COGWAS – A system for mapping genotypes to phenotypes

Patent 1 - First order nucleotide importance for a phenotype

Research Background :

The tool provides a novel methodology and system for Genome Wide Association Studies (GWAS). In this work, we create large scaled multiprocessing systems which can map genotypes to phenotypes. Phenotypes are defined as observed physical attributes of an organism. Within a specie, phenotypes differ between multiple organisms due to the genotype, which is defined as the gene sequence of the organism. Finding mapping between genotypes and phenotypes is the holy-grail of biology due to the following reasons:

1. Generate novel causal hypothesis regarding biological pathways that affect physical trait of the organism.
2. Results of GWAS can be used to predict an individual's biological proclivity towards certain diseases through their gene sequence. For example, this can be used to predict future diseases that a new born child might have.
3. Genes which have a considerable effect on phenotype can be used to engineer crops with desired properties.

This tool maps genotype to phenotype, by taking VCF files as input. The tool then transforms the VCF file into a suitable format for generating the probability with which each position in the genome affects the target phenotype

Claims:

The core problem which GWAS addresses is that gene sequence is very large and it is not clear which part of the gene or which alleles in the gene are responsible for which phenotype. The solution for this problem involves reducing the number of 'candidate alleles' which have a correlation with the phenotype. In prior-art, p-value is used as a metric to shortlist a set of causal genes that affect phenotype. In our system, we present the following claims:

1. We formulate a new metric for quantifying the goodness of a candidate allele to be correlated with phenotype.
2. We provide a methodology that evaluates the individual allele goodness quantification method by using the top k such candidates produced by the method, build a prediction model only using those features and compare the accuracy of the prediction.
3. Our approach in claim 1 is shown to be doing significantly better than p-value (the prior art) w.r.t. the method proposed in claim 2.
4. We have also come up with a very efficient and scalable pipeline using big-data computing for compressing the traditional data into a binary format, generating the candidate scores, and evaluating the top candidates efficiently.

Part 1 – The new measure for goodness of an allele: Single Loci Relevance Calculation

This stage of the patent calculates the importance of each position in terms of probability of its presence in the strains that show a particular phenotype.

A phenotype is a physical trait such as leaf length, grain weight etc. and is a continuous variable. We convert the value of phenotype into a binary variable, 0 or 1. This was performed by strategically choosing a threshold value for the phenotype, and all the strains with a desired trait having larger value than the threshold were considered 1 (or positive sample) and the rest as 0 (or negative samples). This method can also be reversed if the desired trait has a smaller magnitude, hence, samples with phenotype less than a threshold can also be considered as 1 (or positive sample).

All the samples are then aligned together for the process of calculating probabilities.

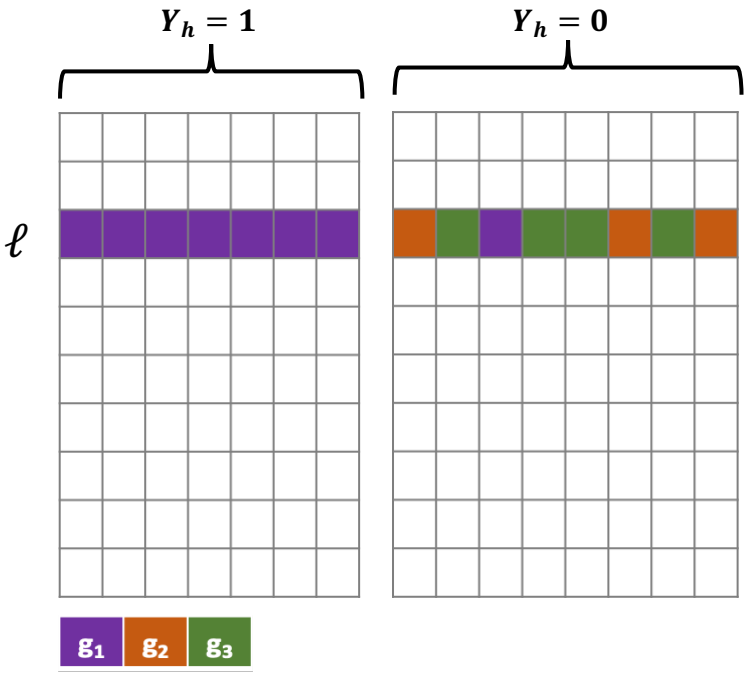

Fig 1 - Alignment of multiple strains of the rice genome. Each row in the figure is a position in the rice genome and each column represents a strain. For each position in each row, there can be three values GT1, GT2 or GT3.
Y_h describes the phenotype value, 0 or 1.

The above process is implemented by aligning individual NumPy files into a single NumPy matrix by their POS ID. Final matrix is of shape L*M*3, where L is the length of reference genome and M is the total number of samples having positive phenotype and negative phenotype, M = P+N.

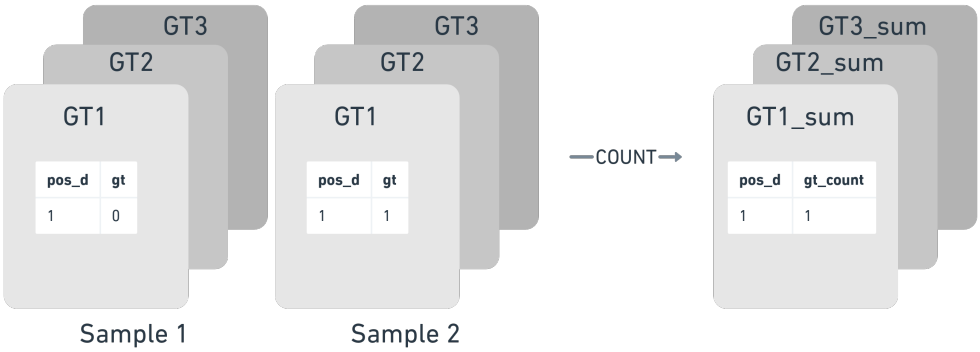

Fig 2 – Illustrated counting algorithm for 2 samples

Before further processing the count matrix, POS_IDs where 80% of the columns are unfilled are dropped.

We calculate COGWAS probability scores for quantifying the relevance of singleton locus. The first step to calculate the score is to calculate the counts of GT values at each position. We calculate 6 set of counts for each position

	$X_\ell = g1$	$X_\ell \neq g1$	$X_l = g_1, g_2, g_3$
$Y_h = 1$	$n(X_\ell = g1, Y_h = 1)$	$n(X_\ell \neq g1, Y_h = 1)$	$n(Y_h = 1)$
$Y_h = 0$	$n(X_\ell = g1, Y_h = 0)$	$n(X_\ell \neq g1, Y_h = 0)$	$n(Y_h = 0)$

Figure 3 – counts for GT1 Genotype.
$g1$ at position ℓ is highly relevant for phenotype h, if both $P(X_\ell = g1|Y_h = 1)$ and $P(X_\ell \neq g1|Y_h = 0)$ are high

Using the counts mentioned in Figure 3, we calculate COGWAS Score for each Genotype for each position by finding probabilities for each genotype

$$P(X_\ell = g1|Y_h = 1) = \frac{n(X_\ell = g1,\ Y_h = 1)}{n(Y_h = 1)}$$

$$P(X_\ell \neq g1|Y_h = 0) = \frac{n(X_\ell \neq g1,\ Y_h = 0)}{n(Y_h = 0)}$$

We further use these probability scores to generate a COGWAS score

$$S_{h,l} = \max\left(S_{h,l}(\mathrm{g}_1),\quad S_{h,l}(\mathrm{g}_2),\quad S_{h,l}(\mathrm{g}_3)\right)$$

	$X_\ell = g_1$	$X_\ell \neq g_1$
$Y_h = 1$	$n(X_\ell = g_1, Y_h = 1)$	$n(X_\ell \neq g_1, Y_h = 1)$
$Y_h = 0$	$n(X_\ell = g_1, Y_h = 0)$	$n(X_\ell \neq g_1, Y_h = 0)$

$$S_{h,\ell}(g_1) = \sqrt{P(X_\ell = g_1|Y_h = 1) \times P(X_\ell \neq g_1|Y_h = 0)}$$

	$X_\ell = g_2$	$X_\ell \neq g_2$
$Y_h = 1$	$n(X_\ell = g_2, Y_h = 1)$	$n(X_\ell \neq g_2, Y_h = 1)$
$Y_h = 0$	$n(X_\ell = g_2, Y_h = 0)$	$n(X_\ell \neq g_2, Y_h = 0)$

$$S_{h,\ell}(g_2) = \sqrt{P(X_\ell = g_2|Y_h = 1) \times P(X_\ell \neq g_2|Y_h = 0)}$$

	$X_\ell = g_3$	$X_\ell \neq g_3$
$Y_h = 1$	$n(X_\ell = g_3, Y_h = 1)$	$n(X_\ell \neq g_3, Y_h = 1)$
$Y_h = 0$	$n(X_\ell = g_3, Y_h = 0)$	$n(X_\ell \neq g_3, Y_h = 0)$

$$S_{h,\ell}(g_3) = \sqrt{P(X_\ell = g_3|Y_h = 1) \times P(X_\ell \neq g_3|Y_h = 0)}$$

Figure 4 - $S_{h,l}$ is calculated for each position l for each genotype g_1, g_2 *and* g_3

Part 2 – The method of evaluating the measure of goodness

The scores calculated from COGWAS are stored in a master table. Top-k positions from the master table are chosen to train classifier models, where the input is loci GT value and the output is phenotype. The pipeline uses Grid Search algorithm to generate optimized hyperparameters in a multiprocessing fashion. Classifier is then further trained using these optimized hyperparameters. Multiple measures of goodness can be compared by looking at the classifiers' accuracy metrics.

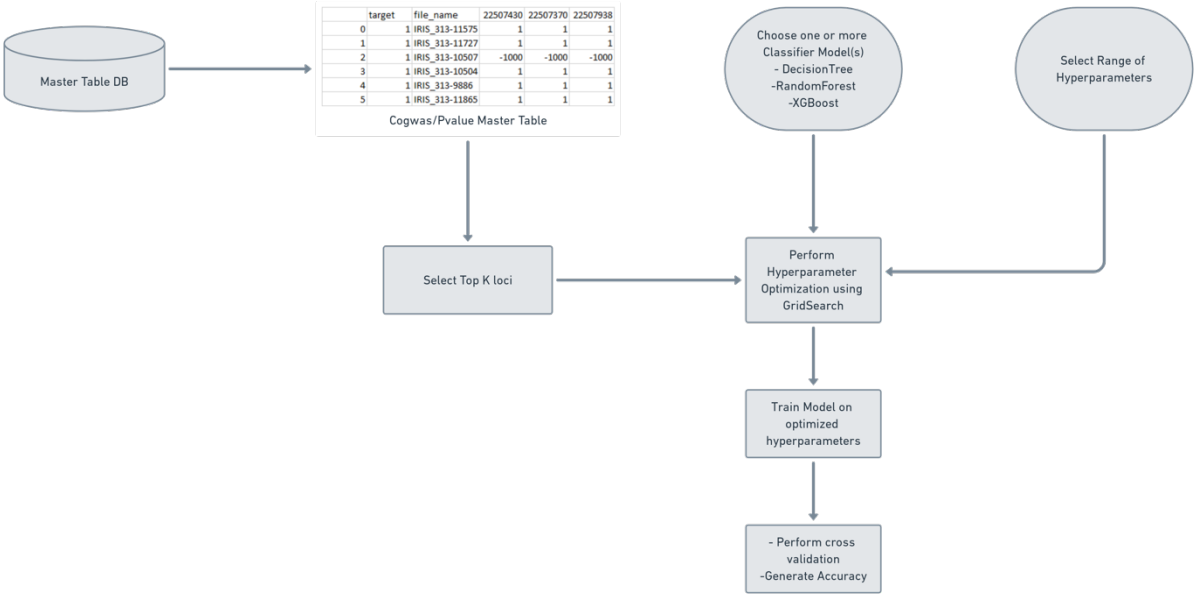

Figure 5 – Flowchart for evaluation of measure of goodness

Part 3 – Comparison of COGWAS and p-value results on Rice genome

Current State of the Art Genome Wise Association Tools such as PLINK use p-values for hypothesis testing. We also calculate p-values using chi-squared test using the three 2X2 contingency matrix shown in figure 4. To calculate p-values using chi squared test, we implement the chi2_contingency tool in the SciPy. Stats library in a multiprocessing fashion. We compare COGWAS score and P-value by training classifier algorithms for multiple K values.

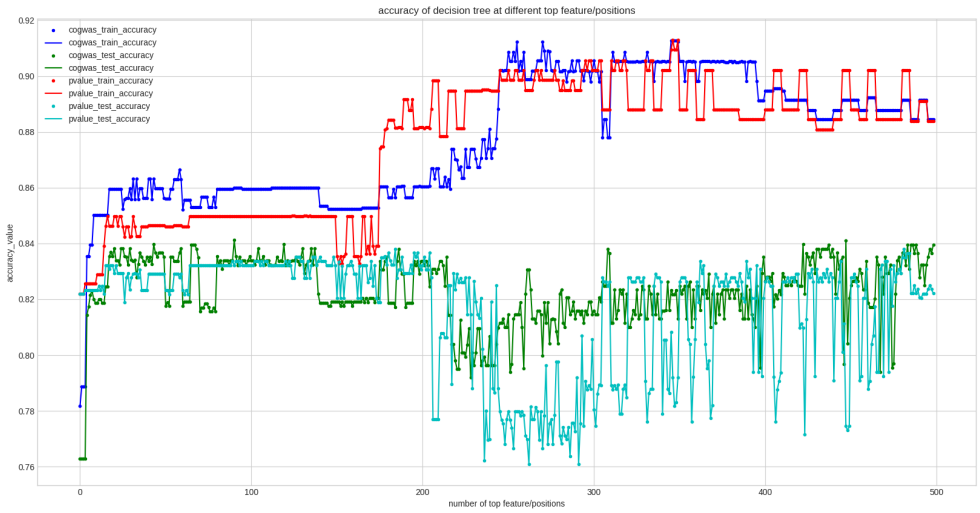

Figure 6 - Comparison of COGWAS score and p-value for multiple k positions by training decision tree classifier

COGWAS score consistently outperforms p-value scores on the test set.
Since K is a hyperparameter, various experiments were performed by taking different values of K and checking which of them are sufficiently able to predict the phenotype. We performed this process by creating a master table, where the rows are sample and columns are SNV positions. To create the master table, we choose K to be fairly large. Then, we vary k from 100 to 1000 at intervals of 100, and for each value of K, we train a decision tree classifier. For example, if K = 100, we select top 100 positions for the 3000 rice genomes, create a feature list of GT values for each sample and keep the target variable as phenotype Boolean. Then for each K, a grid search is performed which helps to find the maximum accuracy the decision tree can provide while taking only k features as input.
In the Figure-7, decision tree gives fairly high accuracy for 400 top positions. Therefore, we run the following pipeline with k = 400.
We were able to reduce the search space for important gene locations from 4M to 400.

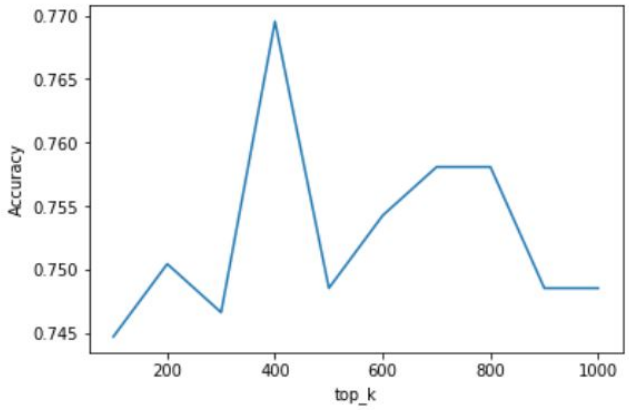

Figure 7 – Test Accuracy of decision tree on k-positions

Once all the 'Single Nucleotide Variations (SNVs)' are scored, either through COGWAS scoring or through chi squared p-values, the SNVs are sorted according to their scores.

Part 4 – The big-data pipeline to solve the problem end-to-end

Data Processing and Compression Algorithm

3000 Variant Call format files for rice were extracted from https://www.irri.org/
Each of the VCF files come with a metadata which contains the phenotype information. We chose 57 phenotypes, hence, every VCF file is annotated with 57 phenotypes and the unit with which corresponding phenotype is measured in. For e.g. – Leaf width (in cms) and Seed Length (in mms)
VCF files are memory heavy, each VCF file takes up 25 Gb. Performing computation on these raw VCF files is challenging. So, we reformat and compress the VCF files for further extensive computation.
Since we perform comparative analysis, we extract, from each VCF file, only the POS ID, Qual and GT. Following describes the three column headers:

1. POS ID – This column contains the position of each nucleotide. For e.g., since on an average, a rice genome consists of 4M base pairs, the values in the column range from 1 to 4M in a sorted manner. Some rows only have a POS ID and no GT value, which means that during the sequencing process, this position either had low quality value or an instrumentation error.
2. Qual – This column describes the quality score which quantifies how confident the sequencing instrument is about the nucleotide value for each position.
3. GT – This column describes any mutation present in the position. This mutation is compared to the reference rice genome. For e.g., if GT value for a sample POS ID is 0/0, this describes that the nucleotide at this POS ID for this sample is same as that of the reference genome.

Data Pre-Processing:

1. We filter out POS ID values with missing GT or Qual values for further analysis.
2. As a threshold, we filter out POS ID where Qual >=30.
3. We encode the GT value with its corresponding encoding shown below

GT Value	GT Value encoded	GT name
0/0	0	gt1
1/0 or 0/1	1	gt2
1/1	2	gt3

4. We drop other columns from the VCF files.
5. GT values are one-hot encoded and bitwise transformed to NumPy format

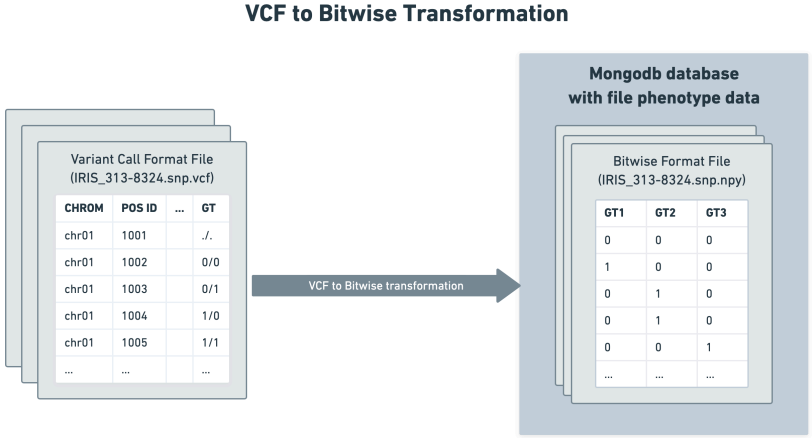

Fig 8 – Transformation of VCF file to Bitwise NumPy format

Due to large number of files, we perform the above compression in a multiprocessing manner. We divide 3000 files into 50 buckets, one for each core. We use 60 cores simultaneously for data compression. For each VCF file, one NumPy file is generated.

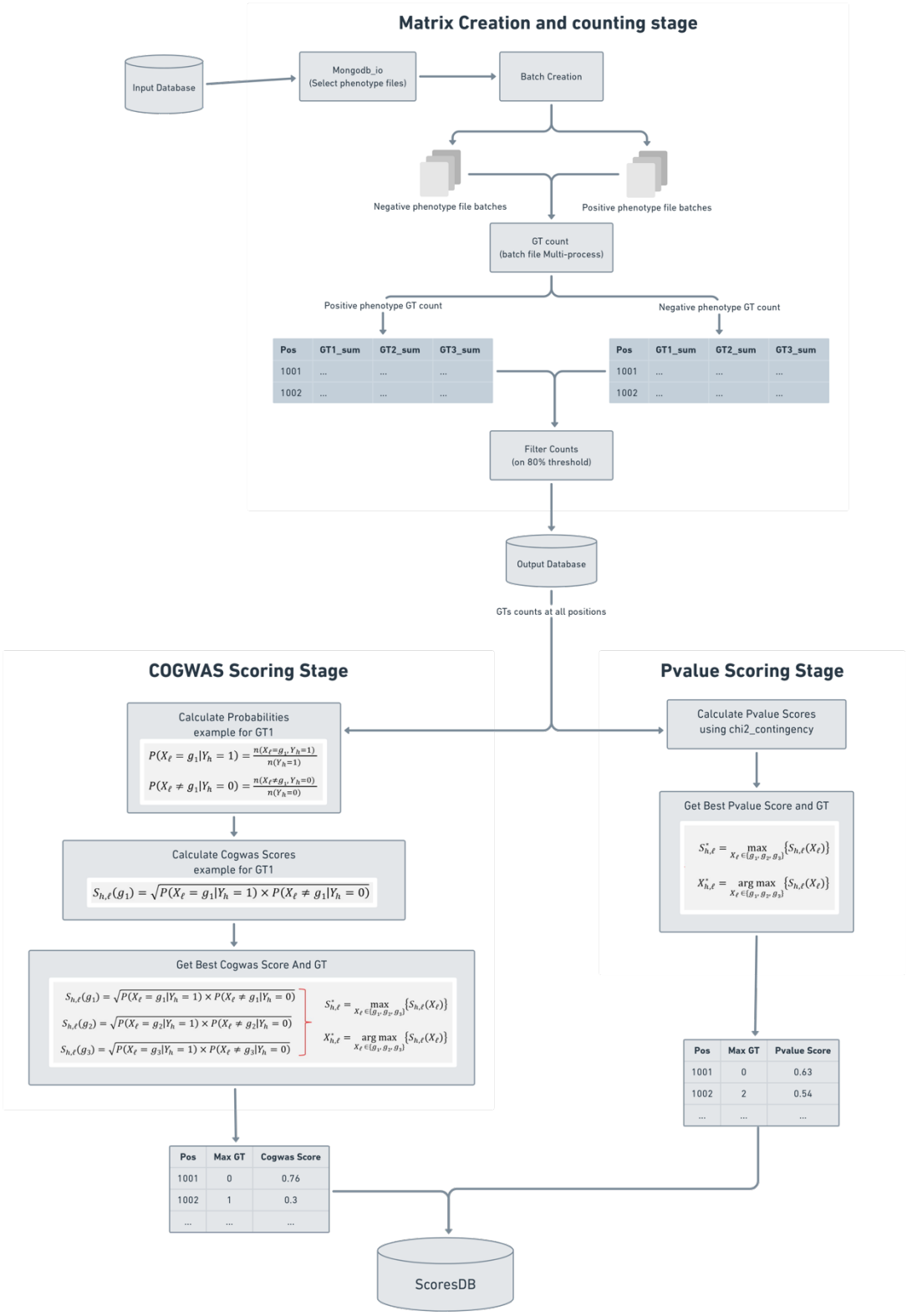

Figure 9-Flowchart for calculating first order nucleotide importance for a phenotype

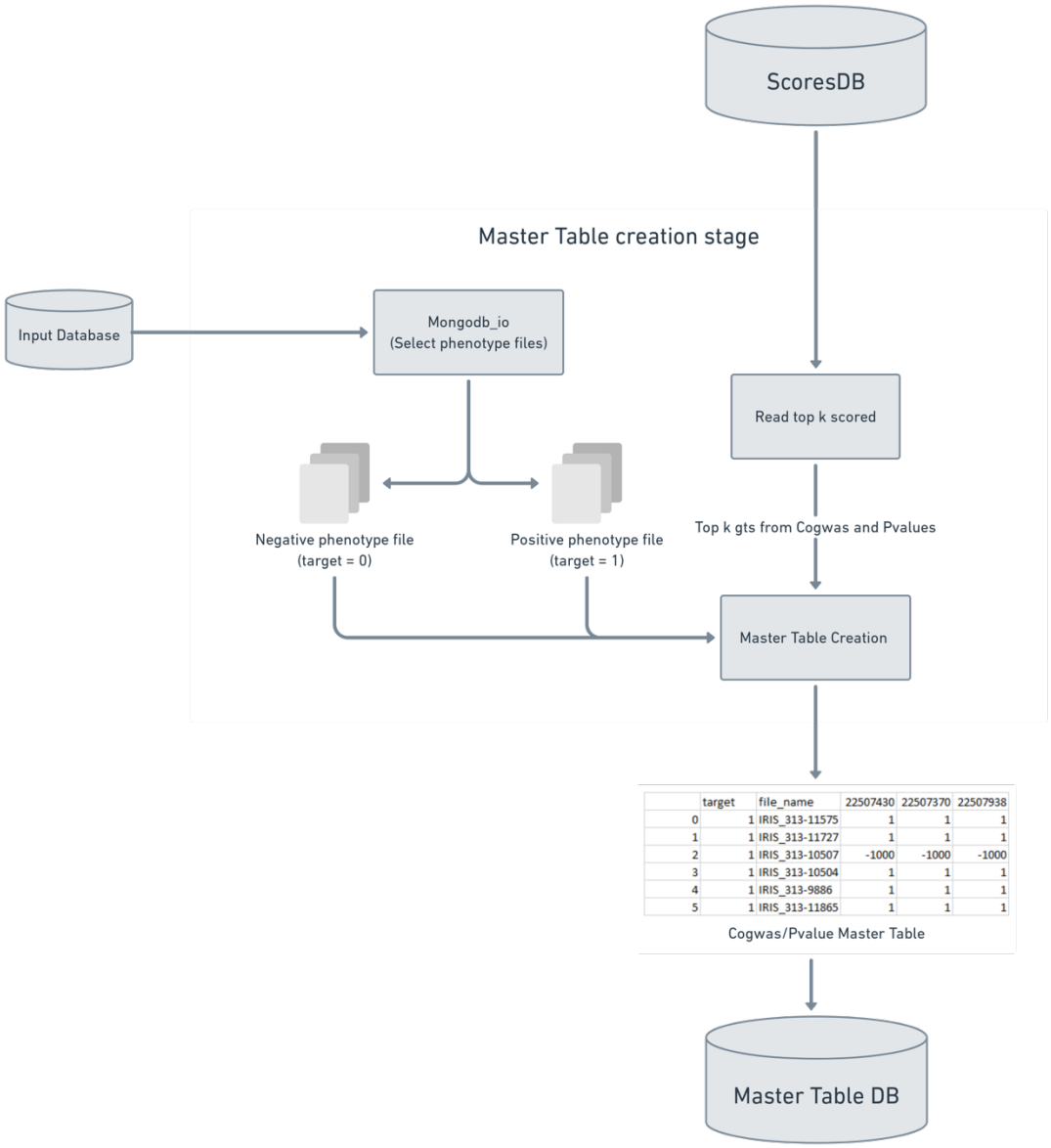

Figure 10 – Master Table creation stage

Post shortlisting of the top 400 loci with significant contribution to Phenotype, we create Entity table and Vocabulary table for COGWAS and P-value scores

Entity Table -
The table represents top-K position selected for each sample

	sample_id	entity_list
0	IRIS_313-11037	[5489084, 5490014, 5504701, 5506033,..., 5092295, 22540942, 22579365]
1	IRIS_313-11661	[22507430, 22507370, 22507938, 22521101,..., 5092295, 22540942, 22579365]

Vocabulary Table –
Top-K positions were queried against BLAST database for rice genome to convert position to gene names

	position	description
1	6453233	{'max_gt': 1, 'location_type': 'CDS', 'gene_id': ['Os01g0218032'], 'protein_id': ['BAS71045.1'], 'note': ['Os01t0218032-01: Similar to Repressor of silencing 1.; start codon is not identified.']}
2	6463385	{'max_gt': 1, 'location_type': 'CDS', 'gene_id': ['Os01g0218150'], 'protein_id': ['BAS71046.1'], 'note': ['Os01t0218150-00: Ab initio predicted gene.']}