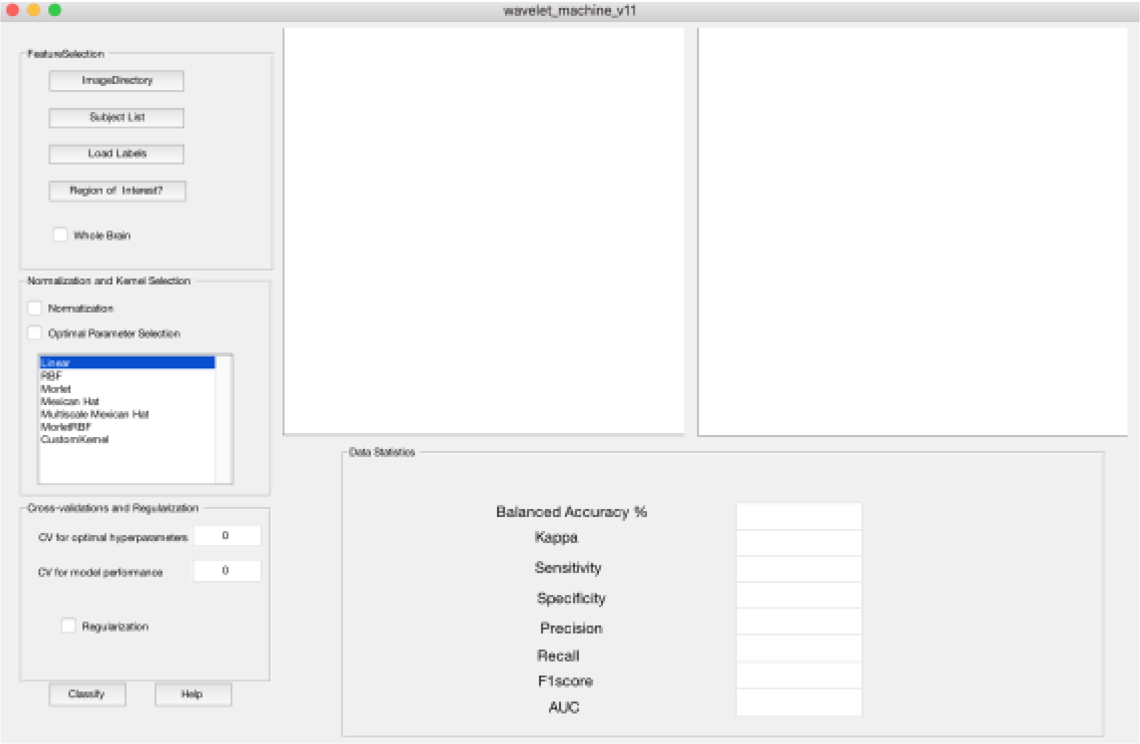
Wavelet Machine GUI Help:

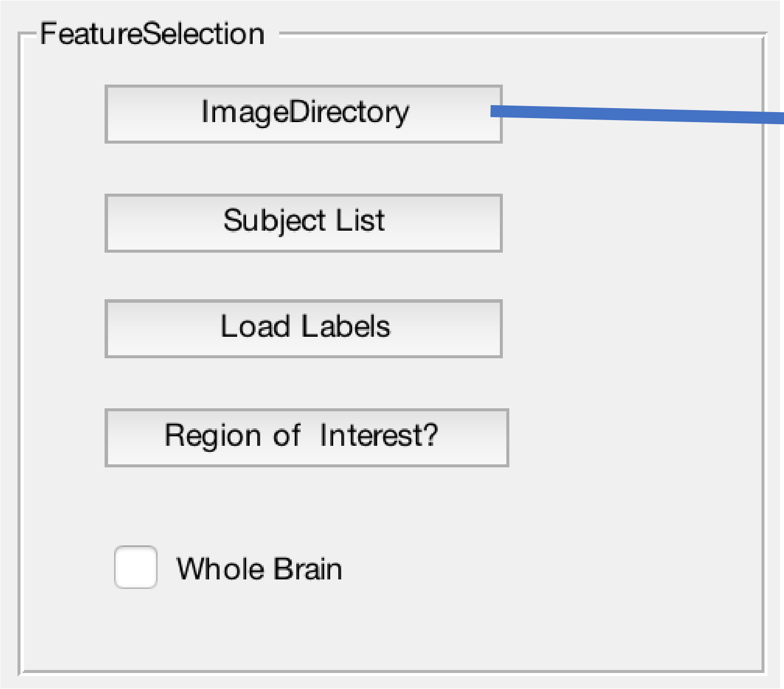


CV for optimal hyperparameters 0

CV for model performance 0

Feature Selection:

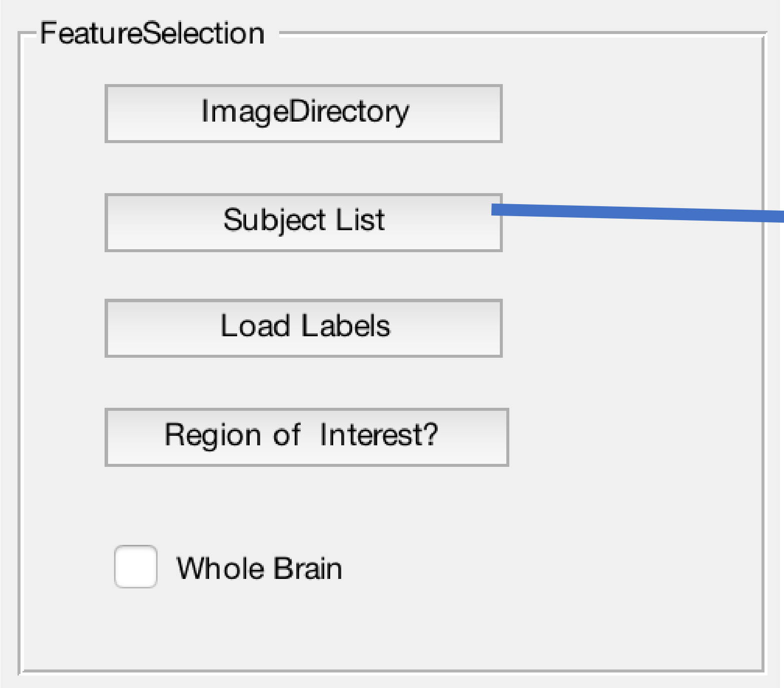
Step 1:



Browse and Select the Directory where Contrast images (the Feature Source) are stored. The contrast images need to be in the following format: con_0002_subjid.img e.g. con_0002_5001.img.

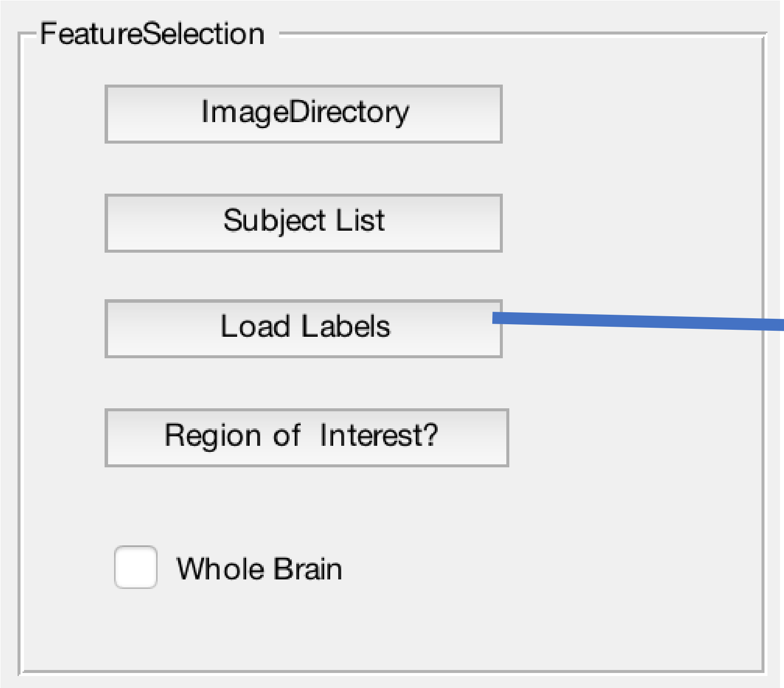
Note the contrast images can also be in a Nifti format: con_0002_subjid.nii

Step 2:



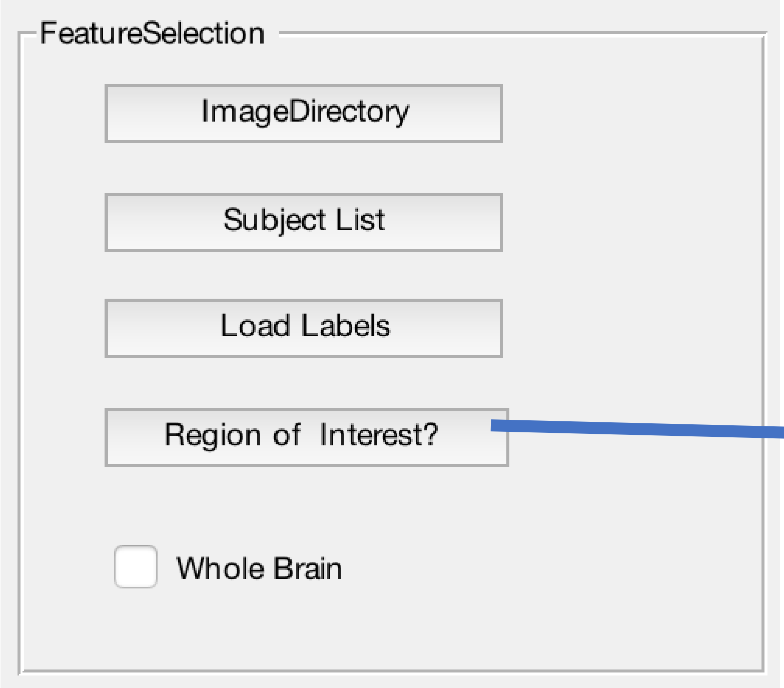
Subject List text files need to be in the same directory where the contrast images are stored. Pick subject list text file.

Step 3:



Pick the labels text file from the same directory. Label groups as “-1” and “1”.

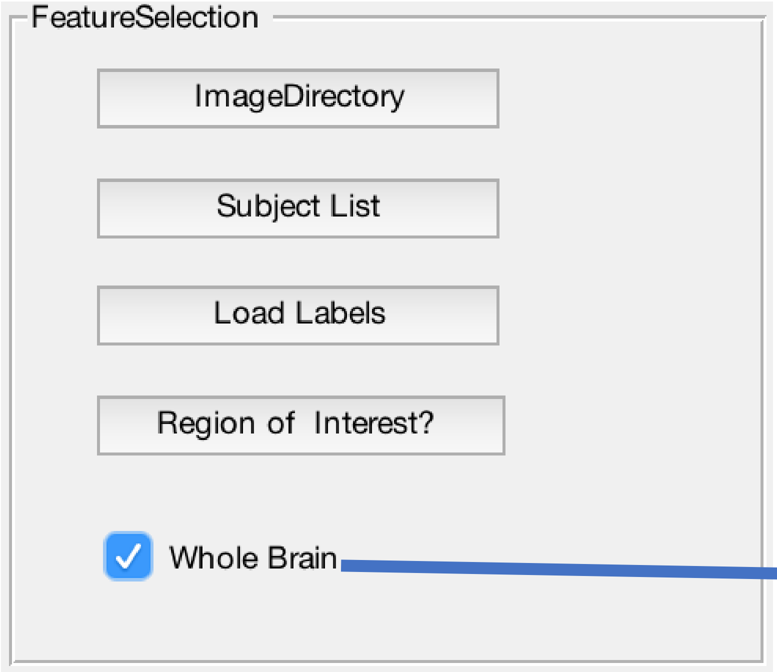
Step 4a:



☐ Whole Brain

Pick region of interest as part of Feature Selection. This can be a Nifti file or Analyze Format file.

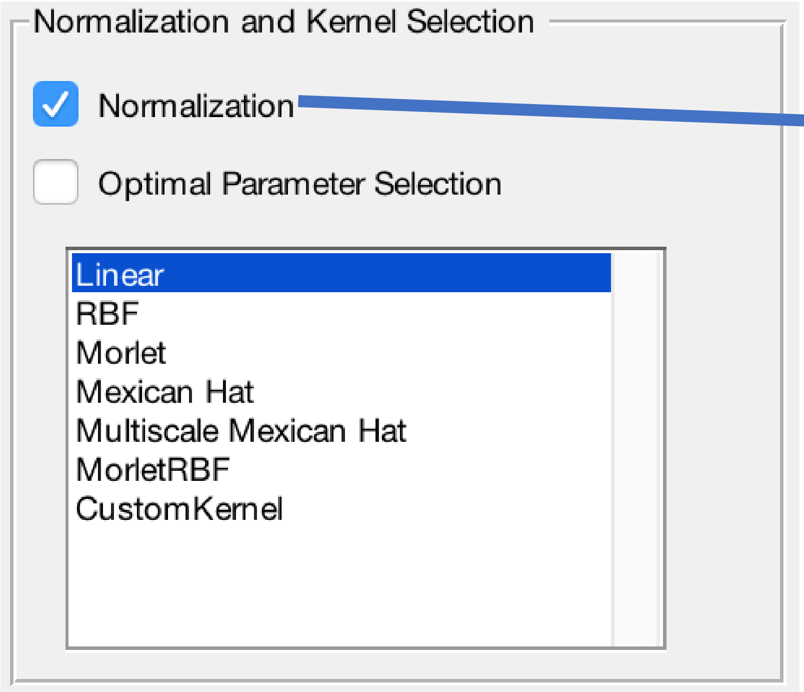
Step 4b:



☑ Whole Brain

If you don’t pick a Region of Interest, you can select the Whole Brain option. The speed of the wavelet classifier varies with number of voxels. So, bigger dimensions will slow the performance of the wavelet kernels.

Normalization and Kernel Selection:
Step 5:



Normalization and Kernel Selection

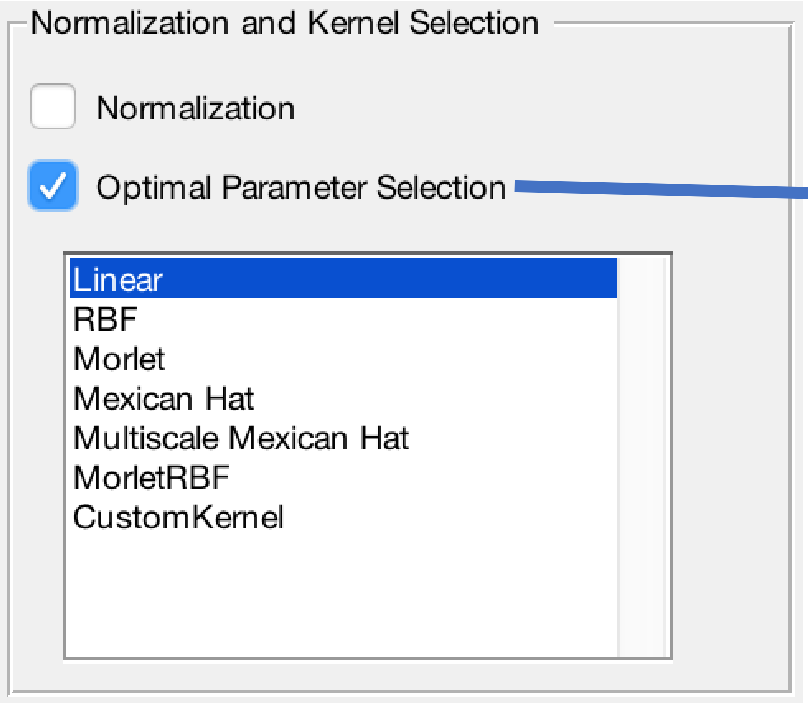
☑ Normalization

□ Optimal Parameter Selection

Linear
RBF
Morlet
Mexican Hat
Multiscale Mexican Hat
MorletRBF
CustomKernel

Normalization is an optional but recommended step. It basically involves subtracting the feature(s) in a column with the mean of the column and dividing that with the standard deviation.

Step 6:



Normalization and Kernel Selection

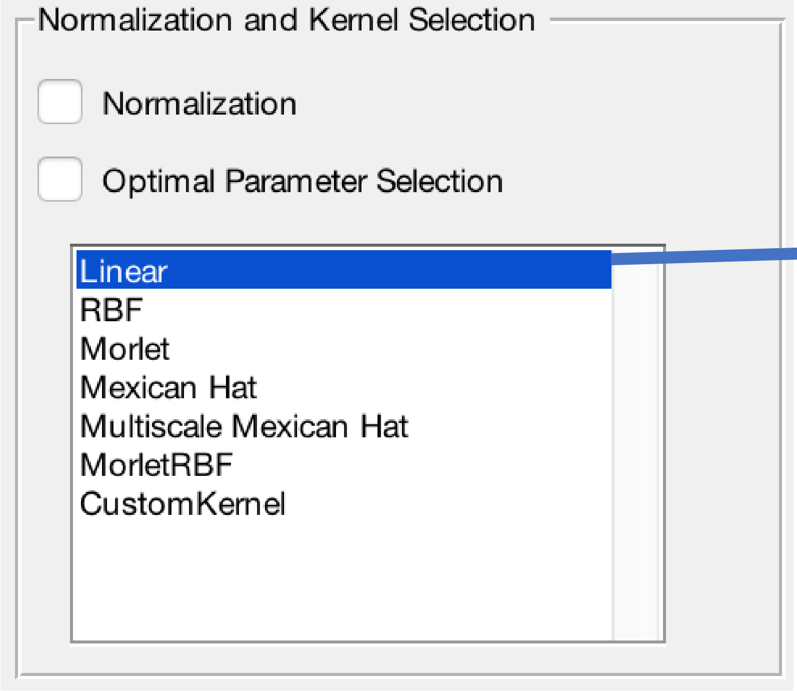
□ Normalization

☑ Optimal Parameter Selection

Linear
RBF
Morlet
Mexican Hat
Multiscale Mexican Hat
MorletRBF
CustomKernel

Optimal Parameter Selection is an optional but recommended step. It basically involves running cross-validation grid search approach to obtain optimal hyperparameters. Note that the grid search range is preselected. To change it, you will have to modify the function “Kernel”HyperPar.m file depending on the “Kernel” picked.

Step 7:



Normalization and Kernel Selection

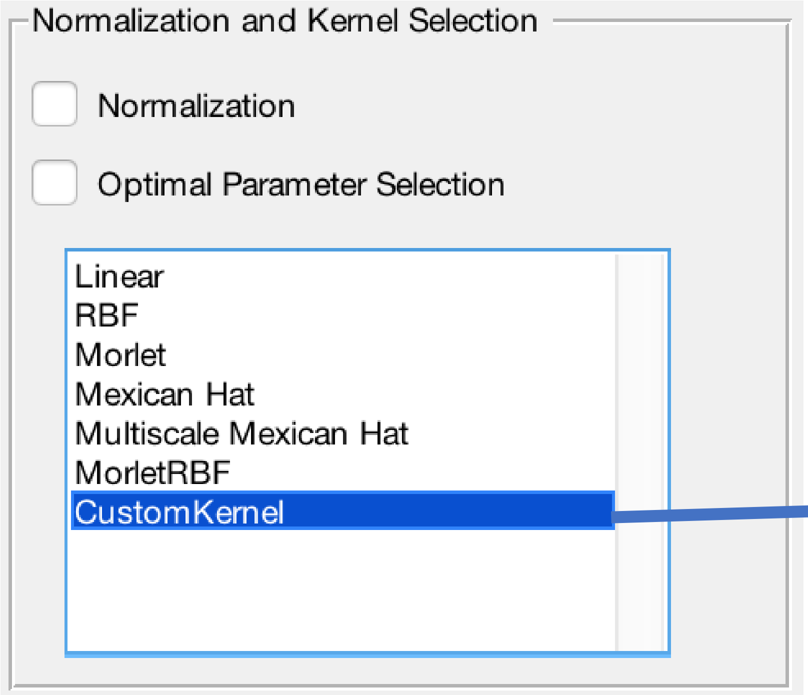
☐ Normalization

☐ Optimal Parameter Selection

- Linear
- RBF
- Morlet
- Mexican Hat
- Multiscale Mexican Hat
- MorletRBF
- CustomKernel

The following list contains the respective kernels. Although Linear by default is highlighted, it's still needs to be selected.

Step 8:



Normalization and Kernel Selection

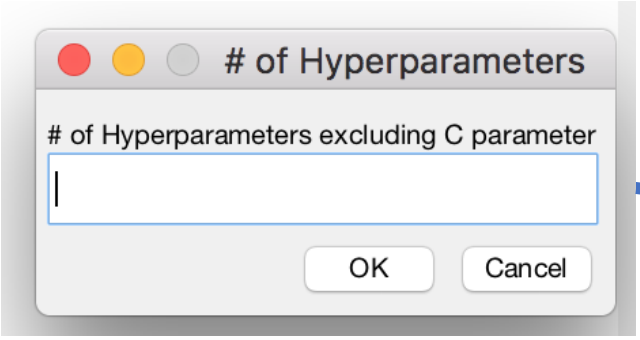
☐ Normalization

☐ Optimal Parameter Selection

- Linear
- RBF
- Morlet
- Mexican Hat
- Multiscale Mexican Hat
- MorletRBF
- CustomKernel

The user can choose to build their own customized kernel. The customized kernel function needs to be in the Matlab function handle format. When the CustomKernel option is selected, an input dialog box opens.

Step 8a:


# of Hyperparameters

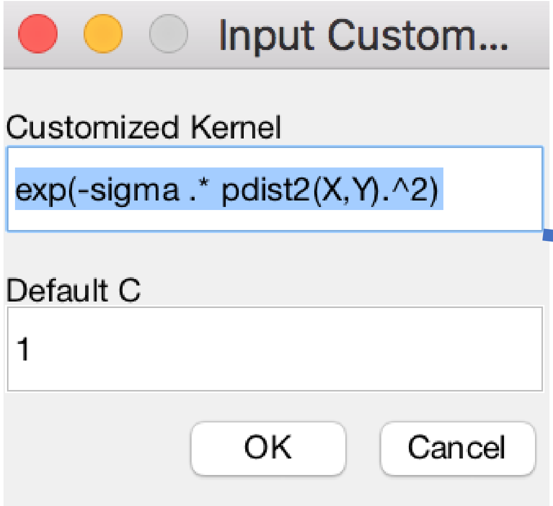
# of Hyperparameters excluding C parameter

Insert the number of hyperparameters. 0 if C is the only parameter, 1 for an additional hyperparameter and so on. Maximum allowable hyperparameters is 3 excluding C. For example, MorletRBF has the C parameter plus three additional parameters. Note once you press "OK" it opens another dialog box with an example kernel functions and hyperparameter values <u>if the optimal parameter selection box is unchecked</u> and kernel function with range for grid search of the hyperparameters if the optimal parameter selection box is checked.

Step 8b:

Examples of the dialog box that opens once you pick the number of hyperparameters and if the optimal parameter selection box is unchecked.

For number of hyperparameters = 0:


Input Custom...

Customized Kernel

exp(-sigma .* pdist2(X,Y).^2)

Default C

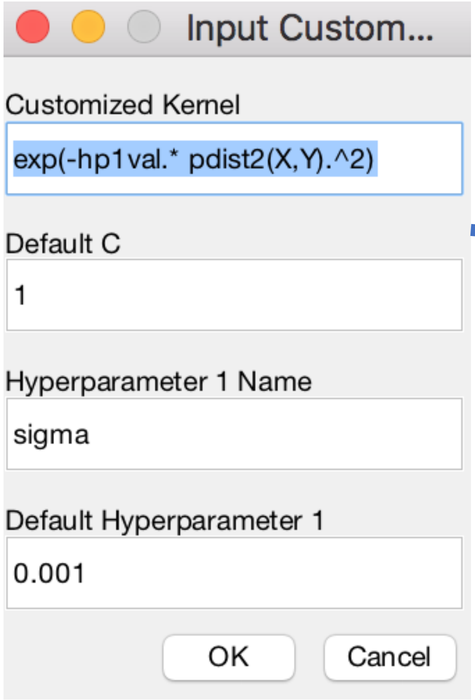
1

The kernel displayed is just an example. In case of 0 hyperparameters (excluding the C parameter), there should be no sigma. Thus, for example the user could replace the default display function by writing his/her own function as @(X,Y) X*Y'

This was just an example of the Linear kernel. However, note that @(X,Y) is added as well. @ is the symbol representing function handle in Matlab and (X,Y) are the number of input parameters.

The Default value of C can be changed to the value user think is a better option.

For number of hyperparameters = 1:



Input Custom...

Customized Kernel

exp(-hp1val.* pdist2(X,Y).^2)

Default C

1

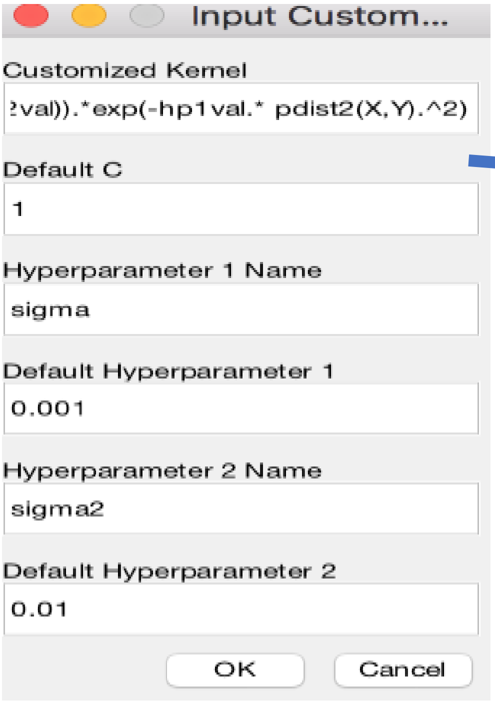
Hyperparameter 1 Name

sigma

Default Hyperparameter 1

0.001

For number of hyperparameters = 2:



Input Custom...

Customized Kernel

?val)).*exp(-hp1val.* pdist2(X,Y).^2)

Default C

1

Hyperparameter 1 Name

sigma

Default Hyperparameter 1

0.001

Hyperparameter 2 Name

sigma2

Default Hyperparameter 2

0.01

The kernel function can again be modified as mentioned above. Including the C parameter, we have 3 parameters. So now the displayed function could be modified or replaced by user. User should again add the function handle symbol "@" before the function expression. So, it should start with @(X,Y, hp1val, hp2val) in this case. Note, the rule in this case is that the name of the additional hyperparameter be "hp1val" and "hp2val".

The Default value of C can be changed to the value user think is a better option.

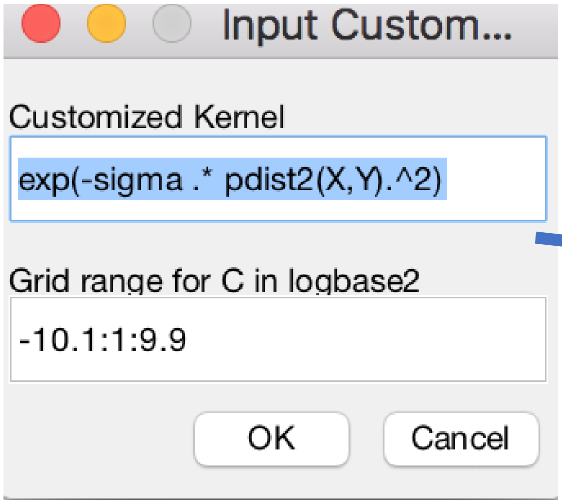
The default Hyperparameter 1 could be a sigma (or bandwidth) parameter of the kernel and again the default values can be replaced.

The default Hyperparameter 2 could be an additional sigma (or bandwidth) parameter of the kernel and again the default values can be replaced.

Step 8c:

Examples of the dialog box that opens once you pick the number of hyperparameters and if the optimal parameter selection box is checked.

For number of hyperparameters = 0:



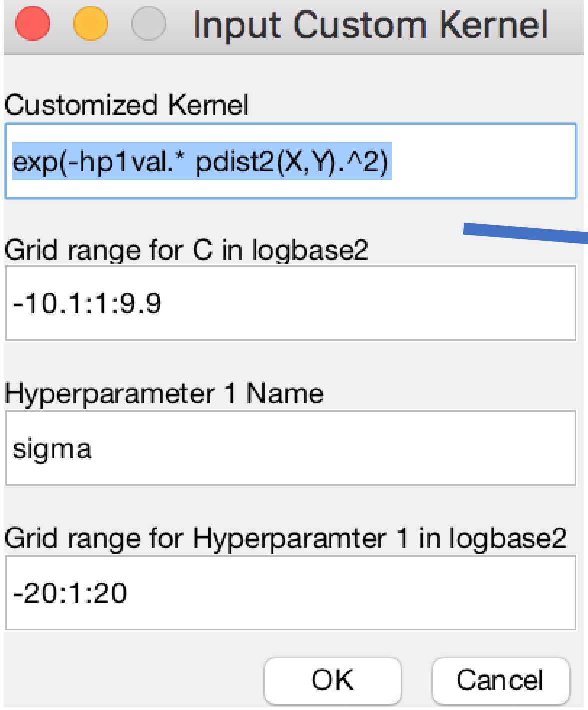
Customized Kernel: exp(-sigma .* pdist2(X,Y).^2)

Grid range for C in logbase2: -10.1:1:9.9

Example of displayed dummy function is shown in previous pages and any functional expression needs to precede with function handle symbol @ and the input parameters in the brackets following @

The grid range of C can be changed to the value user think is a better range option.

For number of hyperparameters = 1:



Customized Kernel: exp(-hp1val.* pdist2(X,Y).^2)

Grid range for C in logbase2: -10.1:1:9.9

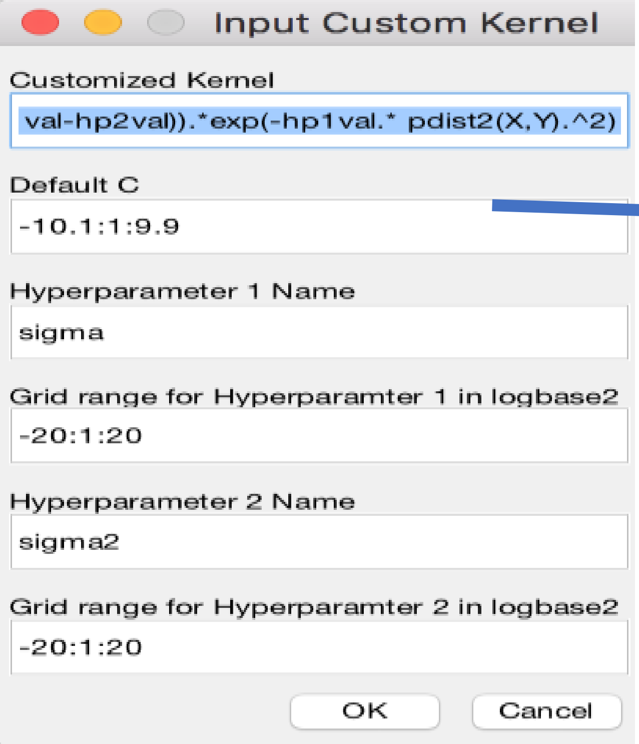
Hyperparameter 1 Name: sigma

Grid range for Hyperparamter 1 in logbase2: -20:1:20

The kernel function can again be modified as mentioned above. Including the C parameter, we have 3 parameters. So now the displayed function could be modified or replaced by user. User should again add the function handle symbol "@" before the function expression. So, it should start with @(X,Y, hp1val, hp2val) in this case. Note, the rule in this case is that the name of the additional hyperparameter be "hp1val".

The grid range of C and the default Hyperparameter 1 can be modified by the user.

For number of hyperparameters = 2:



Customized Kernel
val-hp2val)).*exp(-hp1val.* pdist2(X,Y).^2)

Default C
-10.1:1:9.9

Hyperparameter 1 Name
sigma

Grid range for Hyperparamter 1 in logbase2
-20:1:20

Hyperparameter 2 Name
sigma2

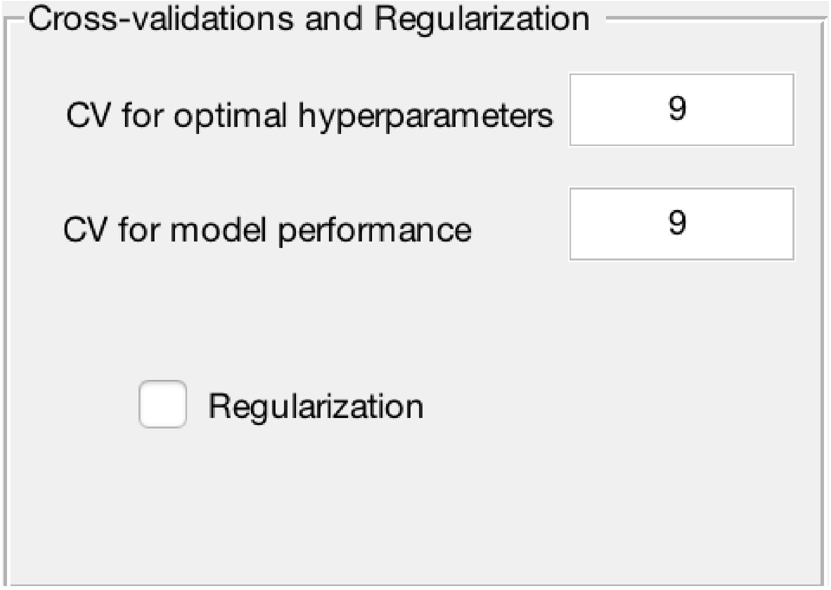
Grid range for Hyperparamter 2 in logbase2
-20:1:20

The kernel function can again be modified as mentioned above. Including the C parameter, we have 3 parameters. So now the displayed function could be modified or replaced by user. User should again add the function handle symbol "@" before the function expression. So, it should start with @(X,Y, hp1val, hp2val) in this case. Note, the rule in this case is that the name of the additional hyperparameter be "hp1val" and "hp2val".

The grid range of C and the default Hyperparameter 1 and Hyperparameter 2 can be modified by the user.

Cross-Validations and Regularization:
Step 9: If the optimal parameter selection is checked on then the user can select how many CV loops to run for optimal hyperparameter selection.

CV loop for model performance is a must.

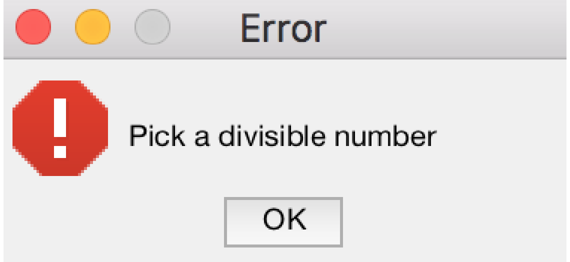


CV for optimal hyperparameters 9

CV for model performance 9

□ Regularization

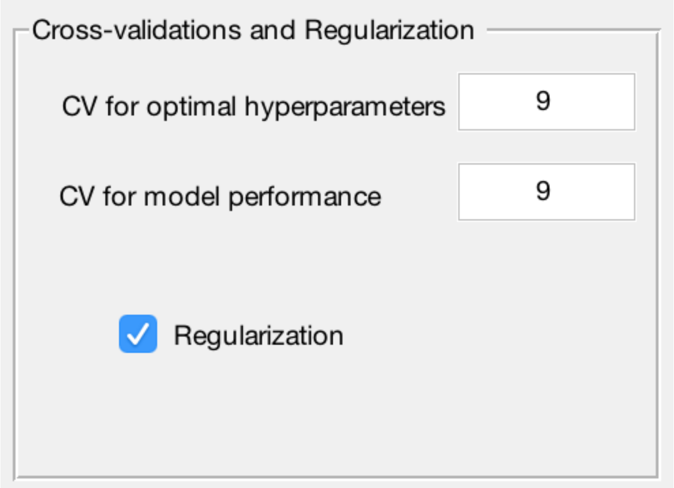
If the number of subjects user picks for CV loops is not fully divisible (i.e. remainder = 0) then there is an error thrown back at the user. The cross validation procedure needs perfect divisible number to function correctly.


Error

Pick a divisible number

Step 10:

The regularization step is optional but highly recommended. Theoretically the Kernel (Gram) matrix needs to be positive definite in order for the Kernel to be in RKHS. In our experience, theoretically the Gram matric of wavelet kernels needs to be positive considering they are translational invariant yet due to numerical instabilities (sometimes due to really small hyperparameter values) can cause the Gram matrix to go negative. Regularization step insures that the Gram matrix never goes negative.

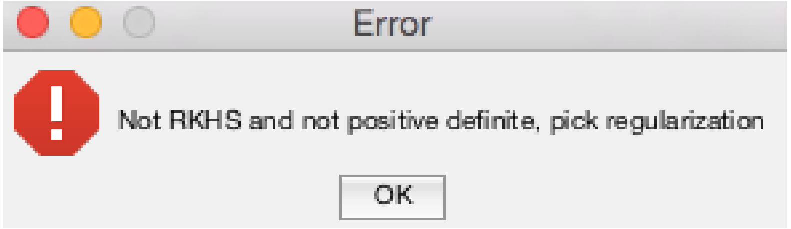

Cross-validations and Regularization

CV for optimal hyperparameters: 9

CV for model performance: 9

☑ Regularization

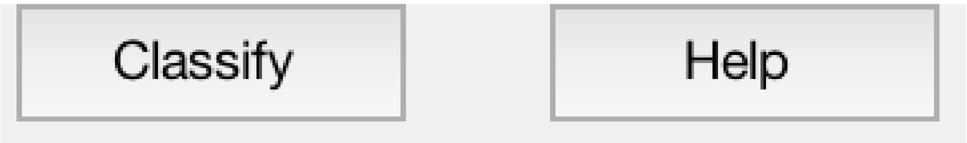
If the user is interested in testing the performance of the customized kernel then the regularization could be switched off and the user will get an error if the Gram matrix is negative.



Error

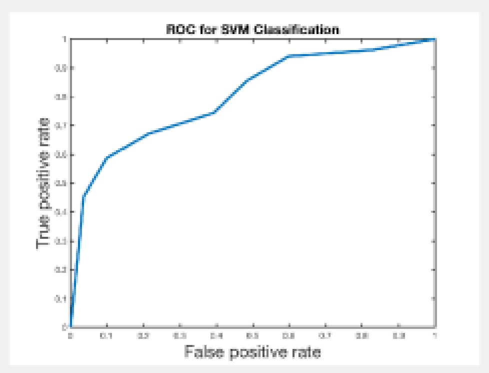
Not RKHS and not positive definite, pick regularization

Classify:
Step 11:
This is the final step and once the user has properly followed the previous 10 steps then he/she can press the Classify button

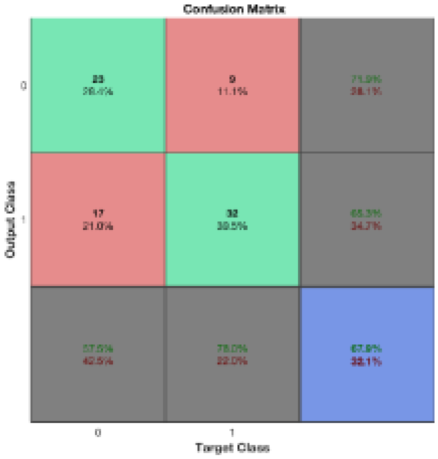


Once the user presses on classification, the GUI displays progress bar for optimal parameter selection (if chosen) and model evaluation. Post classification the results panel show the following graphical outputs:

1. **ROC curve:**

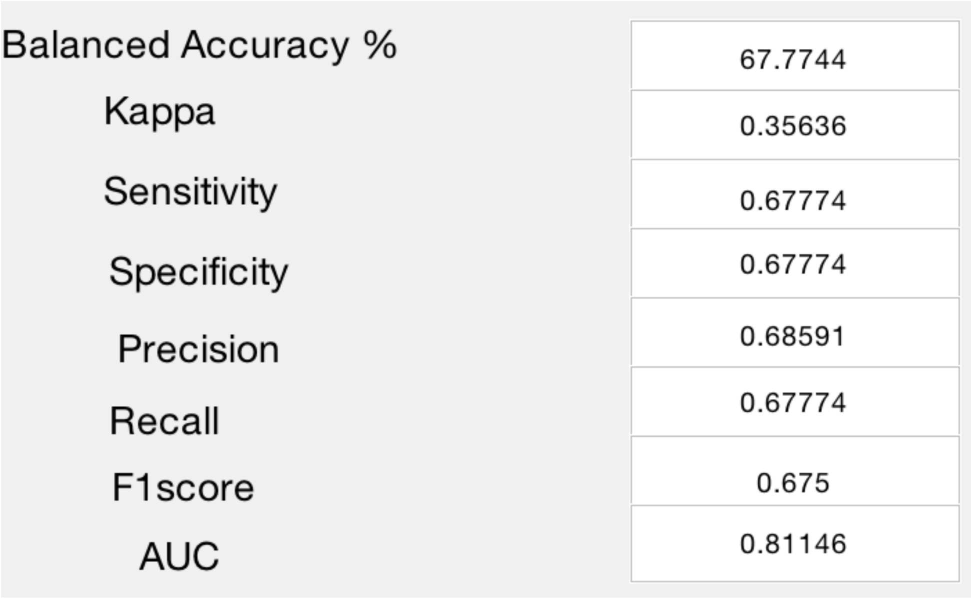


2. **Confusion Matrix**



	0	1	
0	23 28.4%	9 11.1%	71.9% 28.1%
1	17 21.0%	32 39.5%	65.3% 34.7%
	57.5% 42.5%	78.0% 22.0%	67.9% 32.1%

Post classification the GUI also displays the Machine Learning Statistics:



Balanced Accuracy %	67.7744
Kappa	0.35636
Sensitivity	0.67774
Specificity	0.67774
Precision	0.68591
Recall	0.67774
F1score	0.675
AUC	0.81146