Schizophrenia classification using multi-scale functional connectivity

Christian Dansereau

Abstract—The objective of Project 2 is to explore if feature selection based on a margin maximisation criteria would improve the performance of a classification model applied on neuro-imaging data. During the project we have explored normalization procedure and the proper pipeline to identify the right model and parameters.

Keywords—Schizophrenia, classification, multiscale, feature selection.

1 Introduction

The use of machine learning algorithm in the context of neuroimaging is new and pose multiple constrains, like in any other biomedical dataset we are faced with a large number of features and a small number of examples (the number of subjects). Support vector machine (SVM) [1] are attempting to find an hyperplane that will maximize the margin between two classes. To do so SVM use support vectors that are basically representative example. In this paper we are attempting to discriminate two population control subject and patient with Schizophrenia. In order to do so we are recording the brain activity of each individual using a functional magnetic resonance imaging (fMRI) machine. This modality give use a 3D snapshot of the brain activity every 2.5 second. A popular metric to evaluate brain activity pathern is to use connectivity metrics like the temporal correlation of every pair of grey matter region in the brain.

1.1 Objectives

The objectives of the projects are the following: 1) Normalize the data and account for confounding variables; 2) find if there is a functional scale more optimal for classification of the problem at hand; 3) Evaluate the performance of the retained pipeline for in a multi-scale analysis and assess if a boost in performance can be achieved by combining multiple functional scales using a variation of the bagging algorithm; 4) and finality implement and use a margin optimization algorithm for feature selection applied to functional neuro-imaging data.

• *Mr. Dansereau is with the Department of Computer science, University of Montreal, Montreal, CA.*
E-mail: christiandansereau@gmail.com

Manuscript received April 28, 2015

1.2 Public code and data

The code used in this experiment is available on a GitHub repository at the following URL: https://github.com/cdansereau/vision_or/code_project2. A IPython Notebook is also provided with all of the figure generation scripts.

2 Method

The project was realized in python with the following libraries: scipy [2], scikit-learn [3] and matplotlib [4].

2.1 Dataset

The dataset used in this paper is the COBRE (The Center for Biomedical Research Excellence) dataset from the indi initiative. It consist of raw anatomical and functional MR data from 72 patients with Schizophrenia and 75 healthy controls (ages ranging from 18 to 65 in each group). All subjects were screened and excluded if they had; history of neurological disorder, history of mental retardation, history

of severe head trauma with more than 5 minutes loss of consciousness, history of substance abuse or dependence within the last 12 months. Diagnostic information was collected using the Structured Clinical Interview used for DSM Disorders (SCID).

A multi-echo MPRAGE (MEMPR) sequence was used with the following parameters: TR/TE/TI = 2530/[1.64, 3.5, 5.36, 7.22, 9.08]/900 ms, flip angle = 7, FOV = 256x256 mm, Slab thickness = 176 mm, Matrix = 256x256x176, Voxel size =1x1x1 mm, Number of echos = 5, Pixel bandwidth =650 Hz, Total scan time = 6 min. With 5 echoes, the TR, TI and time to encode partitions for the MEMPR are similar to that of a conventional MPRAGE, resulting in similar GM/WM/CSF contrast.

Rest data was collected with single-shot full k-space echo-planar imaging (EPI) with ramp sampling correction using the intercomissural line (AC-PC) as a reference (TR: 2 s, TE: 29 ms, matrix size: 64x64, 32 slices, voxel size: 3x3x4 mm3).

Phenotypic data available for every participant: gender, age, handedness and diagnostic information (control subject of patient with Schizophrenia). An official description of the data and a publicly available version of the dataset is available on the NITRC [1] website.

2.2 Preprocessing and feature extraction

The datasets were analysed using the NeuroImaging Analysis Kit (NIAK[2]) version 0.12.14, under CentOS version 6.3 with Octave[3] version 3.8.1 and the Minc toolkit[4] version 0.3.18. Analyses were executed in parallel on the "Mammouth" supercomputer[5], using the pipeline system for Octave and Matlab [5], version 1.0.2. Brain map visualizations were created using MRICron software [6]. Each fMRI dataset was corrected of inter-slice difference in acquisition time and the parameters of a rigid-body motion was estimated for each time frame. Rigid-body motion was estimated within as well as between runs, using the median volume of the first run as a target. The median volume of one selected fMRI run for each subject was coregistered with a T1 individual scan using Minctracc [7], which was itself non-linearly transformed to the Montreal Neurological Institute (MNI) template [8] using the CIVET pipeline [9]. The MNI symmetric template was generated from the ICBM152 sample of 152 young adults, after 40 iterations of non-linear coregistration. The rigid-body transform, fMRI-to-T1 transform and T1-to-stereotaxic transform were all combined, and the functional volumes were resampled in the MNI space at a 3 mm isotropic resolution. The a censoring method described in [10] called "scrubbing" was used to remove the volumes with excessive motion using a cut-off value of $FD \geq 0.5$. A minimum number of 50 unscrubbed volumes per run, corresponding to ~ 125 s of acquisition for a TR of 2.5 seconds, was then required for further analysis. The following nuisance parameters were regressed out from the time series at each voxel: slow time drifts (basis of discrete cosines with a 0.01 Hz high-pass cut-off), average signals in conservative masks of the white matter and the lateral ventricles as well as the first principal components (95% energy) of the six rigid-body motion parameters and their squares [11],[12]. The fMRI volumes were finally spatially smoothed with a 6 mm isotropic Gaussian blurring kernel.

Functional connectivity matrices were obtained from 9 scales using a functional template based on an independent dataset of ~ 200 subjects form the 1000 functional connectome project (Cambridge dataset). The 9 scales were obtain using the BASC pipeline [13] which is a unsupervised bootstrap clustering procedure for automatic detection of functional scales based on a stability criteria. resulting in 9 partitions of the brain in 7, 12, 20, 36, 64, 122, 197, 325, 444 networks see Figure 1 for an example of the the resulting connectome of one subject.

2.3 Calibrating the problem

Has a first step we need to assess the difficulty of the classification task at hand and to do so

1. http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html
2. http://www.nitrc.org/projects/niak/
3. http://gnu.octave.org
4. http://www.bic.mni.mcgill.ca/ServicesSoftware/ServicesSoftwareMincToolKit
5. http://www.calculquebec.ca/index.php/en/resources/compute-servers/mammouth-parallele-ii

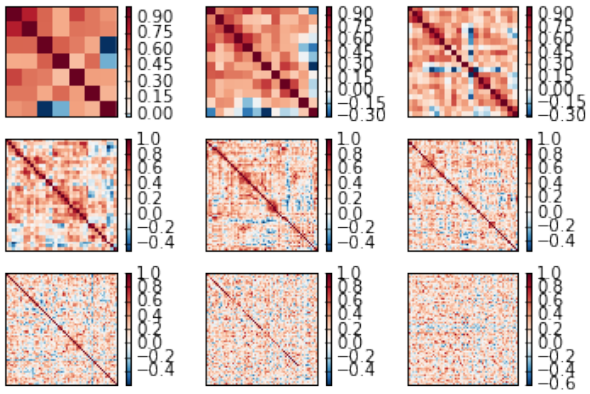

Fig. 1. Exemple of the connectomes for one subject at scale 7, 12, 20, 36, 64, 122, 197, 325 and 444.

we are throwing our problem to a simple off the shelf model with the default parameter. This will define the complexity of the problem. In our case we have used scales 64 which will give us a good reference point in term of the problem complexity. The initial classifier model used was a linear SVM with parameter fixed to $C = 1$ (the default value). In all the subsequent analysis we are performing a stratified 10-fold cross-validation and include a parameter in the SVM classifier to account for the unbalance dataset by automatically re-weight each class by the inverse of its frequency.

The next step is to find the most adapted hyper-parameters (C and/or Gamma if using Gaussian kernel) for our classification problem we choose to use a grid search approach. The grid search parametrised as follow for C (10^-2 to 10^3) and for Gamma (10^-5 to 10^1) The grid search and the final classification were evaluated using a stratified cross-validation due to the unbalanced number of examples available for each class.

2.4 Preprocessing and confounds regression

SVM is sensitive to not normalized data we therefore normalize the features of each subject to zero mean and unit variance. Since some bias can be introduce by confounding factors we account for them by regressing the age and gender contribution based on the training set.

2.5 Optimal functional scale

The fact that we are looking at various level of data abstraction due to the clustering process of BASC in functional scales, we may be more sensitive at some particular scale than other for a given pathology. We have therefore tested the model for each of the 9 scales.

2.6 Multiscale bagging predictions

The idea in this case was to combine the vote of the classifier at each scale using a bagging approach. This ensemble approach combine the predictor of each scale in a bagging procedure as illustrated in Figure 2. We simple take a majority vote from all the classifier. Contrary to most ensemble methods this case may be sensitive to scale that do not have sufficient information to yield a good prediction we therefore took a subset of the scales based on the individual performance obtain in Figure 6.

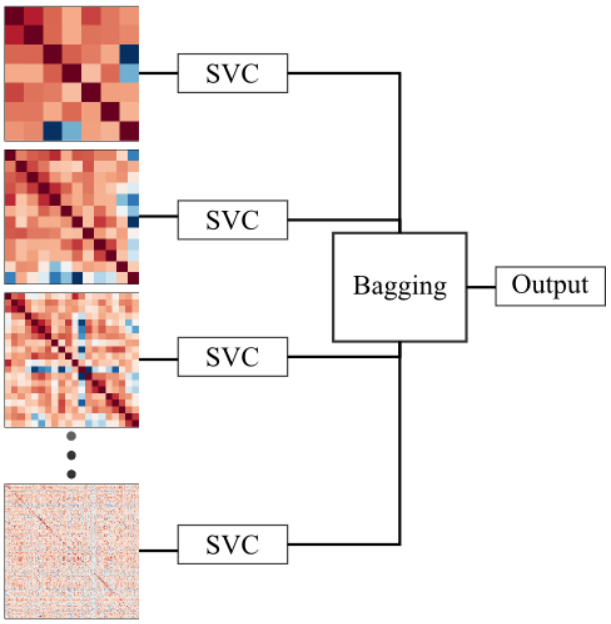

Fig. 2. Pipeline of the modified bagging procedure using multi-scale functional connectomes.

2.7 I-Relief margin optimization

[14] describe feature selection as "the task of choosing a small set out of a given set of features that capture the relevant properties of the data". In our supervised classification problems the relevance is determined by the labels on

the training data. The choice of the features is therefore essential to obtain a compact and accurate classifier. When using an SVM classifier an intuitive question may arise regarding the objective of a support vector machine, it attempt to find the hyperplane that maximize the margin between two classes in a binary task and we can hypothesis that it may exist a subset of feature that optimally discriminate the two classes. If such a subset exist it would 1) be very informative in term of the clinical interpretation and 2) it would potentially require less support vectors and therefore less examples. In order to measure the quality of sets of features a greedy algorithm proposed by [14] and illustrated in Algorithm 1 do iteratively a search of the optimal set of feature by adding and/or removing a feature to/from the subset.

Init set of chosen features $F = \emptyset$
for $t = 1, 2, ...$ **do**
  pick a random permutation s of $1...N$
  for $i = 1$ to N **do**
    $e_1 = e(F \cup s(i))$
    $e_2 = e(F \setminus s(i))$
    if $e_1 > e_2$ **then**
      $F = F \cup s(i)$
    else
      $F = F \setminus s(i)$
    end if
  end for
  if no change made in last step then break
end for

Algorithm 1: Greedy feature flip

Unfortunately the greedy version is not practical for problems with a large feature space like ours we therefore need an alternative. To our knowledge two papers have proposed an online iterative algorithm [14] and [15] that aim at maximizing the margin. We have used an implementation of I-Relief from the package [16] originally proposed by [15] in order to evaluate the potential of this technique.

This procedure give use a weight for each feature we then select the feature with the largest weights. since we do not know the number of features to select we propose to search for the best set based on a nested cross-validation with the criteria to select the best based on how much there weight diverge from the average weight. We search on an standard deviation range of $\alpha \in 0...2.75$ with a step of 0.25. The threshold that we will use to to include all the features that obtained a weight greater or equal to $W_{threshold}$ is defined as $W_{threshold} = W_{std} * \alpha + \overline{W}$.

3 Results

3.1 Calibrating the problem

The figure 3 show the performance of the off the shelf model and we obtained 64.53% with an AUC of 0.70.

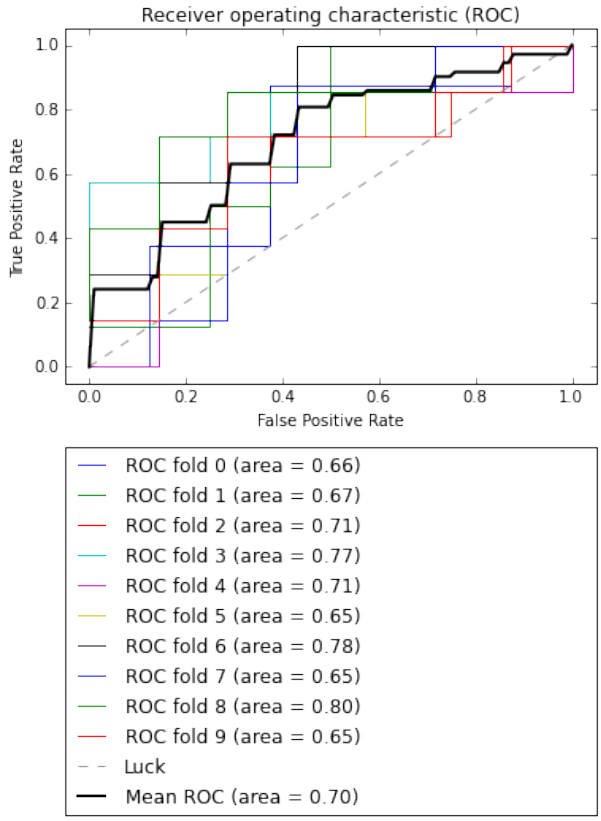

Fig. 3. Calibration of the classification task: Receiver operating characteristic (ROC) curve of the 10 fold cross validation with the average of ROC curve. The legend show the respective AUC (area under the curve) of each fold and the average AUC. The accuracy obtained was 64.53%.

3.2 Normalization and confound variables

When we combine the regression of confounds (age and gender) with the normalization (unit

variance and zero mean for the connectivity of each subject) we have an accuracy of 67.07% and an AUC of 0.75, Figure 4 show the average ROC curve from the 10-fold cross-validation.

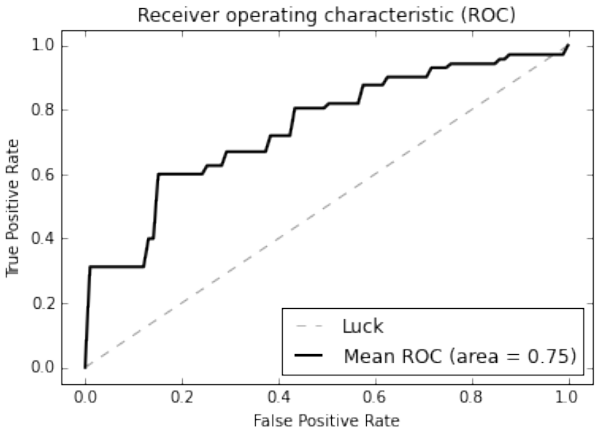

Fig. 4. Receiver operating characteristic (ROC) curve of the average 10-fold cross-validation. The legend show the expected AUC (area under the curve) obtained by luck and the average AUC from the classifier. 67.07% accuracy of the classifier after normalization and regression of the confounding variables.

If we combine the confound regression with the normalization (unit variance and zero mean for the connectivity of each subject) and we search for the optimal parameter C for the linear classifier we obtain 69.89% and an AUC of 0.80 see Figure 5.

3.3 Optimal scale

By selecting the most discriminative scale we where able to achieve a 79.48% accuracy and a AUC of 0.82 as shown in Figure 7.

3.4 Multiscale

Using the 3 scales that individually performed the best (namely scale 122, 197 and 444) we obtained an accuracy of 80.14% std $\pm 8.36\%$ and an AUC of 0.82 see Figure 8 for the ROC curve.

3.5 I-Relief

We obtained an accuracy of 73.94% std $\pm 7.41\%$ with an AUC of 0.82 for the optimized pipeline with feature selection based on I-Relief with optimal threshold.

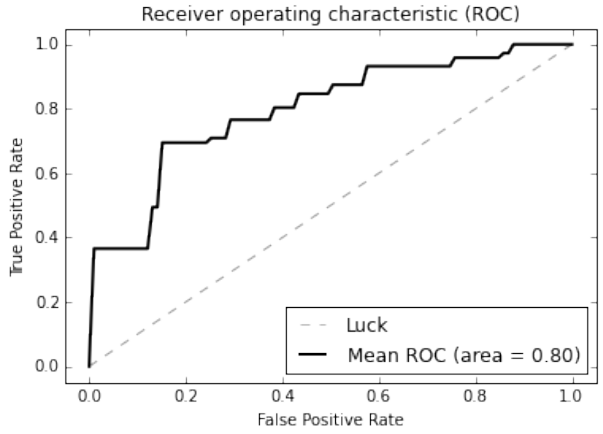

Fig. 5. Receiver operating characteristic (ROC) curve of the average 10-fold cross-validation. The legend show the expected AUC (area under the curve) obtained by luck and the average AUC from the classifier. 69.89% accuracy of the classifier after normalization and regression of the confounding variables and parameter C optimized.

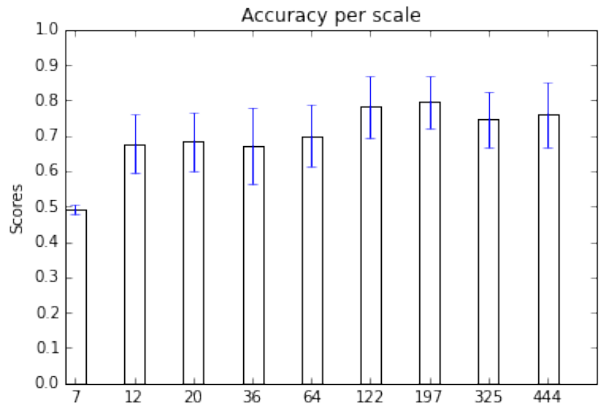

Fig. 6. Accuracy scores of the tuned classifier of each functional scale using a 10-fold cross validation.

3.6 Summary

Table 3.6 show the results of the experiment described above.

4 Discussion and Conclusion

A common mistake made when trying to tackle this kind of problem is to do the feature selection before cross-validation using the labels this may lead in an overestimate of our accuracy. This finding may give very good result that

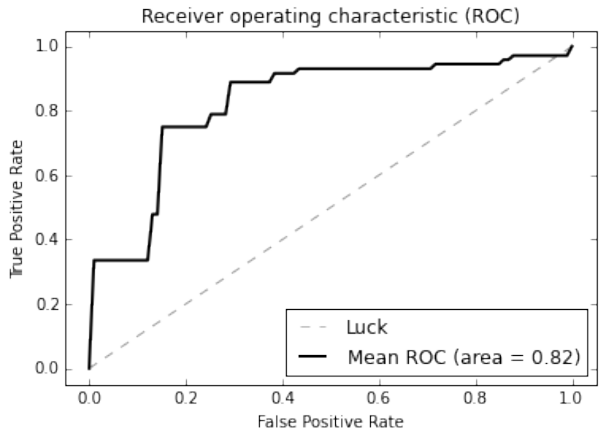

Fig. 7. Receiver operating characteristic (ROC) curve of the average 10-fold cross-validation. The legend show the expected AUC (area under the curve) obtained by luck and the average AUC from the classifier. 79.48% accuracy of the classifier using features from functional scale 197 after normalization and regression of the confounding variables and parameter C optimized using grid search.

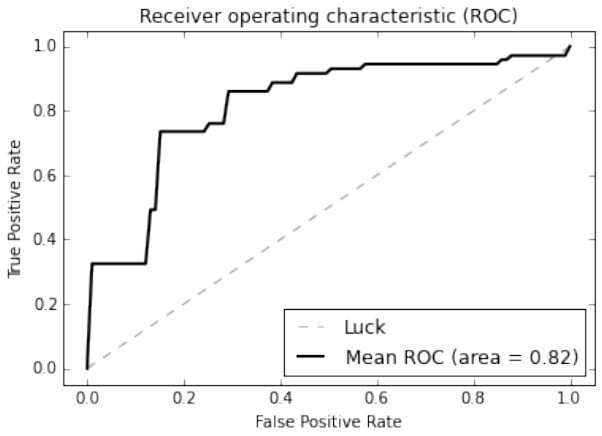

Fig. 8. Receiver operating characteristic (ROC) curve of the average 10-fold cross-validation. The legend show the expected AUC (area under the curve) obtained by luck and the average AUC from the classifier. 80.14% accuracy of the classifier using multiscale bagging from scale 122, 197, 444; after normalization and regression of the confounding variables and parameter C optimized using grid search.

	Accuracy (%)	Std (%)	AUC
SVC linear calib 64x64	64.53	6.86	0.70
NC SVC linear 64x64	67.07	11.08	0.75
Opt NC SVC linear 64x64	69.89	8.69	0.80
Opt NC SVC linear 197x197	**79.48**	**7.50**	**0.82**
Opt NC SVC rbf 197x197	74.61	8.75	0.80
Opt NC Multiscale bagging	**80.14**	**8.36**	**0.82**
Opt NC I-Relief 197x197	73.94	7.41	0.82

TABLE 1
Summary of the performance of each classification pipeline. Acronyms: calib: calibration, NC: normalized and regression of confounds (age and gender), Opt optimisation of the classification parameter using nested 10-fold cross-validation. The multiscale bagging was performed on 3 scales (122, 197 and 444) and I-Relief was perform on the scale 197x197.

will not generalize well as an example we have done this by running our model with the feature selection procedure (I-Relief) in and out of the cross-validation procedure resulting in improved performance when selecting the features outside the cross-validation loop.

In this study, although not shown in the results section we have tested many other classifier models (SVM with Gaussian kernel, LDA, adaboost, bagging, trees and random forest) none of which perform close to the linear SVM except maybe SVM with Gaussian kernel who obtain 74.61% accuracy on the best scale (197) using grid search for the optimal parameters. Our hypotheses for the reason why linear SVM work better than more complex classifier is probably due to the fact that the amount of variability in the data make it difficult for higher order classifier to model. They are more sensitive to outliers and start to focus on noise components that in turn corrupt the classification procedure.

We were able to achieve a decent performance of 79.48% accuracy using a simple model at the appropriate scale and obtain a 80.14% accuracy using the multiscale bagging approach. Unfortunately the I-Relief procedure that we proposed to reduce the feature space did not yield as good results as we were anticipating. It was in fact performing worse then the optimal scale using a linear SVM. For the case of the bagging multiscale it may be good to use this procedure instead of the single scale since it as a very similar accuracy but may be more versatile when presented with

a different pathology. A major advantage is the fact that we do not need to choose the best scale and it use the power of ensemble methods for it's prediction and therefore may perform better than the single scale in some cases. Future work need to be conducted to assess the performance and generalization potential of the leading models to other datasets and pathologies.

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[2] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed 2015-03-15]. [Online]. Available: http://www.scipy.org/

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[5] P. Bellec, F. Carbonell, V. Perlbarg, and A. C. Evans, *A neuroimaging analysis kit for Octave and Matlab*, Accessed August 30th, 2010, 2010. [Online]. Available: http://code.google.com/p/niak/

[6] C. Rorden, H.-O. Karnath, and L. Bonilha, "Improving lesion-symptom mapping." *J Cogn Neurosci*, vol. 19, no. 7, pp. 1081–1088, Jul 2007. [Online]. Available: http://dx.doi.org/10.1162/jocn.2007.19.7.1081

[7] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans, "Design and construction of a realistic digital brain phantom." *IEEE Trans Med Imaging*, vol. 17, no. 3, pp. 463–468, Jun. 1998. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/9735909

[8] V. Fonov, A. C. Evans, K. Botteron, C. R. Almli, R. C. McKinstry, D. L. Collins, and Brain Development Cooperative Group, "Unbiased average age-appropriate atlases for pediatric studies." *NeuroImage*, vol. 54, no. 1, pp. 313–327, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.neuroimage.2010.07.033

[9] A. P. Zijdenbos, R. Forghani, and A. C. Evans, "Automatic "pipeline" analysis of 3-D MRI data for clinical trials: application to multiple sclerosis," *IEEE Transactions on Medical Imaging*, vol. 21, no. 10, pp. 1280–1291, Oct. 2002. [Online]. Available: http://dx.doi.org/10.1109/TMI.2002.806283

[10] J. D. Power, K. A. Barnes, A. Z. Snyder, B. L. Schlaggar, and S. E. Petersen, "Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion." *NeuroImage*, vol. 59, no. 3, pp. 2142–2154, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.neuroimage.2011.10.018

[11] T. E. Lund, K. H. Madsen, K. Sidaros, W.-L. Luo, and T. E. Nichols, "Non-white noise in fMRI: does modelling have an impact?" *NeuroImage*, vol. 29, no. 1, pp. 54–66, Jan. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.neuroimage.2005.07.005

[12] F. Giove, T. Gili, V. Iacovella, E. Macaluso, and B. Maraviglia, "Images-based suppression of unwanted global signals in resting-state functional connectivity studies." *Magnetic resonance imaging*, vol. 27, no. 8, pp. 1058–1064, Oct. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.mri.2009.06.004

[13] P. Bellec and A. C. Evans, "A pipeline system for Octave and Matlab," 2010.

[14] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection - theory and algorithms," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 43–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015352

[15] Y. Sun, "Iterative relief for feature weighting: Algorithms, theories, and applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1035–1051, June 2007.

[16] M. Hanke, Y. Halchenko, P. Sederberg, S. Hanson, J. Haxby, and S. Pollmann, "Pymvpa: a python toolbox for multivariate pattern analysis of fmri data," *Neuroinformatics*, vol. 7, no. 1, pp. 37–53, 2009. [Online]. Available: http://dx.doi.org/10.1007/s12021-008-9041-y