. The most important thing is the following: don't engage in iterative, step-wise procedures to refine your model prediction. Just as with any other regression, your parametrically estimated p-levels that come from the model selected at the end of such a search are almost certainly meaningless and you will have dredged all generalizability out of your result. It's better to define a decision rule clearly *beforehand*; this doesn't have to be a hard-coded rule like 'Take PCs1-3', but it could be algorithmically defined like 'Take all the PCs that collectively account for at least 80% of the variance'.

In our software package, we tried to give you some guidance. We use Akaike's Information Criterion (AIC) to estimate the best model fit, i.e. the best trade-off between over- and under-fitting your meaningful outcome measure. This approach does not take into account how much variance each PC accounts for, but it's a start. Our program lists the best model-fits for you conveniently in one shot, without any data dredging, and then lets you decide which one to use. In our own practice we employ one further restriction: we like to use AIC, but only on contiguous sets of PCs; this means rather than looking at all possible subsets, we only consider the following ones: PC1, PCs 1-2, PCs 1-3, PCs 1-4, etc. This approach ensures that we account for a good amount of variance. Further, any pattern that is constructed from a linear combination of the selected PCs, will give more robust loadings. It is conceivable that meaningful information is contained in PCs of low variance contributions, for instance a combination of PCs 6-8. This might still be worth knowing about. However, it's very unlikely that a pattern made from PCs 6-8 will have any robust loadings at all as it’s contributing very little variance to the data.

6 Working Examples

As a reference for beginning users, six working examples are available in the *Appendix* section. Here is a list of them, with each example's option execution sequence, which typically starts with image upload (Option #2), continues with PCA/best-fit/statistical inference procedures (Options #3 to #11), and finishes with package termination (Option #16):

- PCA on an image dataset with three task levels (inclusive of baseline) and application of user-provided design matrix as part of PLS, followed by generation of a PLS ICV map and estimation of sampling distributions of eigenvalues through a permutation test; simultaneous test of subject *and* task effects

  Location of example: Page 14

  Action sequence: 2 → 3 → 6 → 9 → 13 → 14 → 16

- OrT fit using an image dataset with three task levels (inclusive of baseline), followed by generation of an OrT-fit ICV map and a permutation test on a default OrT-fit statistic; simultaneous test of subject *and* task effects

  Location of example: Page 24

  Action sequence: 2 → 3 → 4 → 7 → 10 → 15 → 16

- Behavioral fit using an image dataset and a behavioral response dataset with one task level (baseline only), followed by generation of a behavioral-fit ICV map and a permutation test on a default behavioral-fit statistic; test of subject effects *only*

  Location of example: Page 38

  Action sequence: 2 → 3 → 5 → 8 → 15 → 13 → 16

- Variance-accounted-for (VAF) in an image dataset by its own covariance patterns

  Location of example: Page 51

  Action sequence: (2 →) 13 → 3 → 13 → 14 → Execute the script *endo_VAF_s.m*

- Variance-accounted-for (VAF) in an image dataset by a set of imported exogenously-derived covariance patterns

  Location of example: Page 56

  Action sequence: 2 → 13 → 14 → Execute the script *pattern_expression_s.m*

- PCA on an image dataset without any design matrix for the purpose of quality control; this use of PCA identifies very high-intensity voxels that might have resulted from faulty pre-processing. In the following working example, we manipulated one scan (among a set of many) to have a voxel with 10,000-fold increased value compared to other voxels in the scan, and showed how to use and interpret the outcome of a script written for this particular application of PCA.

  Location of example: Page 60

  Action sequence: 1 → 3 → 14 → Execute the script *single_vox_anomaly_s.m*

Although Options #11 and #12 are not covered, they should be quite intuitive to understand after reviewing the six existing examples.

In addition to the above examples, a mock repeated-measure dataset (*Mock_Data.zip*) is available for download. As an exercise, a beginning user might want to see if they can replicate the results posted in *Example_OrT_fit_triad_VARS_RESULTS.mat* and *Example_PCA_behav_fit_monad_VARS_RESULTS.mat* on the GCVA Google Groups page. Note that the number of subjects in the dataset is small, with only ten subjects. In general, it is inadvisable to use such a small number of observations for an analysis.

7 Appendix

Several working examples are to follow as announced in the last section. We listed the screen output to make things easier. Occasionally, the wording might be slightly different because of we have made improvements to the text, but there should be no big differences. Since the listing in a Word document cannot do justice to the dynamic interaction between software and user, we have adopted a few helpful conventions:

- Text highlighted in green indicates a command the user has typed into the TTY

- Text that appears in salmon-colored boxes is commentary that does not appear on the screen

- Text in typewriter font does appear on the screen
  - ⇨ Particular commands that are discussed in the current example are bolded to underline their importance, i.e. for instance for Action sequence: 2 → 3 → 6 → 9 → 13 → 14 → 16, these commands are bolded on the main screen prior to their execution in the examples

PCA_suite_f

Type command to activate package. (In general, green highlight represents user input. Light gray highlight represents program outputs that are specific to the current dataset.)


```
  =====================================
   Ordinal Trend (OrT) Analysis Module
  =====================================
---------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

---------------------------------------------------------------------
Please make your selection: 2

Excluding baseline condition, how many task levels are there? 2

Upload paths for target images...
  Read in paths for BASE condition (ends in .txt):
     Probe1.txt

image_paths_for_all_tasks =

/Shape_Sternberg/Probe_Data/550_Probe1.img
/Shape_Sternberg/Probe_Data/553_Probe1.img
/Shape_Sternberg/Probe_Data/554_Probe1.img
/Shape_Sternberg/Probe_Data/556_Probe1.img
```

Upload images for one baseline condition and two task levels. (If there is only one condition, namely, the baseline, then input “0” for the number of task levels.)

<Omission>

```
/Shape_Sternberg/Probe_Data/992_Probe1.img

  Read in paths for TASK condition #1 (ends in .txt):
     Probe2.txt

paths_E1 =

/Shape_Sternberg/Probe_Data/550_Probe2.img
/Shape_Sternberg/Probe_Data/553_Probe2.img
/Shape_Sternberg/Probe_Data/554_Probe2.img
/Shape_Sternberg/Probe_Data/556_Probe2.img
```

<Omission>

```
/Shape_Sternberg/Probe_Data/992_Probe2.img

  Read in paths for TASK condition #2 (ends in .txt):
     Probe3.txt
```

Although these paths are arbitrary, notice the ordering of subjects across task conditions. They are consistent throughout. Any deviation from such proper ordering would render all subsequent results invalid.

For example, if Subject #554 is preceded by subject #553 in the baseline condition (as seen in yellow highlight among the magenta-highlighted pathnames), the same order is strictly maintained throughout path specification for the first and the second task conditions.

```
paths_E2 =

/Shape_Sternberg/Probe_Data/550_Probe3.img
/Shape_Sternberg/Probe_Data/553_Probe3.img
/Shape_Sternberg/Probe_Data/554_Probe3.img
/Shape_Sternberg/Probe_Data/556_Probe3.img

<Omission>

/Shape_Sternberg/Probe_Data/992_Probe3.img

Merging all pathnames to a single array variable...

Pathnames for target images are now merged.

Uploading a raw image from /Shape_Sternberg/Probe_Data/550_Probe1.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/553_Probe1.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/554_Probe1.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/556_Probe1.img

<Omission>

Uploading a raw image from /Shape_Sternberg/Probe_Data/992_Probe1.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/550_Probe2.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/553_Probe2.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/554_Probe2.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/556_Probe2.img

<Omission>

Uploading a raw image from /Shape_Sternberg/Probe_Data/992_Probe2.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/550_Probe3.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/553_Probe3.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/554_Probe3.img
Uploading a raw image from /Shape_Sternberg/Probe_Data/556_Probe3.img

<Omission>

Uploading a raw image from /Shape_Sternberg/Probe_Data/992_Probe3.img

Finished uploading target images.

Provide path information of the mask file you want to apply...

Input threshold level for the mask you want to apply: 0.5

Log-transform target images (negative values will be set to zero; 1==yes/0==no)? 0


Ready to execute a new command...  To proceed, press
any key.
```

```
<Status of a structure in MATLAB workspace
called "vars", which stores updated critical
variables throughout execution; highlighted
fields are those that have been
created/modified after the latest command.>

vars =

target_images:  [517845x75 double]
meanful_set:    [84097x1 double]
dim:            [79 95 69 75]
nOfTasks:       2
image_paths:    [75x85 char]
...
```

The field *dim* provides information on the target image array. The first three numbers represent the voxel dimensions of each image in 3-D, while the fourth entry indicates the number of images in the entire image array. Notice that there are $79 \times 95 \times 69 = 517,845$ voxels in each of the 75 target images (25 for the baseline condition and 25 each for each of the two task levels, with the same subject ordering).

The field *meanful_set* represents the non-zero post-masking voxel subset of those 517, 845 voxels that will be used throughout the analysis. It is a 1-D representation.

Next step: Design matrix specification and principal component analysis (PCA)

```
--------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design
matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 3

Perform a principal component analysis without prior application of design matrix (1==yes/0==no)? 0

Remove task-independent effects from target image data prior to pattern extraction (1==yes/0==no)? 1
Completed background correction.  Task-independent effects have been removed from target image array.

     The size of the target image array is 517845 (# of voxels) by 75 (# of subject-task).
     There is a total of 3 sequential task block(s), each of which consists of 25 subject(s).
     Any design matrix must have 75 row(s).

Use OrT design matrix (1==yes/0==no)? 0
     Provide path information of the file holding the design matrix of your choice...
     Design matrix from file uploaded.
```

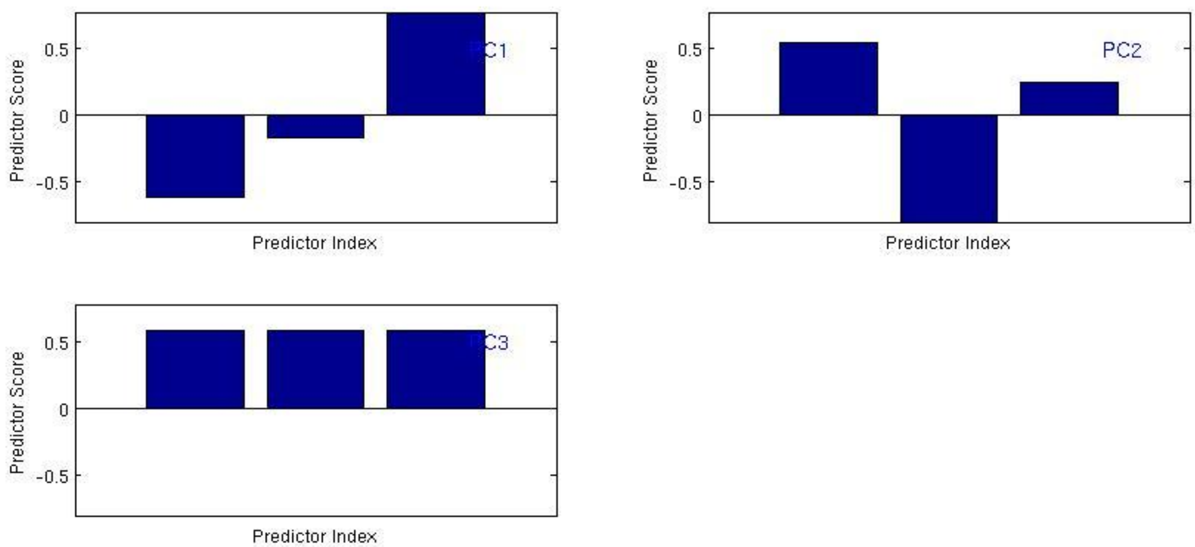
At this point, a SPM window pops up to prompt the user for a design matrix file.

```
Remove mean image from target image data during PCA (1==yes/0==no)? 1

Finished computing eigenvalues, eigenimages, and subject scaling factors.  See plots of brain scores from the
first six PCs.
(Note: Each eigenimage has been normalized to unity.  However, the set of eigenvalues are not normalized.)

Ready to execute a new command...  To proceed, press any key.
```

```
<Status of vars in workspace after completion of a PCA with design matrix>

target_images:          [517845x75 double]          single_PC:                         []
meanful_set:            [84097x1 double]            singlePC_icv_map_noZeroes:         []
dim:                    [79 95 69 75]               OrT_fit_nOfExc:                    []
nOfTasks:               2                           OrT_fit_selected_PCs:              []
image_paths:            [75x85 char]                OrT_fit_composite_PC_image:        []
lambdas:                [2x1 double]                OrT_fit_composite_PC_image_ssfs:   []
eigenimages_noZeroes:   [84097x2 double]            OrT_fit_icv_map_noZeroes:          []
eigenvectors:           [3x3 double]                OrT_fit_nOfExc_p_value:            []
ssf:                    [75x2 double]               behav_fit_rsq:                     []
design_mat:             [75x3 double]               behav_fit_selected_PCs:            []
bg_correct:             1                           behav_fit_composite_PC_image:      []
remove_mean_image:      1                           behav_fit_composite_PC_image_ssfs: []
                                                    behav_fit_contrast_coef:           []
                                                    behav:                             []
                                                    behav_fit_icv_map_noZeroes:        []
                                                    behav_fit_rsq_p_value:             []
                                                    last_time_saved:                   []
```

The design matrix we selected for this example is a 75 × 3 block diagonal matrix with three 25 × 1 blocks. Entries (often derived from behavioral data) in each block are mean-adjusted within their column. The size of the design matrix is reflected in the *design_mat* field of the underlying *vars* structure, as could be seen in the dark orange *vars* status box above.

Below are brain score plots for all three principal components from our PLS. Since there are only three principal components, there are only three (but not six) plots. Notice that the predictor scores for the third principal component are all identical. This is a direct result of mean image removal from target image data prior to a PCA. With mean effects removed, we can focus solely on subject effects, task effects, and their interactions.



```
------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

------------------------------------------------------------------------------------------------------------
Please make your selection: 6

There is a total of 2 eigenimages.
Which eigenimage would you like to use for the bootstrap?  Please select one: 1
Number of resamplings for single-principal-component bootstrapping: 100

Please provide filename prefix for the ICV (inverse coefficient of variation) voxel map:
     TEST
Where would you like to place the file?  Please input target path (ending WITHOUT "/"):
     /Shape_Sternberg/Probe_Data/info_dir

Beginning single-principal-component bootstrap resampling...

     Bootstrap Sample #1 out of 100.

     Bootstrap Sample #2 out of 100.

     Bootstrap Sample #3 out of 100.

     Bootstrap Sample #4 out of 100.

     Bootstrap Sample #5 out of 100.

<Omission>

     Bootstrap Sample #96 out of 100.

     Bootstrap Sample #97 out of 100.

     Bootstrap Sample #98 out of 100.

     Bootstrap Sample #99 out of 100.

     Bootstrap Sample #100 out of 100.

Elapsed time is 642.077958 seconds.
Saving a snapshot of results obtained thus far...
     Done.
Removing saved file from previous snapshot...
     Done.

Extrema on the most current ICV map: -1.3467      1.4312

The final ICV map has been saved to
/Shape_Sternberg/Probe_Data/info_dir/TEST_singlePC_ICVNormMap_100samples_PC1.img.

Completed single-principal-component bootstrap resampling.
```

```
Ready to execute a new command...  To proceed, press any key.
```

```
<Status of vars in workspace after completion of a PLS single-PC bootstrapping>

target_images:         [517845x75 double]         single_PC:                          1
meanful_set:           [84097x1 double]           singlePC_icv_map_noZeroes:          [84097x1 double]
dim:                   [79 95 69 75]              OrT_fit_nOfExc:                     []
nOfTasks:              2                          OrT_fit_selected_PCs:               []
image_paths:           [75x85 char]               OrT_fit_composite_PC_image:         []
lambdas:               [2x1 double]               OrT_fit_composite_PC_image_ssfs:    []
eigenimages_noZeroes:  [84097x2 double]           OrT_fit_icv_map_noZeroes:           []
eigenvectors:          [3x3 double]               OrT_fit_nOfExc_p_value:             []
ssf:                   [75x2 double]              behav_fit_rsq:                      []
design_mat:            [75x3 double]              behav_fit_selected_PCs:             []
bg_correct:            1                          behav_fit_composite_PC_image:       []
remove_mean_image:     1                          behav_fit_composite_PC_image_ssfs:  []
                                                  behav_fit_contrast_coef:            []
                                                  behav:                              []
                                                  behav_fit_icv_map_noZeroes:         []
                                                  behav_fit_rsq_p_value:              []
                                                  last_time_saved:                    []
```


Results from bootstrap resampling are written to an external file every twenty samples. Extrema on the latest inverse-coefficient-of-variation (ICV) map is also displayed on screen. In our case, the absolute value of each of the two extrema is clearly lower than 2, the conventional threshold for statistical significance on an ICV map, indicating the lack of robustness in *every* voxel in the behavioral fit covariance pattern. The name of the external file is also updated to reflect the number of samples used to obtain the ICV map.

After analyzing the user-specified number of bootstrap samples, the final ICV map is written into a field in *vars*.

Generally, if the loading of a voxel in a covariance pattern is robustly positive (> 2 on the ICV map), its activity in the brain is localized and increases with task difficulty level. If the loading is robustly negative (< -2 on the ICV map), its activity is localized and decreases with task difficulty. If the loading is not robust, there is no localization at the voxel *but* the voxel could nonetheless contribute to the overall covariance pattern across task levels in a dispersive fashion in conjunction with other voxels within the pattern.

```
-------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling
distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-------------------------------------------------------------------------------------------------------------
Please make your selection: 9

Number of resamplings for PCA permutation test: 1000

Removing task-independent effects from images...
Completed background correction for all resampling image arrays.

Please provide filename prefix for storing sampling distributions of non-normalized eigenvalues:
     TEST
Where would you like to place the file?  Please input target path (ending WITHOUT "/"):
     /Shape_Sternberg/Probe_Data/info_dir

Beginning PCA permutation test...

     Permutation Sample #1 out of 1000.

     Permutation Sample #2 out of 1000.

     Permutation Sample #3 out of 1000.

     Permutation Sample #4 out of 1000.

     Permutation Sample #5 out of 1000.
```

<Omission>

```
     Permutation Sample #996 out of 1000.

     Permutation Sample #997 out of 1000.

     Permutation Sample #998 out of 1000.

     Permutation Sample #999 out of 1000.

     Permutation Sample #1000 out of 1000.

Elapsed time is 6211.688532 seconds.

Sampling distributions of the eigenvalues have been saved to:
     /Shape_Sternberg/Probe_Data/info_dir/TEST_PCA_lambdas_1001by2.txt

Completed PCA permutation test.


Ready to execute a new command...  To proceed, press any key.
```

Unlike in the previous step of bootstrapping to test voxel weight reliability in a covariance pattern, resampling under a permutation test is performed *without* replacement. The original images are permuted across tasks for each subject to gather information on one default set of PLS statistics, the non-normalized eigenvalues. A PLS analysis is performed on each permuted sample, as on the original image set.

Every twenty samples, sampling distributions of the statistics are written to an external file. Eigenvalues from any given permutation are listed in descending order in one row, with each row representing results from a single permuted sample. The first row carries results from the original non-permuted dataset.

```
--------------------------------------------------------------------------------------------------------------

 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 13

vars =

                         target_images: [517845x75 double]
                           meanful_set: [84097x1 double]
                                   dim: [79 95 69 75]
                             nOfTasks: 2
                           image_paths: [75x85 char]
                              lambdas: [2x1 double]
                   eigenimages_noZeroes: [84097x2 double]
                          eigenvectors: [3x3 double]
                                   ssf: [75x2 double]
                            design_mat: [75x3 double]
                            bg_correct: 1
                     remove_mean_image: 1
                             single_PC: 1
              singlePC_icv_map_noZeroes: [84097x1 double]
                        OrT_fit_nOfExc: []
                  OrT_fit_selected_PCs: []
            OrT_fit_composite_PC_image: []
       OrT_fit_composite_PC_image_ssfs: []
              OrT_fit_icv_map_noZeroes: []
                OrT_fit_nOfExc_p_value: []
                          behav_fit_rsq: []
                behav_fit_selected_PCs: []
          behav_fit_composite_PC_image: []
     behav_fit_composite_PC_image_ssfs: []
            behav_fit_contrast_coef: []
                                  behav: []
            behav_fit_icv_map_noZeroes: []
                 behav_fit_rsq_p_value: []
                       last_time_saved: []


Ready to execute a new command...  To proceed, press any key.
```

A user can inspect the attributes of all fields in *vars* by choosing Option #13. This option is created for quick viewing. For accessing contents of certain fields and performing more involved procedures, a user will instead find Option #14 useful.

In our current example, notice that upon completion of the permutation test, no field in *vars* has been updated. Unlike an OrT-fit permutation test or a behavioral-fit permutation test, there are no *p*-values to report for a general PCA. However, eigenvalues/lambdas from the original analysis and each re-sampled analysis are stored in an external file for subsequent inferences at the user's discretion.

```
-----------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-----------------------------------------------------------------------------------------------------------------
Please make your selection: 14

vars

vars =

                        target_images: [517845x75 double]
                          meanful_set: [84097x1 double]
                                  dim: [79 95 69 75]
                            nOfTasks: 2
                          image_paths: [75x85 char]
                              lambdas: [2x1 double]
                 eigenimages_noZeroes: [84097x2 double]
                         eigenvectors: [3x3 double]
                                  ssf: [75x2 double]
                           design_mat: [75x3 double]
                           bg_correct: 1
                    remove_mean_image: 1
                            single_PC: 1
              singlePC_icv_map_noZeroes: [84097x1 double]
                         OrT_fit_nOfExc: []
                    OrT_fit_selected_PCs: []
               OrT_fit_composite_PC_image: []
          OrT_fit_composite_PC_image_ssfs: []
                  OrT_fit_icv_map_noZeroes: []
                    OrT_fit_nOfExc_p_value: []
                          behav_fit_rsq: []
                   behav_fit_selected_PCs: []
              behav_fit_composite_PC_image: []
         behav_fit_composite_PC_image_ssfs: []
                  behav_fit_contrast_coef: []
                                  behav: []
                 behav_fit_icv_map_noZeroes: []
                      behav_fit_rsq_p_value: []
                           last_time_saved: []

vars.eigenvectors

ans =

   -0.6077    0.5453    0.5774
   -0.1684   -0.7989    0.5774
    0.7761    0.2536    0.5774

return

Ready to execute a new command...  To proceed, press any key.
```

With direct access to the underlying workspace, a user who is familiar with MATLAB and the package can issue any arbitrary MATLAB command and execute external/personalized MATLAB functions/scripts without aborting the program. (Keep in mind that upon exiting the package, all variables and results in the workspace will be lost, unless they have been saved to an external file prior to termination of the package.)

In our current example, after entering the workspace, we type in “vars” and MATLAB prints attributes of all fields in the *vars* structure to the computer screen.

Next, to inspect entries of our eigenvector matrix (which we know is a “3x3 double” from our *vars* structure), we issue the command “vars.eigenvectors”. MATLAB then prints content of *vars.eigenvectors* to the screen, which is consistent with the brain score plots we saw earlier.

Finally, issuing a “return” statement brings us back into the main module with its list of menu options.

```
--------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 16

...Ordinal Trend analysis module terminated.
```

```
PCA_suite_f

  =====================================
   Ordinal Trend (OrT) Analysis Module
  =====================================

----------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

----------------------------------------------------------------------------------------------------------
Please make your selection: 2

Excluding baseline condition, how many task levels are there? 2

Upload paths for target images...
  Read in paths for BASE condition (ends in .txt):
     paths_B.txt

image_paths_for_all_tasks =

B_scan1.img
B_scan2.img
B_scan3.img
B_scan4.img
B_scan5.img
B_scan6.img
B_scan7.img
B_scan8.img
B_scan9.img
B_scan10.img

  Read in paths for TASK condition #1 (ends in .txt):
     paths_E1.txt

paths_E1 =

E1_scan1.img
E1_scan2.img
E1_scan3.img
E1_scan4.img
E1_scan5.img
E1_scan6.img
E1_scan7.img
E1_scan8.img
E1_scan9.img
E1_scan10.img

  Read in paths for TASK condition #2 (ends in .txt):
     paths_E2.txt
```

Type command to activate package. (In general, green highlight represents user input. Light gray highlight represents program outputs that are specific to the current dataset. Boldface is used to emphasize program options that are being chosen.)

Upload images for one baseline condition and two levels of a task parameter, for instance, a parameter for memory load. (If there is only one condition, namely, the baseline, input "0" for the number of task levels. This would represent a cross-sectional analysis; no repeated-measures design would be present.)

Although these paths are arbitrary, notice the ordering of subjects across task conditions. They are consistent throughout. Any deviation from such proper ordering would destroy the repeated-measures design and sever meaningful ties in the data.

For example, if Subject #4 is preceded by subject #3 in the baseline condition (as seen in yellow highlight among the light-gray-highlighted pathnames), the same order is strictly maintained throughout path specification for the first and the second task conditions.

```
paths_E2 =

E2_scan1.img
E2_scan2.img
E2_scan3.img
E2_scan4.img
E2_scan5.img
E2_scan6.img
E2_scan7.img
E2_scan8.img
E2_scan9.img
E2_scan10.img

Merging all pathnames to a single array variable...

Pathnames for target images are now merged.

Uploading a raw image from B_scan1.img
Uploading a raw image from B_scan2.img
Uploading a raw image from B_scan3.img
Uploading a raw image from B_scan4.img
Uploading a raw image from B_scan5.img
Uploading a raw image from B_scan6.img
Uploading a raw image from B_scan7.img
Uploading a raw image from B_scan8.img
Uploading a raw image from B_scan9.img
Uploading a raw image from B_scan10.img
Uploading a raw image from E1_scan1.img
Uploading a raw image from E1_scan2.img
Uploading a raw image from E1_scan3.img
Uploading a raw image from E1_scan4.img
Uploading a raw image from E1_scan5.img
Uploading a raw image from E1_scan6.img
Uploading a raw image from E1_scan7.img
Uploading a raw image from E1_scan8.img
Uploading a raw image from E1_scan9.img
Uploading a raw image from E1_scan10.img
Uploading a raw image from E2_scan1.img
Uploading a raw image from E2_scan2.img
Uploading a raw image from E2_scan3.img
Uploading a raw image from E2_scan4.img
Uploading a raw image from E2_scan5.img
Uploading a raw image from E2_scan6.img
Uploading a raw image from E2_scan7.img
Uploading a raw image from E2_scan8.img
Uploading a raw image from E2_scan9.img
Uploading a raw image from E2_scan10.img

Finished uploading target images.

Provide path information of the mask file you want to apply...

Input threshold level for the mask you want to apply: 0.5

Log-transform target images (negative values will be set to zero; 1==yes/0==no)? 0

Checking the rank of target image array...
     The rank is 30.
The target image array has full rank.  Please proceed.


Ready to execute a new command...  To proceed, press any key.
```

The user has to upload a mask to restrict the number of voxels in the analysis. Here we choose the generic probabilistic gray-matter mask supplied by SPM5 and threshold it at a lower bound of 0.5.

The program does a rank check on all our input files to avoid file repetition. In our example, the rank is 30, i.e. every file is unique.

```
-----------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-----------------------------------------------------------------------------------------------------------
Please make your selection: 13


vars =

                        target_images: [517845x30 double]
                          meanful_set: [73631x1 double]
                                  dim: [79 95 69 30]
                            nOfTasks: 2
                          image_paths: [30x39 char]
                              lambdas: []
                eigenimages_noZeroes: []
                         eigenvectors: []
                                  ssf: []
                           design_mat: []
                           bg_correct: []
                    remove_mean_image: []
                            single_PC: []
           singlePC_icv_map_noZeroes: []
                      OrT_fit_nOfExc: []
                OrT_fit_selected_PCs: []
          OrT_fit_composite_PC_image: []
     OrT_fit_composite_PC_image_ssfs: []
            OrT_fit_icv_map_noZeroes: []
              OrT_fit_nOfExc_p_value: []
                       behav_fit_rsq: []
              behav_fit_selected_PCs: []
        behav_fit_composite_PC_image: []
   behav_fit_composite_PC_image_ssfs: []
             behav_fit_contrast_coef: []
                                behav: []
          behav_fit_icv_map_noZeroes: []
               behav_fit_rsq_p_value: []
                      last_time_saved: []


Ready to execute a new command...  To proceed, press any key.
```

Check the status of a structure in MATLAB workspace called *vars*, which stores updated critical variables throughout execution; cyan-highlighted fields are those that have been created or modified after the latest command.

The field *dim* provides information on the target image array. The first three numbers represent the voxel dimensions of each image in 3-D, while the fourth entry indicates the number of images in the entire image array. Notice that there are 79 × 95 × 69 = 517, 845 voxels in each of the 30 target images (10 for the baseline condition and 10 each for each of the two task levels, with the same subject ordering).

The field *meanful_set* represents the non-zero post-masking voxel subset of those 517, 845 voxels that will be used throughout the analysis. It is a 1-D representation.

Next step: OrT Design matrix specification and principal component analysis (PCA)

```
--------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
```

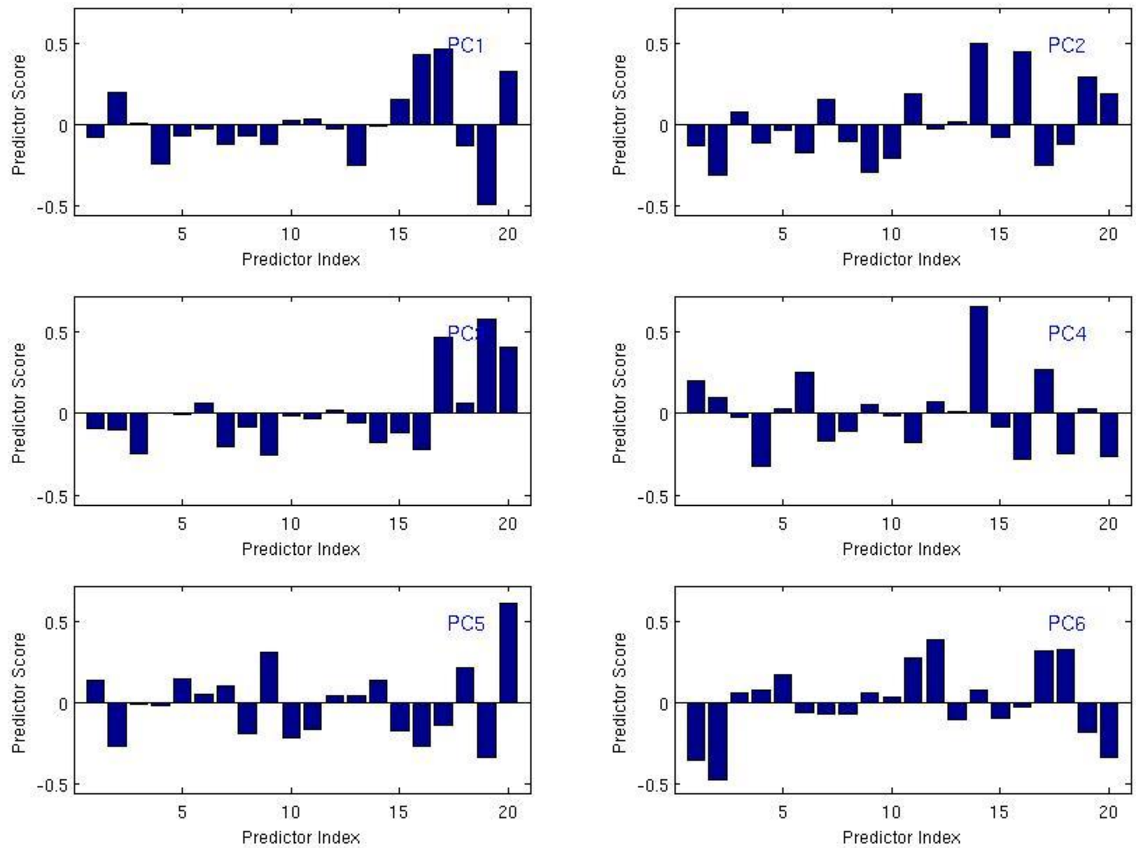
3) Perform a principal component analysis (PCA) on a target image array; design matrix optional

```
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 3

Perform a principal component analysis without prior application of design matrix (1==yes/0==no)? 0

Remove task-independent effects from target image data prior to pattern extraction (1==yes/0==no)? 1
Completed background correction.  Task-independent effects have been removed from target image array.

The size of the target image array is 517845 (# of voxels) by 30 (# of subject-task).
     There is a total of 3 sequential task block(s), each of which consists of 10 subject(s).
     Any design matrix must have 30 row(s).

Use OrT design matrix (1==yes/0==no)? 1
     OrT design matrix will be used.

Finished computing eigenvalues, eigenimages, and subject scaling factors.  See plots of brain scores from the
first six PCs.
(Note: Each eigenimage has been normalized to unity.  However, the set of eigenvalues are not normalized.)


Ready to execute a new command...  To proceed, press any key.
```

← (*)

Select design matrix. In this particular example, the default (OrT) is selected.

(*) Background correction filters the data by removing data variance contributed by effects that vary across subjects, but are *independent* of the task-parameter. Such task-independent effects can be substantial in fMRI and account for as much as 50% or more of data variance. If not removed from the data prior to the application of a design matrix, they will dominate the covariance structure.



Expressions of the first six principal components (PCs) are plotted across the predictors of the design matrix. In this OrT example, they are not particularly instructive since the design matrix has no meaning in terms of behavioral or clinical variables. They simply serve to increase the variance of monotonic effects. For Partial Least Square (PLS), however, which uses a few seed variables, these plots are helpful.

```
----------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

----------------------------------------------------------------------------------------------------------------
Please make your selection: 13


vars =

                        target_images: [517845x30 double]
                          meanful_set: [73631x1 double]
                                  dim: [79 95 69 30]
                             nOfTasks: 2
                          image_paths: [30x39 char]
                              lambdas: [19x1 double]
                 eigenimages_noZeroes: [73631x19 double]
                         eigenvectors: [20x20 double]
                                  ssf: [30x19 double]
                           design_mat: [30x20 double]
                           bg_correct: 1
                    remove_mean_image: 1
                            single_PC: []
            singlePC_icv_map_noZeroes: []
                       OrT_fit_nOfExc: []
                 OrT_fit_selected_PCs: []
            OrT_fit_composite_PC_image: []
       OrT_fit_composite_PC_image_ssfs: []
             OrT_fit_icv_map_noZeroes: []
               OrT_fit_nOfExc_p_value: []
                        behav_fit_rsq: []
               behav_fit_selected_PCs: []
          behav_fit_composite_PC_image: []
     behav_fit_composite_PC_image_ssfs: []
              behav_fit_contrast_coef: []
                                behav: []
           behav_fit_icv_map_noZeroes: []
                behav_fit_rsq_p_value: []
                      last_time_saved: []


Ready to execute a new command...  To proceed, press any key.
```

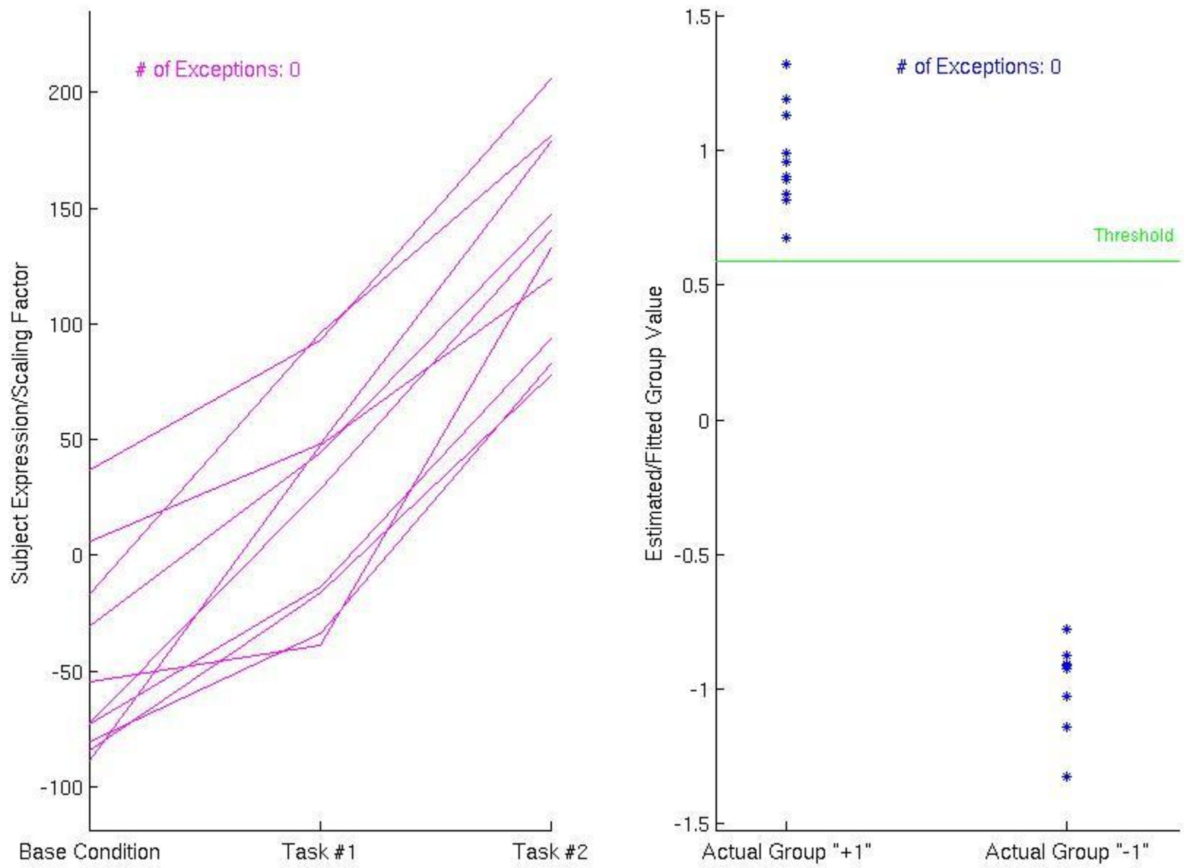
Check the status of the structure *vars* after an OrT PCA. *Vars* contains all pertinent information on the analysis. Up to now, this includes the target image array with all 30 task images and all 19 PCs with their respective expressions by each of the 10 subjects across 3 conditions.

Construct a covariance pattern with the best monotonic trend, using linear regression and a selected set of PCs.


```
--------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 4

Two-group discriminant fit using pre-set contrast scores of OrT eigenimages:
     Would you like to select a subset of eigenimages using Akaike Information Criteria (1==yes/0==no)? 1

Computing Akaike Information Criteria for each linear combination of predictors...
     Note: There are 10 predictors.

1023 sets of predictors were tested.

Under arbitrary combination of predictors, the smallest three Akaike Information Criteria values and their
respective predictors are as follows:
     (1) 14.705 from fitting the following predictors ---> 1    2    3    6    9   10
     (2) 19.134 from fitting the following predictors ---> 1    2    3    5    6    9   10
     (3) 19.2957 from fitting the following predictors ---> 1    2    3    6    7    9   10

Under sequential inclusion of predictors, the smallest three Akaike Information Criteria values and their
respective predictors are as follows:
     (4) 47.618 from fitting the following predictors ---> 1    2    3    4    5    6    7    8    9   10
     (5) 56.2362 from fitting the following predictors ---> 1   2   3
     (6) 60.0339 from fitting the following predictors ---> 1   2

Among sets (1) to (6), which would you like to output for further analysis? 1

Completed two-group discriminant fit using pre-set contrast scores of OrT eigenimages.   See graph.

Output the linear combination of eigenimages associated with the best fit to a file (1==yes/0==no)? 1

Please specify prefix for filename: TEST
Where would you like to place the file?  Please input target path (ending WITHOUT "/"):
     MockData
Best-fit linear combination of eigenimage(s) [1    2    3    6    9   10] derived from best-fit contrast score
coefficients has been written to:
     TEST_contrast_ssf_BestFit_PC1236910.img



Ready to execute a new command...  To proceed, press any key.
```

← (*)

(*) After the program listed a few options for selected sets of principal components, we decided to use set (1). This set gives the best model fit, i.e. the lowest value of the AIC criterion. (Note that in our own practice, we usually pick contiguous sets of PCs, i.e. the kind of sets that are listed under (4), (5), and (6).)



Regarding "`See graph.`":

The graphs above are displayed in a new figure window titled "Separation of Estimated Group Values Derived from Subject Expressions". Notice that in both plots, the minimum number of exceptions to a complete separation of fitted group values derived from the best-fit linear combination of contrast scores is displayed.

The plot on the left provides a means to visually inspect expression of a covariance pattern for each subject across tasks. In this analysis, for any given subject and task, each subject expression/subject scaling factor (SSF) is the degree of alignment between the normalized best linear OrT-fit image (our covariance pattern of interest) and the original pre-background-corrected post-masking target image for that particular subject during that particular task. Each magenta line delineates the progression of subject expression from one condition to the next. As there are 10 subjects, we have 10 lines, each with 3 points, 1 for each task, along the line. An overall upward trend in activation could quite clearly be seen in our example.

In this example (a triadic analysis -- with three conditions, one baseline and two tasks), one can also find a plot on the right with a horizontal green line that shows the threshold associated with the optimal segregation. Each blue asterisk represents a contrast group estimate from a single subject. For a dyadic analysis with one baseline and one task conditions, only the SSF graph on the left is available.

In the ideal empirical scenario, the estimated/fitted group values should cluster closely around the actual group assignments, i.e. the blue asterisks for the "+1" group should be in the vicinity of 1 when viewed along the y-axis and the blue asterisks for the "-1" group should be in the vicinity of -1 when viewed along the y-axis in such a way that each group, "+1" or "-1", has fitted values that are either solely above or solely below the threshold but not scattered across both sides of the threshold, as in our example, which carries zero exceptions.

Check the status of *vars* after a contrast group fit using OrT eigenimages.


```
---------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 13


vars =

                        target_images: [517845x30 double]
                          meanful_set: [73631x1 double]
                                  dim: [79 95 69 30]
                            nOfTasks: 2
                          image_paths: [30x39 char]
                              lambdas: [19x1 double]
                  eigenimages_noZeroes: [73631x19 double]
                         eigenvectors: [20x20 double]
                                  ssf: [30x19 double]
                           design_mat: [30x20 double]
                           bg_correct: 1
                    remove_mean_image: 1
                            single_PC: []
            singlePC_icv_map_noZeroes: []
                        OrT_fit_nOfExc: 0
                  OrT_fit_selected_PCs: [1 2 3 6 9 10]
            OrT_fit_composite_PC_image: [79x95x69 double]
       OrT_fit_composite_PC_image_ssfs: [30x1 double]
             OrT_fit_icv_map_noZeroes: []
               OrT_fit_nOfExc_p_value: []
                        behav_fit_rsq: []
               behav_fit_selected_PCs: []
          behav_fit_composite_PC_image: []
     behav_fit_composite_PC_image_ssfs: []
              behav_fit_contrast_coef: []
                                behav: []
            behav_fit_icv_map_noZeroes: []
                behav_fit_rsq_p_value: []
                      last_time_saved: []


Ready to execute a new command...  To proceed, press any key.
```

Perform a bootstrap estimation of the robustness of voxel loadings in the OrT-fit covariance pattern.

```
------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

------------------------------------------------------------------------------------------
Please make your selection: 7

Number of resamplings for OrT-fit bootstrapping: 1000

Please provide filename prefix for the ICV (inverse coefficient of variation) voxel map:
     TEST
Where would you like to place the file?  Please input target path (ending WITHOUT "/"):
     MockData

Beginning OrT-fit bootstrap resampling...

     Bootstrap Sample #1 out of 1000.
     Note: The number of non-zero eigenvalues is 11.

     Bootstrap Sample #2 out of 1000.
     Note: The number of non-zero eigenvalues is 13.

     Bootstrap Sample #3 out of 1000.
     Note: The rank of eigenimages in the current sample is too low.  A replacement sample will be drawn.

     Note: The rank of eigenimages in the current sample is too low.  A replacement sample will be drawn.

     Note: The number of non-zero eigenvalues is 11.

<Omission>

     Bootstrap Sample #1000 out of 1000.
     Note: The number of non-zero eigenvalues is 11.

Elapsed time is 3184.901241 seconds.
Saving a snapshot of results obtained thus far...
     Done.
Removing saved file from previous snapshot...
     Done.

Extrema on the most current ICV map: -3.3532       3.397

Note: The latest ratio of unusable bootstrap samples (due to low rank) to usable ones is 122-to-1000.  All
unusable samples were replaced.

The final ICV map has been saved to TEST_OrT_fit_ICVNormMap_1000samples_PC1236910.img.

Completed OrT-fit bootstrap resampling.


Ready to execute a new command...  To proceed, press any key.
```

Usually, between 100 and 500 bootstrap samples would suffice at this point. We just want to be conservative and choose the unusually large number of 1000.

Every bootstrap sample is accompanied by its own data rank. Occasionally, the rank might be lower than the number of PCs included for the OrT-fit. (For our example with low data rank and a relatively high number of PCs, this happens rather often.) In this case, the bootstrap sample will be discarded and drawn again.

Check the status of *vars* after an OrT-fit bootstrap.


```
---------------------------------------------------------------------------

 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

---------------------------------------------------------------------------
Please make your selection: 13


vars =

                    target_images: [517845x30 double]
                      meanful_set: [73631x1 double]
                              dim: [79 95 69 30]
                         nOfTasks: 2
                      image_paths: [30x39 char]
                          lambdas: [19x1 double]
              eigenimages_noZeroes: [73631x19 double]
                     eigenvectors: [20x20 double]
                              ssf: [30x19 double]
                       design_mat: [30x20 double]
                       bg_correct: 1
                remove_mean_image: 1
                        single_PC: []
         singlePC_icv_map_noZeroes: []
                   OrT_fit_nOfExc: 0
             OrT_fit_selected_PCs: [1 2 3 6 9 10]
        OrT_fit_composite_PC_image: [79x95x69 double]
   OrT_fit_composite_PC_image_ssfs: [30x1 double]
          OrT_fit_icv_map_noZeroes: [73631x1 double]
           OrT_fit_nOfExc_p_value: []
                    behav_fit_rsq: []
           behav_fit_selected_PCs: []
      behav_fit_composite_PC_image: []
 behav_fit_composite_PC_image_ssfs: []
           behav_fit_contrast_coef: []
                            behav: []
        behav_fit_icv_map_noZeroes: []
            behav_fit_rsq_p_value: []
                  last_time_saved: []



Ready to execute a new command...  To proceed, press any key.
```

In general, for a PCA performed on a bootstrap sample, the number of non-zero eigenvalues is expected to be fewer than the number of non-zero eigenvalues in a similar PCA performed on the original empirical image set, since the resampling is performed with replacement. In the current example, there are 19 non-zero eigenvalues (see the field *vars.lambdas*, which holds the eigenvalues) but there are only 11 non-zero eigenvalues in Sample #1. If the number of non-zero eigenvalues in a given sample drops below the order of the selected highest-order eigenimage (in our case, that number is 10, as in the *max{[1 2 3 6 9 10]}*), the rank of the corresponding set of eigenimages would be too low for a subsequent fit. The sample has to be discarded and new samples are drawn until a set of eigenimages with sufficient rank becomes available, as illustrated in Sample #3.

Extrema on the inverse-coefficient-of-variation (ICV) map are displayed on screen throughout the resampling. In our case, there are at least two voxels in the OrT-fit covariance pattern that are contributing reliably, since |-3.4| and |3.4| are both greater than 2, which is the conventional threshold for robustness on an ICV map.

After analyzing the user-specified number of bootstrap samples, the final ICV map is written into the *OrT_fit_icv_map_noZeroes* field in *vars*.

```
----------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

----------------------------------------------------------------------------------------------------------
Please make your selection: 10

Number of resamplings for OrT-fit permutation test: 1000

Removing task-independent effects from images...
Completed background correction for all resampling image arrays.

Please provide a single filename prefix for storing sampling distributions of two default statistics (number of
exceptions to a complete separation of estimated group assignments, and non-normalized eigenvalues):
     TEST
Where would you like to place the files?  Please input target path (ending WITHOUT "/"):
     MockData

Beginning OrT-fit permutation test...

     Permutation Sample #1 (out of 1000): 2 exceptions

     Permutation Sample #2 (out of 1000): 3 exceptions

     Permutation Sample #3 (out of 1000): 1 exceptions

<Omission>

     Permutation Sample #1000 (out of 1000): 1 exceptions

Elapsed time is 3570.760369 seconds.
p-value of the number of exceptions statistic = 0.084



Sampling distributions of the two default statistics have been saved to:
     TEST_exceptions_to_separation_1001obs_PC1236910.txt
     TEST_OrT_fit_lambdas_1001by19_PC1236910.txt

Completed OrT-fit permutation test.



Ready to execute a new command...  To proceed, press any key.
```

Oops! Although the OrT fit looked quite good with zero exceptions (cf graph above) to the rule of monotonically increasing expression, it is not significant at $p < 0.05$. Using a large number of PCs for such limited data array can obviously produce seemingly impressive ordinal trends purely by chance.

```
------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 13

vars =

                        target_images: [517845x30 double]
                          meanful_set: [73631x1 double]
                                  dim: [79 95 69 30]
                            nOfTasks: 2
                          image_paths: [30x39 char]
                              lambdas: [19x1 double]
                  eigenimages_noZeroes: [73631x19 double]
                         eigenvectors: [20x20 double]
                                  ssf: [30x19 double]
                           design_mat: [30x20 double]
                           bg_correct: 1
                    remove_mean_image: 1
                            single_PC: []
             singlePC_icv_map_noZeroes: []
                       OrT_fit_nOfExc: 0
                  OrT_fit_selected_PCs: [1 2 3 6 9 10]
           OrT_fit_composite_PC_image: [79x95x69 double]
       OrT_fit_composite_PC_image_ssfs: [30x1 double]
              OrT_fit_icv_map_noZeroes: [73631x1 double]
                OrT_fit_nOfExc_p_value: 0.0840
                         behav_fit_rsq: []
                behav_fit_selected_PCs: []
          behav_fit_composite_PC_image: []
     behav_fit_composite_PC_image_ssfs: []
                  behav_fit_contrast_coef: []
                                behav: []
             behav_fit_icv_map_noZeroes: []
                 behav_fit_rsq_p_value: []
                      last_time_saved: []


Ready to execute a new command...  To proceed, press any key.
```

Check the status of *vars* after permutation tests on OrT-fit statistics.

Unlike in the previous step of bootstrapping to test voxel weight reliability in a covariance pattern, resampling under a permutation test is performed *without* replacement. The original images are permuted across tasks for each subject to gather information on two default OrT statistics, the number of exceptions to a complete separation of estimated group assignments and the non-normalized eigenvalues. An OrT analysis is performed on each permuted sample, as on the original image set.

Every twenty samples, sampling distributions of the two statistics are written to an external file. The first row in each file carry results from the original non-permuted dataset. The latest *p*-value is displayed on screen and eventually output to the field *OrT_fit_nOfExc_p_value* in *vars*.

This concludes our example on OrT data upload, analysis, and statistical inference using a mock dataset.

Raw data from this example and the associated *.mat* file that has a record of the results can be downloaded from our "GCVA Google Group" website. The files are saved as a *.zip* file under the name of *Mock_Data.zip*. You will be able to recreate all the steps shown in this example and compare your results with ours. There should be *perfect* agreement of all contents in *vars* that are recorded prior to any bootstrap and permutation procedures. Concretely, these are:

```
                 target_images: [517845x30 double]
                   meanful_set: [73631x1 double]
                           dim: [79 95 69 30]
                      nOfTasks: 2
                   image_paths: [30x39 char]
                       lambdas: [19x1 double]
           eigenimages_noZeroes: [73631x19 double]
                  eigenvectors: [20x20 double]
                           ssf: [30x19 double]
                    design_mat: [30x20 double]
                    bg_correct: 1
             remove_mean_image: 1
                OrT_fit_nOfExc: 0
           OrT_fit_selected_PCs: [1 2 3 6 9 10]
     OrT_fit_composite_PC_image: [79x95x69 double]
OrT_fit_composite_PC_image_ssfs: [30x1 double]
```

The fields populated *after* the bootstrap and permutation procedures, i.e.

```
OrT_fit_icv_map_noZeroes: [73631x1 double]
  OrT_fit_nOfExc_p_value: 0.0840
```

might not show *perfect* agreement, since their computation involves the MATLAB random number generator, which might run with a different initialization on your computer. However, the ballpark results should be similar, i.e. the *p*-level from your permutation procedure should also be around 0.08, and your ICV map should correlate with ours with $R^2 > 0.95$.

```
PCA_suite_f

  =====================================
   Ordinal Trend (OrT) Analysis Module
  =====================================

------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

------------------------------------------------------------------------------------------------------------
Please make your selection: 2

Excluding baseline condition, how many task levels are there? 0

Upload paths for target images...
  Read in paths for BASE condition (ends in .txt):
     paths_B.txt

image_paths_for_all_tasks =

B_scan1.img
B_scan2.img
B_scan3.img
B_scan4.img
B_scan5.img
B_scan6.img
B_scan7.img
B_scan8.img
B_scan9.img
B_scan10.img

Merging all pathnames to a single array variable...

Pathnames for target images are now merged.

Uploading a raw image from B_scan1.img
Uploading a raw image from B_scan2.img
Uploading a raw image from B_scan3.img
Uploading a raw image from B_scan4.img
Uploading a raw image from B_scan5.img
Uploading a raw image from B_scan6.img
Uploading a raw image from B_scan7.img
Uploading a raw image from B_scan8.img
Uploading a raw image from B_scan9.img
Uploading a raw image from B_scan10.img

Provide path information of the mask file you want to apply...

Input threshold level for the mask you want to apply: 0.5
```

Type command to activate package. (In general, green highlight represents user input. Light gray highlight represents program outputs that are specific to the current dataset. Boldface is used to emphasize program options that are being chosen.)

Input “0” for the number of task levels since there are no multiple tasks. All data is taken from the baseline condition/task level.

```
Log-transform target images (negative values will be set to zero; 1==yes/0==no)? 0

Checking the rank of target image array...
     The rank is 10.
The target image array has full rank.  Please proceed.



Ready to execute a new command...  To proceed, press any key.
```

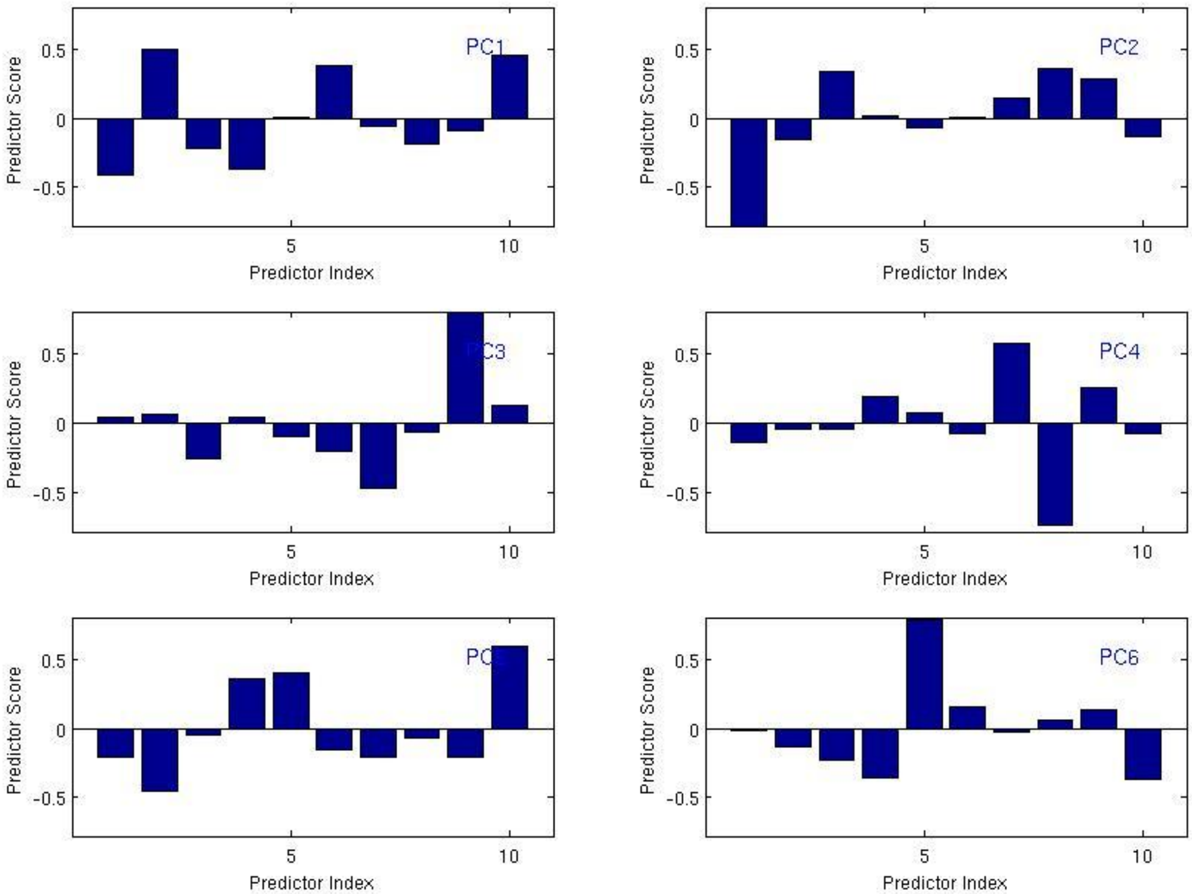

<Status of a structure in MATLAB workspace called "vars", which stores updated critical variables throughout execution; highlighted fields are those that have been created/modified after the latest command. In general, to inspect the vars structure, a user can select Option #13 from the main menu.>

```
target_images:         [517845x10 double]          single_PC:                          []
meanful_set:           [73631x1 double]            singlePC_icv_map_noZeroes:          []
dim:                   [79 95 69 10]               OrT_fit_nOfExc:                     []
nOfTasks:              0                           OrT_fit_selected_PCs:               []
image_paths:           [10x38 char]                OrT_fit_composite_PC_image:         []
lambdas:               []                          OrT_fit_composite_PC_image_ssfs:    []
eigenimages_noZeroes:  []                          OrT_fit_icv_map_noZeroes:           []
eigenvectors:          []                          OrT_fit_nOfExc_p_value:             []
ssf:                   []                          behav_fit_rsq:                      []
design_mat:            []                          behav_fit_selected_PCs:             []
bg_correct:            []                          behav_fit_composite_PC_image:       []
remove_mean_image:     []                          behav_fit_composite_PC_image_ssfs:  []
                                                   behav_fit_contrast_coef:            []
                                                   behav:                              []
                                                   behav_fit_icv_map_noZeroes:         []
                                                   behav_fit_rsq_p_value:              []
                                                   last_time_saved:                    []
```

```
-----------------------------------------------------------------------------------------------------
OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
```
3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
```
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-----------------------------------------------------------------------------------------------------
Please make your selection: 3

No design matrix will be used, since the image array has no multiple task blocks.

Remove mean image from target image data during PCA (1==yes/0==no)? 1

Finished computing eigenvalues, eigenimages, and subject scaling factors.  See plots of brain scores from the first six PCs.
(Note: Each eigenimage has been normalized to unity.  However, the set of eigenvalues are not normalized.)


Ready to execute a new command...  To proceed, press any key.
```

<Below is the status of vars in workspace after a PCA (with pre-PCA removal of mean image and no application of any design matrix)>

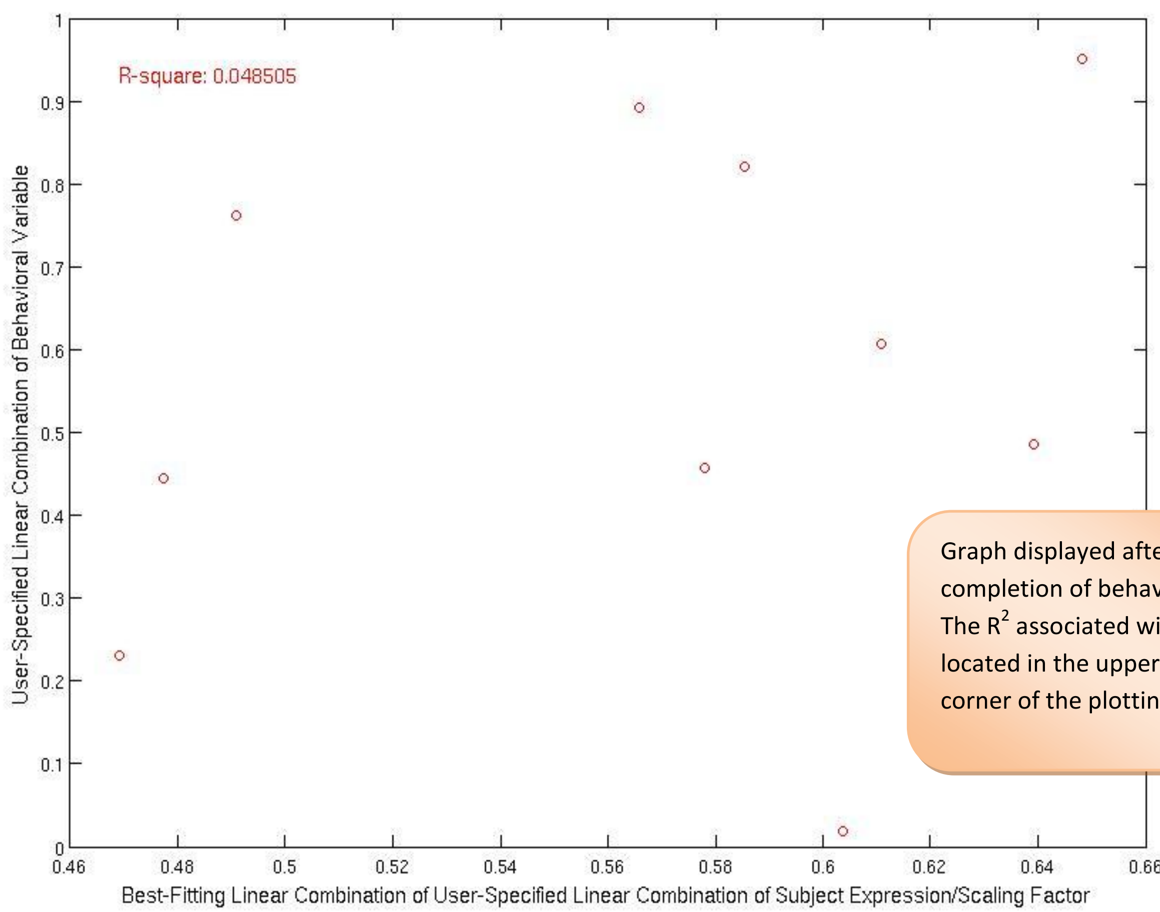
```
target_images:        [517845x10 double]        single_PC:                          []
meanful_set:          [73631x1 double]          singlePC_icv_map_noZeroes:          []
dim:                  [79 95 69 10]             OrT_fit_nOfExc:                     []
nOfTasks:             0                         OrT_fit_selected_PCs:               []
image_paths:          [10x38 char]              OrT_fit_composite_PC_image:         []
lambdas:              [9x1 double]              OrT_fit_composite_PC_image_ssfs:    []
eigenimages_noZeroes: [73631x9 double]          OrT_fit_icv_map_noZeroes:           []
eigenvectors:         [10x10 double]            OrT_fit_nOfExc_p_value:             []
ssf:                  [10x9 double]             behav_fit_rsq:                      []
design_mat:           []                        behav_fit_selected_PCs:             []
bg_correct:           0                         behav_fit_composite_PC_image:       []
remove_mean_image:    1                         behav_fit_composite_PC_image_ssfs:  []
                                                behav_fit_contrast_coef:            []
                                                behav:                              []
                                                behav_fit_icv_map_noZeroes:         []
                                                behav_fit_rsq_p_value:              []
                                                last_time_saved:                    []
```


PC1
PC2
PC3
PC4
PC5
PC6
Predictor Score
Predictor Index
0.5
0
-0.5
5
10

```
------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined
behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

------------------------------------------------------------------------------------------------------------
Please make your selection: 5

Behavioral Fit
     Independent variable: A linear combination of subject scores/subject scaling factors (SSFs)
     Dependent variable  : A linear combination of behavioral data

Which behavioral data set (in ASCII format) would you like to use?
Successfully uploaded behavioral data.

Note: There are no multiple task conditions.  All observations are made in the baseline condition.

Select a subset of all SSF linear combinations (one combination for each eigenimage) using Akaike Information
Criteria (1==yes/0==no)? 1

Computing Akaike Information Criteria for each linear combination of predictors...
     Note: There are 5 predictors.

31 sets of predictors were tested.

Under arbitrary combination of predictors, the smallest three Akaike Information Criteria values and their
respective predictors are as follows:
     (1) 11.1832 from fitting the following predictors ---> 2
     (2) 11.3509 from fitting the following predictors ---> 4
     (3) 12.6707 from fitting the following predictors ---> 1

Under sequential inclusion of predictors, the smallest three Akaike Information Criteria values and their
respective predictors are as follows:
     (4) 12.6707 from fitting the following predictors ---> 1
     (5) 16.5734 from fitting the following predictors ---> 1  2
     (6) 24.9815 from fitting the following predictors ---> 1  2  3

Among sets (1) to (6), which would you like to output for further analysis? 3

Completed behavioral fit through user-specified linear combinations of behaviorial variables and PCA subject
expressions.  See graph.

Output the linear combination of eigenimages associated with the best fit to a file (1==yes/0==no)? 0



Ready to execute a new command...  To proceed, press any key.
```

At this point, a SPM window would pop up to prompt the user for the behavioral data file.

```
<Status of vars in workspace after a behavioral fit>

target_images:          [517845x10 double]        single_PC:                         []
meanful_set:            [73631x1 double]          singlePC_icv_map_noZeroes:         []
dim:                    [79 95 69 10]             OrT_fit_nOfExc:                    []
nOfTasks:               0                         OrT_fit_selected_PCs:              []
image_paths:            [10x38 char]              OrT_fit_composite_PC_image:        []
lambdas:                [9x1 double]              OrT_fit_composite_PC_image_ssfs:   []
eigenimages_noZeroes:   [73631x9 double]          OrT_fit_icv_map_noZeroes:          []
eigenvectors:           [10x10 double]            OrT_fit_nOfExc_p_value:            []
ssf:                    [10x9 double]             behav_fit_rsq:                     0.0485
design_mat:             []                        behav_fit_selected_PCs:            1
bg_correct:             0                         behav_fit_composite_PC_image:      [79x95x69 double]
remove_mean_image:      1                         behav_fit_composite_PC_image_ssfs: [10x1 double]
                                                  behav_fit_contrast_coef:           1
                                                  behav:                             [10x1 double]
                                                  behav_fit_icv_map_noZeroes:        []
                                                  behav_fit_rsq_p_value:             []
                                                  last_time_saved:                   []
```



Graph displayed after completion of behavioral fit. The R^2 associated with the fit is located in the upper left hand corner of the plotting area.

```
--------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 8

Number of resamplings for behavioral-fit bootstrapping: 1000

Please provide filename prefix for the ICV (inverse coefficient of variation) voxel map:
     TEST
Where would you like to place the file?  Please input target path (ending WITHOUT "/"):
     /Test_Folder

Beginning behavioral-fit bootstrap resampling...

     Bootstrap Sample #1 out of 1000.
     Note: The number of non-zero eigenvalues is 7.

     Bootstrap Sample #2 out of 1000.
     Note: The number of non-zero eigenvalues is 7.

     Bootstrap Sample #3 out of 1000.
     Note: The number of non-zero eigenvalues is 6.

     Bootstrap Sample #4 out of 1000.
     Note: The number of non-zero eigenvalues is 5.

     Bootstrap Sample #5 out of 1000.
     Note: The number of non-zero eigenvalues is 6.
```

<Omission>

```
     Bootstrap Sample #1000 out of 1000.
     Note: The number of non-zero eigenvalues is 6.

Elapsed time is 1084.560422 seconds.
Saving a snapshot of results obtained thus far...
     Done.
Removing saved file from previous snapshot...
     Done.

Extrema on the most current ICV map: -1.6443      1.5761

The final ICV map has been saved to /Test_Folder/TEST_behav_fit_ICVNormMap_1000samples_PC1.img.

Completed behavioral-fit bootstrap resampling.


Ready to execute a new command...  To proceed, press any key.
```

```
<Status of vars in workspace after completion of a behavioral-fit bootstrap>

target_images:         [517845x10 double]        single_PC:                          []
meanful_set:           [73631x1 double]          singlePC_icv_map_noZeroes:          []
dim:                   [79 95 69 10]             OrT_fit_nOfExc:                     []
nOfTasks:              0                         OrT_fit_selected_PCs:               []
image_paths:           [10x38 char]              OrT_fit_composite_PC_image:         []
lambdas:               [9x1 double]              OrT_fit_composite_PC_image_ssfs:    []
eigenimages_noZeroes:  [73631x9 double]          OrT_fit_icv_map_noZeroes:           []
eigenvectors:          [10x10 double]            OrT_fit_nOfExc_p_value:             []
ssf:                   [10x9 double]             behav_fit_rsq:                      0.0485
design_mat:            []                        behav_fit_selected_PCs:             1
bg_correct:            0                         behav_fit_composite_PC_image:       [79x95x69 double]
remove_mean_image:     1                         behav_fit_composite_PC_image_ssfs:  [10x1 double]
                                                 behav_fit_contrast_coef:            1
                                                 behav:                              [10x1 double]
                                                 behav_fit_icv_map_noZeroes:         [73631x1 double]
                                                 behav_fit_rsq_p_value:              []
                                                 last_time_saved:                    []
```

In general, for a PCA performed on a bootstrap sample, the number of non-zero eigenvalues is expected to be fewer than the number of non-zero eigenvalues in a similar PCA performed on the original empirical image set, since the resampling is performed with replacement. In the current example, there are 9 non-zero eigenvalues (see the field *vars.lambdas*, which holds the eigenvalues; these were derived during execution of menu Option #3) but there are only 7 non-zero eigenvalues in Sample #1.

Results from bootstrap resampling are written to an external file every twenty samples. Extrema on the latest inverse-coefficient-of-variation (ICV) map is also displayed on screen. In our case, the absolute value of each of the two extrema is clearly lower than 2, the conventional threshold for reliability of voxel-weight contribution to covariance patterns on an ICV map, indicating the lack of robustness in *every* voxel in the behavioral fit covariance pattern. The name of the external file is also updated to reflect the number of samples used to obtain the ICV map.

After analyzing the user-specified number of bootstrap samples, the final ICV map is written into a field in *vars*.

Generally, if the loading of a voxel in a covariance pattern is robustly positive (> 2 on the ICV map), its activity in the brain is localized and increases with task difficulty level. If the loading is robustly negative (< -2 on the ICV map), its activity is localized and decreases with task difficulty. If the loading is not robust, there is no localization at the voxel *but* the voxel could nonetheless contribute to the overall covariance pattern across task levels in a dispersive fashion in conjunction with other voxels within the pattern.

```
----------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of
behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

----------------------------------------------------------------------------------------------------------------
Please make your selection: 11

Note:
     According to previous user selection, which forgoes specification of linear combination for both images and
behavioral responses, there is no segregation of data into task blocks.  Subsequently, the following analysis
pertains to subject effects instead of task effects.

Number of resamplings for behavioral-fit permutation test: 1000

Please provide a filename prefix for storing behavioral fit R-square statistic:
     TEST
Where would you like to place the file?  Please input target path (ending WITHOUT "/"):
     /Test_Folder

Beginning behavioral-fit permutation test...

Note: All subjects virtually belong to only one task condition block, the baseline.  Subject effects, instead of
task effects, will be tested.

     Permutation Sample #1 (out of 1000): R-Square = 0.011844

     Permutation Sample #2 (out of 1000): R-Square = 0.20162

     Permutation Sample #3 (out of 1000): R-Square = 7.9577e-05

     Permutation Sample #4 (out of 1000): R-Square = 0.028689

     Permutation Sample #5 (out of 1000): R-Square = 0.1166
```

<Omission>

```
     Permutation Sample #1000 (out of 1000): R-Square = 0.077262

Elapsed time is 1038.453963 seconds.
p-value of behavioral fit R-square statistic = 0.525

Sampling distribution of behavioral fit R-square has been saved to:
     /Test_Folder/TEST_behav_fit_rsq_1001obs_PC1.txt

Completed behavioral-fit permutation test.
```

From the *p*-value obtained through the permutation test, we can see that the point-estimated behavioral fit R^2 of 4.85% is clearly insignificant.

```
<Status of vars in workspace after completion of a permutation test on behavioral fit R-square>

target_images:         [517845x10 double]         single_PC:                          []
meanful_set:           [73631x1 double]           singlePC_icv_map_noZeroes:          []
dim:                   [79 95 69 10]              OrT_fit_nOfExc:                     []
nOfTasks:              0                          OrT_fit_selected_PCs:               []
image_paths:           [10x38 char]               OrT_fit_composite_PC_image:         []
lambdas:               [9x1 double]               OrT_fit_composite_PC_image_ssfs:    []
eigenimages_noZeroes:  [73631x9 double]           OrT_fit_icv_map_noZeroes:           []
eigenvectors:          [10x10 double]             OrT_fit_nOfExc_p_value:             []
ssf:                   [10x9 double]              behav_fit_rsq:                      0.0485
design_mat:            []                         behav_fit_selected_PCs:             1
bg_correct:            0                          behav_fit_composite_PC_image:       [79x95x69 double]
remove_mean_image:     1                          behav_fit_composite_PC_image_ssfs:  [10x1 double]
                                                  behav_fit_contrast_coef:            1
                                                  behav:                              [10x1 double]
                                                  behav_fit_icv_map_noZeroes:         [73631x1 double]
                                                  behav_fit_rsq_p_value:              0.5250
                                                  last_time_saved:                    []
```

```
--------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 15

Where would you like to save your workspace variables?  Please provide a name for the directory:
(A path could be placed in front of the directory name, i.e. <"path"/"directory name">)
     /Test_Folder
A directory by that name currently exists.  No new directory will be created.

Writing the following structure to a file named workspace_snapshot_20080903.mat...

vars =

                        target_images: [517845x10 double]
                          meanful_set: [73631x1 double]
                                  dim: [79 95 69 10]
                             nOfTasks: 0
                          image_paths: [10x38 char]
                              lambdas: [9x1 double]
                 eigenimages_noZeroes: [73631x9 double]
                         eigenvectors: [10x10 double]
                                  ssf: [10x9 double]
                           design_mat: []
                           bg_correct: 0
                    remove_mean_image: 1
                            single_PC: []
            singlePC_icv_map_noZeroes: []
                       OrT_fit_nOfExc: []
                 OrT_fit_selected_PCs: []
           OrT_fit_composite_PC_image: []
      OrT_fit_composite_PC_image_ssfs: []
             OrT_fit_icv_map_noZeroes: []
               OrT_fit_nOfExc_p_value: []
                        behav_fit_rsq: 0.0485
               behav_fit_selected_PCs: 1
         behav_fit_composite_PC_image: [79x95x69 double]
    behav_fit_composite_PC_image_ssfs: [10x1 double]
              behav_fit_contrast_coef: 1
                                behav: [10x1 double]
           behav_fit_icv_map_noZeroes: [73631x1 double]
                behav_fit_rsq_p_value: 0.5250
                      last_time_saved: [2008 9 3 10 35 24.4746]

Crucial variables in the workspace have been saved to the specified directory and file.


Ready to execute a new command...  To proceed, press any key.
```

```
----------------------------------------------------------------------------------------------------

 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

----------------------------------------------------------------------------------------------------
Please make your selection: 13


vars =

                        target_images: [517845x10 double]
                          meanful_set: [73631x1 double]
                                  dim: [79 95 69 10]
                            nOfTasks: 0
                          image_paths: [10x38 char]
                              lambdas: [9x1 double]
                 eigenimages_noZeroes: [73631x9 double]
                         eigenvectors: [10x10 double]
                                  ssf: [10x9 double]
                           design_mat: []
                           bg_correct: 0
                    remove_mean_image: 1
                            single_PC: []
             singlePC_icv_map_noZeroes: []
                       OrT_fit_nOfExc: []
                 OrT_fit_selected_PCs: []
            OrT_fit_composite_PC_image: []
       OrT_fit_composite_PC_image_ssfs: []
             OrT_fit_icv_map_noZeroes: []
                OrT_fit_nOfExc_p_value: []
                         behav_fit_rsq: 0.0485
                 behav_fit_selected_PCs: 1
            behav_fit_composite_PC_image: [79x95x69 double]
      behav_fit_composite_PC_image_ssfs: [10x1 double]
                 behav_fit_contrast_coef: 1
                                  behav: [10x1 double]
            behav_fit_icv_map_noZeroes: [73631x1 double]
                    behav_fit_rsq_p_value: 0.5250
                          last_time_saved: [2008 9 3 10 35 24.4746]

Important Reminder:
     When "behav_fit_contrast_coef" equals "1", all task condition blocks are merged.  During resampling,
subject image-behavioral response pairs are not permuted across multiple blocks of task condition; subject
images are permuted within the single merged block without being paired up with their corresponding behavioral
responses.  Consequently, output from a permutation test ("behav_fit_rsq_p_value") will be a measure of subject
effects, not of task effects.


Ready to execute a new command...  To proceed, press any key.
```

In general, the module issues an "Important Reminder" under Option #13 only when the *vars* field *behav_fit_contrast_coef* equals 1. In our case, since there are no multiple task conditions, the field was set to 1 by default during the behavioral fit (Option #5).

Technically, the "1" is **not** a coefficient, especially when multiple tasks are specified at the image upload stage of the analysis such that *nOfTasks* is strictly greater than 0. The "1" acts as an indicator when the number of tasks is *effectively* zero, i.e. when the user decides to forgo specification of linear combination and perform a behavioral fit across all task conditions.

```
--------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 16

...Ordinal Trend analysis module terminated.
```

```
------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

------------------------------------------------------------------------------------------------
Please make your selection: 13


vars =

                        target_images: [517845x50 double]
                          meanful_set: [84097x1 double]
                                  dim: [79 95 69 50]
                            nOfTasks: 1
                          image_paths: [50x85 char]
                              lambdas: []
                  eigenimages_noZeroes: []
                         eigenvectors: []
                                  ssf: []
                           design_mat: []
                           bg_correct: []
                    remove_mean_image: []
                            single_PC: []
              singlePC_icv_map_noZeroes: []
                      OrT_fit_nOfExc: []
                   OrT_fit_selected_PCs: []
              OrT_fit_composite_PC_image: []
          OrT_fit_composite_PC_image_ssfs: []
                OrT_fit_icv_map_noZeroes: []
                 OrT_fit_nOfExc_p_value: []
                        behav_fit_rsq: []
                 behav_fit_selected_PCs: []
            behav_fit_composite_PC_image: []
        behav_fit_composite_PC_image_ssfs: []
                behav_fit_contrast_coef: []
                                behav: []
              behav_fit_icv_map_noZeroes: []
                   behav_fit_rsq_p_value: []
                       last_time_saved: []


Ready to execute a new command...  To proceed, press any key.
```

In this example, we start with an imported image dataset whose *vars* structure attributes are as displayed to the left of this box. First, we are going to perform a principal component analysis (PCA) on the dataset to extract a set of covariance patterns – eigenimages/principal components (PCs). Then, we are going to compute the variance-accounted-for in the untransformed dataset by each of those components.

The field *vars.dim* provides information on the target image array. The first three numbers represent the voxel dimensions of each image in 3-D, while the fourth entry indicates the number of images in the entire image array. Notice that there are 79 × 95 × 69 = 517,845 voxels in each of the 50 target images (25 for the baseline condition and 25 for the task condition, with the same subject ordering).

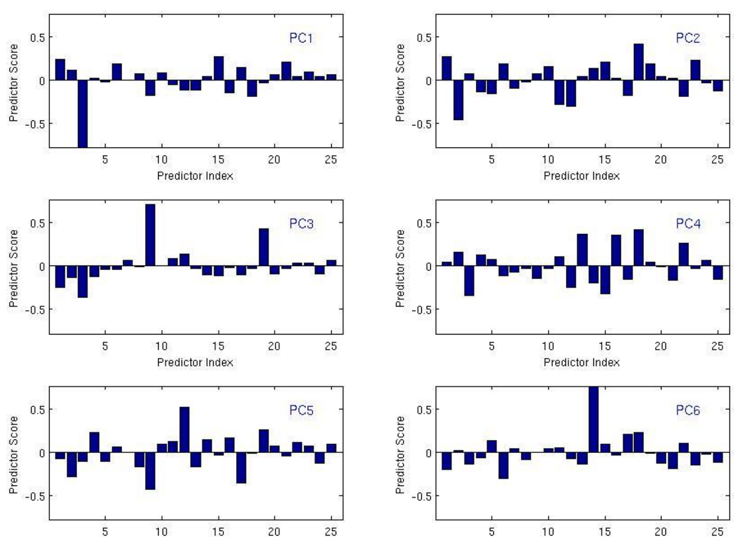
The field *vars.meanful_set* represents the non-zero post-masking voxel subset of those 517, 845 voxels that will be used throughout the analysis. It is a 1-D representation. For this analysis we uploaded the data from a previous application (option #1), rather than starting afresh (option #2).

In general, green highlight represents user input. Light gray highlight represents program outputs that are specific to the current dataset.

```
-------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
```
3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
```
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-------------------------------------------------------------------------------------------------
Please make your selection: 3

Perform a principal component analysis without prior application of design matrix (1==yes/0==no)? 0

Remove task-independent effects from target image data prior to pattern extraction (1==yes/0==no)? 1
Completed background correction.  Task-independent effects have been removed from target image array.

     The size of the target image array is 517845 (# of voxels) by 50 (# of subject-task).
     There is a total of 2 sequential task block(s), each of which consists of 25 subject(s).
     Any design matrix must have 50 row(s).

Use OrT design matrix (1==yes/0==no)? 1
     OrT design matrix will be used.
```

To extract covariance patterns, we perform a PCA. In this case, we opt for an OrT, which requires background correction and an OrT design matrix.

```
Finished computing eigenvalues, eigenimages, and subject scaling factors.  See plots of brain scores from the
first six PCs.
(Note: Each eigenimage has been normalized to unity.  However, the set of eigenvalues are not normalized.)


Ready to execute a new command...  To
proceed, press any key.
```



Let's inspect the *vars* structure to confirm the existence of a set of patterns after the OrT PCA.

```
-----------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-----------------------------------------------------------------------------------------------------------
Please make your selection: 13


vars =

                    target_images: [517845x50 double]
                      meanful_set: [84097x1 double]
                              dim: [79 95 69 50]
                        nOfTasks: 1
                      image_paths: [50x85 char]
                          lambdas: [24x1 double]
              eigenimages_noZeroes: [84097x24 double]
                      eigenvectors: [25x25 double]
                              ssf: [50x24 double]
                       design_mat: [50x25 double]
                       bg_correct: 1
                 remove_mean_image: 1
                        single_PC: []
         singlePC_icv_map_noZeroes: []
                     OrT_fit_nOfExc: []
               OrT_fit_selected_PCs: []
         OrT_fit_composite_PC_image: []
    OrT_fit_composite_PC_image_ssfs: []
          OrT_fit_icv_map_noZeroes: []
             OrT_fit_nOfExc_p_value: []
                      behav_fit_rsq: []
             behav_fit_selected_PCs: []
        behav_fit_composite_PC_image: []
   behav_fit_composite_PC_image_ssfs: []
            behav_fit_contrast_coef: []
                              behav: []
           behav_fit_icv_map_noZeroes: []
               behav_fit_rsq_p_value: []
                    last_time_saved: []


Ready to execute a new command...  To proceed, press any key.
```

After the OrT PCA, seven additional fields in *vars* are filled (the ones in cyan highlight). In particular, the set of covariance patterns (the set of eigenimages/PCs) from the PCA are stored in a field named *vars.eigenimages_noZeroes*. Parameters associated with the PCA are recorded in three separate fields:

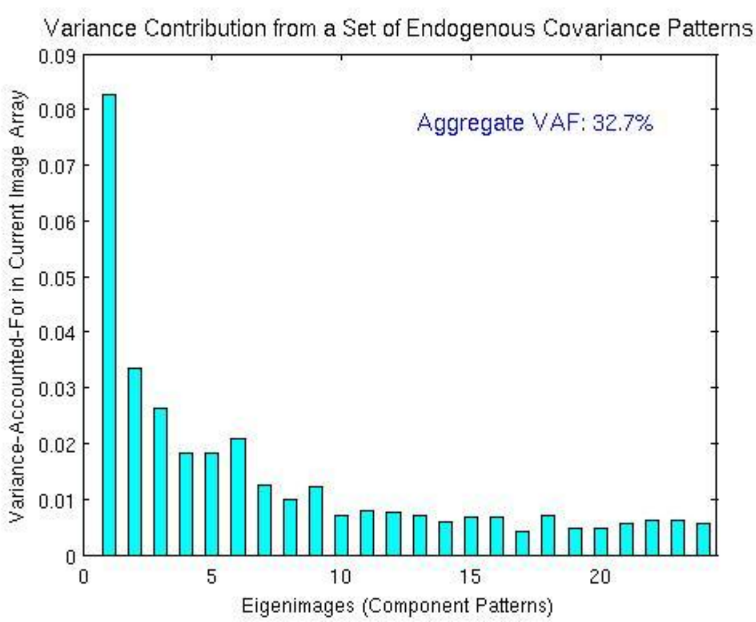
i. *vars.design_mat*
ii. *vars.bg_correct*
iii. *vars.remove_mean_image*

These three fields, along with *vars.eigenimages_noZeroes*, are essential in calculating variance-accounted-for (VAF) in the next step.

Next, we access the workspace to execute the script *endo_VAF_s.m* to compute the VAF by each of the 24 patterns.


```
----------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

---------------------------------
Please make your selection: 14

endo_VAF_s

Extract variance-accounted-for (VAF) from original target image array without any adjustments for background
correction or design matrix (1==yes/0==no)? 1

The aggregate variance-accounted-for (VAF) in the current image array by the endogenous covariance patterns is:
     32.7223%
Contributions from individual component patterns have been placed in a vector called "endo_based_VAF" in the
workspace.

View scree plot of variance-accounted-for (VAF) by each component pattern (1==yes/0==no)? 1
     The plot is on display.
```

With no "adjustments for background correction or design matrix", the script computes VAF in the untransformed images by the covariance patterns.



From the scree plot, one could visually inspect the distribution of VAFs across component patterns of the PCA. In our case, there are 24 component patterns, one for each column in *vars.eigenimages_noZeroes*.

To check for presence of the newly created variable storing the VAF by each pattern, type in *whos*. In addition to the permanent structure of *vars*, a variable named *endo_based_VAF* with 24 columns now resides in the workspace. The scree plot is simply a bar plot of that variable.

The VAFs are characterized as endogenous because they are computed with respect to a set of endogenously-derived covariance patterns, as opposed to an imported set of covariance patterns from an external set of images.

```
 whos
  Name                Size                 Bytes  Class     Attributes

  action              1x2                      4  char
  endo_based_VAF      1x24                   192  double
  vars                1x1              223994344  struct
```

Since we calculated VAF on the set of *untransformed* images, some component patterns contributed more than their higher order counterparts, namely, the ones to their left in the plot. Additionally, the aggregate VAF of 33% is less than 100%. Note that in an ordinary scree plot, principal components are ranked according to their normalized lambdas/eigenvalues, i.e. their variance contribution *to the dataset they were derived from*. The current example is slightly different: here, we plot the variance contribution to the original dataset of untransformed images by the basis set of PCs, which were derived from the data *after* background correction and transformation with the OrT design matrix.

If we were to compute VAF on the *transformed* images, accounting for the effects of background correction and design matrix prior to the PCA, VAF magnitude across component patterns from high to low order (left to right in our plot) would be strictly non-increasing. Furthermore, they would sum to 100%, since the set of PCs is an exhaustive re-expression of the covariance matrix derived from the transformed – background-corrected and design-matrix-modified – images. To illustrate this, we run *endo_VAF_s.m* again, but this time with “adjustments for background correction or design matrix”. The resulting aggregate VAF (in the transformed images) by all pattern components is indeed 100%.

```
endo_VAF_s

Extract variance-accounted-for (VAF) from original target image array without any adjustments for background
correction or design matrix (1==yes/0==no)? 0

The aggregate variance-accounted-for (VAF) in the current image array by the endogenous covariance patterns is:
     100%
Contributions from individual component patterns have been placed in a vector called "endo_based_VAF" in the
workspace.

View scree plot of variance-accounted-for (VAF) by each pattern component (1==yes/0==no)? 0
```

Having completed the VAF calculations, the user can opt to save a snapshot of the workspace, further analyze the data, exit the program, etc.

```
------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

------------------------------------------------------------------------------------------------------------
Please make your selection: 2

Excluding baseline condition, how many task levels are there? 1

Upload paths for target images...
  Read in paths for BASE condition (ends in .txt):
     Target_Data_Base.txt

image_paths_for_all_tasks =

<Omission>

  Read in paths for TASK condition #1 (ends in .txt):
     Target_Data_Task1.txt

paths_E1 =

<Omission>

Merging all pathnames to a single array variable...

Pathnames for target images are now merged.

<Omission>

Finished uploading target images.

Provide path information of the mask file you want to apply...

Input threshold level for the mask you want to apply: 0.5

Log-transform target images (negative values will be set to zero; 1==yes/0==no)? 0

Checking the rank of target image array...
     The rank is 50.
The target image array has full rank.  Please proceed.



Ready to execute a new command...  To proceed, press any key.
```

For the first step in our example, we clear the current workspace and upload a set of target images for which we will calculate subject expressions of a set of exogenous patterns. In the current example, we are using a dataset with one task level, exclusive of baseline.

Note that we assume that a set of exogenous patterns already exist in *.img* format, with dimensions identical to each of our target images. Paths to all of these patterns should be placed in a *.txt* file in a row-by-row format.

In general, green highlight represents user input. Light gray highlight represents program outputs that are specific to the current dataset.

```
-------------------------------------------------------------------------------------------------------------

 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-------------------------------------------------------------------------------------------------------------
Please make your selection: 13

vars =

                   target_images: [517845x50 double]
                     meanful_set: [66128x1 double]
                             dim: [79 95 69 50]
                       nOfTasks: 1
                     image_paths: [50x83 char]
                         lambdas: []
           eigenimages_noZeroes: []
                    eigenvectors: []
                             ssf: []
                      design_mat: []
                      bg_correct: []
               remove_mean_image: []
                       single_PC: []
       singlePC_icv_map_noZeroes: []
                  OrT_fit_nOfExc: []
            OrT_fit_selected_PCs: []
      OrT_fit_composite_PC_image: []
 OrT_fit_composite_PC_image_ssfs: []
        OrT_fit_icv_map_noZeroes: []
          OrT_fit_nOfExc_p_value: []
                    behav_fit_rsq: []
          behav_fit_selected_PCs: []
     behav_fit_composite_PC_image: []
behav_fit_composite_PC_image_ssfs: []
          behav_fit_contrast_coef: []
                            behav: []
      behav_fit_icv_map_noZeroes: []
            behav_fit_rsq_p_value: []
                  last_time_saved: []


Ready to execute a new command...  To proceed, press any key.
```

The field *vars.dim* provides information on the target image array. The first three numbers represent the voxel dimensions of each image in 3-D, while the fourth entry indicates the number of images in the entire image array. Notice that there are 79 × 95 × 69 = 517,845 voxels in each of the 50 target images (25 for the baseline condition and 25 for the task condition, with the same subject ordering).

The field *vars.meanful_set* represents the non-zero post-masking voxel subset of those 517, 845 voxels that will be used throughout the analysis. It is a 1-D representation.

As we can see, the target images have been successfully uploaded, but no PCA has been performed. In fact, in this demonstration, no PCA will be performed explicitly; instead of using a set of endogenously-derived covariance patterns from the existing dataset, we are going to forward-apply an *imported* set of exogenous patterns from another dataset to calculate subject expressions/subject scaling factors (SSFs). Notice that all the fields in *vars* that are associated with covariance patterns (those in cyan highlight) are empty.

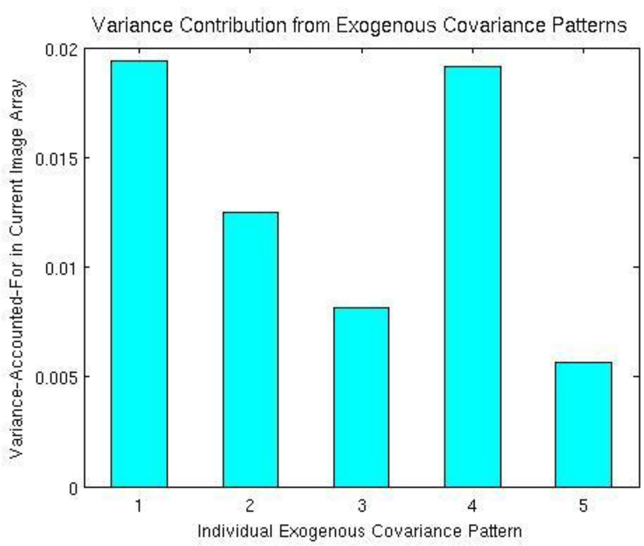
```
--------------------------------------------------------------------------------------------------------------

 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------
Please make your selection: 14

pattern_expression_s

Upload paths for pattern images...
  Read in paths for pattern images (ends in .txt):
     Pattern_Paths.txt

image_paths_for_all_tasks =

/PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC1.img
/PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC2.img
/PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC3.img
/PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC123.img
/PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC8912.img

Uploading a pattern image from /PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC1.img
Uploading a pattern image from /PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC2.img
Uploading a pattern image from /PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC3.img
Uploading a pattern image from /PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC123.img
Uploading a pattern image from /PROJECTION/Exo_Pattern_contrast_ssf_BestFit_PC8912.img

Finished uploading pattern images.  The full path and filename of the file could be found in a string named
"exo_pattern_file" in the workspace.

Remove task-independent effects from target image data prior to pattern projection (1==yes/0==no)? 0

Apply a design matrix to target image array prior to computing variance-accounted-for by patterns
(1==yes/0==no)? 0

Remove mean image from target image data during PCA (1==yes/0==no)? 1

Finished computing subject expressions/subject scaling factors (SSFs) from the exogenous covariance patterns.
The results have been placed in a matrix called "exo_based_ssf" in the workspace.

Contributions from individual component patterns have been placed in a vector called "exo_based_VAF" in the
workspace.
```

Next, we run the script *pattern_expression_s.m* in the workspace to compute expressions of exogenous patterns by each subject in the current image array. Here, the text file *Pattern_Paths.txt* contains five paths (they are listed as part of the output to screen), one for each covariance pattern that we are interested in, listed in rows. The patterns need not to be mutually orthogonal or individually normalized. In fact, they could have originated independently from multiple distinct datasets.

Since we are interested in *subject-task effects* in the *untransformed* images, we have opted for mean image removal (to measure subject-task effects rather than main effects) without the application of background correction (to remove task-independent effects) and design matrix (to accentuate specific patterns of interests associated with the design of an experiment).

```
Examine expression of each exogenous pattern by each subject under each task condition (1==yes/0==no)? 1
     Number of Topographic Profile Rating (TPR) plot(s) on display: 5

If any of the exogenous patterns you uploaded was derived from an Ordinal Trend (OrT) analysis previously, a
permutation test on the prospectively applied OrT number-of-exceptions statistic is available now.  Among
the 5 prospectively applied exogenous patterns, select one on which to perform a permutation test (input "0"
to skip test): 4

Forward application of pattern #4 yields an OrT number-of-exceptions statistic with a point estimated value
of 3 (out of 25 subjects).
     How many permutation samples would you like to draw? 1000
     Computing...
Completed permutation test.  The p-value associated with OrT number-of-exceptions statistic in a forward
application of pattern #4 onto the existing images is 0.023.

Please input a pattern number (between 1 and 5) for the next
OrT statistic permutation test (input "0" to
skip test): 0

View scree plot of variance-accounted-for (VAF) by each
component pattern (1==yes/0==no)? 1
     The plot is on display.
```

For each sample, task ordering of each subject is permuted separately. SSFs and OrT statistic are then computed using the specified OrT pattern.




```
whos
  Name                       Size                   Bytes  Class

  action                     1x2                        4  char
  exo_based_VAF              1x5                       40  double
  exo_based_ssf             50x5                     2000  double
```

Note that there are five patterns, one for each covariance pattern. The first three represent the first three PCs from the original external dataset. The first PC also turns out to be, under optimization using Akaike Information Criteria (AIC), the best-fit linear combination of contiguous PCs. By contiguous PCs, we are referring to a sequential inclusion of lower order PCs starting from the highest order PC, namely, the first PC. Pattern #4 represents an AIC suboptimal case with the best-fit linear combination of the first three PCs. With no contiguity constraint, the best-fit linear combination would have consisted of PCs #8, #9, and #12. This is our pattern #5.

VAF contribution from pattern #5 (about 0.6%) to the existing target image array is the least among all five exogenous patterns, even though the pattern is the AIC-based optimal linear fit from the original external dataset. On the contrary, AIC optimization with contiguity restriction yields more robust results. Both contiguous patterns #1 and #4 clearly have higher VAFs than pattern #5.

Although the VAF by each of these patterns (derived from the original external dataset) is quite low here, one should keep in mind that the patterns were initially obtained by running a PCA on background-corrected images that were also subsequently transformed with a design matrix. VAFs in this *transformed space of the original external dataset* are much larger. In our case, pattern #1 has a VAF of about 37 percent (not shown here), as opposed to the two percent during a forward/prospective application (see pattern #1 in the plot) with no transformations.

Finally, let's revisit the workspace after execution of the script by issuing the command *whos*. In addition to the permanent structure of *vars*, three variables have been created.

A variable named *exo_based_VAF* with five columns now resides in the workspace. The scree plot is simply a bar plot of that variable. *Exo_based_ssf* holds all subject expressions of each of the five exogenous covariance patterns. Since there are 25 subjects and two conditions (one baseline and one task), there are 50 subject expressions for each covariance pattern. Lastly, *exo_pattern_file* simply stores the names and paths of the exogenous patterns that were uploaded and is strictly for recovery purposes.

```
PCA_suite_f

  =====================================
   Ordinal Trend (OrT) Analysis Module
  =====================================
```

Type command to activate package. (In general, green highlight represents user input. Light gray highlight represents program outputs that are specific to the current dataset.)


```
-----------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-----------------------------------------------------------------------------------------------------------
Please make your selection: 1

Where is the file holding the variables for upload into the current workspace?

Uploading variables...
The workspace is now populated with variables from the file you selected.  See below.

vars =

                        target_images: [517845x50 double]
                          meanful_set: [84097x1 double]
                                  dim: [79 95 69 50]
                            nOfTasks: 1
                          image_paths: [50x85 char]
                              lambdas: []
                   eigenimages_noZeroes: []
                         eigenvectors: []
                                  ssf: []
                           design_mat: []
                           bg_correct: []
                      remove_mean_image: []
                            single_PC: []
              singlePC_icv_map_noZeroes: []
                        OrT_fit_nOfExc: []
                   OrT_fit_selected_PCs: []
              OrT_fit_composite_PC_image: []
         OrT_fit_composite_PC_image_ssfs: []
               OrT_fit_icv_map_noZeroes: []
                OrT_fit_nOfExc_p_value: []
                         behav_fit_rsq: []
                 behav_fit_selected_PCs: []
            behav_fit_composite_PC_image: []
       behav_fit_composite_PC_image_ssfs: []
                behav_fit_contrast_coef: []
                                behav: []
             behav_fit_icv_map_noZeroes: []
                  behav_fit_rsq_p_value: []
                      last_time_saved: [2008 7 1 10 53 49.7229]

 Ready to execute a new command...  To proceed, press any key.
```

The field *dim* provides information on the target image array. The first three numbers represent the voxel dimensions of each image in 3-D, while the fourth entry indicates the number of images in the entire image array. Notice that there are 79 × 95 × 69 = 517,845 voxels in each of the 50 target images (25 for the baseline condition and 25 for the task condition, with the same subject ordering).

The field *meanful_set* represents the non-zero post-masking voxel subset of those 517, 845 voxels that will be used throughout the analysis. It is a 1-D representation. For this analysis we uploaded the data from a previous application (option #1), rather than starting afresh (option #2).

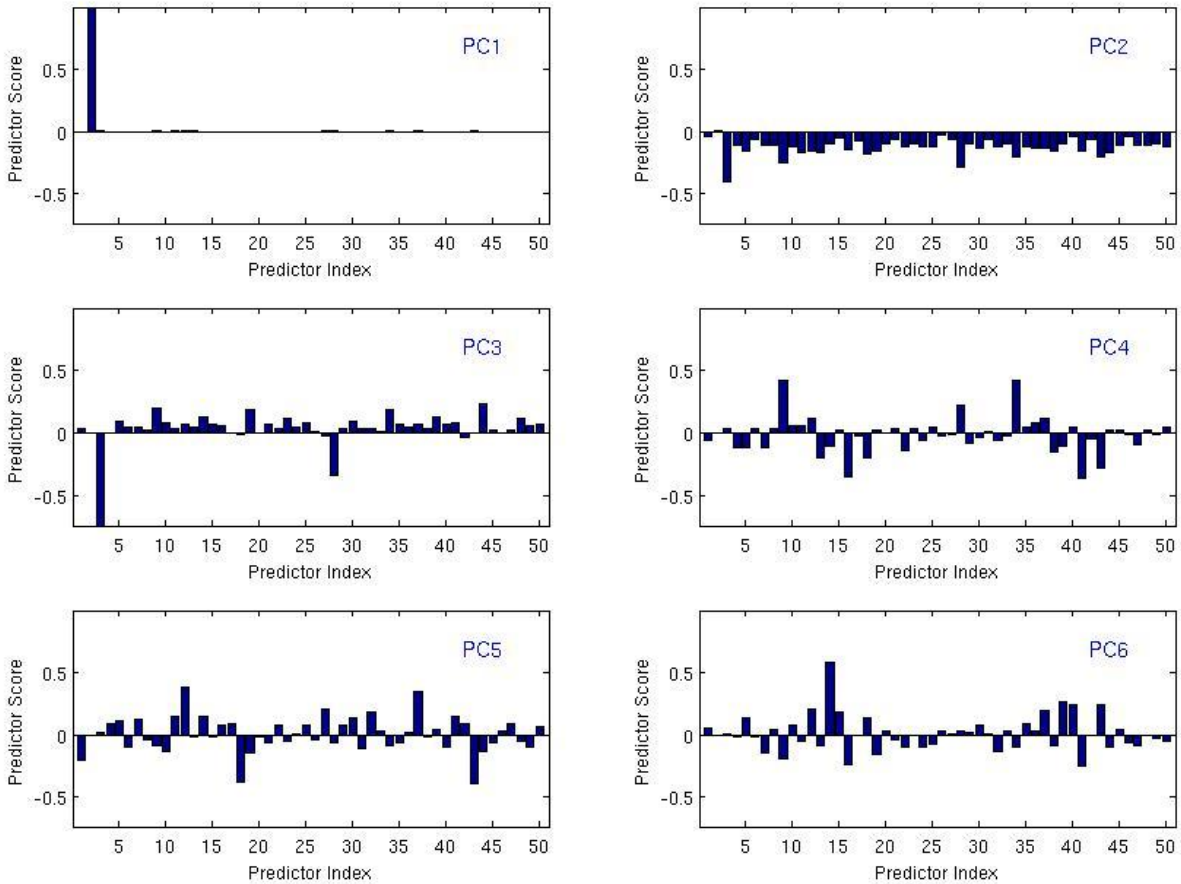
```
--------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design
matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 3

Perform a principal component analysis without prior application of design matrix (1==yes/0==no)? 1

Remove task-independent effects from target image data prior to pattern extraction (1==yes/0==no)? 0

Remove mean image from target image data during PCA (1==yes/0==no)? 0

Finished computing eigenvalues, eigenimages, and subject scaling factors.  See plots of brain scores from the
first six PCs.
(Note: Each eigenimage has been normalized to unity.  However, the set of eigenvalues are not normalized.)


Ready to execute a new command...  To proceed, press any key.
```

The target image array undergoing the PCA has one voxel that has been manipulated to assume a value that is 10,000 times its original value. Through a rudimentary application of PCA – without using any design matrix, background correction, etc. – we hope to identify it. In practice, such an anomalous voxel value could have resulted from an error in a preprocessing step, for instance, a motion artifact with a subsequently misaligned co-registration.

From the graphical output, a plot of brain scores from PC #1, we could see that the 2^{nd} predictor dominates in predictor score. *Heuristically but in no way formally*, this suggests that the anomalous voxel might reside in scan #2 in our target image array and that scan #2 essentially contributes *all* the variance in the data. To ascertain if this is indeed the case and to pinpoint and identify the outlying voxel, we need to run a script to check the following:

1. Does the first normalized eigenvalue (lambda) clearly dominate all succeeding normalized eigenvalues?
2. Is the subject expression of PC1 associated with any particular scan overwhelmingly large (orders of magnitude larger than those associated with other scans)?
3. Is there a virtually perfect correlation between the first eigenimage and one of the scans? If the scans identified in questions #2 and #3 are the same, that scan is likely the faulty one.

Trial #1 – Brain score plots from a plain PCA



```
--------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

--------------------------------------------------------------------------------------------------------------
Please make your selection: 14

single_vox_anomaly_s
```

Run the script *single_vox_anomaly_s* within the workspace to identify a suspect voxel.

```
Testing target image array for a scan with a single anomalous high-intensity voxel:
    The 1st principal component (PC1) accounts for 97% of the total variance.
    The 2nd principal component (PC2) accounts for 1% of the total variance.
    Note: If the total variance is uniformly distributed across all the PCs, each PC would contribute 2%.

    There might be a single anomalous voxel in target image #2:
         vars.target_images(168808, 2) = -117049.6178
    Scan expression of PC1 is 117083.3946, 1472X the next highest subject-scan expression of PC1 in magnitude.
    Scan correlation with PC1 image is 1, 38X the next highest PC1-and-subject-scan correlation in magnitude.

< Omission: Voxel value correction at vars.target_images(168808, 2) >

return


Ready to execute a new command...  To proceed, press any key.
```

As we could see from the script output, scan #2 does contain a potentially erroneous voxel:

1. The first normalized eigenvalue clearly dominates all succeeding normalized eigenvalues with a magnitude of 97%.
2. The subject expression of PC1 associated with scan #2 is orders of magnitude larger than those of other scans.
3. There exists a perfect correlation between the first eigenimage and scan #2, which corroborates our heuristic identification of scan #2 as our faulty scan from brain score plots. Scan #2 *is* the first eigenimage!

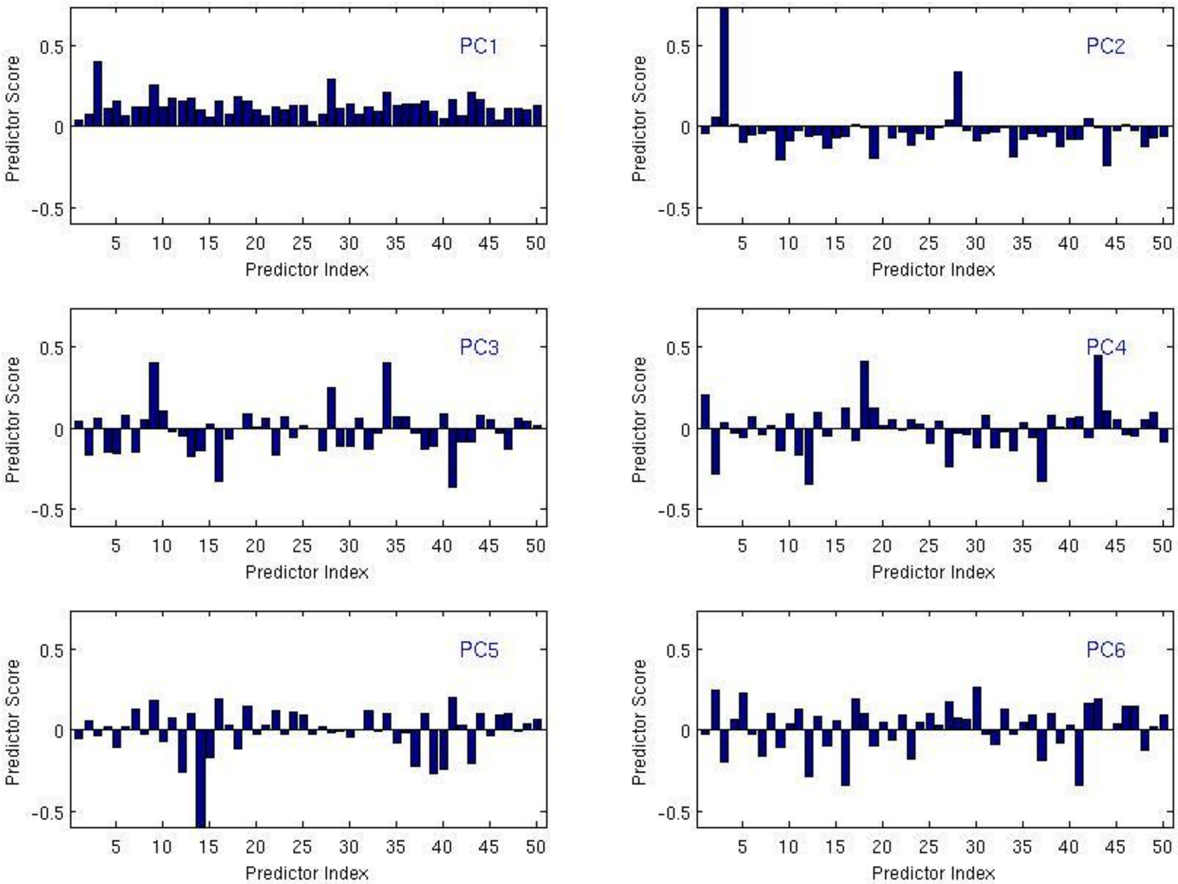
In fact, *vars.target_images(168808, 2)* is indeed the voxel that has been multiplied by 10,000.

Note that if the scans identified in conditions #2 and #3 are different, the script will exit and there will be no report of anomalous voxel or scan.

After correcting for the anomaly, we perform another plain PCA in search of another erroneous voxel (assuming we have no prior knowledge of the total number of outlying voxels).

```
-----------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
```
3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
```
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-----------------------------------------------------------------------------------------------------------
Please make your selection: 3

Perform a principal component analysis without prior application of design matrix (1==yes/0==no)? 1

Remove task-independent effects from target image data prior to pattern extraction (1==yes/0==no)? 0

Remove mean image from target image data during PCA (1==yes/0==no)? 0

Finished computing eigenvalues, eigenimages, and subject scaling factors.  See plots of brain scores from the
first six PCs.
(Note: Each eigenimage has been normalized to unity.  However, the set of eigenvalues are not normalized.)



Ready to execute a new command...  To proceed, press any key.
```

Trial #2 – Brain score plots from a plain PCA



The brain score plots above provide no visual indication of a potentially faulty scan. Let's execute the script in the workspace to ascertain if there are indeed no additional outliers (given our current approach of identification).

```
-----------------------------------------------------------------------------------------------------------------
 OPTIONS:

   Extract/Upload Data
        1) Clear current workspace and upload variables/results from a previous analysis
        2) Clear current workspace and upload new images

   Analysis
        3) Perform a principal component analysis (PCA) on a target image array; design matrix optional
        4) Perform a contrast group fit (for a dyad or a triad) using OrT eigenimages
        5) Fit linearly combined PCA subject expressions against linearly combined behavioral variables

   Statistical Inference
        6) Execute a single-principal-component (PC) bootstrap
        7) Execute an OrT-fit bootstrap
        8) Execute a behavioral-fit bootstrap using behavioral responses and expressions of PCs
        9) Permute subject images across tasks to obtain an eigenvalue sampling distribution for each PC
       10) Permute subject images across tasks to generate null hypothesis for OrT-fit statistic
       11) Permute subject images across tasks to generate null hypothesis for R-square of behavioral fit

   Miscellaneous
       12) List what variables/results are required for running an analysis or a statistical inference
       13) List attributes of critical variables/results in the current workspace
       14) Direct manipulation of workspace

   Storage and Exit
       15) Save all presumably critical variables/results in the workspace to a file
       16) Exit module (WITHOUT saving any variables or results)

-----------------------------------------------------------------------------------------------------------------
Please make your selection: 14

single_vox_anomaly_s

Testing target image array for a scan with a single anomalous high-intensity voxel:
    The 1st principal component (PC1) accounts for 44% of the total variance.
    The 2nd principal component (PC2) accounts for 6% of the total variance.
    Note: If the total variance is uniformly distributed across all the PCs, each PC would contribute 2%.

    There might be a single anomalous voxel in target image #3:
         vars.target_images(113815, 3) = 173.6684
    Scan expression of PC1 is 5482.5917, 1X the next highest subject-scan expression of PC1 in magnitude.
    Scan correlation with PC1 image is 0.81396, 1X the next highest PC1-and-subject-scan correlation in
magnitude.

return


Ready to execute a new command...  To proceed, press any key.
```


Although condition #1 remains valid and the script has identified a candidate scan (#3) that might contain an anomalous voxel, there are breakdowns in conditions #2 and #3. Scan expression of the first PC is no longer orders of magnitude larger than other scan expressions of the same PC. Furthermore, the suspect scan's correlation with PC1 eigenimage is no longer near perfect. At this point, we stop our filtering trials for outlying voxels.

Having identified and corrected for the five outliers, the user can return to the main menu and opt to save a snapshot of the workspace, begin to analyze the data, or exit the program.

- End of Documentation -