

JANIS

Just Another Neuro-Imaging Software

Version 2.3 beta 2013
Documentation

Dipl-Inf. Dominik Grotegerd
dominik.grotegerd@wwu.de
Cognitive And Affective Neuroimaging Group
University-Hospital of Münster

Contents

1	JANIS Software Package	3
1.1	About JANIS	3
1.2	License	3
2	Setup	5
2.1	Requirements	5
2.2	Installation and Configuration	6
3	Usage	7
3.1	Getting started	7
3.2	Setup by providing a convention of folder structure	7
3.3	Choose subjects and define classes	8
3.4	Data Modalities	8
3.5	Training Options	10
3.5.1	Advanced Options	13
3.6	Validation & Results	15
3.6.1	ROC Plot	16
3.6.2	Discrimanative Maps	16
3.7	Hold Out / Test Groups	16
3.8	Ensembles	16
3.8.1	Train a classifier and add it to an ensemble	17
3.8.2	Generate ensembles from SPM.mat conditions	17
3.8.3	Random subspace ensemble	17

1 JANIS Software Package

1.1 About JANIS

JANIS 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.

1.2 License

JANIS 2.3 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 Mnster 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: `janis/extlib/findobj/*`
 - Copyright: (C) 2009 2009, Yair Altman
 - License: BSD-2 / FreeBSD (see `janis/extlib/findobj/license.txt`)
- Files: `janis/extlib/gpml-matlab-v3.1/*`
 - Copyright: (C) 2005-2010 Carl Edward Rasmussen & Hannes Nickisch
 - License: BSD-2 / FreeBSD (see `janis/extlib/gpml-matlab-v3.1/COPYRIGHT`)
- Files: `janis/extlib/libsvm-mat-2.9-1/*`
 - Copyright: (C) 2000-2009 Chih-Chung Chang and Chih-Jen Lin
 - License: BSD-3 (see `janis/extlib/libsvm-mat-2.9-1/COPYRIGHT`)
- Files: `janis/extlib/liblinear/*`
 - Copyright: (C) 2007-2011 The LIBLINEAR Project.
 - License: BSD-3 (see `janis/extlib/liblinear/COPYRIGHT`)

2 Setup

2.1 Requirements

Requirements:

- Linux (Windows might work, but untested; especially third party software needs probably to be recompiled, such as liblinear)
- MATLAB 2009 or above (older versions might work also, but are not tested)
- 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-neighbours)
- MATLAB STATISTICS TOOLBOX (for Naive Bayes and LDA)
- enough RAM (4 GB or above recommended) and a strong multicore CPU is recommended

already included (folder 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 JANIS will handle the path setup of included third party softwares itself. Otherwise you might have to add the path of the respective software-package (`/path/-to/programm`) to your matlab search path. The path of Matlab toolboxes should in general be handled by matlab automatically. After starting JANIS 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 JANIS will work outside of the environment of SPM, it is for full functionality recommended, that the files are moved to the toolbox-directory of SPM. If you do not intend to use JANIS by starting SPM, you have to add the root path of JANIS to the matlab search path, which can easily be done by using the matlab command window:

```
1 addpath('/PATH/TO/janis/');
```

Some classification-algorithms are not included in JANIS. The Bioinformatics and Statistics Toolboxes of matlab come with algorithms for KNN, LDA and Naive Bayes (see 2.1. If your matlab license includes these toolboxes, these algorithms should work out of the box. In the case of libsvm and liblinear there is some configuration (compilation etc.) necessary. Therefore please have a look at the documentation of libsvm, liblinear respectively.

3 Usage

You can use the GUI (recommended). Nevertheless it is possible to use the matlab command line, but until now no full documentation is available.

There are different ways to start the program with a graphical user interface:

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

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*. This brings up a dialog, which asks 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, JANIS will be easy for you. If your study is organized according to the following folder structure (or similar one):

- You have a basedirectory, which contains for every subject a separate folder, like:
 - STUDYFOLDER
 - * SUBJECT1
 - * SUBJECT2
 - * SUBJECT3
 - * ...
- every subjects folder contains sub folder(s), which have the same name
- every SPM.mat are in this folder or such a sub folder
- 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 (fig. 3.2).

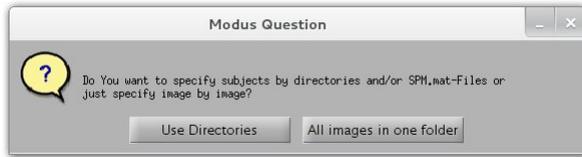


Figure 3.1: If you provide a conventional structure of folders in your study, select “Use directories”. Otherwise you have to select every image by itself.

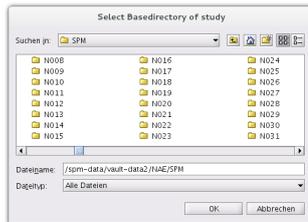


Figure 3.2: At first the main directory of the study should be specified.

3.3 Choose subjects and define classes

A screen will appear, which shows in the middle all directories in the folder you have specified before (fig. 3.3). Now you have to pick every subject and put it in one of two or more classes (left and right box). You can mark multiple subjects by using *CTRL* and the mouse button. Use the arrows *<* and *>* to switch subjects. Maybe you want to order the subjects for a matched cross validation. Therefore you have to mark the subject and push the buttons *up* and *down*. Again you can use *CTRL* to mark multiple subjects.

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 classes, that will be displayed by using the pulldown menu above the subject list.

If you want to reuse this setup in a future session, you might want to save it and reload later. If you are ready, push *Ok*. If something went wrong and you lost the plot, just push the *CLEAR*-Button.

It is recommended, that the number of subjects in both groups is equal.

3.4 Data Modalities

If subjects were defined according to a directory structure, you will have the following options:

There are different ways to handle the type of classification (fig 3.4):

- **Beta-Images:** You will be asked to specify a SPM.mat of the first subject. Please

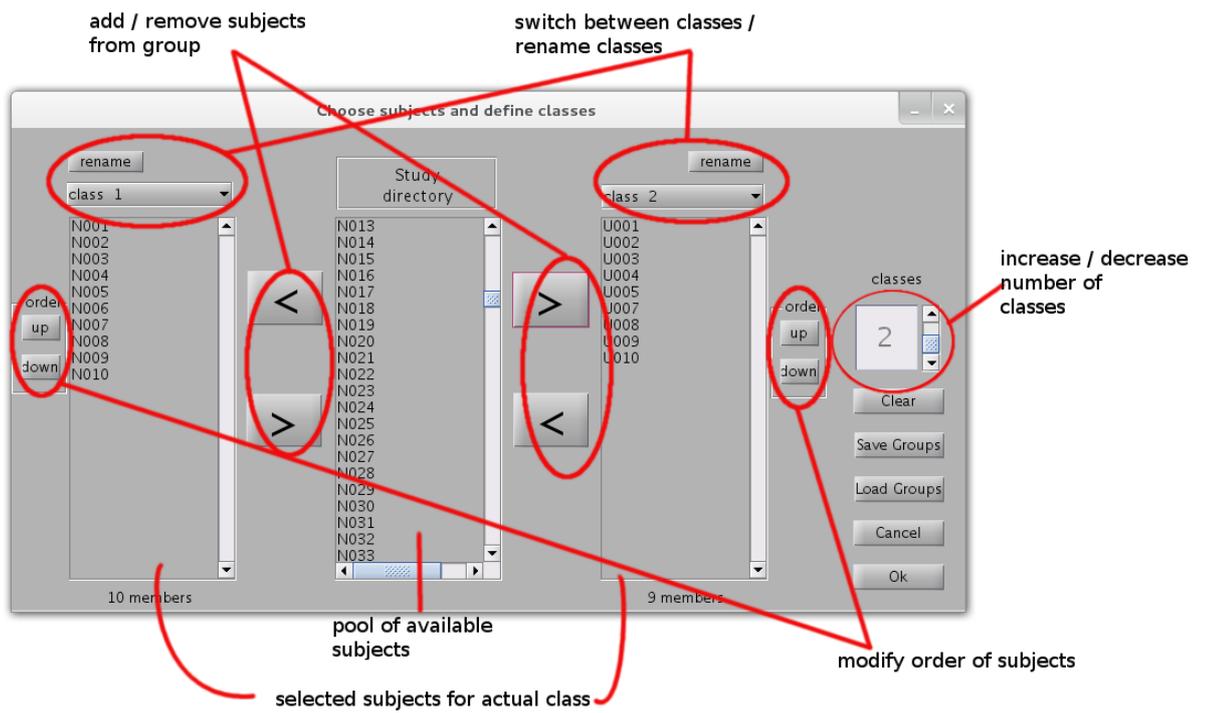


Figure 3.3: Add or remove folders (subjects) to classes.

note, that the name of the SPM.mat and the subfolders have to equal in every subject. Afterwards, all conditions will be displayed which were used to calculate beta-images in SPM-analyses. You should then mark one (or more conditions use CTRL-Button), that should be used for the calculations.

- **Con-Images:** After entering a 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 a SPM.mat-file, but you will have to name a subfolder (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 subfolder.

If images (but not directories) were specified, there are no further options to specify. After pushing the *Apply*-Button, images will be loaded.

Select conditions

3.5 Training Options

For the cross validation process one can choose between a simple **leave one out crossvalidation** and a **leave-one-out-per-group**. 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`

Ensemble Fitter (Statistics toolbox). Creates an fitted ensemble of decision trees. For a detailed description, please see matlab documentation - enter: `doc fitensemble`

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`

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

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

libSVM Support Vector Machine (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>). A manual can be found at the respective website.

liblinear (<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>). A manual can be found at the respective website.

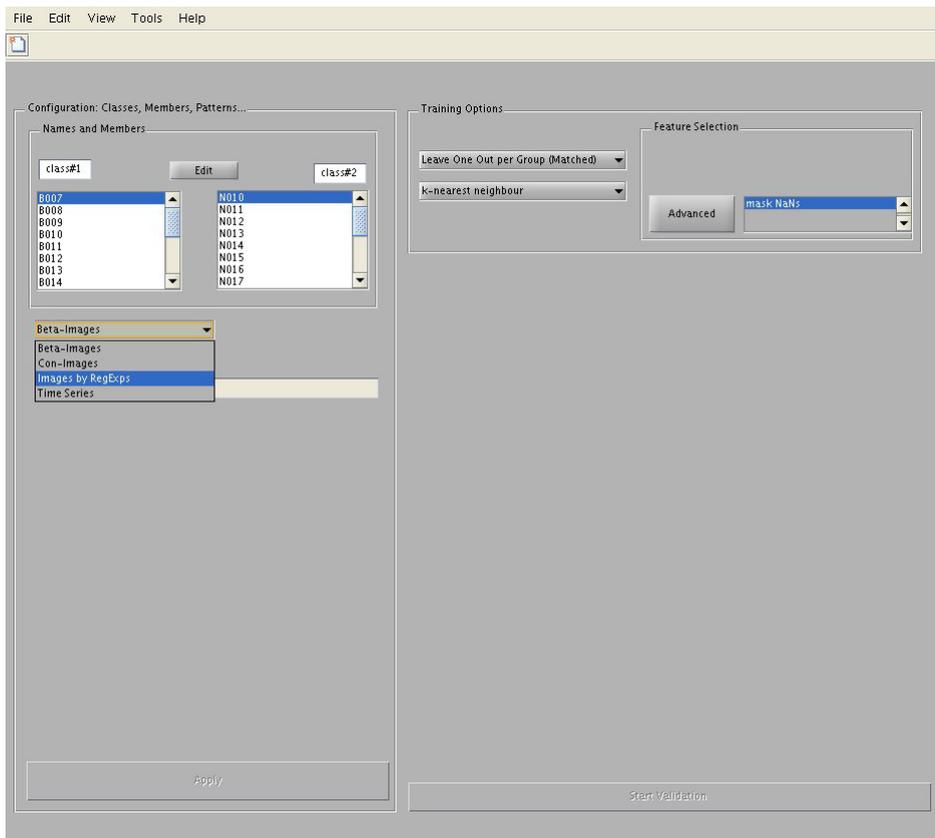


Figure 3.4: Pulldown-Menu: Select type of images / conditions.

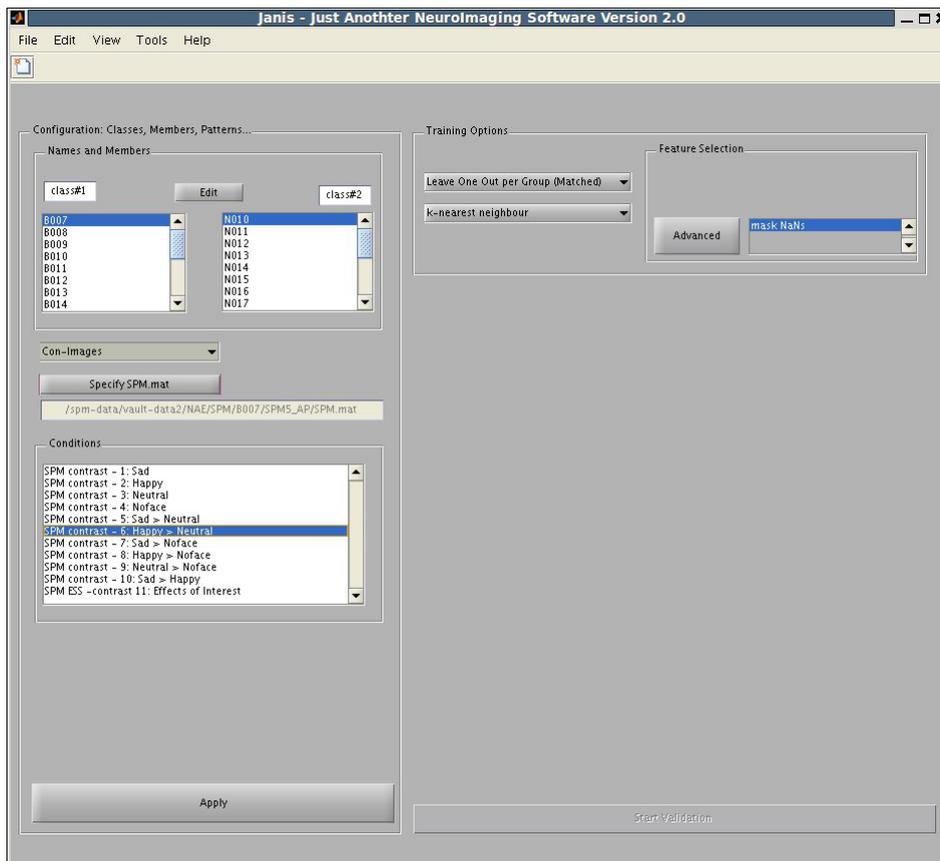


Figure 3.5: List of conditions: Mark the condition(s) which should be used.



Figure 3.6: 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.

Support Vector Machine (Bio-Info-Toolbox) (Bioinformatics toolbox). please see matlab documentation - enter: `doc svmclassify`

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*. These classification algorithm 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. JANIS 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. 3.6). The left column shows up all methods which are selected, the right column shows all remaining methods, which are available. 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. 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 crossvalidation 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).

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

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 JANIS tries to resize the mask itself. If this does not work, try it yourself (e.g. use Realignment Operations from SPM) or *tools* → *wfu-pickatlas* → *resize mask*.

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

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

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

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

Mean Centering Centers all values voxelwise

F-Test Feature Ranking Computes F-Tests for every feature and takes every feature achieving p-value lower than specified.

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`

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'*)

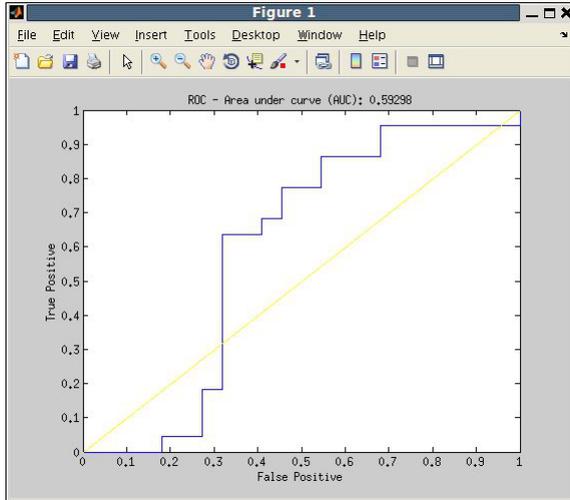


Figure 3.7: Area under curve plotting.

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

SVM RFE (greedy) eliminates features recursive according to SVM weights computed by embedded crossvalidation. 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.

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

Z-Scoring will compute z-values (standard score) for every subjects voxel.

3.6 Validation & Results

If you are ready, you can start 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. One can find 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).

3.6.1 ROC Plot

The area under curve can be plotted after successful training and testing by clicking *tools* → *plot-ROC* (fig. 3.7). 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 make 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* → *Label with aal* → *Discriminative 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 *mricon*. Please note, that only some of the algorithms support extraction of a discriminative map (like libSVM or Gaussian Process Classifiers). You can also rescale and threshold the image with *tools* → *Label with aal* → *Threshold / Scale image*. This might be useful for further analysis like anatomical labeling.

3.7 Hold Out / Test Groups

If in the Training Options the *Test Groups* option was chosen, classes have to be grouped according to training and test samples. Therefore please click on *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 one class for testing.

If you just want to test, which group some specific subjects will be predicted, now further setup is needed. Please note, that in this case the 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.

3.8 Ensembles

You might also want to build ensembles of classifiers. By now, JANIS only supports Majority Vote of classifiers. Therefore please take care, that you have an odd number classifiers for an ensemble. In general there are three different ways to build classifiers.

3.8.1 Train a classifier and add it to an ensemble

You just have to do a normal training and cross validation. After results are finished, click *Classifier Ensemble* → *add to Ensemble*. In the right middle area of the program a new pane will appear, 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 the bottom right 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), different ROIs. You only 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*. The panel and classifiers will then 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. Therefore mark the desired conditions (should be an odd number) and setup everything, like algorithm, ROIs. If you are ready, click *Classifier Ensemble* → *generate from marked conditions*. JANIS 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 does only use 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 and the amount of classifiers, that should be generated (again, please pick an odd number). Finally specify any seed number for computation of pseudo random numbers. By using pseudo random numbers you are able to reproduce your results. JANIS will setup the ensemble automatically. After that again click *Start Majority Vote*. Clear the panel by clicking *reset*.