HAL open science

Fast brain decoding with random sampling and random projections

Andrés Hoyos-Idrobo, Gaël Varoquaux, Bertrand Thirion

To cite this version:

Andrés Hoyos-Idrobo, Gaël Varoquaux, Bertrand Thirion. Fast brain decoding with random sampling and random projections. PRNI 2016: the 6th International Workshop on Pattern Recognition in Neuroimaging, Jun 2016, Trento, Italy. ⟨hal-01313814⟩

HAL Id: hal-01313814

https://inria.hal.science/hal-01313814v1

Submitted on 10 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Autorisation HAL

HAL Authorization

Fast brain decoding with random sampling and random projections

Andrés HOYOS-IDROBO*, Gaël VAROQUAUX*, Bertrand THIRION*,
* Parietal team, INRIA, CEA, University Paris-Saclay, 91191 Gif sur Yvette, France
firstname.lastname@inria.fr

***Abstract*—Machine learning from brain images is a central tool for image-based diagnosis and diseases characterization. Predicting behavior from functional imaging, brain *decoding*, analyzes brain activity in terms of the behavior that it implies. While these multivariate techniques are becoming standard brain mapping tools, like mass-univariate analysis, they entail much larger computational costs. In an time of growing data sizes, with larger cohorts and higher-resolutions imaging, this cost is increasingly a burden. Here we consider the use of random sampling and projections as fast data approximation techniques for brain images. We evaluate their prediction accuracy and computation time on various datasets and discrimination tasks. We show that the weight maps obtained after random sampling are highly consistent with those obtained with the whole feature space, while having a fair prediction performance. Altogether, we present the practical advantage of random sampling methods in neuroimaging, showing a simple way to embed back the reduced coefficients, with only a small loss of information.**

***Index Terms*—Nyström; High-dimensional estimators; machine learning; brain imaging**

I. Introduction

Decoding uses predictive models to link brain regions with an experimental condition or a behavior. It has become a central tool in neuroimage [1]. In particular, linear estimators can highlight the brain maps that lead to the identification of cognitive labels [2][3]. Yet, to date, decoding is still orders of magnitude slower than standard analysis. This discourages the use of non-parametric hypothesis testing (e.g. permutation testing). Additionally, large cohorts are needed to fully tap the potential of decoding, increasing both power and reliability in group studies. A striking example is the Human Connectome Project [4] (30 Terabytes of data and growing).

However, increasing data sizes pose tractability challenges for all processing steps, especially when using multivariate statistics: multivariate estimators entail high computation costs. The literature of machine learning on massive datasets often relies on dimension reductions to mitigate the impact of data size on computational cost. Approximating the data matrix via a *sketch matrix* [5] is one possible technique. It can be used to render tractable in the large-data limit a model like PCA, which has a computational cost that grows super-linearly.

A variety of other approaches build an approximation of the data matrix. *i)* A small subset of rows or columns (samples or features) can approximate the entire matrix [6](e.g. Nyström [7], CUR decomposition). *ii)* Random combination of matrix columns, relying on subspace embedding techniques, give strong concentration phenomena (random projections)[8].

Our contribution: Here we evaluate empirically the difference between random sampling and projections as a strategy of dimension reduction to decode brain images. Additionally, we show that random sampling can be used to approximate the weight maps of a linear estimator in the brain space. Finally, we show the benefit of using these methods to reduce the computation time of statistical analysis of massive datasets.

Notations: Vectors are written using bold lower-case, e.g. $\mathbf{x}$. Matrices are written using bold capital letters, e.g. $\mathbf{X}$.

II. Methods: Random sampling and projections

Given the paired data samples $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, where each $\mathbf{x}_i \in \mathbb{R}^p$ is a brain image (i.e. predictor variable) and each $\mathbf{y}_i \in \mathbb{R}$ is the behavioral/categorical variable to be fit (i.e. the target). The goal is to estimate a function that can be used to predict future responses based on observing only brain images. Henceforth, the data are represented as a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, n observed brain images composed of p voxels.

Kernel-based methods: Here, we give a brief reminder about kernels. A kernel is a function that quantifies the similarity of two observations (e.g. pairwise distance between brain images). Kernel-based methods use a feature mapping $\mathbf{\Phi}$ to reveal the discriminant informations in a high-dimensional space $\mathcal{F}$. In brief, a kernel-method pipeline is: *i)* embedding the data $\mathbf{X}$ into $\mathcal{F}$ using the feature mapping $\mathbf{\Phi}$, and *ii)* performing the estimation (e.g. classification). In neuroimaging, the feature mapping $\mathbf{\Phi} : \mathbb{R}^p \to \mathcal{F}$ is often chosen as linear for interpretability of the weights [2].

Kernel-based methods rely on the idea that inner products in high-dimensional feature spaces can be computed in implicit form via kernel function $\mathbf{K}_{i,j} = \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle$ resulting in the Gram matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$. This is important, because the decision function of many classification algorithms (e.g. SVM and logistic regression) can be carried out just on the basis of the values of the kernel function over pairs of domain points. The kernel function in a linear setting leads to a symmetric positive semidefinite matrix $\mathbf{K}_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$.

Approximating pairwise distances: Random projections reduce the dimension of the input samples $\mathbf{x}$ while incurring a controlled distortion. More formally, to reduce the number of features, the data $\mathbf{x}$ are projected into a k-dimensional random subspace using a random matrix $\mathbf{\Phi}_{\mathsf{RP}} \in \mathbb{R}^{k \times p}$, ensuring that, with high probability, the pairwise distances among a

Algorithm 1 Nyström method: Learning the feature mapping

Require: The training data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, number k of components, where $k < n$.

Ensure: The feature mapping $\mathbf{\Phi}_{\mathsf{Nys}} \in \mathbb{R}^{k \times p}$

1: $\mathbf{r} \leftarrow$ Generate uniform sampling of k components
2: $\mathbf{X_r} \in \mathbb{R}^{k \times p}$ {Subsample of k rows}
3: $\mathbf{K_r} = \mathbf{X_r}\mathbf{X_r}^T$ {Kernel matrix of the subsampled data}
4: $\mathbf{\Phi}_{\mathsf{Nys}} = \mathbf{K_r}^{-1/2}\mathbf{X_r}$ {Normalization}
5: **return** $\mathbf{\Phi}_{\mathsf{Nys}}$

collection n-brain images (i.e. $\mathbf{x}_i$ for $i \in [1, \ldots, n]$) in $\mathbb{R}^p$ are approximately maintained with a distortion at most ϵ [8].

$$(1-\epsilon) \leq \frac{\|\mathbf{\Phi}_{\mathsf{RP}}\ \mathbf{x}_i - \mathbf{\Phi}_{\mathsf{RP}}\ \mathbf{x}_j\|^2}{\|\mathbf{x}_i - \mathbf{x}_j\|^2} \leq (1+\epsilon), \forall (i,j) \in [1, \ldots, n]^2.$$

The matrix $\mathbf{\Phi}_{\mathsf{RP}}$ can be generated by sampling from a given Gaussian distribution with rescaling or, more practically, with binary random variables [9]. This approximation can then be used for further analysis such as kernel-based methods that consider between-sample similarities.

III. Nyström: Approximating brain images

The Nyström method was presented in [10] to speed up kernel-based methods, and has become a standard tool when dealing with large-scale datasets [7]. The idea is to preserve the spectral structure of the kernel matrix $\mathbf{K}$ using a subset of columns of this matrix, yielding a low-rank approximation. This can be cast as building a data-driven feature mapping $\mathbf{\Phi}_{\mathsf{Nys}}$, leading to

$$\mathbf{K}_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle \approx \langle \mathbf{\Phi}_{\mathsf{Nys}}\ \mathbf{x}_i, \mathbf{\Phi}_{\mathsf{Nys}}\ \mathbf{x}_j \rangle .$$

This method is well suited for signal with an underlying structure, like brain images, where the underlying spatial organization (smoothness, network structure) is shared across rows of the data matrix $\mathbf{X}$ and easily captured by $\mathbf{\Phi}_{\mathsf{Nys}}$ after sampling.

The Alg.1 presents the Nyström method, where we build a base, randomly sampling $k \ll p$ rows of $\mathbf{X}$ uniformly, and then we normalize it (i.e. whitening of the subsampled data). In theory, the number k of components should be inversely proportional to the regularity of the signals.

Binging the reduction to the brain space: Given that the feature mapping is normalized, the approximated features can be embedded back into the brain space, using

$$\hat{\mathbf{x}} = \mathbf{\Phi}_{\mathsf{Nys}}^T\ \mathbf{\Phi}_{\mathsf{Nys}}\ \mathbf{x}, \tag{1}$$

This is a direct consequence of $\mathbf{x}$ laying close to $\mathrm{Im}(\mathbf{\Phi}_{\mathsf{Nys}}^T\ \mathbf{\Phi}_{\mathsf{Nys}})$. Indeed, $\mathbf{\Phi}_{\mathsf{Nys}}$ is generated form the data, and hence captures well its structure. This is unlike random projection, that are not tailored to the data and do not lead to good inversions as the data can have a large overlap with the kernel of the projection. Note that using this approach, we can bring back the coefficients of linear estimators to the brain space, and perform further analysis on the resulting maps.

IV. Experiments: Empirical verification

In this section, we investigate decoding after random sampling and random projections. To achieve reliable empirical conclusions, we evaluate the performance across several neuroimaging studies, using both anatomical and functional images. We compare prediction accuracy obtained without compression to that using random projections and Nyström approximation under linear settings. We also quantify and compare the execution time. In all the experiments, $n > 180$ and we split the data into train and test set, changing the proportion of these sets according to the dataset. All dimensionality reduction procedures are calibrated on the train set and used to reduced the test set.

A. Datasets

Haxby [11]: Is a visual object recognition task obtained from 5 subjects. The data consist of 12 sessions, each containing 9 volumes per object category (i.e. face, house, etc) and about $p = 30,000$ voxels per volume. We perform intra-subject discrimination across sessions between various pairs of visual stimuli. The prediction is performed on two left-out sessions.

The Open Access Series of Imaging Studies (OASIS)[12]: This dataset consists of 403 anatomical brain images (Voxel Based Morphometry) of subjects aged between 60 and 96 years old. These images were preprocessed with the SPM8 software to obtain modulated grey matter density maps sampled in the MNI space at 2mm resolution. These images were masked to an average mask of the grey matter, which yields about $p = 140,398$ voxels. We perform across-subject gender discrimination, leaving half of the subjects out to measure the accuracy.

Human Connectome Project (HCP)[4]: We consider a functional Magnetic Resonance Imaging acquired in the HCP This dataset contains 500 participants (13 removed for quality reasons). All of them where unrelated and without psychiatric or neurological history. The primary goal of this dataset is network discovery, which is facilitated by probing experimental task paradigms that are known to tap on well characterized neural networks [13]. We profited from the HCP "minimally preprocessed" pipeline [14] and took images related to 5 tasks: *1)* working memory/cognitive control processing, *2)* incentive processing, *3)* visual and somatosensory-motor processing, *4)* language processing (semantic and phonological processing), *5)* social (theory of mind). We perform across-subject discrimination of 17 experimental conditions selected from the aforementioned task-related datasets.

B. Benchmarking of linear classifiers

We explore well known classifiers in the neuroimaging literature: SVM-ℓ_2 and logistic regression-ℓ_2. Firstly, we analyzed the performance of various standard solvers in the primal and dual space[1]: *i)* for the SVM we use Liblinear and LibSVM,

[1]Note that not all the algorithms are designed to work in both spaces (primal and dual), this is the case of LibSVM which only works on the dual space

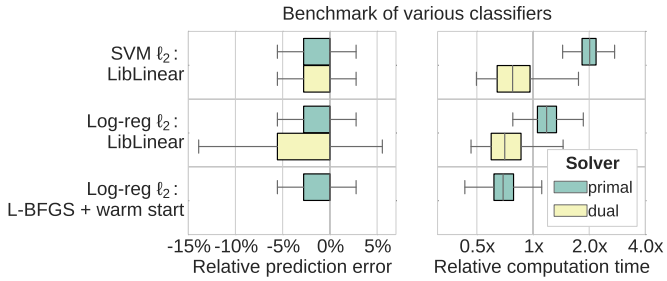

Fig. 1. **Comparison of the performance of various solvers relative to SVM (LibSVM):** Comparison of the performance of the SVM and logistic regression with ℓ_2 penalty on the discrimination of 7 paired visual object recognition tasks for the Haxby dataset, and the discrimination of gender of the OASIS dataset. *(left)* The prediction accuracy of all the classifiers is upper-bounded, they have the same median; *(right)* Regarding the computation time, the logistic regression using L-BFGS with warm start, displays a good trade-off between computation time and accuracy.

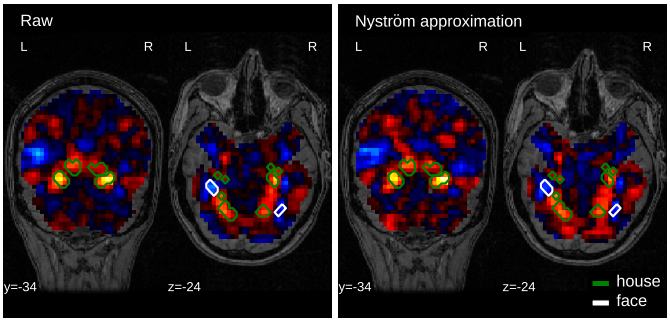

Fig. 2. **Approximation in the brain space:** Weight maps (unthresholded) of a ℓ_2-logistic regression, obtained for the discrimination of face and house on the Haxby dataset. The contours show the FFA (green) and PPA (white) regions, respectively involved in the face and house recognition tasks. These regions are highlighted by the coefficients obtained using two methods: *(left)* the whole feature space, and *(right)* the Nyström method with $k = 100$.

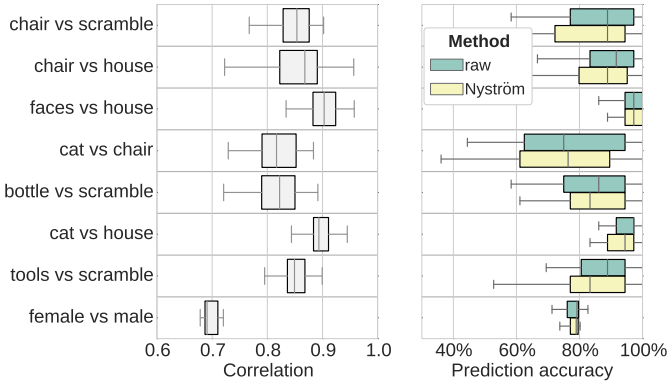

Fig. 3. **Consistency of the discriminative weights after dimensionality reduction:** Discrimination of various conditions using a logistic regression with ℓ_2 penalty. The dimension is reduced to $k = 100$. *(left)* Correlation between coefficients obtained using the raw data and the Nyström method. The weight maps found after approximation are generally consistent with the maps found without reduction; *(right)* Regarding prediction accuracy, Nyström method is comparable with the performance obtained with the raw data.

setting the regularization parameter by inner cross validation. *ii)* For the logistic regression, we use Liblinear with inner cross validation to set the regularization parameter; and L-BGFGS with warm start setting, the parameter via regularization path. To build the confidence interval, we perform a 10-fold cross validation maintaining the proportion between the labels at each iteration. We measure the accuracy[2] obtained on test data and the computation time to train the classifier. We compare the performance of all classifiers with an SVM (LibSVM), which is often used by default.

Fig.1 displays the results of the discrimination of visual objects on the Haxby dataset and the gender prediction on the OASIS dataset. These tasks cover a range from easy to difficult discrimination problems. We can see that the median prediction score is the same for all the classifiers, having an empirical distribution skewed to the right, indicating that the prediction accuracy is often the same. The estimators using a primal solver yield the same distribution.

Regarding computation time, logistic regression-ℓ_2 using L-BFGS with warm start has the best performance: it displays a good trade-off between prediction accuracy and computation time. Henceforth, we refer to this choice simply as logistic regression-ℓ_2, making the solver implicit.

C. Bringing the reduction to the brain space

In this experiment, we show the capability of the Nyström method to approximate the coefficients of a linear estimator in the brain space. We use a logistic regression-ℓ_2 as a classifier, using as a solver a L-BFGS with warm start, to discriminate 7 paired visual objects on the Haxby dataset, and gender on the OASIS dataset. We compare the prediction accuracy and the correlation, between the weight maps obtained using all the voxels with the approximation via Nyström method for $k = 100$.

Fig.2 shows the weight map (hyperplane of discrimination), to discriminate between face and house. We can see that contours show the well-known Fusiform Face Area (FFA) and Parahippocampal Plane Area (PPA) regions, respectively involved in the face and house recognition tasks, are highlighted by the coefficients of the classifier. This is considered as an easy classification task and finding the structure by Nyström approximation only requires a small number k of components ($k > 30$).

Fig.3 shows the consistency of the discriminative weights found after applying the Nyström method and raw data. Nyström displays a high consistency with the raw data, having a correlation score > 0.7 for all the conditions. Regarding the prediction accuracy, we can see that the Nyström method exhibits slightly worst performance than raw.

D. Random sampling, random projections and decoding

Now, we compare the effect of random sampling and projections across different discrimination tasks and datasets. To this end, we use 7 visual object discrimination of the Haxby dataset, gender discrimination of the OASIS dataset, and 17 cognitive tasks of the HCP dataset. We train a binary logistic regression on the first two datasets and multinomial logistic

[2]Accuracy is defined by: $\frac{\text{number of correctly predicted data}}{\text{total number of samples}} \times 100\%$

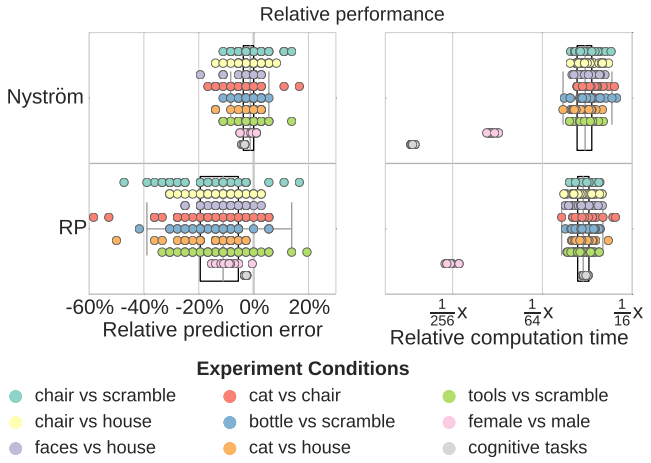

Fig. 4. **Impact of the dimensionality reduction on the prediction performance:** Discrimination using logistic regression of 25 conditions on 3 datasets, after dimensionality is reduction to $k = 100$. *left)* Each point represents the impact of the corresponding dimensionality reduction scheme on the prediction accuracy, relatively to the prediction obtained with raw data. The Nyström approximation method has a better performance than random projections. *right)* In most of the conditions, the time performance of both methods is almost the same, yielding impressive time saving. On the HCP dataset, Nyström is considerably faster.

regression on the last one. We reduce the dimensionality of the feature space form p to k, where k is set to $k = 100$ in this experiment. We use random projections and Nyström method to carry out this task.

The results of the use of random sampling and projections with respect raw data are summarized in Fig.4. We can see that the accuracy of the classifier after dimensionality reduction by Nyström method is close to the one obtained using the whole feature space. This indicates that there is a reliable linear structure underlying the brain images, which is captured by Nyström approximation with only a small number k of components. In contrast, the estimator after random projections shows lower performance. This is because random projections act in the feature direction, needing a larger number k of components to approximate the pairwise distances.

Regarding computational time, both methods have an equal performance in average. Note that using the Nyström method yields impressive time savings on the HCP dataset.

Implementation aspects: We rely on scikit-learn [15] for machine learning tasks (logistic regression and SVM) and on Nilearn to interact with neuroimaging data.

V. Discussion: Decode with random sampling

Our validation over 27 decoding tasks on 3 different datasets, varying from moderate to large size datasets, shows that random sampling overperforms random projections for decoding brain images. The dimensionality reduction by random projections does not take the structure of the signal into account, making it difficult to find an appropriate pseudo-inverse to bring the weight maps back into the brain space. On the other hand, the Nyström method tries to approximate the spectral properties of the data matrix $\mathbf{X}$, relying on a data-driven approach. In this sense, it takes into account the intrinsic structure (e.g. smoothness, latent network structure) being consistent with the feature space and controlling the spatial maps. This leads to an easy scheme to embed the coefficients back, making it possible to perform further analysis.

Regarding computation time, both methods yield impressive speed gains: $>$ 16 times faster. However, the prediction accuracy is not better than that obtained with raw. Indeed, these methods do not separate the signal from the noise.

a) Acknowledgment: The authors acknowledge funding from the EU FP7/2007-2013 under grant agreement 604102 (HBP).

References

[1] A. J. O'Toole, F. Jiang, H. Abdi, N. Pnard, J. P. Dunlop, and M. A. Parent, "Theoretical, statistical, and practical perspectives on pattern-based classification approaches to the analysis of functional neuroimaging data," *J Cogn Neurosci*, vol. 19, 2007.

[2] J. Mouro-Miranda, A. L. Bokde, C. Born, H. Hampel, and M. Stetter, "Classifying brain states and determining the discriminating activation patterns: Support vector machine on functional mri data," *NeuroImage*, vol. 28, 2005.

[3] T. Naselaris, K. N. Kay, S. Nishimoto, and J. L. Gallant, "Encoding and decoding in fmri," *Neuroimage*, vol. 56, pp. 400–410, 2011.

[4] D. V. Essen, K. Ugurbil, E. Auerbach, D. Barch, T. Behrens, R. Bucholz, A. Chang, L. Chen, M. Corbetta, S. Curtiss, S. D. Penna, D. Feinberg, M. Glasser, N. Harel, A. Heath, L. Larson-Prior, D. Marcus, G. Michalareas, S. Moeller, R. Oostenveld, S. Petersen, F. Prior, B. Schlaggar, S. Smith, A. Snyder, J. Xu, and E. Yacoub, "The human connectome project: A data acquisition perspective," *NeuroImage*, vol. 62, 2012.

[5] E. Liberty, "Simple and deterministic matrix sketching," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[6] M. W. Mahoney and P. Drineas, "Cur matrix decompositions for improved data analysis," *PNAS*, vol. 106, 2009.

[7] A. Gittens and M. W. Mahoney, "Revisiting the nystrom method for improved large-scale machine learning." in *ICML*, vol. 28, 2013.

[8] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," in *Conference in modern analysis and probability*, 1984, vol. 26.

[9] D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, 2003.

[10] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," in *NIPS*, 2001, pp. 682–688.

[11] J. V. Haxby, I. M. Gobbini, M. L. Furey *et al.*, "Distributed and overlapping representations of faces and objects in ventral temporal cortex," *Science*, vol. 293, 2001.

[12] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults." *J Cogn Neurosci*, vol. 19, 2007.

[13] D. M. Barch, G. C. Burgess, M. P. Harms, S. E. Petersen, B. L. Schlaggar, M. Corbetta, M. F. Glasser, S. Curtiss, S. Dixit, C. Feldt, D. Nolan, E. Bryant, T. Hartley, O. Footer, J. M. Bjork, R. Poldrack, S. Smith, H. Johansen-Berg, A. Z. Snyder, D. C. V. Essen, and W. U.-M. H. Consortium, "Function in the human connectome: task-fmri and individual differences in behavior." *Neuroimage*, vol. 80, 2013.

[14] M. F. Glasser, S. N. Sotiropoulos, J. A. Wilson, T. S. Coalson, B. Fischl, J. L. Andersson, J. Xu, S. Jbabdi, M. Webster, and J. R. Polimeni, "The minimal preprocessing pipelines for the human connectome project," *Neuroimage*, vol. 80, 2013.

[15] 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," *J. Mach. Learn. Res.*, vol. 12, 2011.