Validate: A Pipeline for Evaluating Classification and Regression Performance for GWAS/QTL Tools Using Known-Truth Datasets

Ann Stapleton and Dustin Landers
University of North Carolina Wilmington

Abstract

Understanding the effectiveness of Genome-Wide Association (GWAS) and Quantitative Trait Loci (QTL) analytical tools under various situations is crucial to deciding which tools are best given a particular problem. Validate provides a way to return classification and regression performance measures for large amounts of tool outputs generated using known-truth simulations. We also provide solutions for aggregating hundreds or thousands of outputs in to a single folder on the iPlant data store, so that Validate can be used.

Contents

1 Introduction

1.1 About the Developers

Project Lead and Principal Investigator

Dr. Ann Stapleton works at the interfaces, such as the junctions between research and teaching, individual research projects and large collaborative projects, the organization of international meetings and high-school teacher inquiry labs. Most recently, she has worked at the interface between plant biology and software engineering, leading the way to broad methods applicable to evaluating genotype-to-phenotype analytical methods.

Statistical Analyst and Developer

Dustin Landers received a BS from Appalachian State and continued his education in Applied Statistics at UNCW. Moving from world of survey statistics to the that of Big Data, he has an interest in bridging the gap between statistics and software engineering.

1.2 Why Do We Need Validate?

Measuring the predictive performance of GWAS and QTL models and applications presents a unique challenge to the predictive modeler. A typical method for evaluation of predictive models is to assess their performance through a process known as cross-validation. In cross-validation, the initial data is broken down in to a training data set and test data set. Using the training data set, the actual predictive model is built. We then pass the test data set through the model, and compare the predicted values with the known-truth values. Doing so allows us to assess if our initial model was over or under fit.

However, with GWAS and QTL models, we are not so much interested in the predicted phenotypes. Instead, we are interested in what the models tell us about how we arrived at the predicted phenotypes. Did the SNP produce an effect significantly different from zero? What was the effect? Are the SNPs that correlate with phenotypic effects clustered in a general region of the genome or do they vary in location? Are there lots of small effects or a few large effects? Are the effects factorial in nature (epistasis) or are they additive?

In order to assess the performance of these kinds of models in giving us the answers to these questions, we rely on large simulated data sets that are generated with known random effects from identical stochastic processes. By using models to predict data sets for which already know the processes underlying their development, we can get an idea of how accurate the models are at identifying real processes in nature and empirically estimate the models error using even basic statistical tools.

1.3 What is the GWAS/QTL Problem in Genetics and How Do We Approach It?

We begin by introducing what we considered in developing an application to test GWAS classification and effect size estimation performance, and how we approached a universal method for measuring classifier and estimation performance. We assume that GWAS models and algorithms are typically interested in two separate but intimately related statistical and machine learning problems.

First, GWAS applications can be seen as a classification problem under which we are only interested in estimating whether or not a marker has an effect or not: a binary response of true or false. Or, in terms of effect sizes, is the true effect of a marker different from zero?

Second, GWAS applications can be seen as an estimation or regression problem. In the latter case, we use relatively simple measures such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to evaluate effectiveness. Validate treats this as an optional parameter, and the focus of Validate is on estimating GWAS applications' effectiveness as a classifier.

Usually, the classifier problem can be formulated such that we are interested in understanding how $\mathbf{Y}$, a vector of some (usually quantitatively) observed trait, is a function of $\mathbf{X_i}$, a vector of a qualitatively observed genotype for some given SNP or marker i. We are interested in establishing a new vector $\mathbf{T}$:

$$T_i = c(s(i); t), T_i \in \{0, 1\}$$

where $s(x)$ is a scoring function and t is an arbitrary threshold. $c(x; t)$ returns a predicted class based on the score. We assume that the end goal is a vector that indicates which markers are associated with change in the quantitative trait.

By cross-validating models with data that we have generated, we can test the effectiveness of models by comparing a known version of $\mathbf{T}$, which we will denote $\mathbf{K}$, with the $\mathbf{T}$ vector generated from the GWAS application. For some measures, such as AUC, $\mathbf{K}$ will remain the same while $\mathbf{T}$ changes under new parameter settings for t.

For this reason, for the purposes of Validate, we assume GWAS applications are serving the function of $s(x)$. In other words, we view GWAS applications' primary goal as assigning scores to markers. An application which returns $c(x; t)$ is limited to other measures of performance, such as misclassification error, true positive rate, and so on.

1.4 How to Use This Manual

Evaluating a GWAS/QTL tool with Validate is essentially three major steps after your application is installed on either the iPlant Foundation API or the Agave API:

→ Section 2

1. **Running the Tool With Simulations As Inputs.** This step involves deciding what simulations to use and then iterating over those simulations and submitting them as job requests through the API. We recommend using some sort of scripting method that you are comfortable with. At this point, we don't have a standalone application for submitting jobs. We use rPlant, which is freely available R package that allows you to connect with iPlant's API layer to submit job requests. We also recommend sampling about 300-500 simulations per tool tested.

→ Section 3

2. **Aggregating the Outputs into a Single Folder.** This part is a bit of a logistical exercise. You need to put all your tool outputs that you want to be analyzed using the same known-truth metadata in to aggregate folders. For example, say we are using simulations that are generated using varying levels of heritability. Since the varying heritability values in essence produce different SNP effects, we need to essentially run Validate three separate times. Validate requires an input on an entire folder, and then iterates over those tool outputs. So the first step here, is to decide how many different runs we need to do and then create that many folders. We provide a GUI tool, *Aggregate* that allows you to select files from multiple

folders on your iPlant data store (multiple runs of a single tool) and move those (or aggregate them) in to a single folder.

→ Section 4

3. **Running *Validate* on the Aggregated Folder.** Once you have all your outputs in aggregated folders. You can simply log in to the iPlant Discovery Environment and run *Validate* on that folder. You must know two column names in the outputs: The name of the SNP column, and the name of threshold column (such as P-value). Further, if you wish the get back effect size estimation errors, you must also know the column on the estimated effect size column (for example, PLINK's is BETA). Once you submit Validate, you will receive a notification when it is completed. The *Validate* output will be columns of performance measures for each tool output.

→ Section 5

4. **Running *Demonstrate* on the Validate Output.**