In silico phenotyping via co-training for improved phenotype prediction from genotype[*]

Damian Roqueiro [1†], Menno J. Witteveen [1†], Verneri Anttila [2,3,4], Gisela M. Terwindt [5], Arn M.J.M. van den Maagdenberg [5,6], Karsten Borgwardt [1]

[1] *Machine Learning and Computational Biology Lab, Dept. of Biosystems Science & Engineering, ETH Zurich, Switzerland* [2] *Analytical and Translational Genetics Unit, Dept. of Medicine, Massachusetts General Hospital and Harvard Medical School, Boston, Massachusetts, USA* [3] *Program in Medical and Population Genetics and* [4] *Stanley Center for Psychiatric Research, Broad Institute of MIT and Harvard, Cambridge, Massachusetts, USA* [5] *Dept. of Neurology and* [6] *Dept. of Human Genetics, Leiden University Medical Center, Leiden, The Netherlands*

[*]Accepted for oral presentation at ISMB 2015. Contact: menno.witteveen@bsse.ethz.ch
[†]Both authors contributed equally to this work

ABSTRACT

Predicting disease phenotypes from genotypes is a key challenge in medical applications in the postgenomic era. Large training datasets of patients that have been both genotyped and phenotyped are the key requisite when aiming for high prediction accuracy. With current genotyping projects producing genetic data for hundreds of thousands of patients, large-scale phenotyping has become the bottleneck in disease phenotype prediction.

Here we present an approach for imputing missing disease phenotypes given the genotype of a patient. Our approach is based on *co-training*, which predicts the phenotype of unlabeled patients based on a second class of information, e.g. clinical health record information. Augmenting training datasets by this type of *in silico* phenotyping can lead to significant improvements in prediction accuracy. We demonstrate this on a dataset of patients with two diagnostic types of migraine, termed migraine with aura and migraine without aura, from the International Headache Genetics Consortium.

Imputing missing disease phenotypes for patients via co-training leads to larger training datasets and improved prediction accuracy in phenotype prediction.

Motivation Co-training (Blum and Mitchell, 1998) is an instance of semi-supervised learning, which is often employed in scenarios where the number of labeled examples ($\mathcal{L}$) is low and the number of unlabeled instances ($\mathcal{U}$) is large. The reason for this imbalance is simply due to the high cost of labeling the data. The co-training method benefits from a natural split of the feature space. An instance x is described by the set $\mathcal{X}$ of all features, comprised of two mutually exclusive "views" $\mathcal{X}_1$ and $\mathcal{X}_2$. A labeled object x is referenced as $((x_1, x_2), y)$ where x_1 and x_2 are the values for the features in $\mathcal{X}_1$ and $\mathcal{X}_2$, and y is the class label. The algorithm then learns two classifiers h_1 and h_2, one for each view of $\mathcal{L}$, followed by an iterative bootstrapping in which instances of $\mathcal{U}$ are labeled and the most confident ones are moved to $\mathcal{L}$.

In this study, and following the spirit of co-training, the two exclusive views of the data are the clinical covariates and the genotype data of patients with one of two different types of migraine: a) migraine with aura and b) migraine without aura.

Results Our analysis was conducted by partitioning the entire dataset into three groups: Set **I**, the *training dataset*: contains a subset of the patients for which all available information is present, i.e.: a disease phenotype, a set of clinical covariates and genotype data in the form of single-nucleotide polymorphisms (SNPs); Set **II**, the *co-training dataset*: similar to the training set but with a much larger number of patients. Here the patients lack a disease phenotype (unlabeled); Set **III**, the *evaluation dataset*: is used to evaluate the method. It does not contain clinical covariates. This is depicted in Fig. 1.a

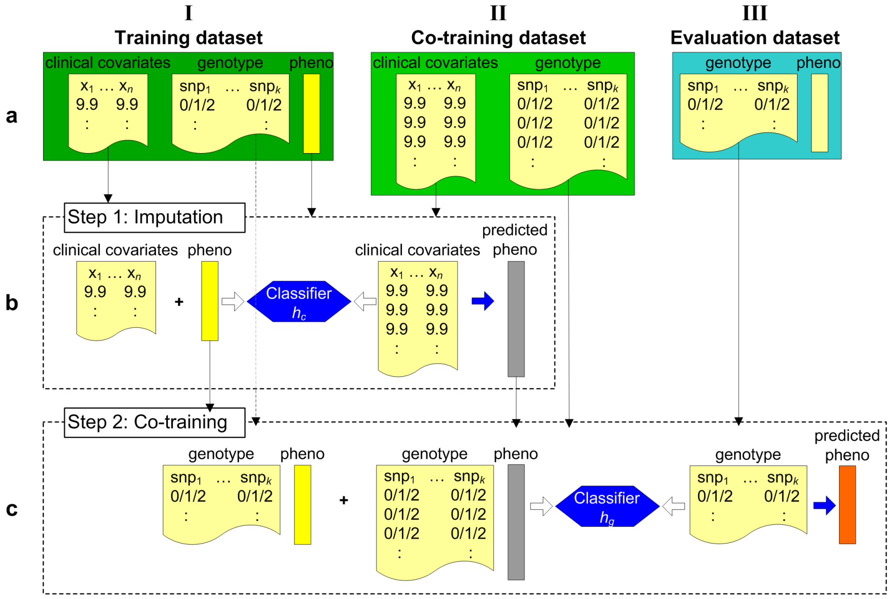

Figure 1: Data partioning and the proposed two-stage approach to co-training

The algorithm was then applied in two sequential steps (Fig. 1.b-c): Step 1: predict a disease phenotype for the patients in set II by learning a classifier h_c from the clinical covariates of the patients in set I; Step 2: the previous predictions are used to augment the pool of labeled examples. Then, a genotype classifier h_g is constructed via co-training. Finally, h_g is tested on III to obtain an AUC score.

Four metrics were used to compare the prediction performance of the algorithm. These metrics corresponded to different cases that ranged from using the least possible amount of data for training (to compute a lower bound) to using all available data (upper bound). Between these two ranges, the actual prediction performance was reported and all these values are shown in Table 1.

Table 1: Bounds and prediction performance of *in silico* phenotyping. Partition of the data into: set I = 10%, set II = 70% and set III = 20%; 100 random folds.

	AUC scores	
Metric	μ	σ
Lower bound, training only on I	0.574	0.034
Univariate featire selection on I, training on I+II	0.608	0.035
In silico phenotyping (co-training)	0.646	0.029
Upper bound, I+II with true labels	0.689	0.025

References

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, New York, NY, USA. ACM.