Random subsampling techniques for sea bass mortality prediction

Giovanni Gaio, Simone Moretti

July 18, 2025

1 Introduction

The use of genomes to predict the effects of disease has been largely celebrated. But actually finding such links is usually not easy, and an automatic data-driven technique to predict disease outcome would be very useful. Given the size of genomes it is not always easy to understand the genes that have an phenotype related to the disease effects. The size of the datasets hinders the efficacy of standard machine-learning techniques, as few example genomes are available but each is composed of hundreds of thousands of bases, if not many millions.

The scope of this work is to experiment with some simple random subsampling techniques. The idea is to keep only a subset of the genome bases while keeping all the individual genomes sequenced. The hope is that this will allow a standard machine-learning algorithm to better understand the structure and importance of the gene with respect to the disease outcome.

2 Methodology

We tried to evaluate the effectiveness of random subsampling of genes in improving predictions of mortality.

2.1 Dataset

The main dataset we used was a table filled with the genome of 990 individual sea basses. The sequenced genome at our disposal was made up of 6072853 individual genes.

In addition to the set of genes for some individuals, we had a set of annotated genes. For a subset of genes some further information is known: a function (either `Open_chromatin`, `Enhancer` or `Promoter`) and a tissue number (between 0 and 25).

2.2 Data pipeline

1. Subsampling. The idea is to help our estimator to not get lost in the $\sim 10^6$ genes, but allowing it to work with only a smaller number of genes at a time. This step is thourghly explained in subsection 2.3
2. Train-test split.
3. Training. Once we subsampled the dataset, we used the `XGBoost`[1] library to construct a predictor in a fast and easy manner. We always used the parameter `method="hist"`, as we didn't find any meaningful difference between the options and this offers the widest compatibility across devices.
4. Evaluation.

randomisation...

2.3 Subsampling techniques

We always subsampled selecting a subset of genes in the whole dataset.

Uniform subsampling. The first and simplest thing we tried was to randomly and uniformly subsample the genes on the entire genome. We selected a given fraction of the genes, keeping or discarding each gene with fixed and uniform probability.

Uniform subsampling on chromosomes. A second possibility is to uniformly and randomly sample a fixed number of genes on each chromosome.

2.4 Annotated genes

Using the additional information at our disposal we used only a subset of annotated genes. Among these we further subsampled to obtain a small feature set.

Subsampling using annotations. Some genes have been linked to specific organs, tissues or functions. This information can be used to select and sample genes with only some function or organ and subsample among these.

3 Results

In this section we'll look at the results we got from each of the different techniques outlined in section 2.

Execution notes. All the results were obtained by running our program with about 16 GB of working memory and 1 core per instance. The system running our program was equipped with an Intel Xeon [1]. The runtime for a sweep on the selected subsample rates is on the order of a few hours. Though note that for different subsampling ratios memory requirements and runtime changed by over an order of magnitude in the extreme cases.

Baseline results. To understand the improvement given by our proposed techniques, in every plot we compared our scores with some "baseline scores". These baseline scores are the score of a "trivial" classifier, that always reports the most common class. Our data samples were in 54.32% of cases of one class, thus the baseline accuracy is this value.

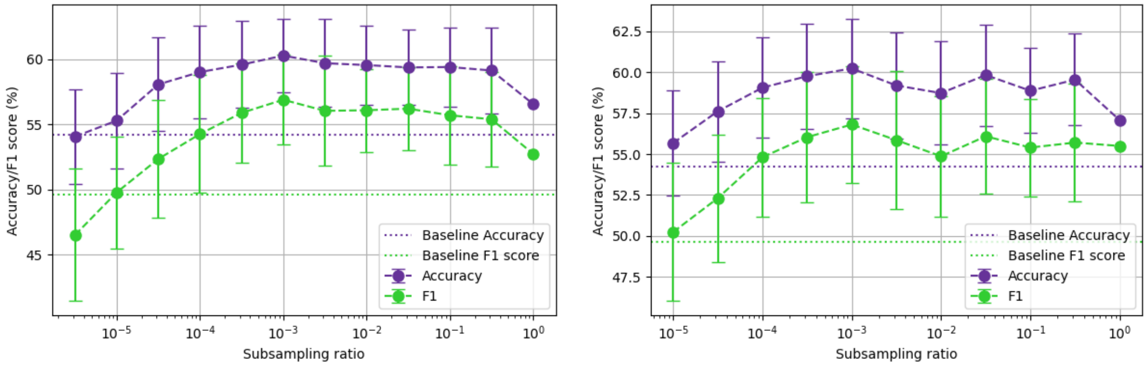
3.1 Subsampling on the whole genome

The first simple tests were done selecting a random subset of genes among all the SNPs that we had. The results for each subsampling rate are shown in Figure 1. These plots show a mostly flat picture: both for the accuracy and F1 score there isn't a definite trend or variation as a function of the subsampling rate.

The only exception to this is Figure 1a, that shows a diminishing trend with the smallest subsampling rates. Nevertheless both of the last two data points are compatible with the middle points.

Remark that our experimental setup is designed to have no random components out of the subsampling step, thus no subsampling implies no randomization. For this reason the values for subsampling rate $1 = 10^0$ (no subsampling) are the same in the two plots of Figure 1 and have no error bars. This single values could be slightly far from than the true expected value because of random variation in

[1] The cluster we worked on, is more thoroughly described here: `https://docs.dei.unipd.it/en/CLUSTER/Overview`



(a) Plot of the scores when subsampling uniformly on the whole genome.

(b) Plot of the scores when subsampling uniformly on each chromosome.

Figure 1: Plots of accuracy and F1 scores for different subsampling rates on the whole genome.

the results. Thus the slight decrease between the ”middle” subsampling rates and this last data point is probably not meaningful.

3.2 Annotated genes subsampling

The second set of tests was done using only the annotated genes. The results on each set of similarly annotated genes are shown in Figure 2: Figure 2a shows the results using function annotations, and Figure 2b the results when selecting genes with tissue number in different intervals.

As before, these plots show a mostly flat picture, both for the accuracy and F1 score.

3.3 Considerations on the results

Overall the results do not show any meaningful trend. The most likely pattern to be inferred from our data is a flat trend, meaning that the scores and subsampling rates are nearly independent. This trend seems to hold well outside the extreme ends of our sampled interval of subsampling rates.

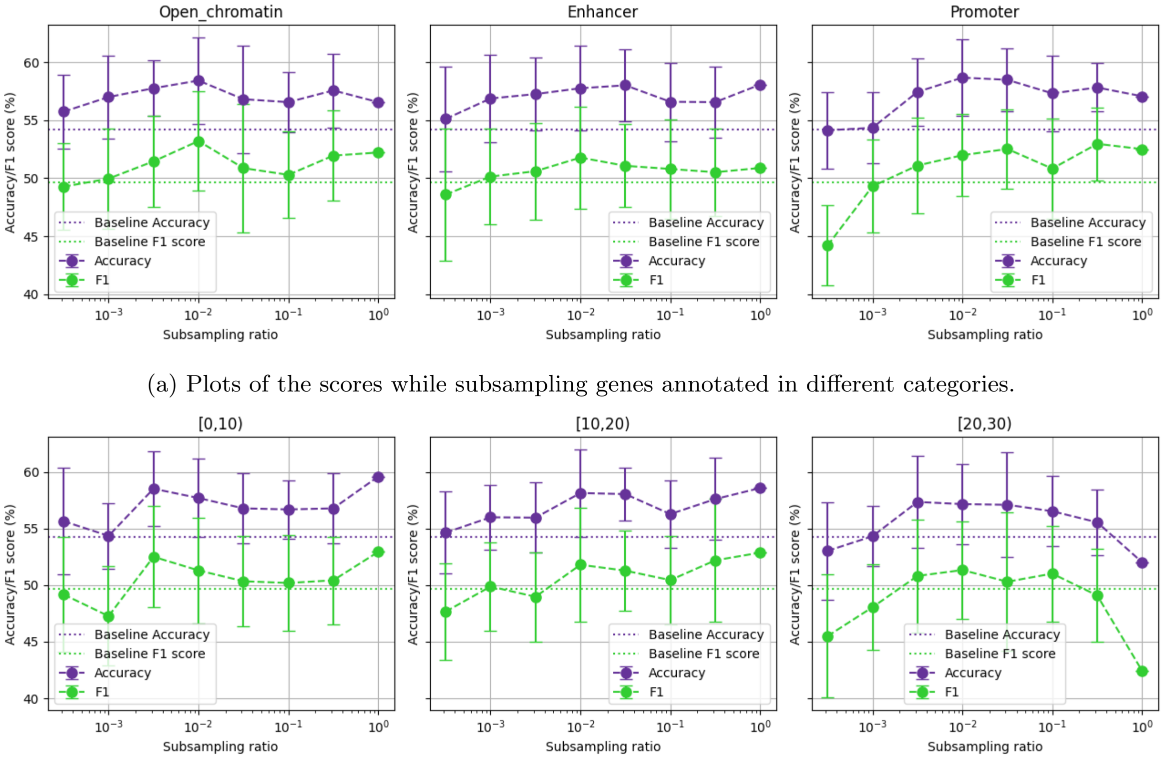
In fact, observing the lower extreme, a simple observation can be drawn from our results: keeping only a very small subset of genes (in the order of tens or less) has a negative effect on the scores. This can be seen especially in the ”Promoter” plot of Figure 2a or in Figure 1a, but is probably the cause of the lower scores for lower subsampling rates in other plots. More extreme values of the subsampling rates have been tried and support this hypothesis, though the results are not included in this work.

The flat trend could be ultimately a good thing: this could allow future experiments of better models, hyperparameters or techniques to use only a subsample of the data without losing quality in the results. This would enable the use more complex models and faster testing.

4 Conclusions

References

[1] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: http://doi.acm.org/10.1145/2939672.2939785.



(a) Plots of the scores while subsampling genes annotated in different categories.

(b) Plots of the scores while using only genes with tissue number in a given range.

Figure 2: Plots of the accuracy and F1 scores for the different subsampling rates on the annotated genes. Each single plot is done using only the genes annotated with the indicated values.