A Network Approach to Genetic and Expression based Phenotype Prediction on Mouse

Gianni Giusto
gianni.giusto@epfl.ch

Yann Mentha
yann.mentha@epfl.ch

Raphaël Reis
raphael.reisnunes@epfl.ch

Lucas Zweili
lucas.zweili@epfl.ch

Abstract

Personalized medicine has gain much attention in recent years with the drastic reduction in genome sequencing prices and systems genetics approaches such as GWAS or PheWAS have proven to be suitable tools for genome analysis. Only recently, the use of graph methods have been gaining popularity due to their inherent flexibility and high representative power. In this project we focus on the BXD mouse genetic dataset and use network properties as well as graph signal processing method to select and infer a phenotype based on the underlying genotype and/or protein expression levels.

1 Introduction

In the past decades, genetic medicine has made huge progress by allowing one to diagnose illnesses such as cancer or heart failure early enough in their development by sequencing the patient DNA and systematically analyzing it. However, such techniques are not totally foolproofed yet: indeed, the genetic background of a patient represents only one of the multiple complex causes for a given pathological phenotype, alongst with the epigenetic, the regulated gene expressions and other biological processes. Developing new tools able to take these other factors into account is therefore the new challenge scientists will face in the future in the domain of personalized medicine.

In this project we are aming at investigate phenotype prediction using the omic dataset[1] from the BXD mice family[2]. This dataset contains informations about genetics, protein expression and phenotypes for various mouse strains. Our goal consisted in a first step in finding suitable phenotypes to predict, before comparing multiple classifiers with and without the signal imputation and monitor the prediction improvement. Unfortunately, the dataset presented an extensive amount of missing values, challenge we did have to deal with throughout the project. In order to produce an accurate prediction, several different methods were implemented in order to estimate missing information by taking advantage of the network structure of the data. To do so, we heavily used graph signal processing (GSP) and Tikhonov regression methods.

2 Background and definitions

2.1 BXD strain context

BXD mouse family is a recombinant inbred (RI) strain. It means that each mouse has chromosomes that incorporates a permanent set of recombination event between chromosomes inherited from their genetically pure parents. These RI families are often used to map the locations of DNA sequence differences that contributed to differences in phenotype. To achieve such quantitative task, the strain is uniquely defined in the BXD family from a genetic point of view.

2.2 Single Nucleotide Polymorphism

A single nucleotide polymorphism (SNP) is a relatively common genetic variation of a single base pair in the genetic code which is responsible for large part of the genetic variation among a specie. However, to be considered as a SNP, the mutation must be relatively common within a strain (typically it must be at least in 1 percent of the population). Since these SNPs are very precisely located on the chromosomes and genes, they can somehow be used to determine the allele of a given gene, and more specifically from which parent the gene in question was inherited from. [3]

[1] Original dataset: http://www.genenetwork.org

[2] The BXD family is a set of mice which were derived by crossing two genetically specific mice (C57BL/6J (B6) and DBA/2J (D2)). Each offspring is uniquely defined in genetic terms.

[3] : https://www.ncbi.nlm.nih.gov/books/NBK44417/

3 Acquisition

The data consists mainly of three files: a *genetic* file where a binary value indicates from which parental genome a certain SNP is inherited from; a *phenotype* file recording the phenotype expression level (if continuous) or a class (if discrete) and several *expression* files where multiomic clinical and molecular phenotypes are recorded.

To build our graph, we used the *genetic* file where nodes represent mouse strains and edges, a similarity measure based on genetic information (*c.f.* Section 4 for details). Both continuous and discrete signals are defined for each mouse strain (node) as the expression level of a gene/protein or the class for a given phenotype (*e.g* hair color: brown, dark, etc.)

4 Exploration

4.1 Build the network

We built the graph based on the genetic information: this way, mice having a similar genetic material are linked stronger than mice with low genetic resemblance.[4] To do so, we used the cosine similarity measurement to compute the genetic distance between two mice, because of the categorical nature of genetics features. We then applied a gaussian kernel ($\sigma = 0.53$) and set the value of the resulting weighted edges below a certain threshold to zero ($\epsilon = 0.27$), to get a sparser graph. The parameters were carefully selected through grid search (cf section 4.3) in order to obtain a connected network required to perform further computations. The network is characterized in Table 1. We then performed basic community detection algorithms to grasp the way the network was connected (Figure 1). It showed four relevant communities in terms of modularity maximization. If we look closer, we see that the mice at the external zones share less edges with other communities than the nodes at the center. An interpretation could be that mouse strains in the middle inherited more likely SNPs pool than external ones. In a sense, border strains are "more extrem" genetic pools. However, since we have low average clustering coeff. and high diameter, we cannot say that the graph has a small-world property.

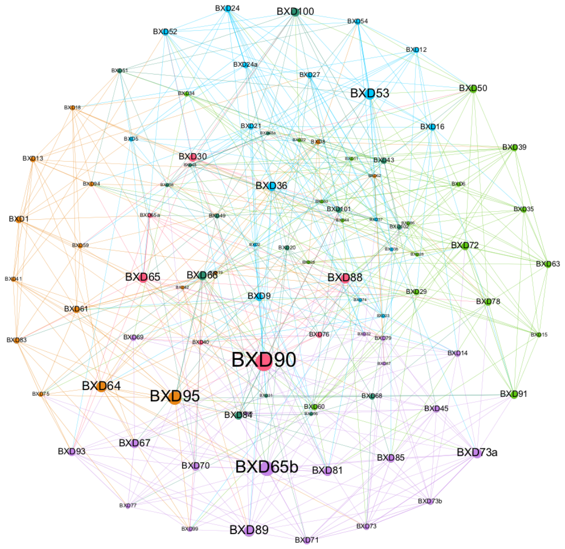


Figure 1: **Network's community detection visualization with Gephi [1]**. Each node represents a strain whose size is proportional to node degree. The color assignment is made after modularity maximization with a resolution of 1.3, randomization and usage of weights.

Table 1: Network characteristics.

Genetic graph	
Number of nodes	93
Number of edges	373
Graph density	8.72%
Average degree	8.02
Nb of connected components	1
Diameter of the network	6
Avg clustering coefficient	0.26

4.2 Network simulation

To acquire some insights on the nature of the graph we used genetic variance based on strain SNP distance. Since SNPs are randomly inherited from one parent or the other, it could make sense to argue that relations amongst the strains of the F1 generation are random as well, hence presuming for a random graph.

To verify our guess, we ran two network model simulations. The first model was an Erdős–Rényi (ER) random graph generator with probability for edge creation $p = \frac{2m}{n(n-1)}$ where n is the number of nodes and m the number of edges. The second model is a Barabási-Albert (BA) scale-free graph generator with a parameter $q = 2$ where q is the number of edges to attach from a new node to existing nodes during building phase.

Our network obtained the distribution shown in Figure 2. If we compare it to the ER and BA sim-

[4] *Note*: Building the graph on the expression data was tested as well (*c.f.* Section 4.3 for details), leading to overall poorer and less stable results for predicting the phenotypes of interest. The genetic-based graph was therefore preferred.

ulations from Figure 3, the network behavior from the genetic graph is clearly closer to the ER model than the BA one. Yet, this evidence is not sufficient to conclude that the genetic similarity graph is actually a random network, however we can assess that its node degree distribution is close to the latter.

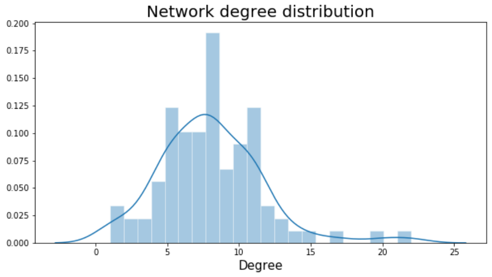


Figure 2: **Network degree distribution**.

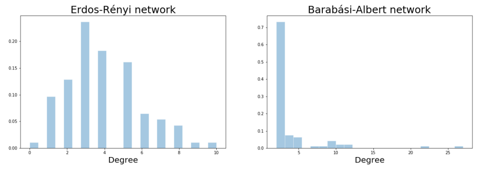


Figure 3: **Simulated degree distribution for two theoretical network models**. (Left) ER model with parameter n equal to the number of nodes and m to the number of edges. (Right) AB model with parameter $q = 2$.

4.3 Grid Search Approach

As the number of available phenotypes was consequent (5092) and that there was no guarantee for all of the phenotypes to be predictable from the provided dataset, we had to make a selection on the list of observable features we would try to predict. Only the most present phenotypes (the ones presenting the smallest amount of missing values) were predicted, due to the already low number of samples of the dataset (94 mouse strains). The phenotype are displayed in Table 2. Since no relevant result was found for continuous phenotypes, these are not displayed.

Table 2: Most present discrete phenotypes

Phenotype	Phenotype codes
Hair coat color	X62 X63 X64
Mitochondrial strain origin	X4473
Transthyretin protein serum band	X61
GABRA2 receptor expression	X76
Epoch or phase of production	X152 X111 X546 X1012

To do so, we implemented a first grid-search system performing Tikhonov regression/classification on a graph built either from the genetic or the expression dataset, following the method described above. After that, the graph nodes were divided into training and testing sets (0.8/0.2 ratio) on which we ran a Tikhonov regression/classification depending on the nature of the phenotype to predict(continuous/discrete). The grid search was executed over the metric, σ^2 the variance of the kernel, ϵ the cutoff and τ the Tikhonov regularization parameter.
One of the encountered challenge consisted in finding suitable metrics to compare the goodness of fit for both the continuous and discrete phenotypes. R2 score was used for the regressions tasks, but no significant results were found. Concerning classification, Matthew correlation coefficient (MCC) was chosen as the reference metric as it is suitable both for binary and multiclass classification, since it takes into account the balance ratios of the four confusion matrix categories. In addition, many classes were highly imbalanced making accuracy an inappropriate metric for methods comparison. The sorted results of the grid-search for the expression-based and genetic-based graphs are respectively shown in Table 3 and 4

Table 3: Expression Based GS

PhenoCode	Phenotype	MCC	Accuracy
X111	Epoch/phase production	1.0	1.0
X152	Epoch/phase production	0.88	0.93
X546	Epoch/phase production	0.84	0.93
X1012	Epoch/phase production	0.79	0.91
X62	Hair color	0.39	0.54

Table 4: Genetic Based GS

PhenoCode	Phenotype	MCC	Accuracy
X111	Epoch/phase production	0.61	0.8
X62	Hair color	0.38	0.55
X64	Hair color	0.36	0.52
X63	Hair color	0.36	0.51

Using the above described approach, two families of phenotypes could be isolated: the first one was the ”BXD epoch or phase of production” for several traits, the second is the hair coat color. For the ”Epoch or phase production” phenotypes (phenotypes X111, X152, X546 and X62), an unexpectedly high accuracy was obtained (1.0 for X111) classifying the data perfectly into 2 sets of distinct mice. However, this result was obtained through a simple logistic regression as well.

Unfortunately, the description of the phenotype in question was not explicit about its nature, and no additional resource could be found concerning the subject. The phenotype was therefore assumed to be irrelevant and no further investigations were made in that direction.

Concerning the hair coat color (a discrete phenotype with 4 categorical values) an average accuracy of 0.55 was obtained on 1000 random train/test sets with the optimal parameters of the grid search, based on the genetic graph. Similar results (0.59 of accuracy on 1000 random train/test sets) were obtained using a logistic regression based on the genetic dataset. But the phenotype for which high chances to obtain relevant results for was set at that point, it remained to increase this accuracy and assess whether graph methods would be able to overpass linear methods.

5 Exploitation

In order to find relevant results for the X62 phenotype (hair coat color) two distinct approaches were followed:

- **Feature imputation**: The underlying assumption relies on the fact that expression information does not suffice to explain the concerned phenotype, and aims at using the genetic data in addition to it. Since it missed a lot of expression data, we imputed it using Tikhonov regression method. To assess improvement, we compared it to naive feature imputation with an empirical mean.

- **Phenotype prediction**: The underlying assumption was that genetic data sufficed to explain the hair color. Several graph-methods and linear models were tested to do so.

5.1 Feature Imputation

5.1.1 Baseline

The current method does not take the graph into account and is hence referred as *baseline*. We first perform a feature reduction using recursive feature elimination (RFE). The method build a first model using all the available feature and progressively prune the least important ones. The selected features are then passed as input to the classifier in order to determine the categorical phenotype X62 responsible for hair color (for the reasons mentioned in section 4.3). At this step, missing values are simply inferred using the mean of each feature (*i.e.* gene expression level) prior feeding the data to the classifier, mean imputation has the advantage to not change the sample mean of the missing feature for other samples. For this classification task we test one linear model, namely *logistic regression*. Results are shown in Table 9.

5.1.2 Tikhonov regression imputation

Then we imputes the expression data missing values, we used the graph structure with a Tikhonov regression. Meaning that for each node, the missing expression feature is imputed solving the following minimization problem:

$$argmin_x \|Mx - y\|_2^2 + \tau\, x^T Lx \qquad (1)$$

5.1.3 Logistic regression following

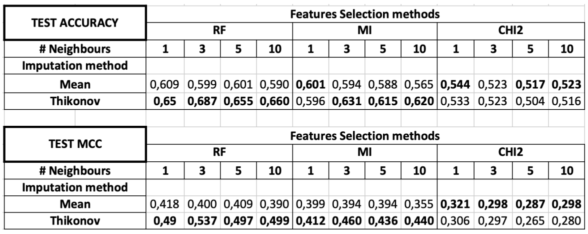
As a second step, we build a classifier that aims at predicting node phenotype (label) based on molecular expression and linkage properties of the network. We expect a result suggesting that missing values inference from Tikhonov regression brings relevant information from the graph structure to our classifier model. Unfortunately, for a similar logistic regression model, we reach an accuracy of 0.29 compared to 0.37 for the baseline. This motivates the use of GCN (*c.f.* 5.1.4).

5.1.4 Graph Neural Network

Finally, we also test a Graph Neural Network to see if it is capable to better exploit, due to its Graph nature, the inferred values from Tikhonov regression. The idea behind the GNN is to do a weighted average of a certain number of neighbours features to predict the class of the actual node. To get robust results we test our model on three different subsets generated by three features selection methods; Random-Forest (RF), Mutual information (MI) and Chi-squared (CHI2). We also vary the number of neighbours taken into account. As we can see in figure 4, the model performs better mostly under Thikonov imputation method, as expected. Best results: 0.687 Acc. 0.537 MCC (imputation=Thikonov, subset=RF, Neighbours=3).

5.2 Phenotype Prediction

As accuracy seemed to reach a ceiling in Section 5.1, the efforts were concentrated on simpler models taking exclusively the genetic information into account.



TEST ACCURACY	Features Selection methods											
	RF				MI				CHI2			
# Neighbours	1	3	5	10	1	3	5	10	1	3	5	10
Imputation method												
Mean	0,609	0,599	0,601	0,590	**0,601**	0,594	0,588	0,565	**0,544**	0,523	**0,517**	**0,523**
Thikonov	**0,65**	**0,687**	**0,655**	**0,660**	0,596	**0,631**	**0,615**	**0,620**	0,533	0,523	0,504	0,516

TEST MCC	Features Selection methods											
	RF				MI				CHI2			
# Neighbours	1	3	5	10	1	3	5	10	1	3	5	10
Imputation method												
Mean	0,418	0,400	0,409	0,390	0,399	0,394	0,394	0,355	**0,321**	**0,298**	**0,287**	**0,298**
Thikonov	**0,49**	**0,537**	**0,497**	**0,499**	**0,412**	**0,460**	**0,436**	**0,440**	0,306	0,297	0,265	0,280

Figure 4: Accuracy and MCC comparison for Mean and Thikonov imputation methods

.

5.2.1 Harmonic function

This method is based on the algorithm proposed by Zhu *et al.* [2]. In their work, they describe a semi-supervised approach based on a Gaussian random field model to infer the class of unlabeled nodes in weighted graph where edges represent similarity between two nodes (as in our case). This method only exploit link properties between nodes but does not take into account features related to those nodes. Nevertheless, it achieves 56% accuracy (4-fold cross-validation) on the grid-search optimized genetic-based network ($\sigma = 0.52$, $\epsilon = 0.27$) (Table 10).

5.2.2 Logistic regression and RFE

As 7324 genes were present in the genotype dataset, it was assumed that most of the features were irrelevant and only adding some noise to both the linear models and the graph methods. In order to reduce this number, we used Recursive Feature Elimination (RFE) using logistic regression as our reference method. By keeping only 10 features, the MCC and the accuracy for the $X62$ phenotype (hair color) passed respectively from 0.47 and 0.6 to 0.98 and 0.99. By further isolating the two features which obtained the highest absolute-valued weights in the logistic regression, we reduced our feature space to only two SNPs (rs30336558 and rs32862298), without decreasing neither the MCC nor the accuracy. A quick search on our reference database [5] allowed us to locate these SNPs in the genome. Further researches on hair-color responsible genes within BXD strains [6] highlighted essentially two genes, Myo5a and Tyrp1, whose positions in the genome were determined in a similar manner.

As shown above, it seems like the two SNPs are located exactly on the concerned genes, justifying such high accuracies on test sets. Interestingly, most of the measurements for these two genes on BXD strains were coming from the EPFL/LISP laboratory, justifying further such results.

The second step consisted in defining whether graph methods would benefit of such feature reduction methods as well. To do so, a graph was built and optimized through grid-search as described previously on the genetic data first with the 7324 features and then with the rs30336558 and rs32862298 ones exclusively. Results are shown in Figure 6 and 5.

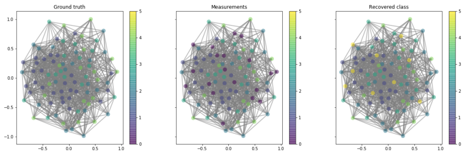


Figure 5: Graph obtained on the 7324 features ($\sigma = 0.43, \epsilon = 0.1$) The figure shows one run of the cross-validation. (Legend) test set=0, dilute=1, brown=2, grey=3 black=4 correct, classification=5

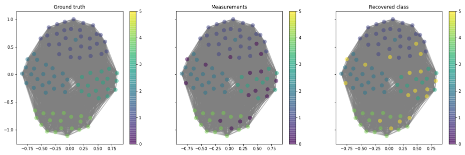


Figure 6: Graph obtained on rs30336558 and rs32862298 features ($\sigma = 0.31, \epsilon = 0.004$)

As we can see, graph methods (tikhonov classification in this case) do benefit from feature selection passing from an (accuracy/mcc) of (0.52/0.4) to (0.98/0.98) (*1000 random train/test sets 0.8/02 ratio*) reaching similar results as logistic regression after a bit of tuning, reflecting the representational power of such methods. A similar approach for the Transthyretin protein serum band (X61) phenotype was followed and ended up on the result shown in Figure 7 and 8 where the location of the gene was gain appropriately estimated.

6 Discussion

Missing values are common when we work with data. In our case, it is exacerbated since the data from different laboratories are put together. In such cases, it's important to researchers to get powerful and accurate way to recover missing values. Our network-based approach on GNN classifier showed encouraging improvement in both prediction accuracy and MCC score (see Fig. 5)

[5] http://www.genenetwork.org/

[6] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3185026/

compared to feature imputation by the mean. The benefit of the method used here could be that it used other relevant data. However, lots of methods are out and would deserve a comparison to assess quality of the method we used. This could be a matter for later work.
As the genetic-based graph results showed, too many features do not necessarily benefit the prediction of a given phenotype: it rather adds undesired noise, both for linear and graph-based methods. In such cases, RFE showed particularly good results, allowing one to track down the essential features and rediscovering the locations of responsible genes for various phenotypes. Once isolated, graph methods proved doing at least as well as linear ones.
However, in the present investigation we managed to obtain such results mainly due to the discrete nature of the studied phenotypes and their bijective relation with the respective genetic background: a lot of phenotypes do not show such behavior, especially continuous phenotypes. The skin color in humans is regulated for instance by more than eleven loci in the genome[7]. A network approach could potentially lead to relevant results, and take advantage of this diversity when assessing the similarity between individuals.

References

[1] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. *Gephi: An Open Source Software for Exploring and Manipulating Networks*. 2009.

[2] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions". en. In: (), p. 8.

7 Appendix

Table 5: SNPs positions for hair color

SNP	Location
rs30336558	Chr9: 74.557367
rs32862298	Chr4: 81.455696

Table 6: Myo5a and Tyrp1 locations

Gene	Location	Max LRS Location
Myo5a	Chr9 75.223099	Chr9: 74.557367
Tyrp1	Chr4: 81.455696	Various chrom

Table 7: SNPs positions for Transthyretin

SNP	Location
rs8271271	Chr12: 104.154182
rs29190933	Chr12: 103.283977

Table 8: Location of the Transthyretin responsible gene

Gene	Location	Max LRS Location
Trait 13034:	Chr12:104.2-105	Various chrom

Table 9: Performances of logistic regression

Method	Accuracy	MCC
Log reg (7324 features)	0.60	0.47
Log reg (2 features)	0.98	0.99

Table 10: Performances of different graph methods.

Method	Accuracy	MCC
Harmonic (7324 features)	0.56	0.39
Tikhonov (7324 features)	0.52	0.4
Tikhonov (2 features)	0.98	0.98

[7] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3317488/