Introduction

It has become possible to acquire high-dimensional neuroimaging and genetic data on relatively large cohorts of subjects, which provides novel means to understand the large between-subject variability observed in brain organization. Genetic association studies aim at unveiling correlations between the genetic variants and the numerous phenotypes extracted from brain images and thus face a dire multiple comparisons issue.

Prerequisites

- Tractable Voxelwise GWAS: while statistics can be accumulated across the brain volume for the sake of sensitivity [2], the significance of the resulting summary statistics can only be assessed through permutations.
- Scientific Python: mainstream tools such as Numpy/Scipy.

The computational framework: MapReduce

Optimizations

- Good granularity: computing over phenotypes or explanatory variables one by one is inefficient. Matrix-based operations should thus be used instead of vector-based operations.
- Filtering: keeping all the associations represents petabytes of output. For this reason, we store only p-values lower than a threshold in single precision.
- Linear algebra properties and caching: the norm of a column does not change after a permutation on rows, orthogonality is preserved too. Results can be cached to speed up regressions on permuted data.
- I/O issues: requiring all the data for each subtask is a waste of resources. By essence, permutations imply computations on the same data after shuffling. So the permutation procedure should be embedded in the inner loop, such that all permutations are done on the same node for a given dataset.

Results on real data

- Comparison of execution time and speed of Plink and our mapper.
- Speedup 1 is the speedup against the performance reported in [1], while Speedup 2 is the speedup against Plink with our settings.

Performance evaluation of the procedure

Sequential

<table>
<thead>
<tr>
<th>Procedure</th>
<th>GWAS time (sec)</th>
<th>Speedup 1</th>
<th>Speedup 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plink v1.06</td>
<td>~540</td>
<td>8.30 x 10^6</td>
<td>1</td>
</tr>
<tr>
<td>Plik v1.06</td>
<td>~2</td>
<td>2.24 x 10^6</td>
<td>1</td>
</tr>
<tr>
<td>Our mapper</td>
<td>0.803</td>
<td>5.50 x 10^6</td>
<td>6667</td>
</tr>
</tbody>
</table>

Parallel

- Setting and execution of the Map-Reduce algorithm on a HPC cluster.

References and acknowledgments


Conclusion

- We present an efficient and scalable framework that can deal with cluster-based inference and permutations.
- Results on real data highlight the potential of our method.
- This framework runs on HPC cluster or cloud.
- Paper available at http://hal.inria.fr/hal-00720265