PRoNTo

Software resources

Edinburgh
April 2013

Christophe Phillips, Ir. PhD.

Cyclotron Research Centre, ULg, Belgium
http://www.cyclotron.ulg.ac.be

PRoNTo

- Software
- Conclusion

Existing stuff

Princeton Multi-Voxel Pattern Analysis (MVPA) Toolbox (http://code.google.com/p/princeton-mvpa-toolbox/)

- **(+)** Matlab based, AFNI/BV oriented but some SPM5 compatibility,
- **(-)** only(?) neural network classification, only(?) fMRI data, last version from 2009.

Multivariate Pattern Analysis with Python – PyMVPA (http://www.pymvpa.org/)

- **(+)** looks powerful with loads of tools,
- **(-)** Python based, OS specific (Linux).

Brain Voyager MVPA – BV-MVPA (http://www.brainvoyager.com/bvqx/)

- **(+)** search-light and SVM,
- **(-)** proprietary code, fMRI only, no way of adding own machine

OUR GOAL

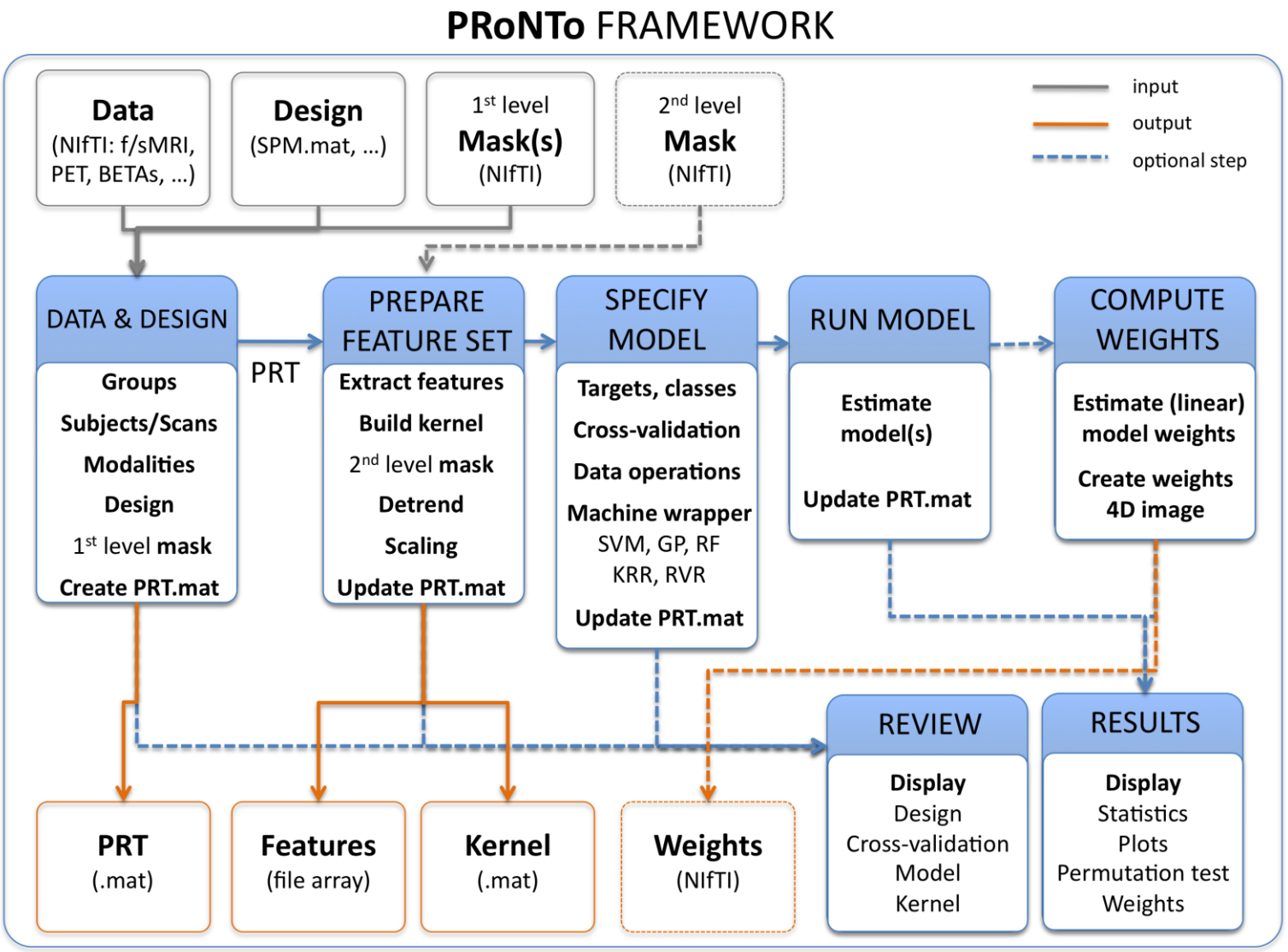
"develop a toolbox based on machine learning techniques for the analysis of neuroimaging data"

BUT

Matlab based, easy to use,
multiple modalities (fMRI/sMRI/PET),
various machines, modular code

→PRoNTo

PRoNTo


PRoNTo FRAMEWORK
Data
(NIfTI: f/sMRI, PET, BETAs, ...)
Design
(SPM.mat, ...)
1st level
Mask(s)
(NIfTI)
2nd level
Mask
(NIfTI)
input
output
optional step
DATA & DESIGN
Groups
Subjects/Scans
Modalities
Design
1st level mask
Create PRT.mat
PRT
PREPARE FEATURE SET
Extract features
Build kernel
2nd level mask
Detrend
Scaling
Update PRT.mat
SPECIFY MODEL
Targets, classes
Cross-validation
Data operations
Machine wrapper
SVM, GP, RF
KRR, RVR
Update PRT.mat
RUN MODEL
Estimate model(s)
Update PRT.mat
COMPUTE WEIGHTS
Estimate (linear) model weights
Create weights 4D image
REVIEW
Display
Design
Cross-validation
Model
Kernel
RESULTS
Display
Statistics
Plots
Permutation test
Weights
PRT
(.mat)
Features
(file array)
Kernel
(.mat)
Weights
(NIfTI)

Requirements

- A recent version of Matlab
- SPM8: Wellcome Trust Centre for Neuroimaging, University College London, UK. *http://www.fil.ion.ucl.ac.uk/spm/software/spm8/*

Some code in C/C++

Available for: Windows XP (32 bits), Windows 7 (64 bits), Mac OS 10, *some* Linux flavours (32/64 bits).

If more "exotic" $\Rightarrow$ compile for your OS !!!

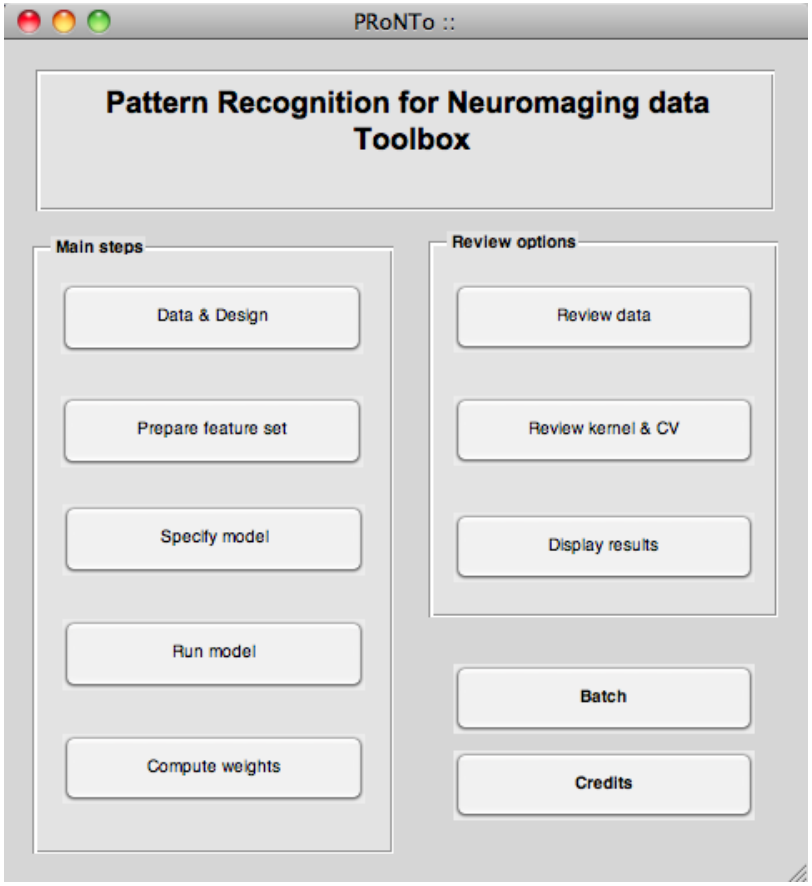
PRoNTo

For users

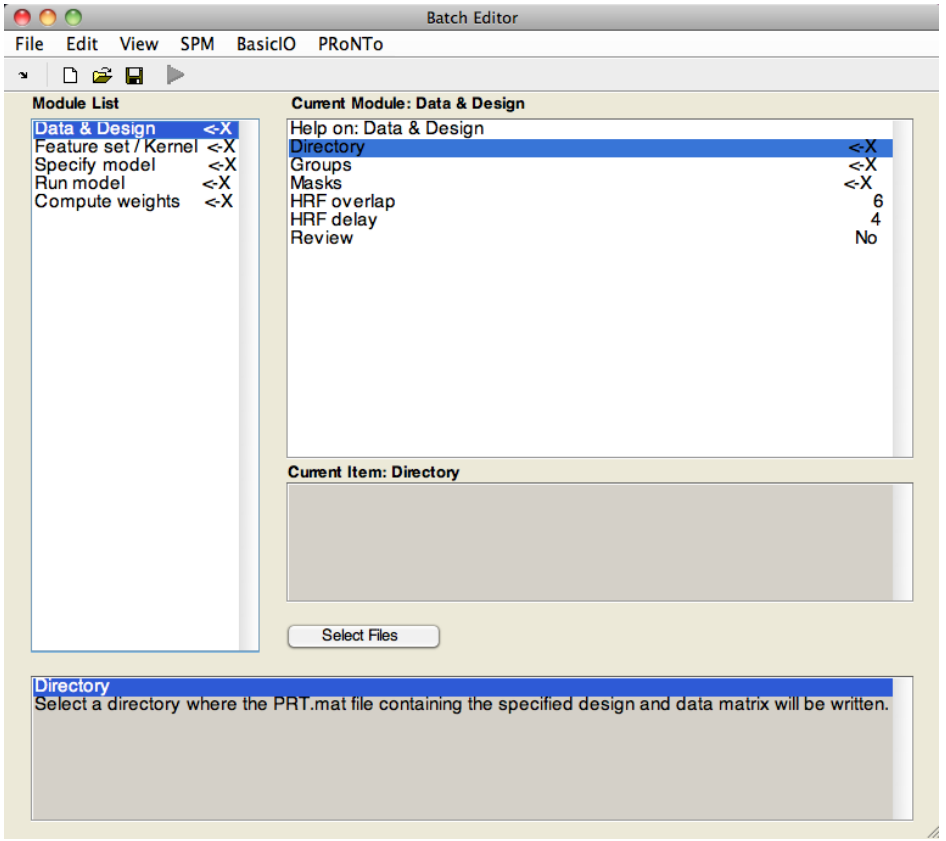
User Interface	Matlab Batch	Functions
Easy to use	Very efficient	User-specific analysis
Click buttons	Can be saved and copied	More programming skills required
	Compatible with SPM	

User point of view

User Interface



Matlab Batch



PRoNTo

For developers

User Interface

Specific GUI

Batch system

Script

Machine learning

Features

Kernel

Model

Training

Validation

Wrapper

Machines

Classification (SVM, GPC, RF)

Regression (KRR,RVR)

Machine Learning tools, classification

- Support Vector Machines (SVM)

  LibSVM implementation,
  http://www.csie.ntu.edu.tw/~cjlin/libsvm

  Nello Cristianini and John Shawe-Taylor. An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge University Press, New York, NY, USA, 2000.

- Gaussian Process Classifier (GPC)

  GPLML implementation,

  http://www.gaussianprocess.org/gpml/code/matlab/doc/

  Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning. the MIT Press, 2006.

Machine Learning tools, regression

- Kernel Ridge Regression (KRR)

  John Shawe-Taylor and Nello Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, 2004.

- Relevance Vector Regression (RVR)

  Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. Journal of Machine Learning Research, 1:211(244), 2001.

- Gaussian Process Regression (GPC)

  GPLML implementation,

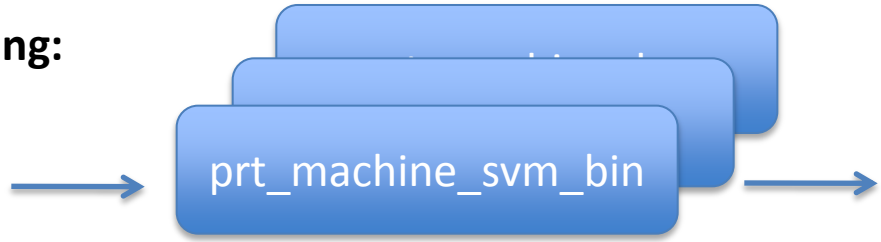
Developer point of view

Structure containing:
-Data/Kernel
-Labels
-Etc
Structure containing:
-Function name (machine)
-Arguments

Structure containing:
-Predictions
-Coefficients/Weights
-etc

Machine Library
(classification and regression models)

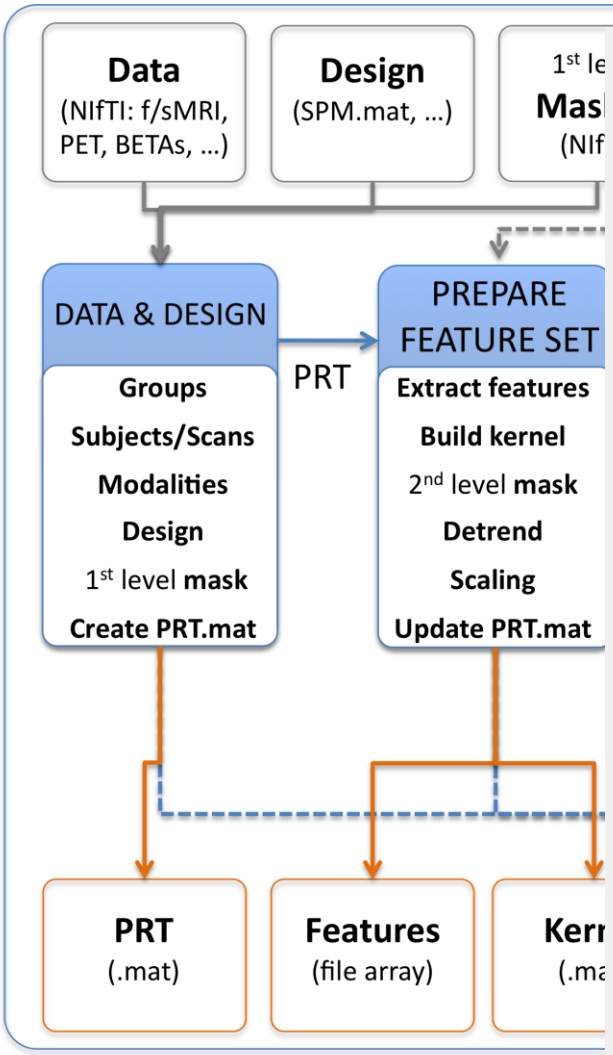
Structure containing:
-Data/Kernel
-Labels
-Etc



Structure containing:
-Predictions
-Coefficients/Weights

PRoNTo

PRoNTo FRAMEWORK




```
>> load('PRT.mat')
>> PRT

PRT =

    group: [1x1 struct]
    masks: [1x1 struct]
       fs: [1x1 struct]
      fas: [1x1 struct]
    model: [1x4 struct]

>> PRT.group

ans =

       gr_name: 'TODOS_172'
       subject: [1x172 struct]
    hrfoverlap: 0
      hrfdelay: 0

>> PRT.group.subject(1).modality(1).mod_name

ans =

T1
```

PRT	group	gr_name				
		subject	subj_name()			
			modality()	mod_name		
				TR		
				scans		
				design	conds	cond_name()
						onsets()
						durations()
						rt_trial()
						scans()
						blocks()
						discardedscans()
						hrfdiscardedscans()
					stats	overlap
						goodscans
						discscans
						meanovl
						stdovl
						mgoodovl
						sgoodovl
						goodovl
					TR	
					unit	
					covar	

PRoNTo

PRT	mask	mod_name		
		fname		
	fs	fs_name		
		k_file		
		id_col_names		
		fas	im	
			ifa	
		modality	mod_name	
			Detrend	
			param_dt	
			mode	
			idfeat_fas	
			normalise	type
				scaling
		id_mat		

PRoNTo

fas	mod_name	
	Dat	
	Detrend	
	param_dt	
	hdr	fname
		dim
		mat
		pinfo
		dt
		n
		descrip
		private
	idfeat_img	

PRT	model	model_name()					
		input()	use_kernel				
			type				
			machine	function			
				Args			
			class	class_name			
				group()	gr_name()		
					subj	num()	
						modality	
			fs	fs_name			
			samp_idx				
			Targets				
			targ_allscans				
			cv_mat				
			operations				
			cv_type				
		output()	fold	targets()			
				predictions()			
				stats()	con_mat		
					acc		
					c_acc		
					b_acc		
					acc_lb		
					acc-_ub		
				func_val()			
				type()			
				alpha()			
				b()			
				totalSV()			
			stats	con_mat			
				acc			
				c_acc			
				b_acc			
				acc_lb			
				acc_ub			

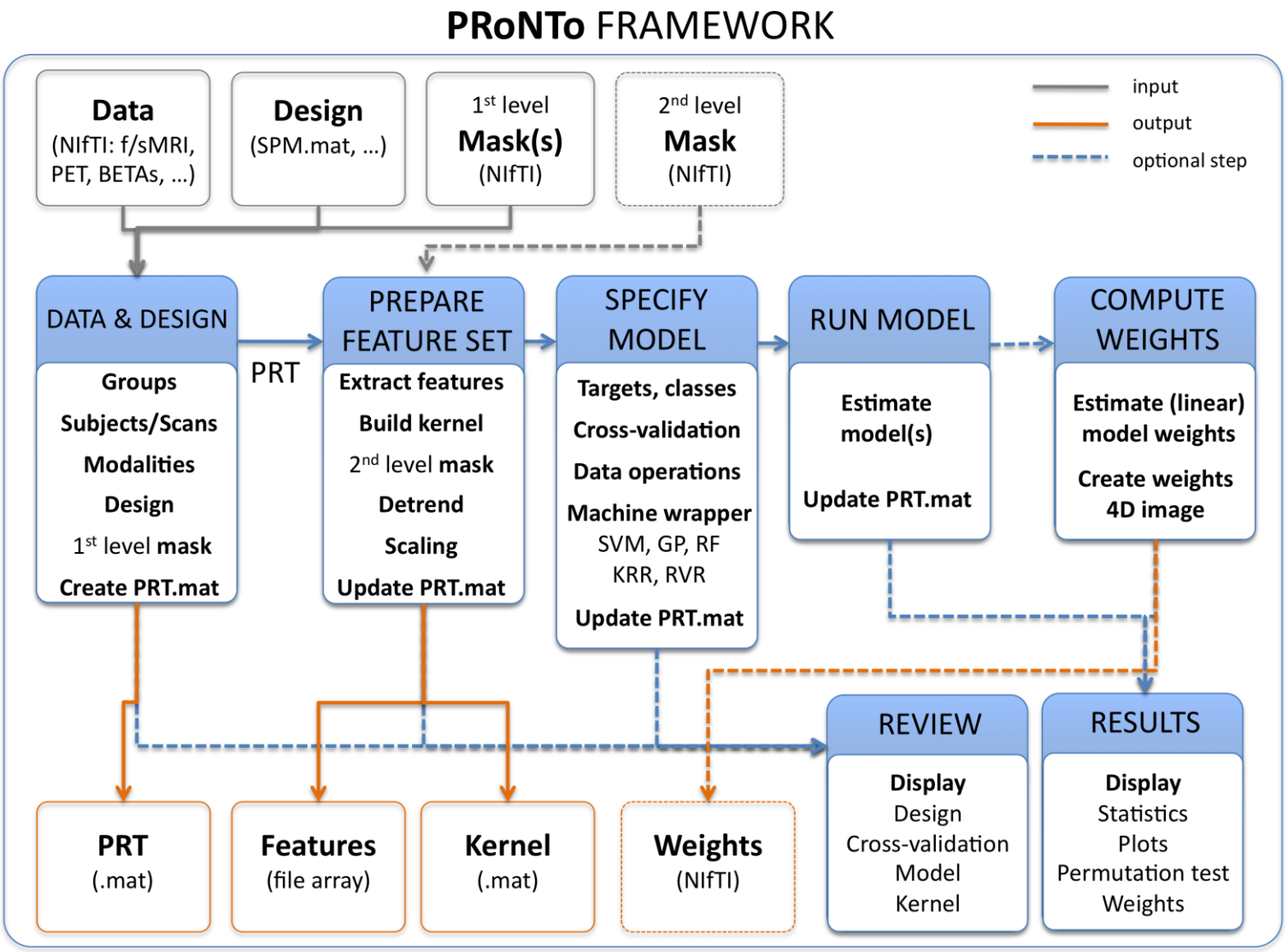
Data format:

- Input images in NIfTI (Neuroimaging Informatics Technology Initiative), http://nifti.nimh.nih.gov/

- Other format could be supported but this would require coding the I/O routine…

Note: sometimes it's simpler to turn your data into a NIfTI 'image'

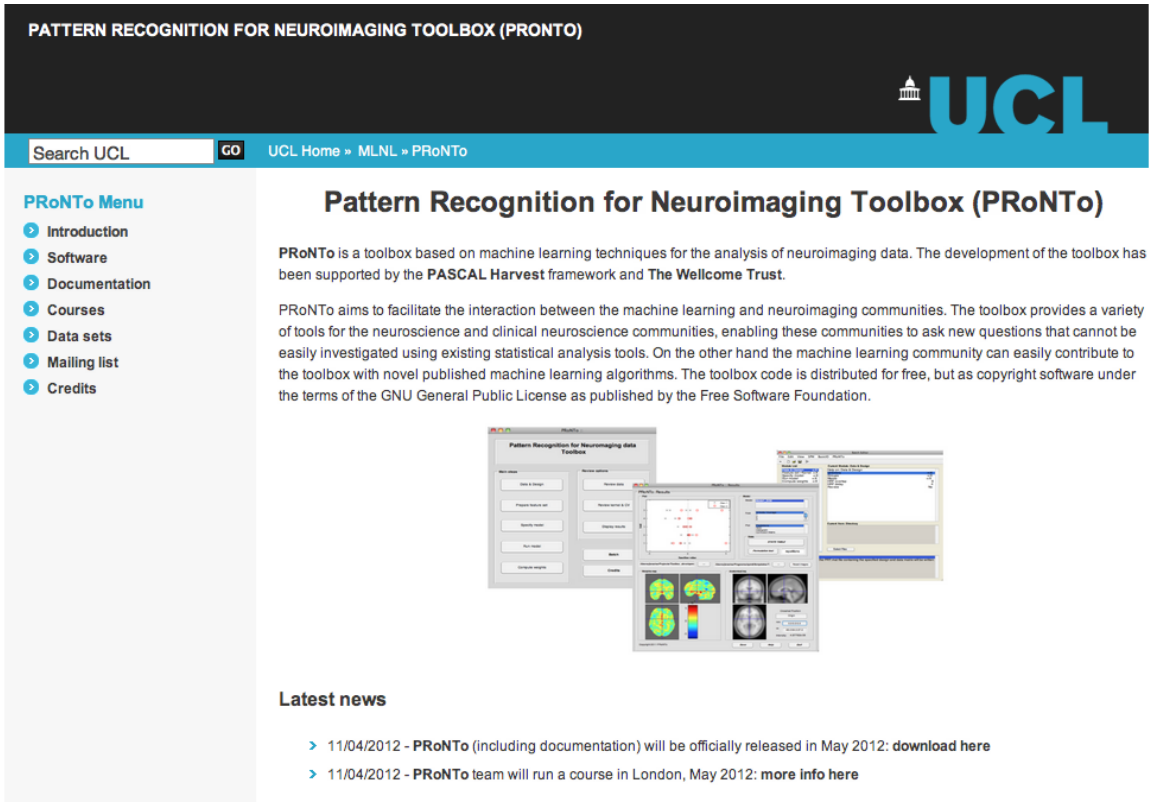
PRoNTo


PRoNTo FRAMEWORK
Data
(NIfTI: f/sMRI, PET, BETAs, ...)
Design
(SPM.mat, ...)
1st level
Mask(s)
(NIfTI)
2nd level
Mask
(NIfTI)
input
output
optional step
DATA & DESIGN
Groups
Subjects/Scans
Modalities
Design
1st level mask
Create PRT.mat
PRT
PREPARE FEATURE SET
Extract features
Build kernel
2nd level mask
Detrend
Scaling
Update PRT.mat
SPECIFY MODEL
Targets, classes
Cross-validation
Data operations
Machine wrapper
SVM, GP, RF
KRR, RVR
Update PRT.mat
RUN MODEL
Estimate model(s)
Update PRT.mat
COMPUTE WEIGHTS
Estimate (linear) model weights
Create weights 4D image
REVIEW
Display
Design
Cross-validation
Model
Kernel
RESULTS
Display
Statistics
Plots
Permutation test
Weights
PRT
(.mat)
Features
(file array)
Kernel
(.mat)
Weights
(NIfTI)

Download

Available here:

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



Credits

The development of PRoNTo was possible with the financial and logistic support of:

- PASCAL Harvest Programme (http://www.pascal-network.org/)
- the Department of Computer Science, University College London (http://www.cs.ucl.ac.uk);
- the Wellcome Trust;
- PASCAL2 (http://www.pascal-network.org/);
- the Fonds de la Recherche Scientifique-FNRS, Belgium (http://www.fnrs.be);
- The Foundation for Science and Technology, Portugal (http://www.fct.pt);
- Swiss National Science Foundation (PP00P2-123438) and Center for Biomedical Imaging (CIBM) of the EPFL and Universities and Hospitals of Lausanne and Geneva.

PRoNTo


PRT.group(g).subject(s).modality(m)
PRT.mask(m)
PRT.fs(f)
PRT.model
PRT.fmris
PRT.kernel(k)
TO DO LIST
- Multi-subject (Probrd)
- Betos
- SPM.mat
- sMRI
classif/regression

PRoNTo