Pattern Recognition for Neuroimaging Toolbox: PRoNTo

Jessica Schrouff

PRNI 2018

June 14th

NUS, Singapore

Outline

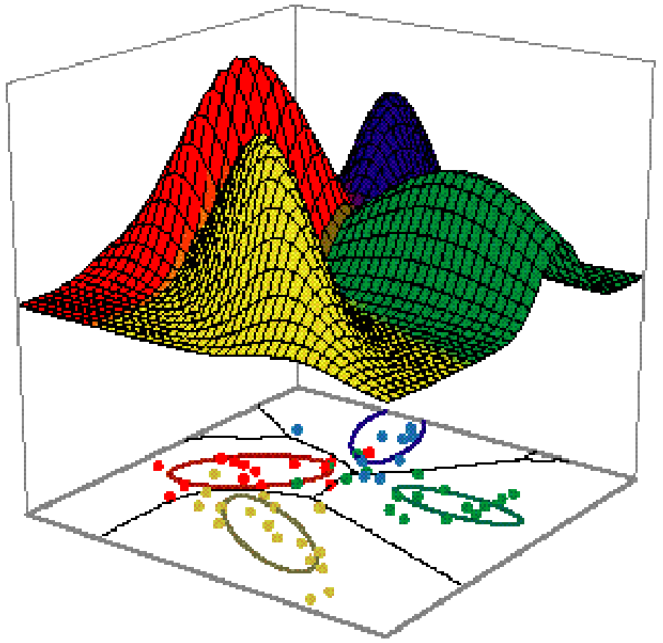
- PRoNTo's goals and history
- PRoNTo for users
  - ✓ General framework
  - ✓ Modules
- PRoNTo for developers
  - ✓ Implementation
  - ✓ Scripting examples
- PRoNTo v3
- Hands-on session with Fabio Ferreira

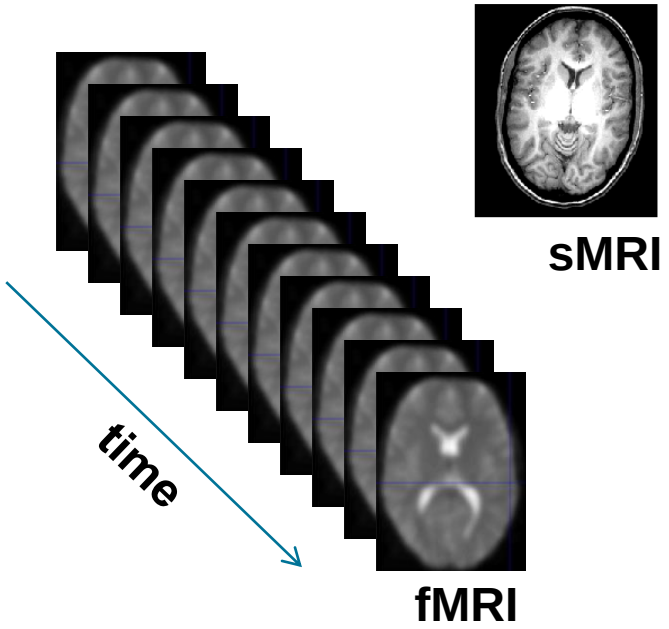
Aims

Machine Learning community

PRoNTo

Neuroscience and Clinical Neuroscience communities





PRoNTo's history

- Pascal Harvest project: August 2011, v1.0
- Team from different backgrounds and institutions
- Releases since: v1.1, v2.0, v2.1, v3.0 to come

PRoNTo features

- Matlab based
- Open-source
- Interfaced: GUI and batch
- SPM compatible
- Multiple classification and regression algorithms
- Includes developments from the team member's work

Available at: http://www.mlnl.cs.ucl.ac.uk/pronto/

Questions it addresses

- Classification
  - ➢Can we use brain scans to diagnose psychiatric or neurological disorders?
  - ➢Can we decode from the brain scans information about the stimuli being processed?

- Regression
  - ➢ Can we predict clinical scores from brain scans?

PRoNTo for users

Jessica Schrouff

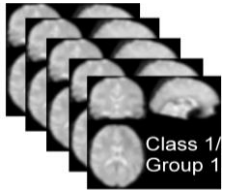
PRNI 2018

June 14th

NUS, Singapore

Framework

Data
Design
1st level
mask



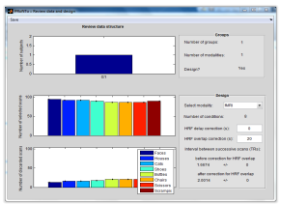
Atlas
2nd level
mask (ROI)

Data & Design

- Groups
- Subjects/scans
- Modalities
- Conditions

PRT.mat

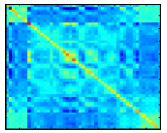
Review Design



Feature set

- Extract features
- Build kernel(s)
- Detrend
- Scaling

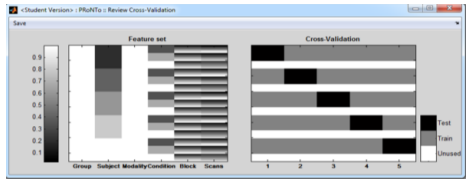
Review Kernel



Specify model

- Targets/classes
- Cross-Validation
- Hyper-parameter optimization
- Machines
- Data operations

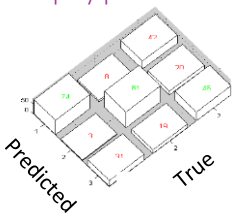
Review Model, CV



Run model

- Estimate model
- Assess performance
- Assess significance

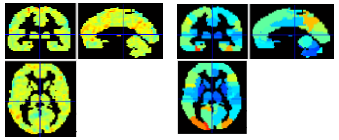
Display performance



Weights

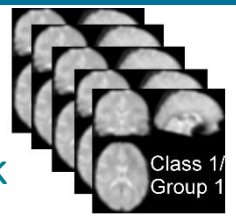
- Feature weights
- Kernel contribution

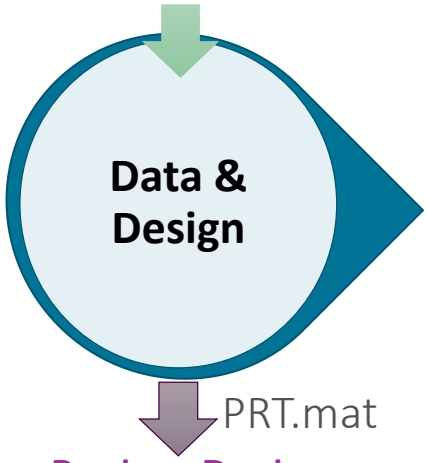
Display weights

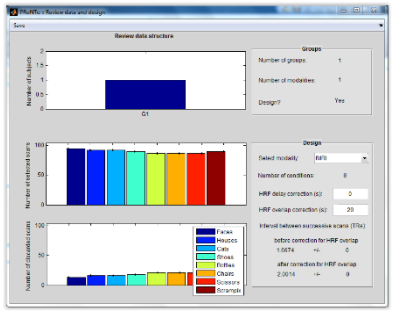


Data & Design

Data
Design
1st level mask

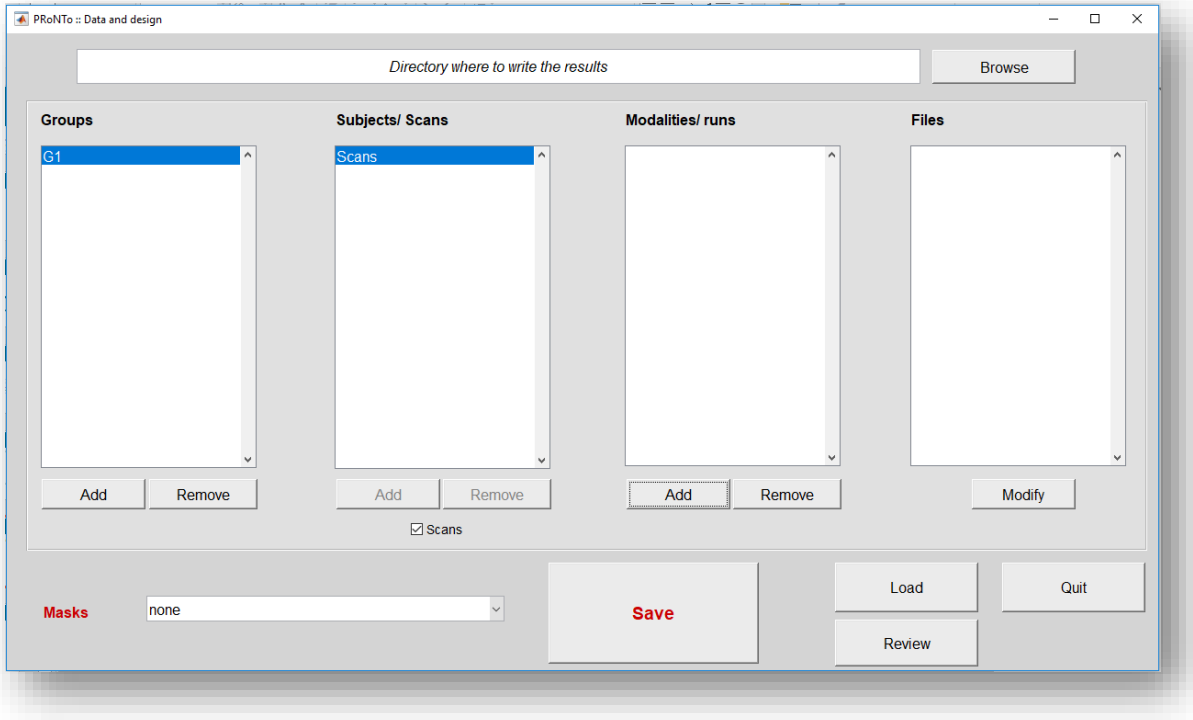






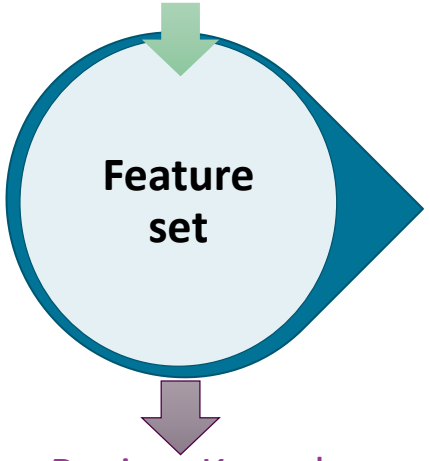
- User inputs:

✓ Groups
✓Subjects
✓Modalities
✓Design
✓Covariates
✓Targets
✓Masks

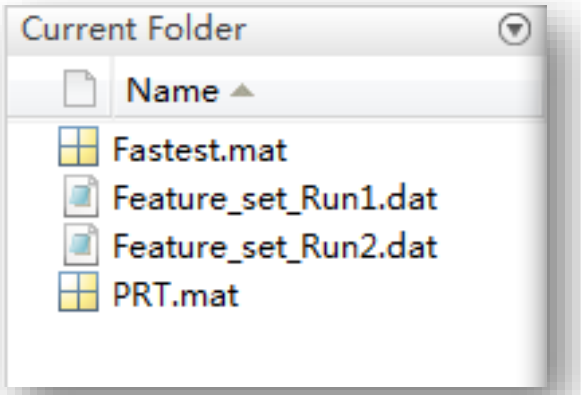


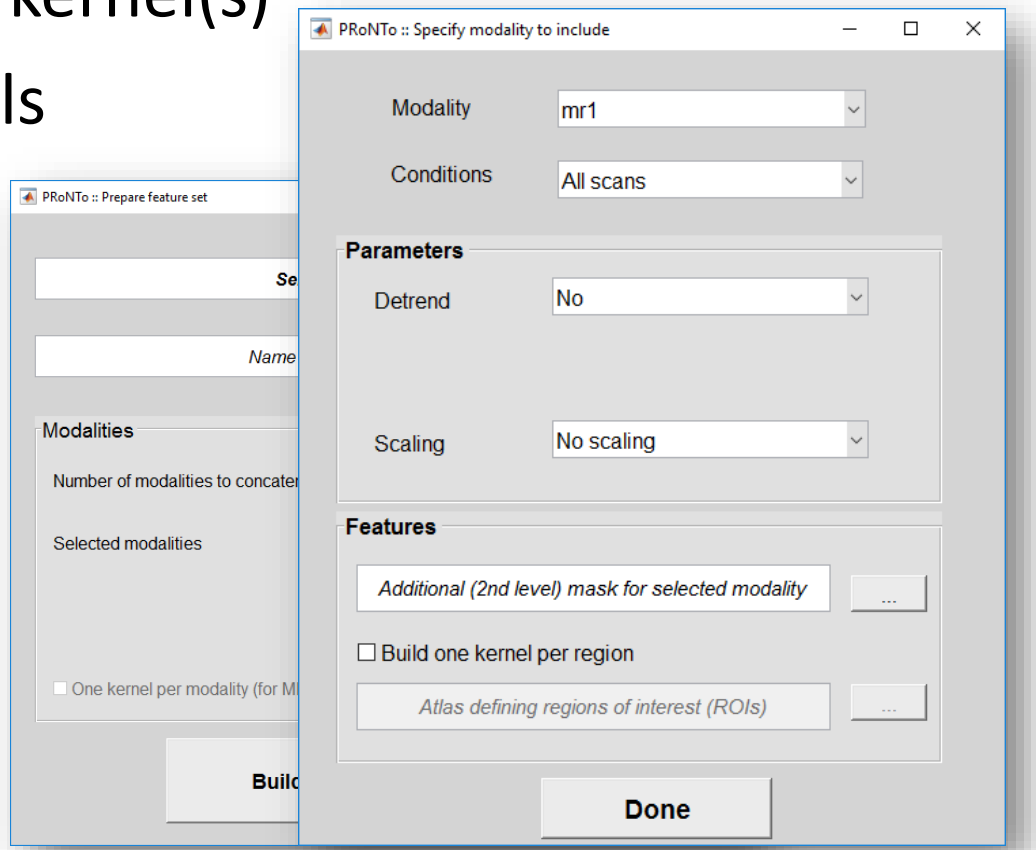
Feature set

Atlas
2nd level
mask (ROI)

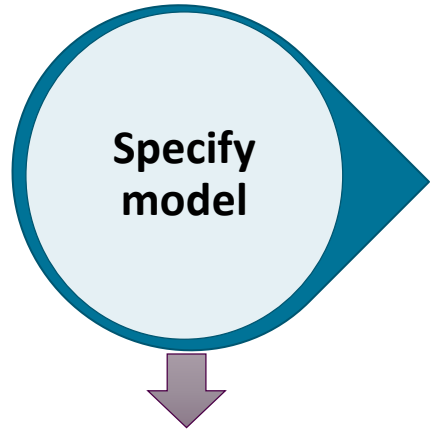


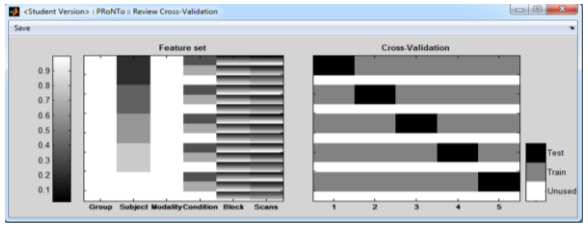
- Reads through all images in modality
- Selects features for kernel(s)
- v2: combines kernels



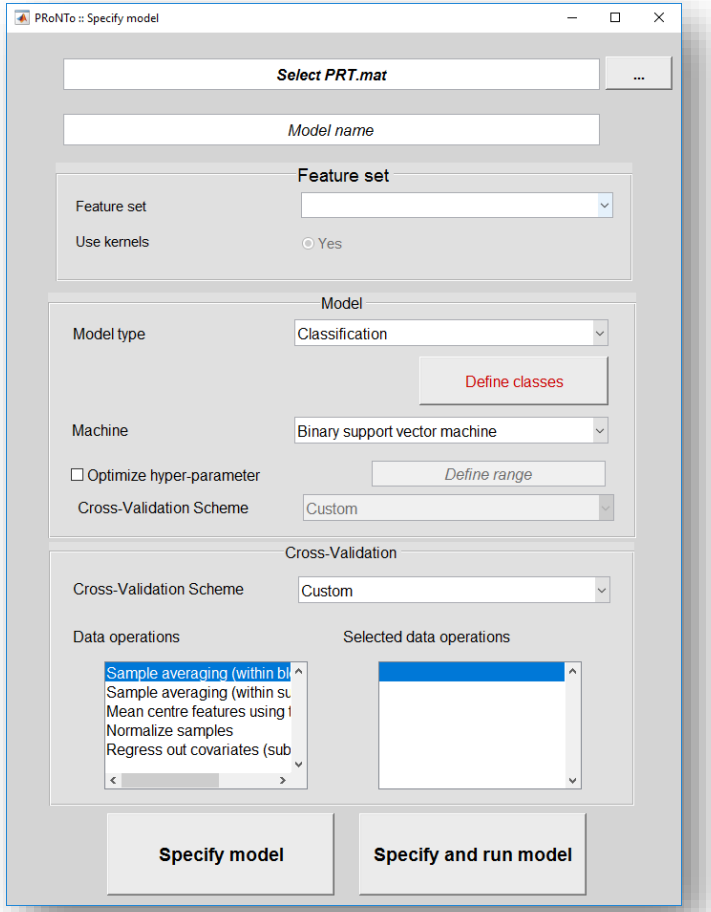


Specify model





- Define the question:
- ✓ feature set
- ✓ classes / samples
- ✓ machine and parameters
- ✓ cross-validation (inner & outer)
- ✓ data operations



Machines

- Classification:

✓ SVM (LIBSVM)

✓ Gaussian Processes (GPML)

✓ L1-MKL (simpleMKL)

- Regression:

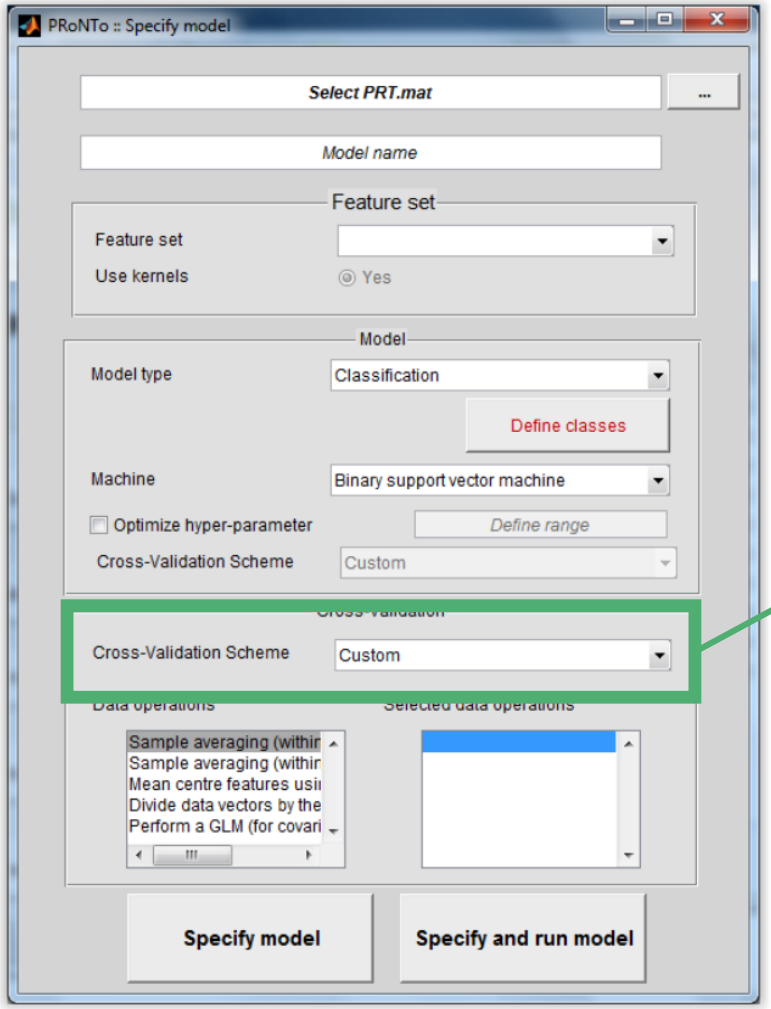
✓ RVR

✓ KRR

✓ Gaussian Processes (GPML)

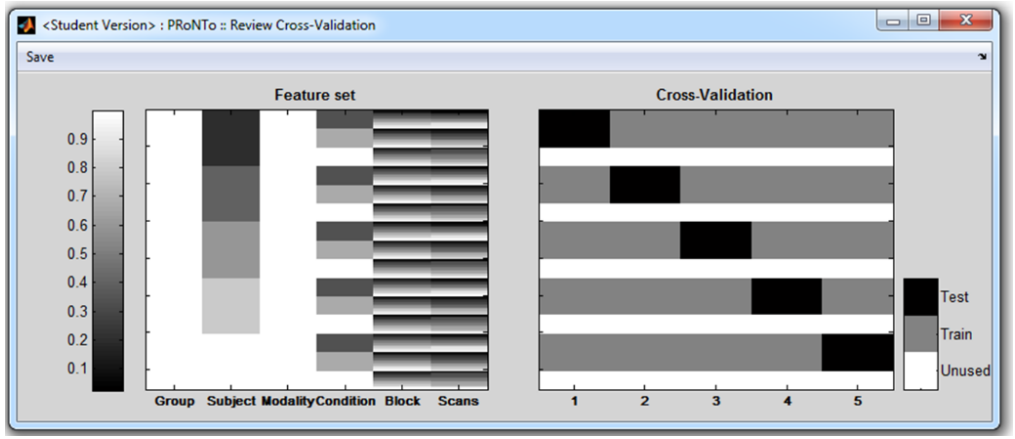
✓ L1-MKL (simpleMKL)

Cross-validation

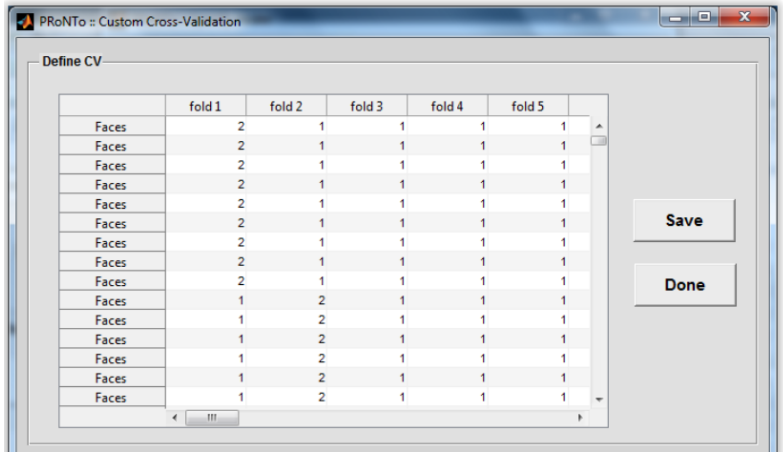


Standard approaches:

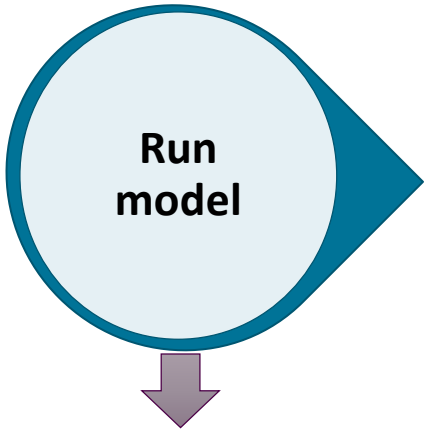
- LOSO
- LOBO
- LORO
- LOSCO
- k-fold CV



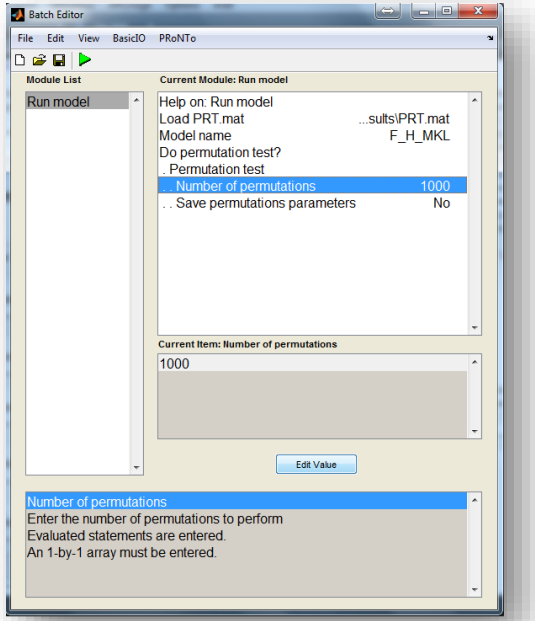
Flexible CV schemes allowed

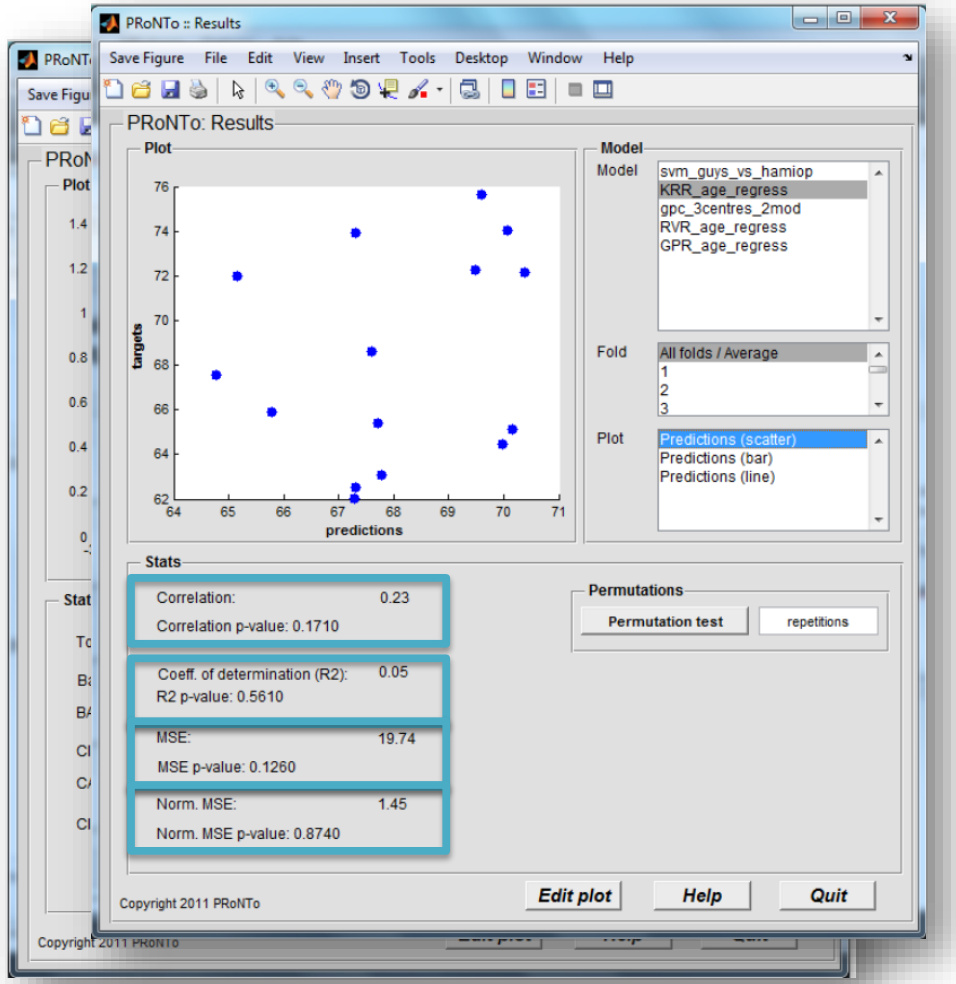


Run model



- Estimates the model
- Batch: runs permutations



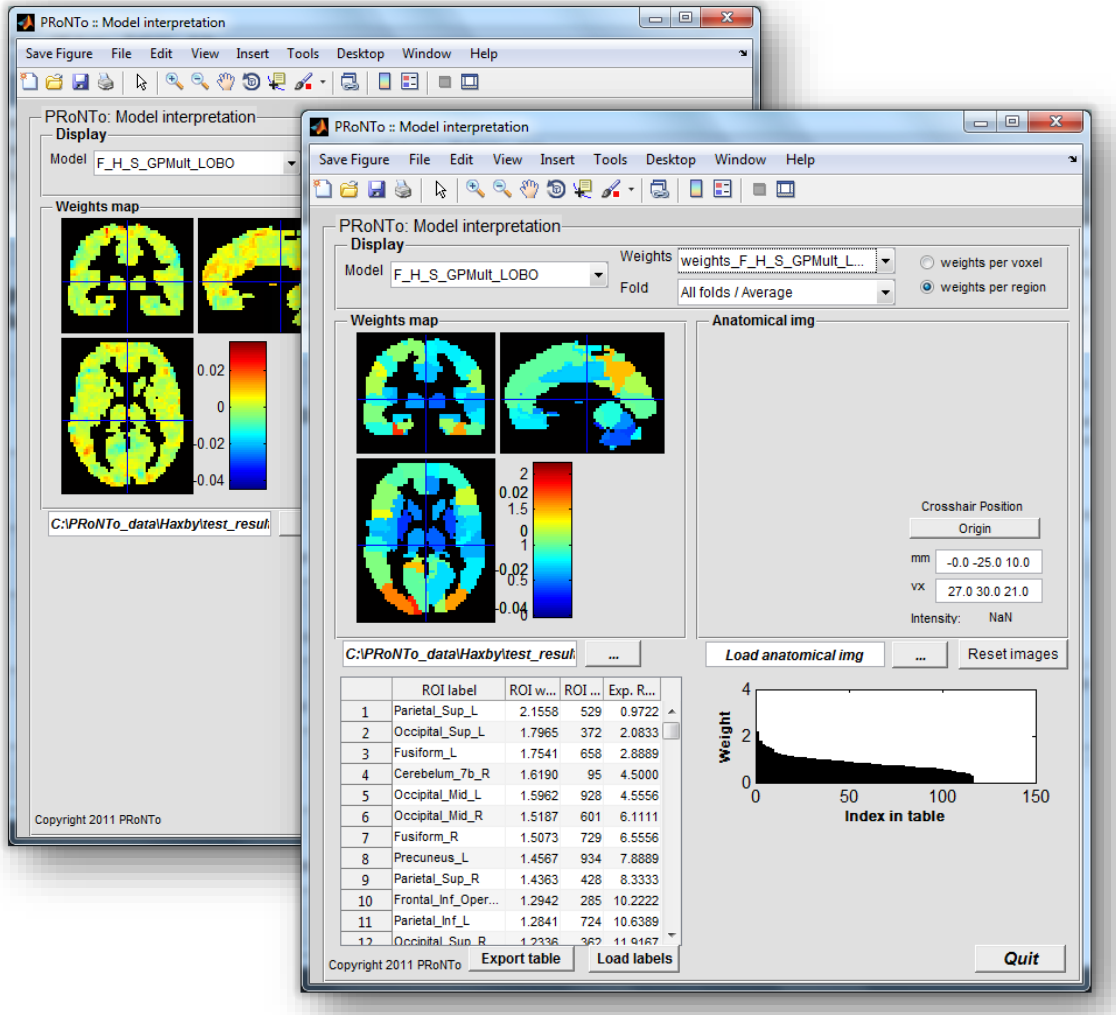


Compute weights

- Build voxel weights
- Maps back to original image
- ROI contributions:

✓ average from atlas (post-hoc)

✓ kernel contribution (MKL)



Thank you!

Questions?

PRoNTo for developers

Jessica Schrouff

PRNI 2018

June 14th

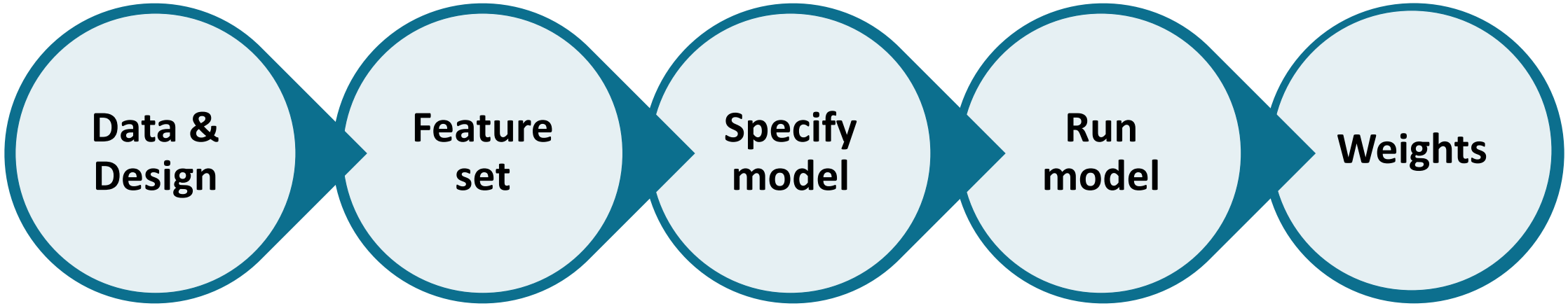
NUS, Singapore

PRoNTo for developers

- Tool box: use PRoNTo as a basis for developments
- Avoids to re-code e.g. cross-validation when new data/model
- Established framework: less prone to errors, double dipping

PRoNTo architecture

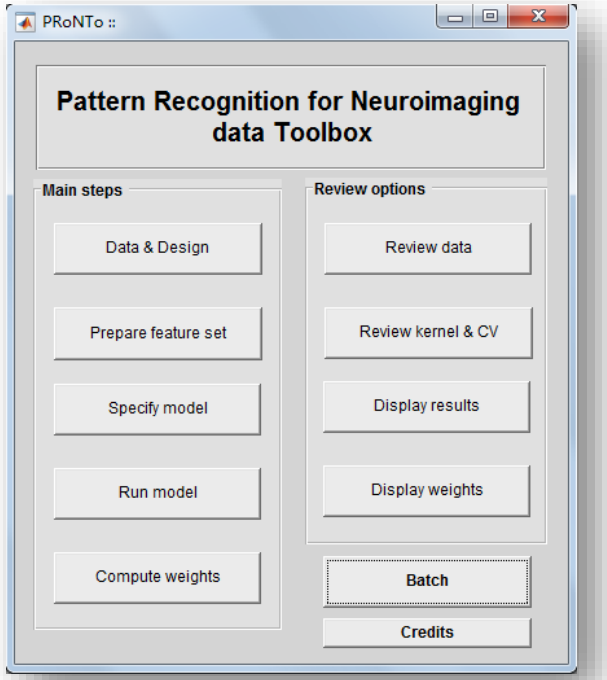
- Modular



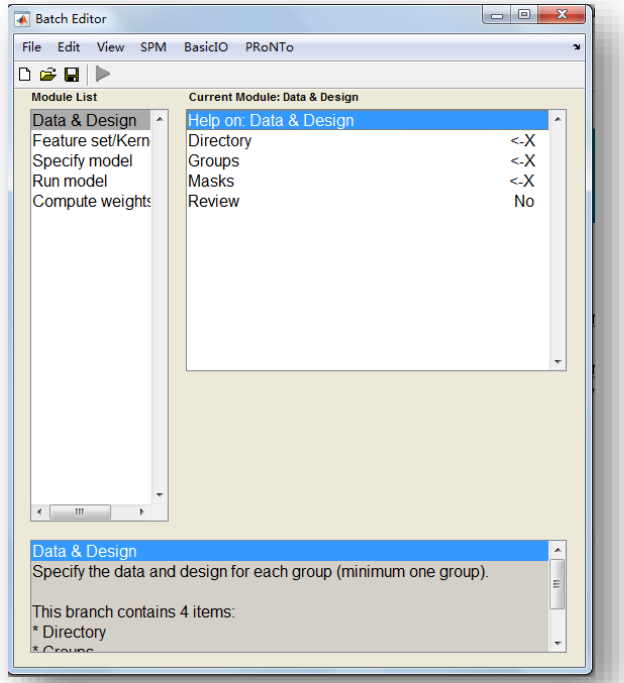
- User interfaces
- Core functions
- All written in Matlab with occasional mex interfaces

PRoNTo architecture

GUI



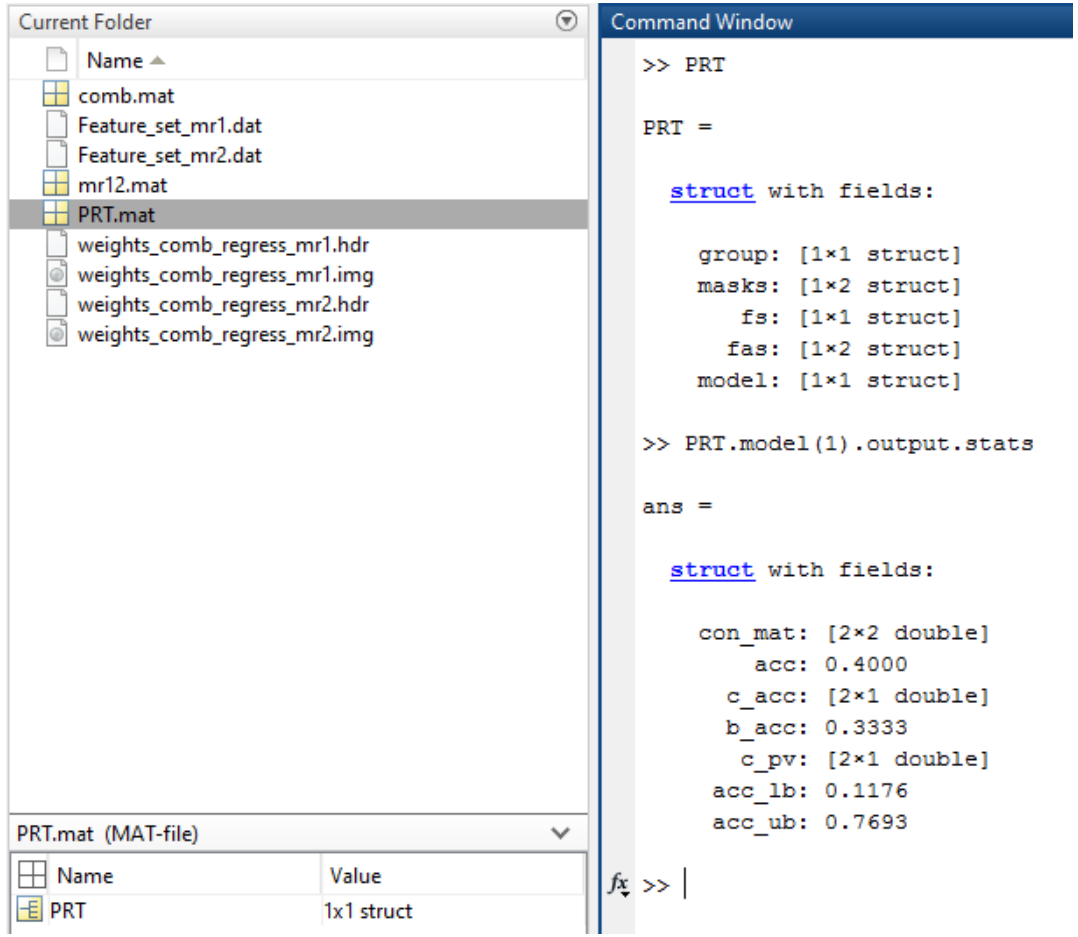
Batch



Core functions

PRT structure

- Similar to SPM structure
- Saved in folder as ‘PRT.mat’
- PRT has one or 2 fields per module
- Load and dig:



Core functions

- One function per module
- Ideal for scripting:

- Not for Data & Design step
- Best for running models

```
function [outfile] = prt_cv_model(PRT,in)
% Function to run a cross-validation structure on a given model
%
% Inputs:
% -------
% PRT:              data structure
% in.fname:         filename for PRT.mat (string)
% in.model_name:    name for this model (string)
%
% Outputs:
% --------
% Writes the following fields in the PRT data structure:
%
% PRT.model(m).output.fold(i).targets:     targets for fold(i)
% PRT.model(m).output.fold(i).predictions: predictions for fold(i)
% PRT.model(m).output.fold(i).stats:       statistics for fold(i)
% PRT.model(m).output.fold(i).{custom}:    optional fields
%
% Notes:
% ------
% The PRT.model(m).input fields are set by prt_init_model, not by
% this function
%
%_____________________________________________________________________
% Copyright (C) 2011 Machine Learning & Neuroimaging Laboratory
```

Scripting example - 1

- Simulated data: one feature set per SNR and sparsity
- 19 * 15 = 285 models
- Power using PRoNTo:
  - Home script
  - Batch to create feature set
  - Load PRT
  - Use a pre-defined model as basis
  - Replace the feature set
  - Run and extract performance

```
function [PRT] = pronto_analysis(fname,prtname,channel_file)

load(prtname)
path = spm_fileparts(prtname);
% Replace file name by new file name
PRT.group.subject.modality.scans = fname;
PRT = rmfield(PRT,{'fs','fas','model'});
save(fullfile(path,'PRT.mat'),'PRT')

% Build feature set
batch_create_feature_set(prtname,channel_file)
load(prtname)

% Specify models
batch_specify_models_pronto(prtname)
load(prtname)

% Run models
in.fname = prtname;
in.model_name = 'SVM';
prtname = prt_cv_model(PRT,in);
```

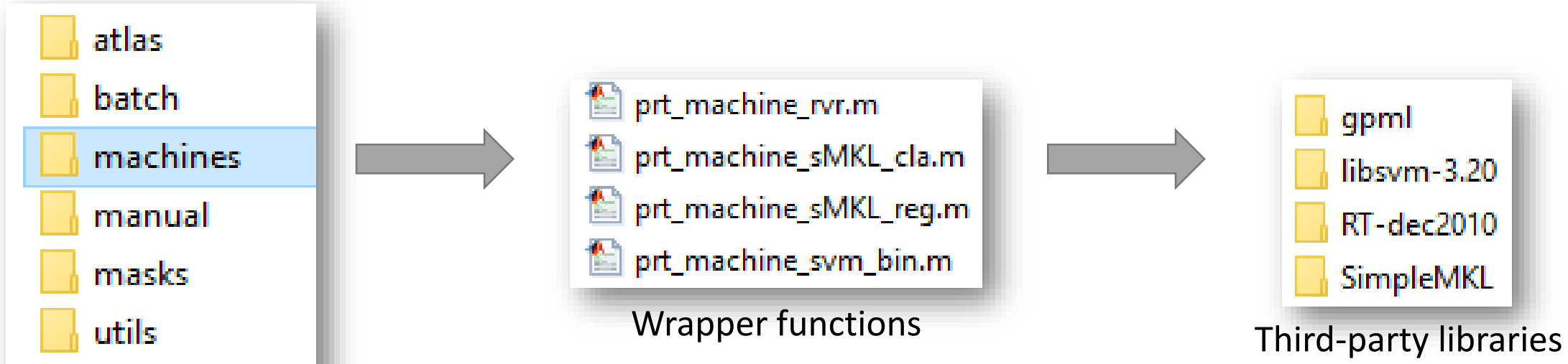
Scripting example - 2

- Permutations can be lengthy
- Can run code on cluster
- Different strategies:
  - one job per xx permutations
  - using parfor
  - spmd
  - Batch or prt_permutations
  - Careful with graphical/workspace output

```
function [] = prt_permutation(PRT, n_perm, modelid, path, flag)
% Function to compute permutation test
%
% Inputs:
% -------
% PRT:      PRT structure including model
% n_perm:   number of permutations
% modelid: model ID for model to test and model to copy permutations from
%          (optional). Can hence be of size 1x1 or 1x2.
% path:     path
% flag:     boolean variable. set to 1 to save the outputs for each
%           permutation. default: 0
%
```

PRoNTo machines

- PRoNTo interfaces libraries



New machine

- Can interface your classification/regression algorithm

prt_machine_esvr.m

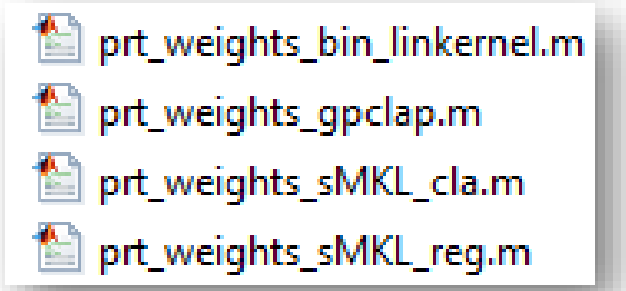
```
function output = prt_machine_esvr(d,args)
% Run Epsilon-SVR - wrapper for libSVM
% FORMAT output = prt_machine_esvr(d,args)
% Inputs:
%   d           - structure with data informa
%     .train      - training data (cell arr
%                   each [Ntr x D]). each m
%                   of the data. This is us
%                   multiple kernel learnin
%     .test       - testing data  (cell arr
%                   [Nte x D])
%     .tr_targets - training labels (for cl
%                   regression) (column vec
%     .use_kernel - flag, is data in form o
%                 form of features (false)
%   args      - libSVM arguments
% Output:
%   output  - output of machine (struct).
%     * Mandatory fields:
%      .predictions - predictions of classi
%     * Optional fields:
%      .func_val - value of the decision fu
%      .type     - which type of machine th
%
```

Re-arrange inputs/outputs and arguments

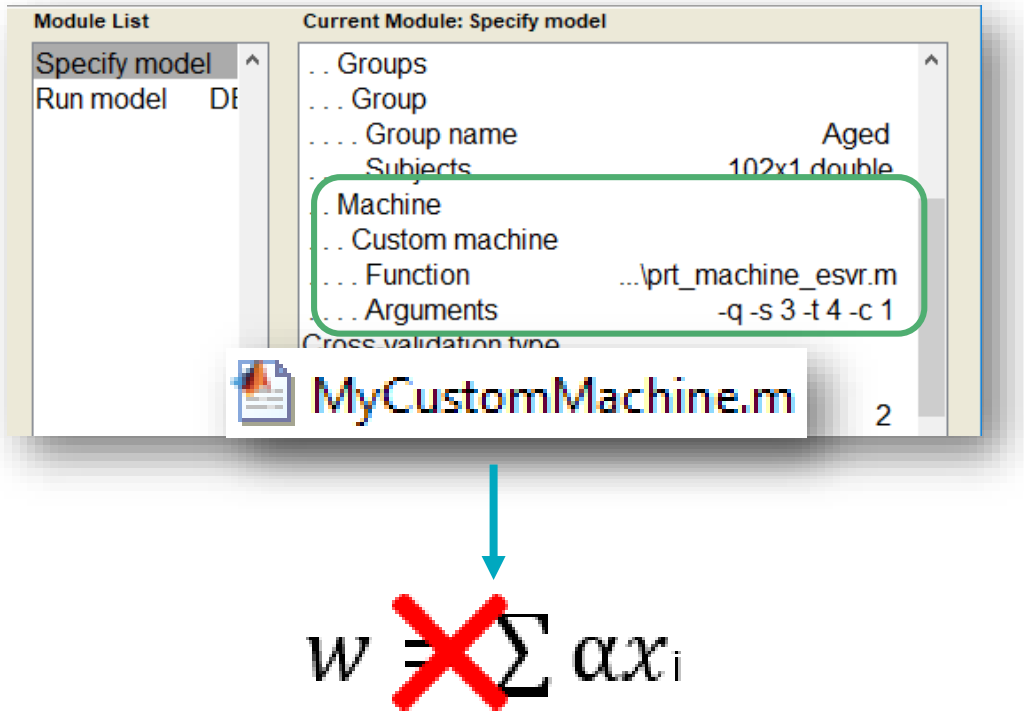
esvr_train.m
esvr_predict.m

New machine: further considerations

- labels: 1 for class 1, 2 for class 2, …
- Hyper-parameter optimization: careful with string arguments
- Use ‘custom machine’
- Compiled machine?
- Weight computation:
  - Default: linear kernel
  - If not: need to specify weight function



→ $w = \sum \alpha x_i$



$w \not\equiv \sum \alpha x_i$

Contributing

- Yes!
- Next releases in Github
- Need to interface: GUI and batch
- Backwards compatibility
- Matlab backwards compatibility and toolbox codes

- Coming soon: developer's manual

Thank you!

Questions?

PRoNTo v3.0

Jessica Schrouff

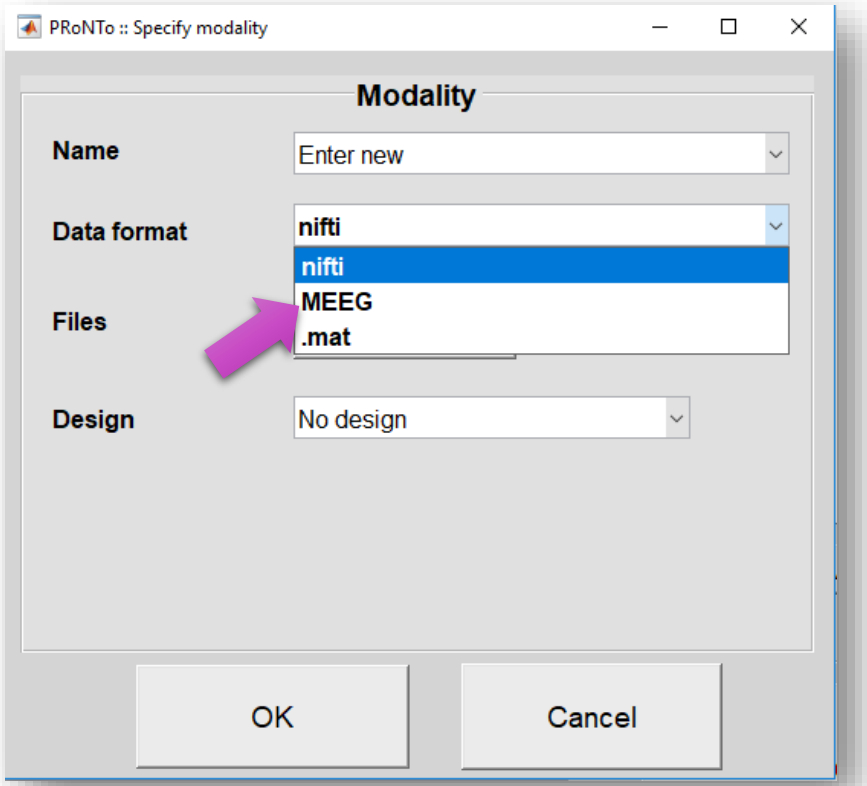
PRNI 2018

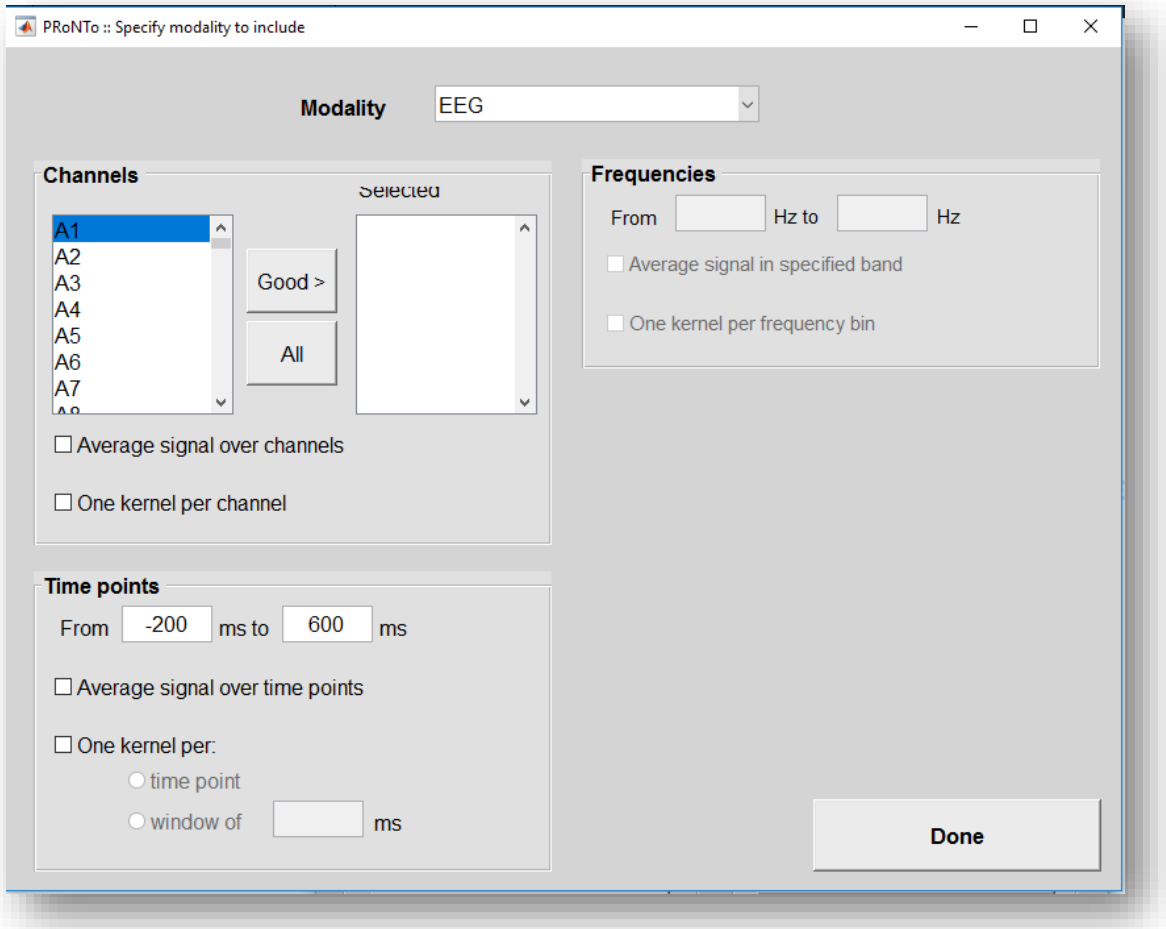
June 14th

NUS, Singapore

New input formats

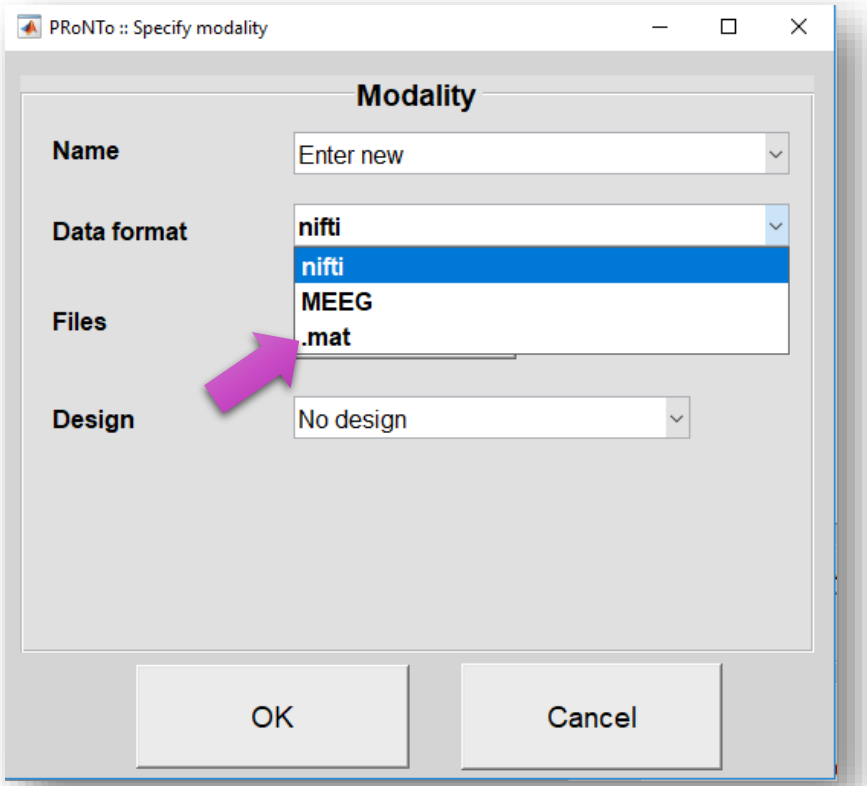
- SPM MEEG

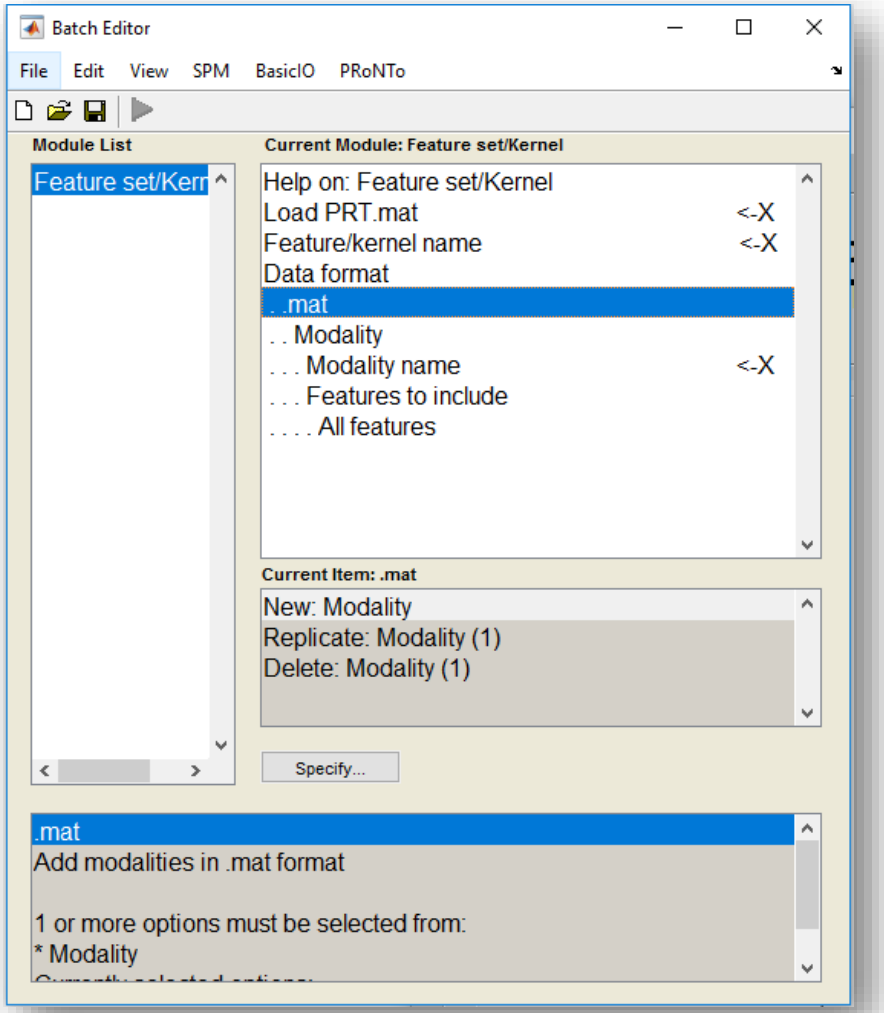




New input formats

- .mat





New functionalities

- Non-kernel methods
- Combining different data formats (e.g. EEG and fMRI)
- Multi-Task Learning
- Test model module: share PRT and weight maps

- and much more …

Thank you!

Suggestions?

Hands-on session

Jessica Schrouff

PRNI 2018

June 14th

NUS, Singapore

Demo

- OASIS data set
- 50 demented – 50 non-demented
- Confounds: age and gender (boolean)
- Two features extracted: grey and white matter

- **Goal**: classify demented from non-demented