# A MapReduce Approach for Ridge Regression in Neuroimaging-Genetic Studies

Benoit Da Mota[1,3], Michael Eickenberg[1], Soizic Laguitton[2], Vincent Frouin[2],
Gaël Varoquaux[1], Jean-Baptiste Poline[2], and Bertrand Thirion[1]

[1] Parietal Team, INRIA Saclay, France
[2] CEA, DSV, I2BM, Neurospin, France
[3] Parietal Team, MSR-INRIA joint centre, France
{benoit.da_mota,michael.eickenberg,
gael.varoquaux,bertrand.thirion}@inria.fr
{soizic.laguitton,jbpoline}@gmail.com
{vincent.frouin}@cea.fr

**Abstract.** In order to understand the large between-subject variability observed in brain organization and assess factor risks of brain diseases, massive efforts have been made in the last few years to acquire high-dimensional neuroimaging and genetic data on large cohorts of subjects. The statistical analysis of such high-dimensional and complex data is carried out with increasingly sophisticated techniques and represents a great computational challenge. To be fully exploited, the concurrent increase of computational power then requires designing new parallel algorithms. The MapReduce framework coupled with efficient algorithms permits to deliver a scalable analysis tool that deals with high-dimensional data and hundreds of permutations in a few hours. On a real functional MRI dataset, this tool shows promising results.

**Keywords:** Bio-statistics, Neuroimaging, Genetics, Ridge regression, Permutation Tests.

## 1 Introduction

Using genetics information in conjunction with brain imaging data is expected to significantly improve our understanding of both normal and pathological variability of brain organization. It should lead to the development of biomarkers and in the future personalized medicine. Among other important steps, this endeavor requires the development of adapted statistical methods to detect significant associations between the highly heterogeneous variables provided by genotyping and brain imaging, and the development of the software components that will permit large-scale computation to be done.

In current settings, neuroimaging-genetic datasets consist of a set of *i)* genotyping measurements at given genetic loci, such as Single Nucleotide Polymorphisms (SNPs) that represent a large amount of the genetic between-subject variability, on the one hand, and *ii)* quantitative measurements at given locations (voxels) in three-dimensional images, that represent e.g. either the amount of functional activation in response to a certain task or an anatomical feature, such as the density of grey matter in the corresponding brain region.

Most of the efforts so far have been focused on designing association models, and the computational procedures used to run these models on actual architectures have not been considered carefully. For instance, permutation tests of simple linear association models have been deemed as inefficient in some of these studies, e.g. [11]; however, they can be replaced by analytical tests only in very specific cases and under restrictive assumptions. Gains in sensitivity might be provided by multivariate models in which the joint variability of several genetic variables is considered simultaneously. Such models are thought to be more powerful [13, 1, 5, 7], because they can express more complex relationships than simple pairwise association models. The cost of unitary fit is high due to high-dimensional, potentially non-smooth optimization problems and various cross-validation loops needed to optimize the parameters; moreover, permutation testing is necessary to assess the statistical significance of the results of such procedures in the absence of analytical tests. Multivariate statistical methods require thus many efforts to be tractable in this problem on both the algorithmic and implementation side, including the design of adapted dimension reduction schemes. In this work we will consider the simplest approach, ridge regression [5], that is powerful for detecting multivariate associations between large variable sets, but does not enforce sparsity in the solution.

Working in a distributed context is necessary to deal with the memory and computational loads, and yields specific optimization strategies. Once the unitary fit cost has been minimized, the main task when implementing such natural data parallel applications is to choose *how to split the problem into smaller subproblems* to minimize computation, memory consumption and communication overhead. For the first time, we propose an efficient framework that can manage ridge regression with numerous phenotypes and permutations.

In Section 2, we present our sequential algorithm, then we describe our framework to distribute efficiently the computation on large infrastructures. Experimental results on simulated and real data are presented in Section 3.

## 2 Methods : the computational framework

Ridge regression of neuroimaging genetics data is clearly an embarrassingly parallel problem, which can be easily split into smaller tasks. Our computational framework relies on an adapted workflow summarized in Fig. 1, in which subtasks are optimized for the sake of efficiency. To simplify the presentation we first describe the core algorithm and then the workflow.

### 2.1 Optimizing the ridge regression algorithm

The *map* step, i.e. the scoring of ridge classifiers, is the most demanding in computation time ($< 99.9\%$ in our final implementation) and thus has to be optimized in priority. The computational burden mostly depends on the ridge regression step. Our algorithm performs Ridge Regression for multiple targets and multiple individual penalty values. It solves the following problem:

$$\hat{\beta}_{ij} = \text{argmin}_\beta \|y_i - X\beta\|_2^2 + \lambda_{ij}\|\beta\|_2^2,\ i \in [1,p], j \in [1,J]$$

where $X \in \mathbb{R}^{n \times p}$ is the gene data matrix, $y_i \in \mathbb{R}^n$ is a variable extracted from brain images, $\hat{\beta}_{ij} \in \mathbb{R}^p$ is the estimated coefficient vector, and $\lambda_{ij} > 0$ is the penalty term where $j$ indexes $J$ different penalties for the target $y_i$. We obtain the solution using the singular value decomposition (SVD) of $X$, which we write $X = USV^T$, truncated to non-zero singular values. In the full rank case and for $p > n$ we have $U \in \mathbb{R}^{n \times n}$ and $V^T \in \mathbb{R}^{n \times p}$, while $S$ is a diagonal matrix with entries $s_k, 1 \leq k \leq n$. For one $\hat{\beta}_{ij}$ we have

$$\hat{\beta}_{ij} = V \text{diag}_k \left( \frac{s_k}{s_k^2 + \lambda_{ij}} \right) U^T y_i$$

All $\hat{\beta}_{ij}$ are calculated with the same SVD, it is reused (and cached). For all $i$, $U^T y_i$ is pre-calculated, which is conveniently and effectively done by multiplying the matrices $U^T$ and $Y$ where the columns of $Y$ are the $y_i$. Since for a given $j$ every target $i$ potentially has a different penalty associated, the shrinkage operation $\frac{s_k}{s_k^2+\lambda_{ij}}$ is not writable as a matrix multiplication against $U^T Y$. However, it is a linear operation on matrices, and by defining $\varSigma \in \mathbb{R}^{n \times p}$ with $\varSigma_{ki} = \frac{s_k}{s_k^2+\lambda_{ij}}$ for a fixed $j$, it can be written as the pointwise matrix product

$$\hat{\beta} = V(\varSigma \circ U^T Y).$$

These are the operations implemented by our algorithm for the $J$ different sets of penalties, using $J$ different matrices $\varSigma$. With a pre-calculated SVD and $U^T Y$, the cost of this operation is $\mathcal{O}(npN)$, where $N$ is the number of target variables.

Care was taken of computational/hardware sources of optimization, like CPU cache issues. For instance, matrix-based operations are used instead of vector-based operations to optimize the use of advanced vector extensions instructions set in new CPU. Our Python code uses the Numpy/Scipy/Scikit-learn scientific libraries, which rely on standard and optimized linear algebra libraries (Atlas or MKL) that are several order of magnitude faster than naive code. Next, we need to consider evaluation and parameter setting procedures:

- the power of the procedure is measured by the ratio of explained variance, computed within a *shuffle-split* loop that leaves 20% of the data as a test set at each of the ten iterations;
- to select the optimal shrinkage parameters, $J = 5$ values are tested first, then a grid refinement is performed where five other parameters are tested ;
- each shrinkage parameter of the ridge regression is evaluated using an inner 5-folds cross validation loop.

This setting thus needs approximately 500 ridge regressions for one phenotype and one permutation.

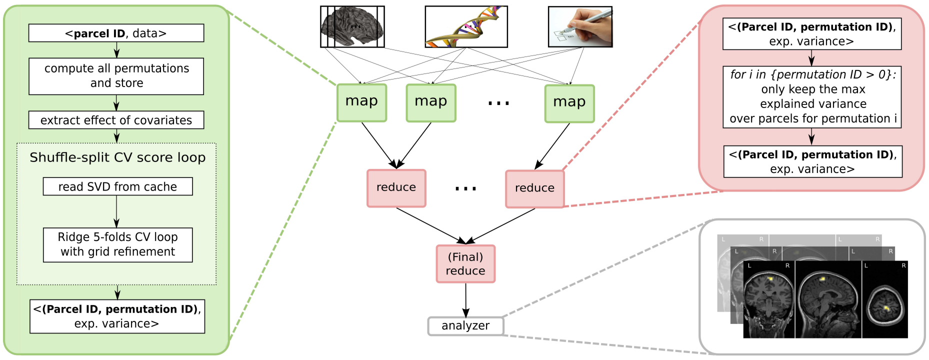
<parcel ID, data>
compute all permutations and store
extract effect of covariates
Shuffle-split CV score loop
read SVD from cache
Ridge 5-folds CV loop with grid refinement
<(Parcel ID, permutation ID), exp. variance>
map
map
...
map
reduce
...
reduce
(Final) reduce
analyzer
<(Parcel ID, permutation ID), exp. variance>
for i in {permutation ID > 0}: only keep the max explained variance over parcels for permutation i
<(Parcel ID, permutation ID), exp. variance>

**Fig. 1.** Overview of the Map-reduce framework for the application of Ridge Regression in neuroimaging-genetics. *Permutation ID* is 0 for not permuted data.

### 2.2 The distributed algorithm

*The MapReduce framework* [2, 3] seems the most natural approach to handle this problem and can easily harness large grids. The *Map* step yields explained variance for an image phenotype and for each permutation, while the *reduce* step consists in collecting all results to compute statistic distribution and corrected p-values. Sub-tasks are created in a way that minimizes inputs/outputs (I/O). By essence, permutations imply computations on the same data after shuffling. The permutation procedure is thus embedded in the mapper, so that all permutations loops are run on the same node for a given dataset and the problem is split in the direction of the brain data. Figure 1 gives an overview of our framework.

*Shared cache* is a crucial feature since the *Map* step is dominated by costly SVDs. For instance, with 1,000 phenotypes and 1,000 permutations, each SVD in the inner CV loop is required 2 millions times and costs few tens of second. A shared cache on NFS, provided by the Joblib Python library [12], coupled with system cache saves many computations.

## 3 Results

We present three types of results. First, we present the performances of our distributed framework. Then, we illustrate the interest of our approach on simulated data with known ground truth. Finally, we present results on a real dataset.

### 3.1 Performance evaluation of the procedure

To illustrate the scalability of our Map-Reduce procedure, we execute the whole framework on a cluster of 20 nodes; each one is a 2 × Intel(R) Xeon(R) CPU X5650 (6 cores) @ 2.67GHz with 48GB of memory, connected with Gigabit Ethernet LAN; all files were written on the NFS storage file-system; our mapper

| | |
|---|---|
| Nb of parcels : | 1,000 |
| Nb of SNPs : | 31,790 |
| Nb of samples : | 1,229 |
| Nb of permutations : | 200 |
| Nb of tasks : | $1,000 + 1$ |
| Theoretical sequential time : | 359h 33min |
| Total time : | 1h 48min |
| Max cores : | 240 |
| Speed-up : | 200 |

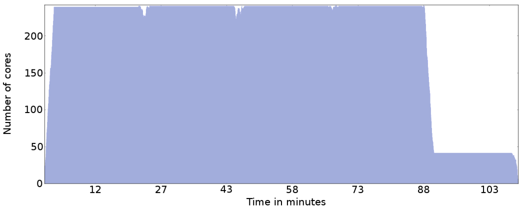

**Fig. 2.** Setting and execution of the MapReduce algorithm on the cluster

runs with the Enthought Python Distribution (EPD 7.2-2-rh5 64 bits) with scikits-learn 0.11 [8] and with the MKL as linear algebra library with OpenMP parallelization disabled; the workflow is described and submitted with the soma-workflow software [6]. This framework makes it possible *i)* to describe a set of independent tasks that are executed following an execution graph and *ii)* to execute the code by submitting the graph to classical queuing systems operating on the cluster. We report in Fig. 2 the result of an execution with almost all the 240 cores available during all the run. The workflow is composed by 1,000 mappers and 1 reducer tasks. The mappers represent 99.9% of the total of serial computation time. Once the SVD are cached, the execution time of a map task is around 20 minutes. We can see in Fig. 2 that after 88 minutes, we use only few cores, but all the unused cores are available for other users. This comes from the number of tasks: on 240 cores, after 4 batches of 240 tasks, only 40 are left. To improve the global speedup, we could split the problem into smaller pieces to decrease the time of the mappers or we could choose a more optimal splitting. We have not explored these possibilities yet.

### 3.2 Simulated Data

We simulate functional Magnetic Resonance Images (fMRI) from real genetic data obtained from the Imagen database [10]. We use the number of minor alleles for each SNP and we assume an additive genetic model. We use only the first chromosome in which ten random SNPs produce an effect in a spherical brain region, centered at a random position in the standard space, then intersected with the support of grey matter using a mask computed for the Imagen dataset (see below). We add i.i.d. Gaussian noise, smoothed spatially with a Gaussian kernel ($\sigma = 3mm$), to model other variability sources. The effect size and the Signal-to-Noise Ratio (SNR) can vary across simulations. Then 1,000 imaging phenotypes are obtained by computing the mean signal in brain *parcels* that were created using a *Ward Agglomeration* clustering.

To assess our approach, ten different datasets were generated and were run on our framework with P=200 permutations to estimate the distribution of the maximum explained variance under the null hypothesis. Results are given in Table 1 and show that our method detects 8 effects among 10 simulations with a p-value $p < .05$. The results do not give evidence of the influence of the SNR simulation nor of the volume of the effect on the test sensitivity.

| Simul. | Volume | Average | Best Parcel | |
|---|---|---|---|---|
| # | ($mm^3$) | SNR | explained variance | p-value |
| 1 | 3375 | 0.19 | 0.022 | 0.005 |
| 2 | 3348 | 0.66 | 0.042 | 0.005 |
| 3 | 2457 | 0.30 | 0.056 | 0.005 |
| 4 | 2754 | 0.54 | 0.033 | 0.005 |
| 5 | 3213 | 0.22 | 0.007 | 0.35 |
| 6 | 3348 | 1.50 | 0.027 | 0.005 |
| 7 | 1431 | 0.55 | 0.031 | 0.005 |
| 8 | 1890 | 0.66 | 0.005 | 0.5 |
| 9 | 3375 | 0.19 | 0.036 | 0.005 |
| 10 | 3132 | 0.41 | 0.026 | 0.005 |

**Table 1.** Results on the simulated datasets *p-value* the p-value corresponding to the given ratio of explained variance, obtained by 200 permutations)

### 3.3 Results on a real dataset

We used data from Imagen, a large multi-centric and multi-modal neuroimaging database [10] containing functional magnetic resonance images (fMRI) associated with 99 different contrast images in more than 1,500 subjects. The dataset is built on the first batch of subjects of the study. Regarding the fMRI data, the protocol in [9] was used, which yields the *[angry faces - neutral]* functional contrast (i.e. the difference between watching angry faces or neutral faces).

*Imaging phenotype.* Standard preprocessing, including slice timing correction, spike and motion correction, temporal detrending (functional data), and spatial normalization (anatomical and functional data), were performed using the SPM8 software and its default parameters; functional images were resampled at 3mm resolution. Obvious outliers detected using simple rules such as large registration or segmentation errors or very large motion parameters were removed after this step. The *[angry faces - neutral]* contrast was obtained using a standard linear model, based on the convolution of the time course of the experimental conditions with the canonical hemodynamic response function, together with standard high-pass filtering procedure and temporally auto-regressive noise model. The estimation of the model parameters was carried out using the SPM8 software. A mask of the grey matter was built by averaging and thresholding the individual grey matter probability maps. Subjects with too many missing data (imaging or genetic) or not marked as *good* in the quality check were discarded. An outliers detection [4] was run and 10% of the *most outlier* subjects were eliminated.

*Genotype.* We keep only SNPs in the first chromosome with less than 2% missing data. All the remaining missing data were replaced by the median over the subjects for the corresponding variable. The age, the sex and the acquisition center were taken as confounding variables.

The final dataset contains 1,229 subjects, 1,000 brain parcels, 31,790 SNPs and 10 confounding variables. Our Map-Reduce framework was run with P=1,000 permutations to assess statistical significance. The workflow takes approximately

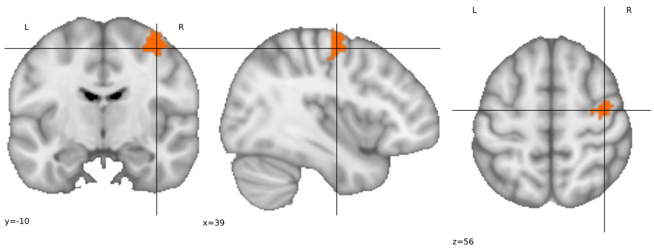

**Fig. 3.** Location of the brain parcel with a significant explained variance ratio (exp. var. = 0.019, $p \simeq 0.048$, corrected for multiple comparisons) on the real dataset.

9 hours on the previously described 240 cores cluster, for a theoretical serial time around 75 days (i.e. a speed-up of approximately 200). Only one parcel is detected with a corrected p-value $\leq 0.05$. A view of the location of the detected parcel is reported in Fig. 3.

## 4 Conclusion

Penalized linear models represent an important step in the detection of associations between brain image phenotypes and genetic data, which faces a dire sensitivity issue. Such approaches require cross validation loops to set the hyper-parameters and for performance evaluation. Permutations have to be used to assess the statistical significance of the results, this yielding prohibitively expensive analyses. In this paper, we present an efficient and scalable framework that can deal with such a computational burden and that we used to provide a realistic assessment of the statistical power of our approach on simulations. Our results on simulated data highlight the potential of our method and we provide promising preliminary results on real data, including one multivariate association that reaches significance. To the best of our knowledge, this is the first result of that kind in a brain-wide chromosome-wide association study, although it needs to be reproduced to be considered as meaningful.

*Acknowledgment* Support of this study was provided by the IMAGEN project, which receives research funding from the European Community's Sixth Framework Program (LSHM-CT-2007-037286) and coordinated project ADAMS (242-257) as well as the UK-NIHR-Biomedical Research Centre Mental Health, the MRC-Addiction Research Cluster Genomic Biomarkers, and the MRC program grant Developmental pathways into adolescent substance abuse (93558). This research was also supported by the German Ministry of Education and Research (BMBF grant # 01EV0711). This manuscript reflects only the author's views and the Community is not liable for any use that may be made of the information contained therein. This work was supported in part by the joint INRIA - Microsoft Research Center. The experiments presented in this paper were carried out using the computing facilities of the CATI.

## References

1. F. Bunea, Y. She, H. Ombao, A. Gongvatana, K. Devlin, and R. Cohen. Penalized least squares regression methods and applications to neuroimaging. *Neuroimage*, 55(4):1519–1527, Apr 2011.
2. C-T. Chu, S. K. Kim, Y-A. Lin, Y. Yu, G. R. Bradski, A. Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *NIPS*, pages 281–288. MIT Press, 2006.
3. J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.
4. V. Fritsch, G. Varoquaux, B. Thyreau, J-B. Poline, and B. Thirion. Detecting outlying subjects in high-dimensional neuroimaging datasets with regularized minimum covariance determinant. *Med Image Comput Comput Assist Interv*, 14(Pt 3):264–271, 2011.
5. O. Kohannim, D. P. Hibar, J. L. Stein, N. Jahanshad, C. R. Jack, M. W. Weiner, A. W. Toga, and P. M. Thompson. Boosting power to detect genetic associations in imaging using multi-locus, genome-wide scans and ridge regression. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1855 –1859, 30 2011-april 2 2011.
6. S. Laguitton, D. Rivière, T. Vincent, C. Fischer, D. Geffroy, N. Souedet, I. Denghien, and Y. Cointepas. Soma-workflow: a unified and simple interface to parallel computing resources. In *MICCAI Workshop on High Performance and Distributed Computing for Medical Imaging*, Toronto, Sep. 2011.
7. N. Meinshausen and P.Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.
8. 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*, 12:2825–2830, 2011.
9. S. D. Pollak and D. J. Kistler. Early experience is associated with the development of categorical representations for facial expressions of emotion. *Proc Natl Acad Sci U S A*, 99(13):9072–9076, Jun 2002.
10. G. Schumann, E. Loth, T. Banaschewski, A. Barbot, G. Barker, C. Bchel, P. J. Conrod, J. W. Dalley, H. Flor, J. Gallinat, H. Garavan, A. Heinz, B. Itterman, M. Lathrop, C. Mallik, K. Mann, J-L. Martinot, T. Paus, J-B. Poline, T. W. Robbins, M. Rietschel, L. Reed, M. Smolka, R. Spanagel, C. Speiser, D. N. Stephens, A. Strhle, M. Struve, and I. M. A. G. E. N. consortium. The IMAGEN study: reinforcement-related behaviour in normal brain function and psychopathology. *Mol Psychiatry*, 15(12):1128–1139, Dec 2010.
11. J. L. Stein, X. Hua, S. Lee, A. J. Ho, A. D. Leow, A. W. Toga, A. J. Saykin, L. Shen, T. Foroud, N. Pankratz, M. J. Huentelman, D. W. Craig, J. D. Gerber, A. N. Allen, J. J. Corneveaux, B. M. Dechairo, S. G. Potkin, M. W. Weiner, P. Thompson, and Alzheimer's Disease Neuroimaging Initiative. Voxelwise genome-wide association study (vGWAS). *Neuroimage*, 53(3):1160–1174, Nov 2010.
12. G. Varoquaux. Joblib: running python function as pipeline jobs. `http://packages.python.org/joblib/`.
13. M. Vounou, T. E. Nichols, G. Montana, and Alzheimer's Disease Neuroimaging Initiative. Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach. *Neuroimage*, 53(3):1147–1159, Nov 2010.