Bioinformatics, 31, 2015, i303–i310
doi: 10.1093/bioinformatics/btv254
ISMB/ECCB 2015

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] and Karsten Borgwardt[1,*]

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

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Abstract

Motivation: 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.
Results: 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.
Conclusions: Imputing missing disease phenotypes for patients via co-training leads to larger training datasets and improved prediction accuracy in phenotype prediction.
Availability and implementation: The code can be obtained at: http://www.bsse.ethz.ch/mlcb/research/bioinformatics-and-computational-biology/co-training.html
Contact: karsten.borgwardt@bsse.ethz.ch or menno.witteveen@bsse.ethz.ch
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Predicting disease phenotypes from genotypic information of a patient is a key question in medical research, with implications for disease diagnosis, prognosis and therapy. Any prediction system, or *classifier*, relies critically on the existence of a training dataset which includes labeled examples, that is to say, patients for which both genotypic and phenotypic data are present. The limiting factor when creating such training datasets used to be the low number of patients for which genotypic or even full-genome data were available. But experimental advances in genotyping, using genotyping chips (Wellcome Trust Case Control Consortium, 2007) or next-generation sequencing (Davey *et al.*, 2011), plus the advent of many large-scale sequencing studies (1000 Genomes Project Consortium *et al.*, 2012) and biobanks (Allen *et al.*, 2014) that store genotypic information, are steadily changing this situation; gradually, the availability of disease phenotypes is turning into the bottleneck when collecting training datasets for phenotype prediction.

Automated approaches to phenotyping, such as image phenotyping which extracts features from images, are currently gaining popularity, e.g. in model organisms (Karaletsos *et al.*, 2012) and plant genetics (Bucksch *et al.*, 2014), but not for all kinds of phenotypes such images are available, e.g. for many human diseases. Health record information on patients is collected in growing numbers, both manually and by electronic devices (Gagnon, 2014).

These records have been used, for example, to examine the extent of correlation between different diseases and strong correlations have been found (Roque *et al.*, 2011). An open question is whether correlations between individual clinical covariates and overall disease diagnoses can be exploited to impute missing phenotypes.

In this study, we propose an algorithm that can use clinical-side information on a patient to impute missing disease phenotypes. We show that augmenting training datasets through this kind of *in silico* phenotyping may improve phenotype prediction accuracy.

The approach we propose is an instance of co-training (Blum and Mitchell, 1998), a well-studied machine learning algorithm that assigns class labels to unlabeled data points via a classifier that is trained on a second view of the data, i.e. a different set of features. Augmenting training datasets in this way has led to numerous successful applications, for instance in website category classification. The growing size of genotypic and clinical covariate datasets in genetics now enable us to examine whether co-training can improve disease phenotype prediction as well.

We explore the impact of our approach in a case study using two Dutch cohorts of migraine patients (Anttila *et al.*, 2010; Freilinger *et al.*, 2012). These patients were diagnosed as being affected by one of two subtypes of migraine known as migraine with aura and without aura. Two types of data were collected for all patients: clinical covariates and genotype data. In the original studies these data were used to find susceptibility loci for each subtype of migraine. Our goal with phenotype prediction is to apply the philosophy of co-training to construct a classifier on one view of the data to boost the prediction accuracy of another classifier constructed on the second view. The final classifier will predict if a patient should be diagnosed as having migraine with aura or without aura. Our analysis shows that even with a modest amount of labeled data, the approach we propose provides a massive improvement when predicting disease phenotypes from genotypes.

The remainder of this paper is organized as follows. We proceed to discuss related work before presenting our co-training approach to *in silico* phenotyping in Section 2. In Section 3, we employ our approach in order to improve disease phenotype prediction in two Dutch migraine cohorts that have been used by the International Headache Genetics Consortium (IHGC). We conclude by discussing the conditions that must be met for co-training to work, and give an optimistic outlook to its applicability for disease phenotype prediction in the future.

In Machine Learning, semi-supervised learning is a class of algorithms that utilize unlabeled data as well as labeled examples when training a model. 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 is low and the number of unlabeled instances is large. The reason for this imbalance is simply due to the high cost of labeling the data. This fits perfectly the description of the problem we previously outlined in which a multitude of genomic data is readily available but only a small fraction of it contains manually curated disease phenotypes.

For notation purposes, we will assume that a dataset $\mathcal{D}$ contains two classes of data: labeled ($\mathcal{L}$) and unlabeled ($\mathcal{U}$). In semi-supervised learning, the family of bootstrapping algorithms that learn from unlabeled data in an iterative manner, proceed in the following way (Dasgupta *et al.*, 2002): (i) build a classifier on $\mathcal{L}$, (ii) use it to assign labels to some instances in $\mathcal{U}$ and include these newly labeled data in $\mathcal{L}$. The classifier in (i) is then retrained and the process is iterated until a given stopping criterion is met (normally when $\mathcal{U}$ is empty or when a certain number of iterations is reached).

The co-training method (Blum and Mitchell, 1998) is a specific implementation of this generic approach which benefits from a natural split of the feature space in $\mathcal{D}$. In essence, an instance x is described by the set $\mathcal{X}$ of all available features in $\mathcal{D}$. For co-training, $\mathcal{X}$ is comprised of two mutually exclusive 'views' $\mathcal{X}_1$ and $\mathcal{X}_2$. In this way, a labeled object x can be referenced as $((x_1, x_2), y)$ where x_1 and x_2 are the values for the features in $\mathcal{X}_1$ and $\mathcal{X}_2$ respectively, 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 $\mathcal{L}$. Two important conditions must be met for co-training to be applicable: (i) x_1 and x_2 should be conditionally independent of each other given y and (ii) $\mathcal{X}_1$ or $\mathcal{X}_2$ are sufficient to train a classifier h_1 or h_2 that could classify the data points in $\mathcal{D}$. When this is the case, a good predictor h_2 can be learned from random classification noise from an initial weak predictor h_1 (Blum and Mitchell, 1998).

The work closest to our approach in genetics is missing phenotype imputation (Bobb *et al.*, 2011; Zhou and Stephens, 2014), in particular when dealing with high-dimensional phenotypes as in eQTL (expression quantitative trait loci) studies. These approaches differ from ours in that they try to predict individual missing phenotypes in a large set of phenotypes for some individuals. In contrast, we systematically annotate an entire unlabeled dataset with a one-dimensional binary phenotype.

2 *In silico* phenotyping

2.1 Setup

In order to illustrate the applicability of our *in silico* phenotyping approach in a clinical scenario, we will use a dataset of patients with two diagnostic types of migraine, termed migraine with aura and migraine without aura. In this dataset, a clinical diagnosis (disease phenotype) as well as clinical covariates and genotype data are available for each patient.

The methodology is built around three main partitions of the entire dataset, each of them consisting of different types and amounts of data:

- Set **I**, the training dataset: contains a subset of the patients for which all available information is present. For each patient there is a disease phenotype, a set of clinical covariates and genotype data surveyed for a specific set of single-nucleotide polymorphisms (SNPs). This complete set of information is normally hard or expensive to acquire, therefore the number of instances in this set will be relatively small compared to the sets described below.
- Set **II**, the co-training dataset: is similar to the training dataset, with the important difference that the patients lack a disease phenotype. Nevertheless, it has the advantage of being relatively large since its instances are deemed to be easier or less expensive to obtain.
- Set **III**, the evaluation dataset: is used for evaluation of the method. For each patient, it contains their genotype and known disease phenotype. It does not contain clinical covariates.

The above mentioned partitions of the data are illustrated in Figure 1a. Additionally, each partition shows the type of information it contains: clinical covariates, genotype data or disease phenotype information.

To understand the rationale of our method it is worth mentioning that the types of data mentioned above have an implicit price-tag

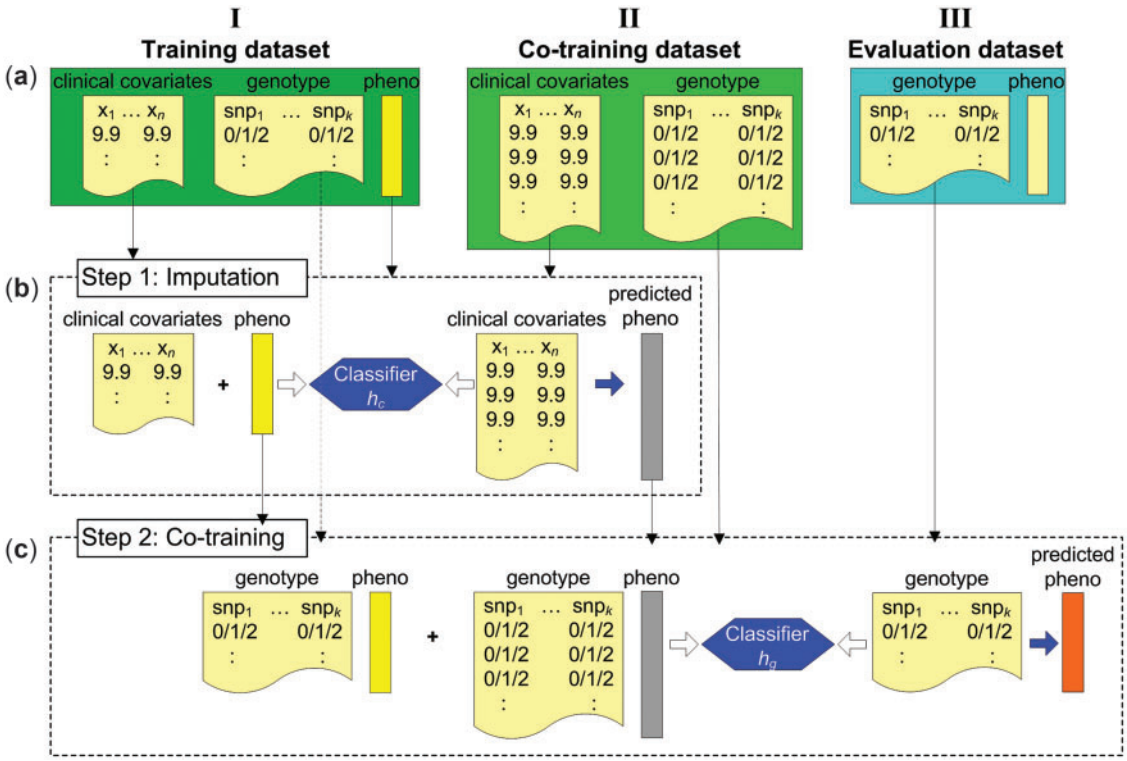

Fig. 1. Partitioning of the data and the proposed two-stage approach to co-training. Information about the disease phenotype (diagnosis) indicated as "pheno" and colored in *gold* (true labels) and *silver* (imputed labels). Clinical covariates shown as a generic real value (9.9). Genotype information coded as a model of additive effects {0, 1, 2}

associated to them. We assume that the *phenotype* information is the one with the highest cost as it involves multiple visits to a physician in order to obtain a clinical diagnosis. Normally, this diagnosis cannot be obtained without performing tests on the patient. The results of these tests are compounded into the *clinical covariates*, which constitute our second most expensive source of data. In addition to the normal tests that are conducted in a clinical setting, there may be a need to obtain genetic information about the patient's DNA. This is what we call *genotype data* and we consider it the least expensive of the three. This last assumption may be debatable at the time this manuscript was written but it is strongly supported by a trend in which the cost of whole-genome sequencing continues to dramatically decrease (Mardis, 2011; Wetterstrand, 2013).

Following the spirit of co-training, the two exclusive views of the data are the clinical covariates and the genotype data. Both of them comprise the full dimensionality of the dataset.

The sets I, II and III were obtained from performing 100 random splits of the entire migraine dataset. The relative sizes of the sets were 10% for I, 70% for II and 20% for III. In the case when patients were randomly assigned to set II, their true disease phenotypes were assumed to be unknown. The 100 splits were then used to determine the feasibility of the model as described in the next section.

2.2 Approach

The previous section described the structure of the data used by our *in silico* phenotyping approach. This section provides details of all the steps implemented in our methodology. These main steps are presented in Algorithm 1.

Algorithm 1: ***In silico*** **phenotyping via co-training.**

Data: Set I: training set with clinical covariates, genotype data and class labels $y = \{\texttt{aura}, \texttt{no\_aura}\}$
Set II: co-training dataset with clinical covariates and genotype data. No class labels
Set III: independent evaluation dataset with genotype data and class labels $y = \{\texttt{aura}, \texttt{no\_aura}\}$

Result: Two classifiers:
a clinical covariate classifier h_c to impute labels on II
a genotype classifier h_g to be tested on III

1. Use I to train a classifier h_c on the clinical covariates
2. Apply h_c on the clinical covariates in II. Predict the class labels for all elements in II
3. Use the true labels in I and the imputed labels in II to train h_g on the genotype data
4. Apply h_g to the genotypes in III and compare the predicted labels to the true labels

An important first distinction that can be made between the original co-training algorithm (Blum and Mitchell, 1998) and our method is that we do not follow an iterative approach. Our method consists of a two-stage process illustrated in Figure 1b and c.

- Step 1: the goal is to predict a disease phenotype for the patients in set II. To that effect, a classifier h_c is learned from the clinical covariates of the patients in set I. The classifier is then applied to the clinical covariate view of all instances in set II and disease phenotypes are predicted for each instance. These predictions, colored in grey in

Figure 1b, are considered 'imputations' of the missing disease phenotypes in II because of how they are utilized in the next step.

- Step 2: the disease phenotypes predicted in the previous step are used to augment the pool of labeled examples. A second classifier h_g is constructed via co-training by using: (a) the genotype data and true disease phenotypes in I and (b) the genotypes and predicted disease phenotypes in II. Finally, this classifier is tested on III to obtain the area under the ROC curve (AUC score). This is shown in Figure 1c.

The disease phenotypes for sets I and II in the figure are colored differently to distinguish the true (*gold*) labels from the imputed ones (*silver*).

2.2.1 Construction of the clinical covariate classifier h_c

An ensemble strategy, known as bagging predictors, was used to learn the clinical covariate classifier h_c. Bagging predictors (Breiman, 1996) generates multiple versions of a predictor and uses them to get an aggregated predictor. The aggregation typically outputs an average over all values when predicting a numerical output or conducts majority voting when predicting a class. Bagging predictors can be an effective methodology to generate high accuracy predictors (Breiman, 1996) and was the method of choice for our algorithm. In order to generate such an aggregated predictor for the clinical covariates, a machine learning model for the predictors had to be selected. In the case of the clinical data at hand, empirical evaluation led to the choice of logistic regression for the predictors. A vital element for the bagging predictors is that the predictors show instability (Breiman, 1996), which in our case was achieved by applying the following two techniques:

- the learning set was perturbed by randomly assigning 63.2% of it to every individual predictor. A total of 5000 of these predictors were generated.
- for every predictor, a random selection of $\sqrt{c}$ from all c features was assigned to the model, which is a common heuristic for generating these random subspaces.

These techniques created sufficient variability in the predictors such that, in the end, an effective aggregated prediction was made. For the aggregation of the logistic regressors, the mean over the class probabilities was used (Skurichina and Duin, 2002). As mentioned before, 5000 bagged predictors were averaged to get the final prediction.

2.2.2 Univariate feature selection, choosing the top k SNPs

Since our genotype data is of very high dimensionality ($\sim$ 500 000 SNPs), we conducted univariate feature selection before constructing the genotype-based classifier, in order to retain only the most relevant features. We performed this dimensionality reduction by determining the Pearson correlation coefficient between each SNP and the class labels (see Supplementary materials, Section S1.3). For each SNP j, the genotypes of all patients (coded as {0, 1, 2}) and the binary 0/1 class vector indicating migraine without aura and with aura were compared. This 0/1 vector with true labels is only present in the training dataset. For the co-training dataset this class vector comes in the form of soft labels with values $v_i \in [0, 1]$ for patient i. The soft labels were obtained using the aggregated logistic regression predictors described in Section 2.2.1. The Pearson correlation was able to deal with the two types of label vectors in concert by simply including them into the same computation. A P-value of the correlation was then computed for each SNP and all SNPs were ranked by their P-values. A low P-value corresponded to a high degree of association between the state of a SNP and the occurrence of the two subtypes of migraine. Then, the top k SNPs with the lowest P-values were selected as features for the h_g classifier. We set k to a default value of 2000 and the effect of varying k is described in Section 3.3.5.

2.2.3 Construction of the genotype classifier h_g

In order to train the classifier h_g on the genotype data, the top 2000 SNPs obtained in Section 2.2.2 were used as attributes. Another element that was necessary to train h_g were the class labels. Since most classification algorithms only accept clearly defined classes, the predicted (soft) labels in the co-training dataset that were obtained from h_c had to be binarized. This was done by ranking these prediction scores and selecting a cut-off such that the class priors, $p(y = \texttt{aura})$ and $p(y = \texttt{no\_aura})$ as estimated on the training dataset, were preserved. This is a grounded assumption by the fact that the data in the co-training dataset were sampled from the same distribution as the training dataset.

Due to this binarization, a supervised learning method that is both accurate and robust to potential mislabelings in the co-training dataset was needed. An attractive candidate was random forest (Breiman, 2001), which was ultimately used as our genotype classifier h_g. To train this model 10 000 weak predictors in the form of trees were created, using a 63.2% random data split. At each node in the tree, the common heuristic of selecting $\sqrt{k}$ random features was used ($k = 2000$ in our case).

3 Experiments

We explored the use of *in silico* phenotyping on a dataset of migraine patients.

3.1 Datasets

The migraine patients were part of the Leiden University Medical Center Migraine Neuro-analysis (LUMINA) programme. The recruitment of participants and the methods to collect the clinical and genotype data can be found in Anttila *et al.* (2010) and Freilinger *et al.* (2012) (see summary in Supplementary materials, Section S1.1). There were 1938 patients in total, of which 820 were clinically diagnosed as having migraine with aura and 1118 without aura. The two labels `aura` vs. `no_aura` constitute the patients' disease phenotypes.

The data available for each patient can be viewed as two mutually exclusive views: clinical covariates and genotype data.

Clinical covariates. They consist of binary labels extracted from questionnaires based on the International Classification of Headache Disorders (ICHD-II) guidelines (Headache Classification Subcommittee, International Headache Society, 2004) and aimed at diagnosing the subtype of migraine. The traits are: attack length, pulsation, unilaterality, aggravation by physical exercise, intensity of pain, photophobia, phonophobia, nausea and vomiting. For example, attack length is the answer to the following question: "Do you have headache attacks lasting between 4 and 72 hours?". Information on gender and age of onset was also collected.

Genotype data. Patients were genotyped using Illumina arrays that covered $\sim$500 000 SNPs. The two alleles reported for patient i and SNP j were coded according to a model of additive effects.

For each SNP j with $j \in \{1, 2, \ldots, s\}$, where s is the total number of SNPs, the genotype of patient i was coded as:

- 0 if the patient is homozygous for the major allele,
- 1 if heterozygous,
- 2 if homozygous for the minor allele.

The SNPs were pre-processed and removed from the analysis if they did not meet the following criteria:

- minor allele frequency > 0.01,
- Hardy-Weinberg equilibrium $> 1.0e^{-6}$.

SNPs were further filtered out if they were present in one cohort and not in the other, i.e. the intersection between the SNP arrays was preserved. After the filtering steps, a total of 463 825 SNPs were analyzed for each patient.

Systematic differences in allele frequency between patients with aura and without aura can result in misleading associations of certain SNPs. This problem is referred to as population stratification and is measured by the genomic control factor λ_{GC} defined as a median of χ^2 statistics (Devlin and Roeder, 1999). For the Dutch cohorts $\lambda_{GC} = 1.0529$ and, as a result of this, we safely assumed that population stratification was not a concern in this dataset (a value of $\lambda_{GC} < 1.1$ normally indicates absence of stratification). The filtering of SNPs, their coding to additive effects and the computation of λ_{GC} were performed with PLINK (Purcell *et al.*, 2007).

All data analyzed in this work were obtained from the IHGC with the appropriate approval from the centers in charge of their collection.

3.2 Experimental setup

The entire migraine dataset was partitioned into the training, co-training and evaluation datasets (sets I, II and III, respectively, as described in Section 2.1). Each sample could only belong to one of the sets. For the samples assigned to II, their disease phenotype was assumed missing and was therefore ignored. In a similar manner, samples assigned to III were assumed to only have genotype data.

One hundred (100) random partitions of the data were generated and the method was executed on each of them. From each partition, an area under the curve (AUC) score was reported. The final classification performance of the 100 random permutations was obtained as the mean AUC score accompanied by its standard deviation.

The experiments were run using Python (version 2.7.6) with Scikit-learn (version 0.15.2; Pedregosa *et al.*, 2011). Plots and other additional results were created in R (version 3.1.2).

3.3 Results

In order to systematically evaluate the utility of our *in silico* phenotyping approach, we conducted the following experiments: Firstly, we determined the lower and upper bounds for the prediction performance of the algorithm assuming perfect labeling of the data, We then examined how the actual prediction performance of *in silico* phenotyping compared to these bounds. Secondly, the effect of varying the amount of data in the training and co-training sets on the prediction performance of *in silico* phenotyping was established. Finally, using the relative sizes of data splits detailed at the end of Section 2.1, we empirically examined the effect of varying the number of SNPs that are selected in the univariate feature selection (Section 2.2.2).

3.3.1 Upper/lower bounds and the algorithm's performance

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 with their respective receiver operating characteristic (ROC) curves in Figure 2.

Lower bound. The genotype classifier h_g was trained only on I and evaluated on III. This was the least amount of data that could be made available to the classifier.

Upper bound. When h_g was trained on I + II using all the true labels (bear in mind that although II was assumed to have no disease phenotypes, these data were available to us). This was the best case scenario on which h_g was fed with the best possible data. It assumed that the clinical covariate classifier h_c performed a perfect imputation of the disease phenotypes on II.

Univariate feature selection on I. This refers to applying the entire co-training method as described in Algorithm 1 when the top SNPs used in h_g were selected only from I (true labels).

In silico phenotyping (co-training). This is the methodology we propose in this paper and also corresponds to Algorithm 1. Here, the

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 feature 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

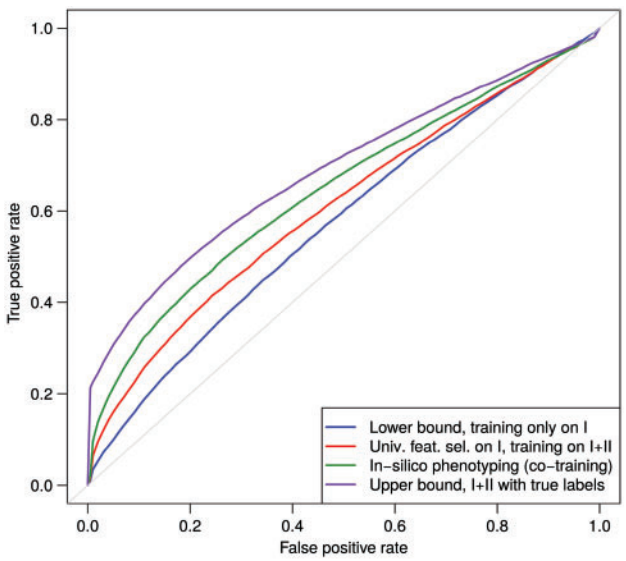

Fig. 2. ROC curves of bounds and prediction performance of *in silico* phenotyping. Partition of data into sets: I = 10%, II = 70% and III = 20%; 100 random folds

univariate feature selection was done on the augmented dataset I + II. In other words, the top SNPs used as attributes in h_g were chosen when a bigger dataset with true labels and predicted labels was available.

It can be seen in Table 1 that the *in silico* phenotyping approach via co-training (highlighted) is able to substantially improve the classification performance compared to the lower bound. Further, if the univariate feature selection is not performed on the co-training dataset the performance of the method is adversely affected. This leads to the conclusion that feature selection is an important performance driver in our co-training approach.

3.3.2 Varying the size of I (true labels)

In order to explore how the initial size of the training dataset affected the performance of our method, we conducted an analysis in which the size of the co-training dataset was fixed and the size of training dataset varied. In formal terms, 100 random partitions of the entire dataset into I, II and III were created such that their relative sizes were: 10% for set I, 70% for set II and 20% for set III. Sets II and III were fixed and the size of I was gradually decreased (from 193 samples to 19). Table 2 shows the results of this analysis.

On the one hand, the expected trend that more training data leads to a better performance is indeed observed. On the other hand, radically decreasing the size of the training dataset does not lead to the steep decline in performance that one might expect. Most notably is the fact that for smaller sizes of I the performance stays above the lower bound shown in Table 1. This indicates that even with a smaller labeled set, our co-training approach clearly brings benefits to the final performance of the classifier h_g.

3.3.3 Varying the size of II (unlabeled data)

Similarly to what was done in Section 3.3.2, we wanted to analyze the performance of our method when the size of the co-training dataset was varied. In this case, 100 random partitions were created with set sizes of 10% for I, 40% for II and 20% for III. Sets I and III were fixed and II was extended from 774 to 1356 samples (70% of the data). The results of this analysis are shown in Table 3.

The trend displayed in Table 3 summarizes the motivation of our approach: as the unlabeled set increases, the co-training approach benefits from a larger pool of unlabeled samples and the performance of the final classifier improves.

3.3.4 Varying simultaneously the sizes of I and II (comparison to lower bound)

The results shown in Tables 2 and 3 present a snapshot of the performance of co-training when one set is varied while the other one is fixed. To further understand the dynamics at play, we conducted a more comprehensive analysis in which we explored the entire spectrum of variations of sets I and II.

Table 2. Varying sizes of I and its AUC scores when II is fixed (size of II = 1356 samples, 70% of the data; 100 random folds)

Number of samples in I		AUC scores μ	AUC scores σ
193	10% of the data	0.646	0.029
96	5%	0.619	0.034
19	1%	0.605	0.035

To that effect, we created 100 random partitions of the data into sets I = 40%, II = 40% and III = 20%. For each of these random partitions, sets I and II were subdivided into a 10-by-10 grid corresponding to 10% subdivisions of the original data (40% in this case). The entire co-training analysis was conducted on each cell of the grid, for all random partitions. This amounted to a total of 10 000 executions of the method. Then, the mean AUC scores for every cell were computed.

This exact same procedure was repeated on the same random partitions and grid subdivisions mentioned above, but computing the lower bound instead. As described in Section 3.3.1, the lower bound was obtained by training h_g on set I and by testing on set III. The mean AUC scores were also computed for each cell. Finally, a delta AUC (ΔAUC) was obtained for each cell in the grid by subtracting the lower bound AUC from the AUC obtained through co-training. This ΔAUC represents the improvement brought by co-training for a given partition of the data. Figure 3 shows these results. The ΔAUC was chosen because of its robust properties in relation to other measures used to compare model performance (Hilden and Gerds, 2014). From the figure it is clear that the co-training approach outperforms the lower bound when the size of set I is relatively small compared to the size of II. This is seen by larger ΔAUCs in the top-left corner of Figure 3. With set I getting larger, the benefit of co-training is diminished.

3.3.5 Choosing the value of *k* for univariate feature selection

As described in Section 2.2.2, we set the number of selected SNPs by default to $k = 2000$. We examined the effect of this choice by the two types of analyses described below.

Table 3. Varying sizes of II and its AUC scores when I is fixed (size of I = 193 samples, 10% of the data; 100 random folds)

Number of samples in II		AUC scores μ	AUC scores σ
774	40% of the data	0.597	0.038
969	50%	0.604	0.035
1162	60%	0.611	0.035
1356	70%	0.646	0.029

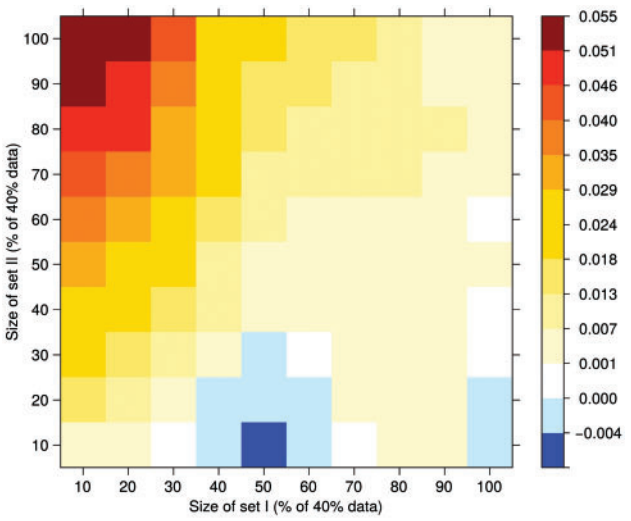

Fig. 3. Delta of mean AUC (*in silico* phenotyping vs. lower bound) for varying sizes of I and II

Table 4. Varying the number of selected features for the genotype classifier h_g (mean AUC μ and standard deviation σ). Row marked with asterisk (*) indicates the optimal value of k reported by the internal cross-validation

	AUC scores	
Number of top k SNPs	μ	σ
200	0.624	0.031
400	0.631	0.032
800	0.638	0.031
1600	0.644	0.028
2000	0.646	0.029
3200	0.648	0.028
6400	0.651	0.030
12 800	0.650	0.027
*25 600	0.648	0.029
51 200	0.643	0.026

Firstly, we applied our full co-training methodology by constructing classifiers h_g with varying numbers of k features. ($k \in \{200, 400, 800, 1600, 2000, 3200, 6400, 12\,800, 25\,600, 51\,200\}$). For each k, we computed the mean AUC of 100 random partitions of the data and report these values in Table 4. The results indicate that the mean AUC does not drastically improve for choices of k larger than 2000. In fact, the performance for $k > 1600$ flattens out and does not show a strong decreasing or increasing trend. Similar results have also been observed for a large number of other genotype-based classifiers on other datasets (Manor and Segal, 2013).

Secondly, we also performed an objective analysis to select the optimal number of features, k. To this effect, we compared different choices of k by performing internal cross-validation on the training data. The results of this analysis can be found in the Supplementary materials (Section S1.7, Table S4). A value of $k = 25\,600$ was found to be optimal in terms of mean AUC in the internal cross-validation. Still, when this k is used on the independent evaluation data (set III), the mean AUC is only slightly better than the mean AUC obtained for $k = 2000$ (0.2% improvement, see Table 4).

4 Discussion and conclusion

In this article, we have presented an approach to *in silico* phenotyping. It is the problem of imputing a disease phenotype for an individual with known genotype and side-information, such as clinical covariates or health records. We have shown that *in silico* phenotyping can be employed to systematically augment datasets on which models for phenotype prediction can be trained. This augmentation of the training dataset led to a drastic improvement in prediction quality when predicting subtypes of migraine phenotypes on patients from a subprogramme of the International Headache Genetics Consortium.

Several factors affect the ability to improve disease phenotype prediction through co-training. First, the original training dataset must be small enough to allow for any improvement by augmenting the training dataset. Otherwise, the classifier trained on the original dataset will already achieve a performance that can hardly be improved.

Second, the dataset on which *in silico* phenotyping is performed must be large enough compared to the original training dataset. Adding only very few new samples to the original training dataset will not significantly change the performance of the classifier.

Third, the side information that is used to predict missing disease phenotypes, such as clinical covariates or health record data, must be predictive for the given phenotype, while not being completely redundant to the genetic data. This means, a classifier trained on the side information may not give predictions that are highly correlated with those of the classifier trained on the original dataset (see Supplementary materials, Section S1.2). Otherwise, both will mis-classify exactly the same points.

All of these constraints seem rather restrictive, but in fact, we believe that they are met not only in our case study here, but also in many current datasets. In most problems of disease phenotype prediction, we are far from having enough samples to reach a point at which prediction could not be improved. Large sequencing projects create more and more genotypic data, and biobanks and electronic health record databases are collecting more and more clinical information on samples and patients.

It is important to note that we here focus on improving disease phenotype prediction through augmenting the genotypic training dataset. Despite the significant improvement that we obtain, further improvements may be achieved not only by another increase in the training dataset size, but also by including environmental or epigenetic factors, such as DNA methylation, into our model. That said, co-training can still be applied, even when working with these richer models, and may contribute to reaching the ultimate goal of making disease phenotype predictions accurate enough for clinical applications.

Acknowledgements

We thank the International Headache Genetics Consortium and everyone who has contributed to the collection and genotyping of the individual cohorts, as well as all the study participants.

Funding

This work was funded in part by the Alfried Krupp von Bohlen und Halbach-Stiftung (K.B.), and the Marie Curie Initial Training Network MLPM2012, Grant No. 316861 (M.J.W., K.B.). Additional funding was obtained from EU FP-7 EUROHEADPAIN (Grant No. 602633) (A.v.d.M.) and by the Netherlands Organization for Scientific Research [VIDI 917.11.319] (G.M.T.)

Conflict of Interest: none declared.

References

1000 Genomes Project Consortium *et al.* (2012) An integrated map of genetic variation from 1,092 human genomes. *Nature*, **491**, 56–65.

Allen,N.E. *et al.* (2014) UK biobank data: come and get it. *Science Trans. Med.*, **6**, 224ed4–224ed4.

Anttila,V. *et al.* (2010) Genome-wide association study of migraine implicates a common susceptibility variant on 8q22.1. *Nat. Genet.*, **42**, 869–873.

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. ACM, New York, NY, USA. pp. 92–100.

Bobb,J.F. *et al.* (2011) Multiple imputation of missing phenotype data for QTL mapping. *Stat. Appl. Genet. Mol. Biol.*, **10**: Article 29.

Breiman,L. (1996) Bagging predictors. *Mach. Learn.*, **140**, 123–140.

Breiman,L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.

Bucksch,A. *et al.* (2014) Image-based high-throughput field phenotyping of crop roots. *Plant Physiol.*, **166**, 470–486.

Dasgupta,S. *et al.* (2002) PAC generalization bounds for co-training. In: Dietterich,T. *et al.* (eds) *Advances in Neural Information Processing Systems 14*. MIT Press, Cambridge, MA, USA, pp. 375–382.

Davey,J.W. *et al*. (2011) Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nat. Rev. Genet.,* **12**, 499–510.

Devlin,B. and Roeder,K. (1999) Genomic control for association studies. *Biometrics,* **55**, 997–1004.

Freilinger,T. *et al*. (2012) Genome-wide association analysis identifies susceptibility loci for migraine without aura. *Nat. Genet.,* **44**, 777–782.

Gagnon,M.-P. (2014) A systematic review of factors associated to m-health adoption by health care professionals. In *Medicine 2.0 Conference*. JMIR Publications Inc., Toronto, Canada.

Headache Classification Subcommittee, International Headache Society. (2004) The International Classification of Headache Disorders: 2nd edition. *Cephalalgia,* **24**, 9–160.

Hilden,J. and Gerds,T.A. (2014) A note on the evaluation of novel biomarkers: do not rely on integrated discrimination improvement and net reclassification index. *Stat. Med.,* **33**, 3405–3414.

Karaletsos,T. *et al*. (2012) ShapePheno: unsupervised extraction of shape phenotypes from biological image collections. *Bioinformatics,* **28**, 1001–1008.

Manor,O. and Segal,E. (2013) Predicting disease risk using bootstrap ranking and classification algorithms. *PLoS Comput. Biol.,* **9**, e1003200.

Mardis,E.R. (2011) A decade's perspective on DNA sequencing technology. *Nature,* **470**, 198–203.

Pedregosa,F. *et al*. (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.,* **12**, 2825–2830.

Purcell,S. *et al*. (2007) PLINK: a toolset for whole-genome association and population-based linkage analysis. *Am. J. Hum. Genet.,* **81**, 559-575.

Roque,F.S. *et al.* (2011) Using electronic patient records to discover disease correlations and stratify patient cohorts. *PLoS Comput. Biol.,* **7**, e1002141.

Skurichina,M. and Duin,R. (2002) Bagging, boosting and the random subspace method for linear classifiers. *Pattern Anal. Appl.,* **5**, 121–135.

Wellcome Trust Case Control Consortium (2007) Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature,* **447**, 661–678.

Wetterstrand,K.A. (2013) DNA sequencing costs: data from the NHGRI Genome Sequencing Program (GSP). *National Human Genome Research Institute*.

Zhou,X. and Stephens,M. (2014) Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nat. Methods,* **11**, 407–409.