Analyzing Large Scale Genotype Datasets With Gnocchi

Frank Austin Nothaft, fnothaft@berkeley.edu

Background

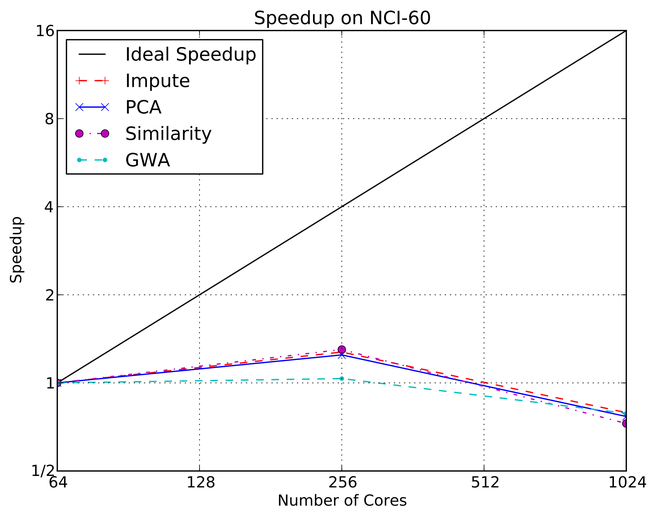
Although the ADAM project defines parallel APIs for processing large genomic datasets, these APIs are mostly designed for running ETL jobs against read data. While some of these APIs can be reused for variant data, genomic variants can benefit from different APIs:

- Variant datasets need to be analyzed in the context of a population.
- Genotype datasets have a natural mapping to matrix-like structures.

In Gnocchi, we introduce a variety of APIs for running matrix-like calculations against large, genomic variant datasets.

Performance

- Evaluated for speedup on 1,024 core cluster
- Run against NCI-60 cell line dataset + phenotypes
- 16GB of memory per core, 0.75TB HDD, 10G interconnect



Architecture

Matrix methods:

Traditionally, genomic variant storage formats have stored genotypes as "row"-oriented data. In this representation, all of the genotypes for a single variant are stored in a single record. While this makes matrix calculations straightforward, it limits scalability, as the size of a single record grows with the number of individuals we are looking at.

ADAM uses a "cell" oriented approach, where genotypes belong to independent records. As such, we must be able to reconstruct genotype matrices. We do this by running a `groupBy` on each position.

Frequently, people want to run PCA on genotype matrices, however, this is difficult to do in Spark because genotype matrices are flat and wide. Instead, we transpose the genotype matrix and compute PCA via the SVD:

$$A = U\Sigma V^{\mathsf{T}}$$
$$A^{\mathsf{T}} = V\Sigma U^{\mathsf{T}}$$
$$T_A = U\Sigma$$

Aggregation queries:

While matrices are used to run many machine learning kernels that are used to identify population structure, not all algorithms map to matrices. Another common pattern is per-variant aggregation:

pos	ref/alt	NA1	NA2	NA3	NA4	...	NA999
chr1:1,000	T/A	0/0	0/1	1/1	0/1		0/0

$$a_0 = f(i, g_0)$$
$$a_i = f(a_{i-1}, g_i), \forall i \in [1, N)$$

We use this pattern to implement a case-control genome wide association study (GWAS) test. This kernel applies a χ^2 test to the cross-join of all genotypes and phenotypes for a population to test for the association of a phenotype with an individual variant.

This pattern is typically used to iteratively train models, such as a linear regression at each variant, as well as to compute per-site statistics.

Future Work

- Many of these stages operate on a matrix of genotypes. Currently, we build this matrix from the input genotypes each time via a groupBy.
  - In practice, we frequently work on a sparse representation of the matrix, which is much smaller than the full matrix. We can materialize this.
  - Additionally, the groupBy cost can be minimized through better partitioning strategies (e.g., coordinate sort).
- We are working on more clustering variants (e.g., K-means) and regression tests.
- Additionally, the regression kernels expose interesting join patterns.