MANIA
-
Machine learning Application for NeuroImaging Analyses

Version 2.5 beta 2013
Manual

`https://bitbucket.org/grotegerd/mania`

Dipl.-Inf. Dominik Grotegerd
dominik.grotegerd@wwu.de
Cognitive And Affective Neuroimaging Group
Department of Psychiatry
University-Hospital of Münster
`http://www.uni-muenster.de/NeuroImage`

Contents

1 MANIA Software Package

1.1 About MANIA

MANIA (Machine learning Application for NeuroImaging Analyses) is a tool focused on group-based pattern classification approaches. It uses third party software libraries (such as libSVM) to classify neuroimaging data. The aim of this software is to deliver a broad, extensible repertory of pattern classification algorithms and feature selection methods. Anatomical MRI data are supported as well as functional MRI data.

The latest version can be downloaded at:
`https://bitbucket.org/grotegerd/mania/downloads`

1.2 License

MANIA 2.5 is published under BSD License Version 3 (http://opensource.org/licenses/BSD-3-Clause). Any other usage is not permitted without explicit authorization by the author.

> All rights reserved.
>
> Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
>
> * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
>
> * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
>
> * Neither the name of the University of Münster nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
>
> THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY

DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Third Party Software

- Files: mania/extlib/findobj/*
  - Copyright: (C) 2009 2009, Yair Altman
  - License: BSD-2 / FreeBSD (see mania/extlib/findobj/license.txt)
- Files: mania/extlib/gpml-matlab-v3.1/*
  - Copyright: (C) 2005-2010 Carl Edward Rasmussen & Hannes Nickisch
  - License: BSD-2 / FreeBSD (see mania/extlib/gpml-matlab-v3.1/COPYRIGHT)
- Files: mania/extlib/libsvm-mat-2.9-1/*
  - Copyright: (C) 2000-2009 Chih-Chung Chang and Chih-Jen Lin
  - License: BSD-3 (see mania/extlib/libsvm-mat-2.9-1/COPYRIGHT)
- Files: mania/extlib/liblinear/*
  - Copyright: (C) 2007-2011 The LIBLINEAR Project.
  - License: BSD-3 (see mania/extlib/liblinear/COPYRIGHT)

2 Setup

2.1 Requirements

Requirements:

- Linux (Windows might work, but untested; especially third party software needs maybe recompilation, such as liblinear and libsvm)
- Matlab 2009 or above (older versions might work also, but are not tested; older versions might lack some features, e.g. ReliefF algorithm)
- SPM8 http://www.fil.ion.ucl.ac.uk/spm/software/
  (SPM5 might work also, but was not tested)

Recommended:

- Matlab Bioinformatics toolbox
  (for k-nearest-neighbors, SVM)
- Matlab Statistics toolbox (for Naive Bayes, LDA and more)
- sufficient RAM (at least 4 GB, the more the better) and strong multicore CPU

already included (folder mania/external):

- libsvm at least version 2.9:
  http://www.csie.ntu.edu.tw/∼cjlin/libsvm/
  (you have to use and compile the matlab interface)
- liblinear
- gpml (Gaussian Process Classifiers):
  http://www.gaussianprocess.org/gpml/code/matlab/doc/

Usually MANIA will handle the path setup of included third party software itself. Otherwise you might have to add the path of the respective software-package (/path/-to/`program`) to your matlab search path. The path of Matlab toolboxes should in general be handled by matlab automatically. After starting MANIA you can check the installation by clicking *tools* → *check Installation.* This will show up any severe or important missing libraries.

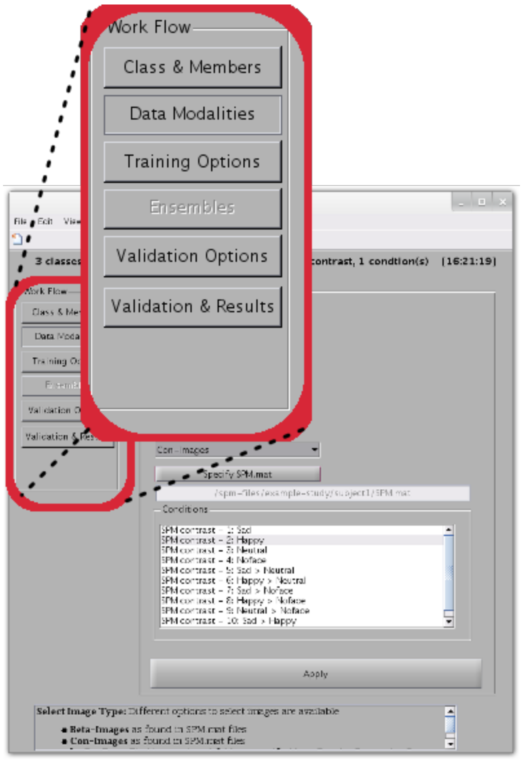
2.2 Installation and Configuration

Although most of the functionality of MANIA will work outside of the environment of SPM, it is recommended for full functionality that files are moved to the toolbox-directory of SPM. If you do not intend to use MANIA by starting SPM, you have to add the root path of MANIA to the matlab search path, which can easily done by using the matlab command window:

```
addpath('/PATH/TO/mania/');
```

Some classification algorithms are not included in MANIA. The Bioinformatics and Statistics toolboxes of Matlab include algorithms for KNN, LDA and Naive Bayes (see 2.1). If your Matlab license includes the required toolboxes, these algorithms should work out of the box. In the case of libsvm and liblinear there might be some configuration (compilation etc.) necessary depending on your operation system. Therefore, please have a look at the documentation of libsvm or liblinear, respectively.

3 Usage



3 classes ... contrast, 1 condtion(s) [16:21:19]

/spm-files/example-study/subject1/SPM.mat

SPM contrast - 1: Sad
SPM contrast - 2: Happy
SPM contrast - 3: Neutral
SPM contrast - 4: Noface
SPM contrast - 5: Sad > Neutral
SPM contrast - 6: Happy > Neutral
SPM contrast - 7: Sad > Noface
SPM contrast - 8: Happy > Noface
SPM contrast - 9: Neutral > Noface
SPM contrast - 10: Sad > Happy

Select Image Type: Different options to select images are available

- **Beta-Images** as found in SPM.mat files
- **Con-Images** as found in SPM.mat files

In general, all operations in MANIA are performed using the GUI (Graphical User Interface). Until now, there is no documentation about command line usage, which might follow in future.
There are different ways to start the program with a graphical user interface:

- Start from SPM Toolbox-Bar → *mania* (the recommended way)
- Type `mania` in the matlab command windows (for most of the functionality also sufficient)

In MANIA, different steps of the workflow are arranged according to different panels. The box *workflow* at the left side of the GUI holds buttons, which bring up these different panels.

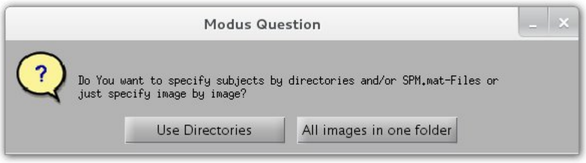
3.1 Getting started

3.2 Setup by providing a convention of folder structure

To start a new classification, click on the *New*-Icon or *File* → *new Classification*. The following pop-up will ask if you want to setup the project by a specific directory structure or if you want to specify each image hand by hand (fig. 3.1).

If you are using SPM and you / your working group keeps a convention of a folder structure, MANIA will be easy for you. If your study is organized according to the following folder structure (or similar one):

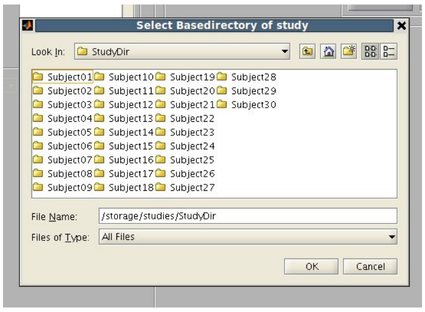
- You have a base-directory, which contains for every subject a separate folder,



Modus Question

Do You want to specify subjects by directories and/or SPM.mat-Files or just specify image by image?

Figure 3.1: If you provide a convention of a directory structure in your study, select "Use directories". Otherwise you have to select every image by itself.

like: 

Select Basedirectory of study

Look In: StudyDir

File Name: /storage/studies/StudyDir

Files of Type: All Files

- every subjects folder contains sub folder(s), which have the same name (e.g. Subject10/SPMcalculations/ Subject10/rawData/ ...)
- for every subject, the SPM.mat is in this folder or such a sub folder (e.g. Subject10/SPMcalculations/SPM.mat)
- conditions of first subject analysis have got the same names and order for every subject (i.e. every names for beta-images are equal in every subject)

If you have such a convention (or a very similar one):

At first, you have to select a directory, which contains sub directories of the subjects.

3.2.1 Choose subjects and define classes

A window will appear, which shows in the middle all directories in the folder you have specified before (fig. 3.2). Now, select subjects and put them in one of two or more classes (left and right box). Instructions:

- You can mark multiple subjects by using *CTRL* and the mouse button.
- Use the arrows < and > to switch subjects. It is recommended, that the number of subjects in both groups is equal.
- Maybe you want to order the subjects for a matched cross validation. Therefore you have to mark the subject and press the buttons *up* and *down*. Again you can use *CTRL* to mark multiple subjects.

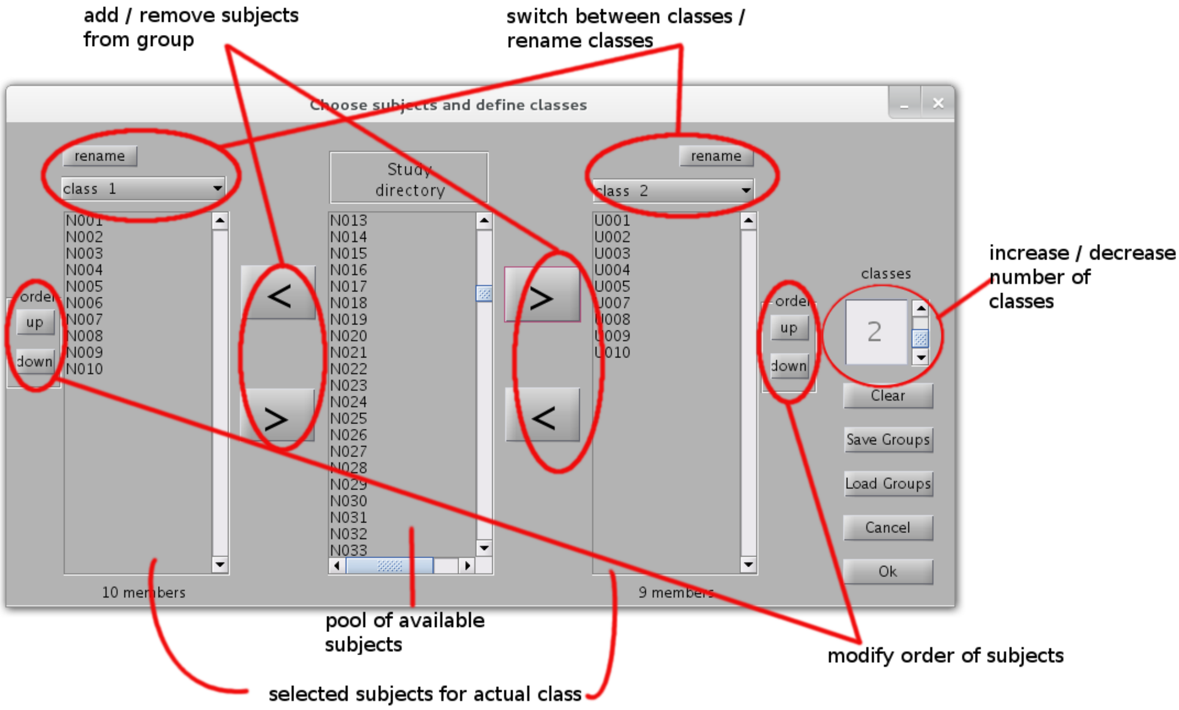


Figure 3.2: Add or remove folders (subjects) to classes.

- By default there are two groups, which can be increased using the slider on the right side. In any case only two groups will be displayed at a time. Switch the displayed classes by using the pull-down menu above the subject list.
- It might be convenient to give classes an appropriate name. To change the selected class, use the *Rename*-button above the pull-down menu.
- If you want to reuse this setup in a future session, you can save (*Save Groups*) it and reload later (*Load Groups)*.
- If you are ready, press *Ok*. If something went wrong and you lost the plot, just press the *Clear*-Button.

3.3 Setup by providing images

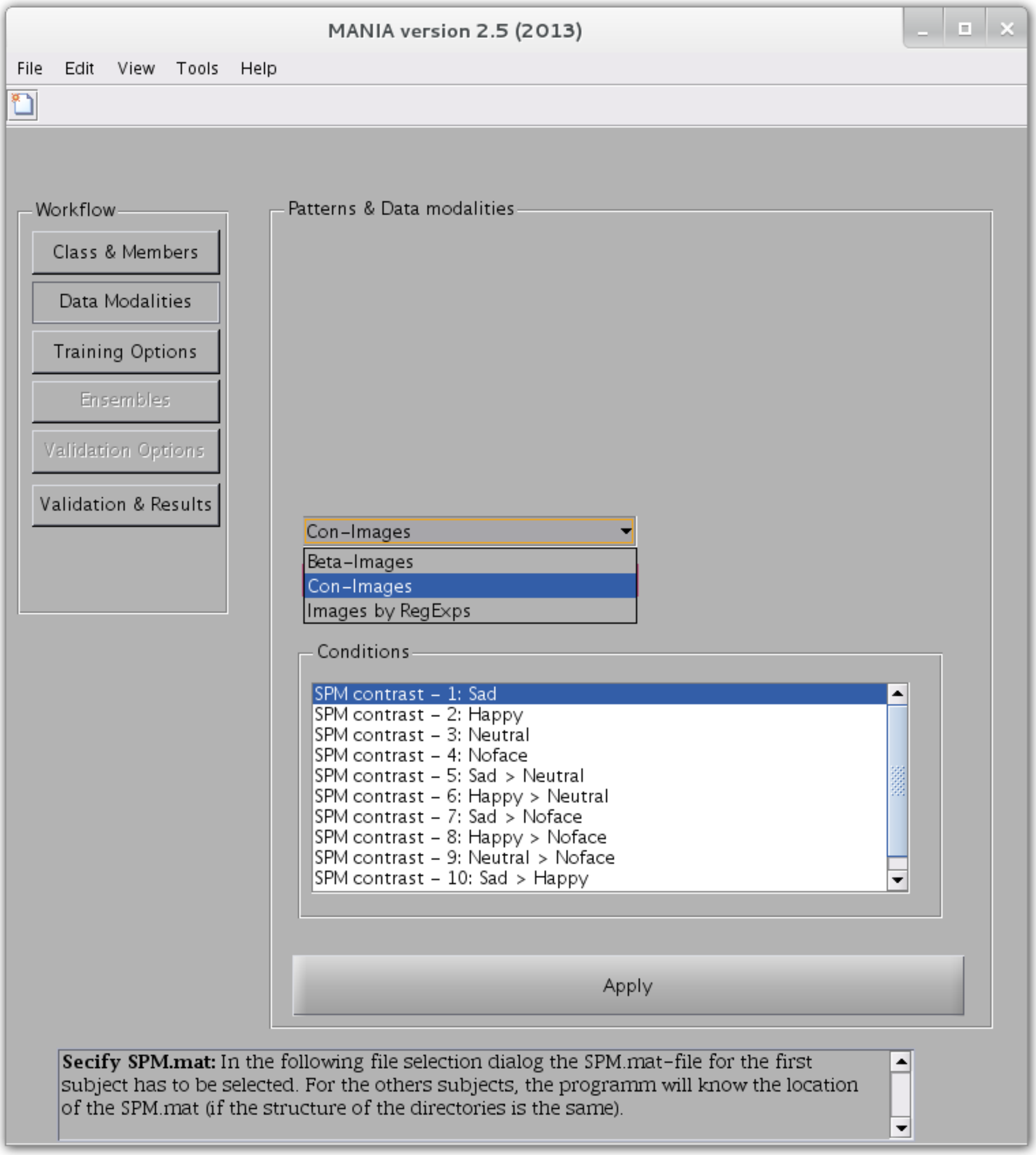
Prerequisites: All images can be found in one single folder. Click on the *New*-Icon or *File → new Classification*, then "All images in one folder". Basically the procedure is very similar to 3.2.1 – instead of folders, image files will be displayed, that again have to be assigned to different classes. Please note, this method has one shortcomming: Only one image per subject can be specified.

3.4 Data Modalities

There are different ways to handle the type of classification (fig 3.3). If subjects were defined according to a directory structure, the following options are available:

- `Beta-Images`: Refers to beta weights maps calculated by SPM using a General Linear Model. You will be asked to specify an SPM.mat of the first subject. Please note, that the name of the SPM.mat and the sub folders have to equal in every subject. Afterward, all conditions will be displayed which were used to calculate beta-images in SPM-analysis. You should then mark one (or more conditions using *CTRL*-Button), which should be used for the calculations.
- `Con-Images`: Refers to contrast images calculated by SPM. After specifying an SPM.mat, all contrasts calculated by SPM will be displayed. You should then mark one (or more conditions).
- `Images by regular expressions`: In this case you do not have to specify an SPM.mat-file, but you will have to name a sub folder (which has the same name for every subject) and to specify an expression to catch all images in this subfolder. E.g. .*.nii will take all nii-images in that sub-folder.

If the "images"-option (but not directories) was selected in the previous step, no further options to specify are available here.
After pressing the *Apply*-Button, images will be loaded.



SPM contrast – 1: Sad
SPM contrast – 2: Happy
SPM contrast – 3: Neutral
SPM contrast – 4: Noface
SPM contrast – 5: Sad > Neutral
SPM contrast – 6: Happy > Neutral
SPM contrast – 7: Sad > Noface
SPM contrast – 8: Happy > Noface
SPM contrast – 9: Neutral > Noface
SPM contrast – 10: Sad > Happy

Secify SPM.mat: In the following file selection dialog the SPM.mat-file for the first subject has to be selected. For the others subjects, the programm will know the location of the SPM.mat (if the structure of the directories is the same).

Figure 3.3: Pull-down menu: Select type of images / conditions. After that, mark the condition(s) which should be used. Press *Apply* in order to load selected images / conditions.

3.5 Training Options

For the cross validation process, three options are available: a simple *leave one out cross-validation*, a *leave-one-out-per-group*, and *Test Groups*. Leave-one-out-per group is recommended, when you have balanced group sizes (which is always recommended) and the sample sizes are small. Please note, that you cannot do a Leave-one-out-per-group, if the group sizes differ. A third options is called *Test Groups*, which will be explained in 3.7.

The second pull-down menu shows you a list of different classification algorithms:

Naive Bayes (Statistics toolbox). For a detailed description, please see matlab documentation - enter: `doc NaiveBayes`; Parameters:

- distribution function: default is Gaussian distribution; also available: Kernel density estimation

Ensemble Fitter (Statistics toolbox). Creates an fitted ensemble of decision trees. For a detailed description (i.e. which ensemble method can/should be used), please see matlab documentation - enter: `doc fitensemble`; Parameters:

- ensemble method, e.g. AdaBoost, LogitBoost; for non-binary classifications e.g. Bag
- learning cycles (NLEARN): depending on the problem a few hundreds up to a few thousands will be sufficient.

GPML Gaussian Process Classifier (http://www.gaussianprocess.org/gpml/code/matlab/doc/). For a detailed description you will find the manual in `extlib/gpml-matlab-v3.1/doc/manual.pdf`; Parameters:

- Mean Functions: default is meanZero; also available: MeanOne
- Covariance function: currently only covLin (linear covariance) available
- Likelihood functions: default is likErf (Error function or cumulative Gaussian likelihood function) - also available: likLogistic

k-nearest neighbors (Bioinformatics toolbox). For a detailed description, please see matlab documentation - enter: `doc knnclassify`; Parameters:

- k: how many nearest neighbors
- distance measure: default is euclidean distance; also available: cityblock, cosine, correlation and Hamming distance.

Linear Discriminant Analysis (Statistics toolbox). For a detailed description, please see matlab documentation - enter: `doc classify`; Parameters:

- discrimant function: default is diagLinear - also available: diagQuadratic.

libSVM Support Vector Machine (http://www.csie.ntu.edu.tw/~cjlin/libsvm/). A manual can be found at the respective website. Parameters are just command line parameters of libsvm as string, e.g. "-t 0 -c 1". Description of parameters will be shown in the GUI. For detailed description, please see libSVM documentation.

liblinear (http://www.csie.ntu.edu.tw/∼cjlin/liblinear/). A manual can be found at the respective website. Parameters are similar to libSVM, also parsed as a string.

Support Vector Machine (Bioinformatics toolbox). Please see matlab documentation - enter: `doc svmclassify`; Parameters:

- kernel function, default linear; also available: quadratic, polynomial, rbf, mlp
- BoxConstraint Value, C for the soft margin, default 1.
- Method (default: SMO = Sequential Minimal Optimization (SMO); QP = Quadratic programming; LS = Least Squares)
- Tolerance checking for Karush-Kuhn-Tucker (KKT) criterion (default 1e-3)
- KKT Violation Level - fraction of alphas that are allowed to violate the KKT conditions; default 0
- Kernel Cache Limit - default 5000

Most of the classifiers have a list of options and parameters. To change the parameters, click *edit settings*. Please note that changed parameters are lost after switching to another classification algorithm. To show the selected parameters go to *view settings*. Classification algorithms are quite different; the options vary from algorithm to algorithm. The most parameters can be accessed and changed using the GUI. We refer for a fully understanding of the algorithms and the respective parameters to the algorithm specific documentation and do not cover it here.
Please note that the SVM algorithm of the statistics toolbox and the Gaussian Process Classifier of GPML do not support multiclass classifications itself. MANIA uses the *One-vs-One*-method for the SVM (Bioinformatics toolbox) for multiclass classifications. Since it relies only on the predicted labels, in some cases prediction is not possible. GPML multiclass classifications are realized using the *One-vs-All*-method, using decision values.

3.5.1 Advanced Options

There are a couple of feature selection / preparation operations available. A click on the *advanced* button inside the feature preparation / selection pane brings up a new window (fig. 3.4). The left column shows up all methods which are selected, the right column shows all remaining available methods. Switch them by marking the desired methods and then use the *add* and *remove* buttons.
Most of the methods need a setup. Mark one method on the left column and click *Edit Settings* to setup and change parameters for the specific method. Current parameters can be reviewed by clicking on *View Settings*. Please note, that operations will be executed according to their listed order. After finishing this setup and clicking *Apply* the methods will also be listed in the feature preparation / selection pane.

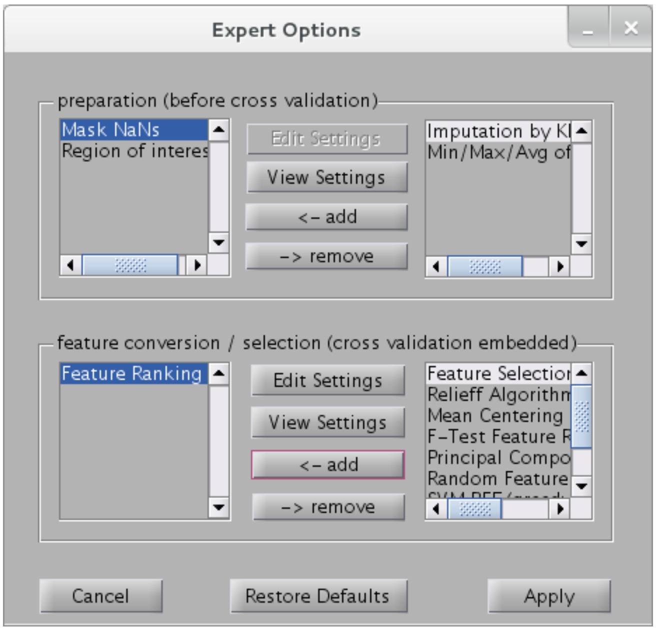


Figure 3.4: Advanced options for feature selection approaches. Right: available operations; Left: selected operations. Top: operations applied on all subjects, before cross-validation; bottom: operations embedded in cross-validation.

After removing an operation from the list, the changed parameters will be lost and have to be edited next time again.

Preparation - Every operation that is done before cross-validation for all subjects - therefore independent from class labels:

mask NaNs Will remove every voxel from all subjects, which has at least one appearance of an unknown value (NaN - not a number). This is for most cases essential (otherwise you will get an error, if there are still NaNs and the algorithm does not support unknown values). No Parameters.

Imputation by KNN (Bioinformatics toolbox) Tries to estimate unknown values (NaN) by looking at the k-nearest-neighbors of that subject (alternative to masking out NaNs). See matlab documentation, enter: `doc knnimpute`. Parameters:

- K:number of nearest neighbors
- distance measure: default is euclidean distance; also available: cityblock, cosine, correlation and Hamming distance.

Region of interest Uses only voxels, which are in a specified Region of interest. Please specify file in nifti-format. **Important:** The ROI should be the same size and dimensions like the specified patterns. Otherwise MANIA tries to resize the mask itself. If this does not work, try it your self (e.g. use Realignment Operations from SPM) or *tools → wfu-pickatlas → resize mask*.

- Mask: Brain mask in nifti format. Should be resliced with SPM to match resolution of sample images.

Min/Max/Avg Computes average, minimal and/or maximal values of specified ROIs for every subject. These values are the new features.

- ROIs: Specify multiple Nifti files of brain masks
- "Compute average of n max/min. values": Specify a number n, if you want to compute an average of the n maximum values (minimum respectively), instead of just using the maximum / minimum.
- Min: Yes/No - if minimum should be used
- Mean: Yes/No - if average should be computed and used
- Max: Yes/No - if maximum should be used

feature conversion / selection - Every operation that is embedded in cross-validation process - and therefore in general only computed on the training sample and in a reasonable way repeated on the test data:

Sequential Features Selection (Statistics toolbox) A recursive process in which voxels of little relevance are removed step by step. See matlab documentation, enter: `doc sequentialfs`; Parameters:

- Feature Selection Direction: Foward = start with one good feature and add in every iteration features; Backward = start with all features and remove irrelevant features in every iteration.
- Criteria function: Can be used to specifiy other criteria functions. default: @critfun
- Tolerance function: absolute or relative
- Maximum Deviance - default $chi2inv(.95, 1)$

Relieff Algorithm (Statistics toolbox) The ReliefF algorithm computes importance of features. See matlab documentation, enter: `doc relieff`. Parameters:

- Number of features, that should finally be used
- K: Number of nearest neighbors

Feature Ranking (Bioinformatics toolbox) ranks features according to a specified criterion (for instance t-test) and keeps only a user-defined amount of most relevant voxels. Works only for binary classifications. See matlab documentation, enter: `doc rankfeatures`. Parameters:

- Number of features, that should finally be used
- selection criterion: default is ttest (two-sample t-test). Also available: entropy (Kullback-Liebler distance), bhattacharyya, roc (Area under curve), wilcoxon.

F-Test Feature Ranking Computes F-Tests for every feature and takes every feature that achieved a p-value lower than the specified threshold. Works also for non-binary classifications. Parameters:

- p-value: all features, that achieve during the F-test a p-value smaller than this one specified will be selected.

Principal components (PCA) Computes the first n PC's on the training data (where n is the sample size). This algorithm uses Matlab's *princomp* with option *'econ'*. No Parameters.

Random Feature Selection performs randomized subset feature search reinforced by classification. See matlab documentation, enter: `doc randfeatures`

- Number of features, that should finally be used
- Subset size: Number of features in every subset
- Pool size: Number of subsets for final pool
- classifier: KNN (default) oder LDA

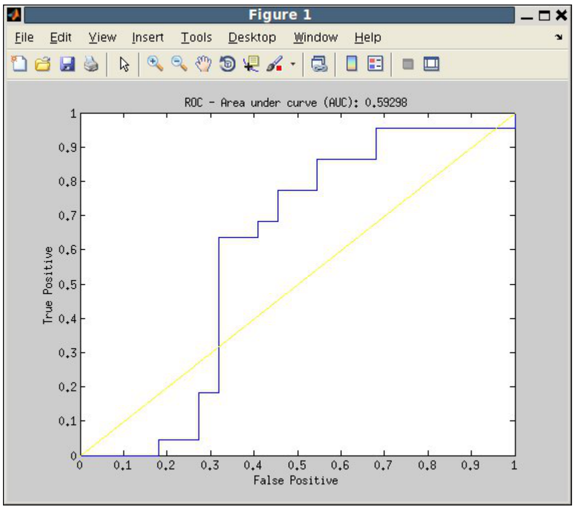


Figure 3.5: Area under curve plotting.

SVM RFE (greedy) eliminates features recursively according to SVM weights computed by embedded cross-validation. For performance reasons, this method removes multiple features in every step. In the beginning many features are removed. With every iteration, less features are removed, until specified subset size will be achieved. Parameters:

- Number of features, that should finally be used

SVM weights computes SVM weights on training data and then removes low relevance features (one step).

- Number of features, that should finally be used

Z-Scoring computes z-values (standard score) for every subjects voxel. No parameters.

3.6 Validation & Results

After setting all parameters, you are ready to start the training with *Start Validation*. A small window will give you information about progress, current detection rates and remaining time. Finally, a results pane shows the results including further information. Here, computed (mean) accuracy, sensitivity, specificity as well as a confusion matrix (columns: true output, rows: predicted output; see `http://en.wikipedia.org/wiki/Confusion_matrix`) are located.

3.6.1 ROC Plot

The area under ROC curve (AUC) can be plotted after successful training and testing by clicking *tools* → *plot-ROC* (fig. 3.5. In the resulting window you will also find the size of the area under curve (on top).
ROC plot is only available for binary classifications.

Permutation Test

A permutation test is available for significance testing. But be careful, this might take a very long time! And you should at least run 1000 permutations or more. After a classifier has been trained and tested you can click *tools* → *Permutation test.* A p-value appears after successful computation in the results pane.

3.6.2 Discriminative Maps

Weight Maps can be extracted, too. Click on *tools* → *Weight Map* → *Extract W-Map.* In the appearing window you have to pick a specific support-vector-machine (from a specific leave out). After picking a cross validation step you will be asked for a location and name to save. The generated image can be inspected with any tool, which is able to read img- and hdr-files, like *mricron.* Please note, that only some of the algorithms support extraction of a discriminative map (in the current version only linear SVMs).

3.7 Hold Out / Test Groups

If *Test Groups* was chosen in training options previously, classes have to be grouped according to training and test samples. Therefore, go on with *Validation Options.* The classes will be listed here. For every class, you have to specify, if it should be used for training or testing the classifier. There should be at least two classes for training and at least one class for testing.
If you just want to test, to which group some specific subjects will be predicted, no further setup is needed. Please note, that in this case an accurate prediction of an accuracy value is not possible!
If accuracy values should be predicted, the test samples need to be relabeled according to the relationship to the training classes (i.e. the classifier needs some information about which test group is related to which training group).

3.8 Ensembles

You might also want to build ensembles of classifiers. By now, MANIA only supports Majority Vote of classifiers. Therefore, please take care that you have an odd number of classifiers for an ensemble. If a standoff situation occurs, the algorithm tries relay on decision values of the classification algorithm, if possible.

3.8.1 Train a classifier and add it to an ensemble

At first, just do a normal training and cross validation with a desired candidate for a classifier ensemble. When cross validation is finished, click *Tools → Classifier Ensemble → add to Ensemble.* The *Ensembles* tabbed pane will get available, showing a list with the recently added classifier. Repeat the procedure of computing and adding classifiers to the ensemble until you are ready to cross-validate the whole ensemble. Therefore just press *Majority Vote Start.* The result of the ensemble will be printed as usual in *Validation & Results* pane. Furthermore, for every single classifier the accuracy will be printed in the table, too.

Please note that you can setup every classifier with different algorithms (libSVM, GPC...), different conditions (even different images), and different ROIs. But you have to take care, that the subjects and the order of the subjects are always the same! If you want to get rid of the ensemble and its panel just click *reset.* Then, the panel and classifiers will be deleted.

3.8.2 Generate ensembles from SPM.mat conditions

If you are using the conditions specified in the SPM.mat, you can generate automatically classifier ensembles from these conditions. At first, setup all desired options for a full classification, like algorithm, ROIs and just do a normal training and cross validation with an example candidate for a classifier ensemble. Afterwards, to generate the ensemble, mark the desired conditions (should be an odd number). If you are ready, click *Classifier Ensemble → generate from marked conditions.* MANIA will setup the ensemble automatically. Finally click *Start Majority Vote.* Clear the panel by clicking *reset.*

3.8.3 Random subspace ensemble

It is also possible to generate random subspace ensembles. This is basically an ensemble of equal classifiers, but each classifier only uses a random subspace of the available features (a user-defined amount). Therefore you have to setup everything (ROI, algorithms), mark one condition and click *Classifier Ensemble → Random Subspace Ensemble.* You will be asked to:

- specify the amount of features every classifier should get randomly
- specify the amount of classifiers that should be generated (again, please pick an odd number)
- specify any seed number for computation of pseudo random numbers. By using pseudo random numbers you are able to reproduce your results.

MANIA will setup the ensemble automatically. Proceed by clicking *Start Majority Vote*, again. Clear the panel by clicking *reset.*