Modeling Gene Transcriptional Regulation by Means of Hyperplanes Genetic Clustering

Fabrizio Frasca*, Matteo Matteucci†, Marco Masseroli‡ and Marco Morelli§
*†‡Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza L. Da Vinci 32, 20133 Milan, Italy
§Center for Genomic Science of IIT@SEMM,
Istituto Italiano di Tecnologia (IIT), 20139 Milan, Italy
*Email: fabrizio.frasca@mail.polimi.it †Email: matteo.matteucci@polimi.it
‡Email: marco.masseroli@polimi.it §Email: marco.morelli@iit.it

***Abstract*—In the wide context of biological processes regulating gene expression, transcriptional regulation driven by epigenetic activity is among the most effective and intriguing ones. Understanding the complex language of histone modifications and transcription factor bindings is an appealing yet hard task, given the large number of involved features and the specificity of their combinatorial behavior across genes. Genome-wide regression models for predicting mRNA abundance quantifications from epigenetic activity are interesting in an exploratory framework, but their effectiveness is limited as the relative predictive power of epigenetic features is hard to discern at such level of resolution. On the other hand, an investigative analysis cannot rely on prior biological knowledge to perform sensible grouping of genes and locally study epigenetic regulative processes. In this context, we shaped the "gene stratification problem" as a form of epigenetic feature-based hyperplanes clustering, and proposed a genetic algorithm to approach this task, aiming at performing data-driven partitioning of the whole set of protein coding genes of an organism based on the characteristic relation between their expression and the associated epigenetic activity. We observed how, not only the hyperplanes described by the resulting partitions significantly differ from each other, but also how different epigenetic features are of diverse importance in predicting gene expression within each partition. This demonstrates the validity and biological interest of the proposed computational method and the obtained results.**

***Keywords*—Gene expression, Epigenetic transcriptional regulation, K-planes regression, Genetic clustering**

I. Introduction

Gene expression regulation is an essential, yet very sophisticated mechanism to increase or decrease the production rate of gene products such as proteins or, more generally, RNA. This regulatory phenomenon is important for both simple and more complex forms of life, providing the ability to dynamically adapt to mutating environmental conditions and enabling cellular differentiation. Gene regulation actually involves the coordination of several complex mechanisms acting at different locations and stages of the expression of a gene. Among the biological processes most effective in this task there are the ones involved in transcription initiation [1], the majority of them occurring at the epigenetic level, i.e., just acting 'on top' of the genome by means of chemical modifications and protein bindings. In this context, the main role is played by the combinatorial interactions between transcription factors [2], [3], and histone modifications [4], [5] (respectively TF and HM in the following). The challenge of understanding how histone modifications and transcription factors regulate gene expression in a cell is particularly enticing as many human diseases have been shown to be caused by the alteration of the expression levels of some genes by abnormalities in TF and HM combinatorial patterns [3].

Recent advancements in sequencing technologies, along with joint efforts from laboratories and research centers all over the world, have enabled extensive genome-wide measurements of epigenetic processes and the sharing of such measurements in an organized and catalogued manner. The ENCODE (ENCyclopedia Of DNA Elements) consortium is building a comprehensive parts list of functional elements in the human genome [6]; it has set up a large repository to make the results of various sequencing experiments publicly available, along with essential metadata dealing with experiment settings and quality. Currently, the ENCODE repository hosts measurements from Chromatine Immuno-Precipitation sequencing (ChIP-seq) experiments for a large number of epigenetic 'features' such as TFs and HMs.

Taking advantage of such available data, attempts have been made to conceive statistical predictive models for the mRNA abundance of a cell from corresponding ChIP-seq data on TFs and HMs at a genome-wide level [7]–[9]. The importance of genome-wide models as a powerful explorative framework has been remarked in [10], as they allow drawing more general and fundamental conclusions on the roles and interplay of TFs and HMs. On the other hand, these two classes of features seem to be statistically redundant for this task. In [10] statistical redundancy has been well distinguished from *functional redundancy* and has been shown how redundancy at genome-wide level breaks down at the resolution of groups of ontology-classified biological processes, where variations in the relative predictive power of TFs and HMs are observed. Thus, it is interesting and useful to design statistical models still on a genome-wide scale, but taking into account the characteristics of these epigenetic features to have different relative predictive capabilities for different gene subsets.

The comprehension of important epigenetically-driven dynamics in the mRNA production of a cell can be cast as the task of finding *groups* of HMs and TFs which sufficiently well explain measured mRNA quantifications for *groups* of genes. In this case, it is important to stress not only how the information about the mRNA gene quantifications must consistently drive the gene partitioning procedure, but also how both the grouping for genes and epigenetic features should arise directly from data, when analysis is carried out with investigative purposes. With respect to the first point, we do not aim at finding genes with similar input patterns or expression levels, but rather, genes with common kinds of *correlations* between their epigenetic status and expression. As for the second point, focusing on ontology-classified biological pathways is not suitable, if not in contrast, with an exploratory analysis, as it requires prior biological knowledge on the cell under study.

It is then within the general context of regression analysis, where we frame the described problem, considering genes of interest as samples, epigenetic measurements as input features and mRNA quantifications as target values. In this context, and aiming at retaining maximum model interpretability, finding groups of genes/samples corresponds to clustering *hyperplanes*, each representing the solution of the ordinary least squares problem over a certain group of genes.

In this paper, we present the application of a genetic algorithm to perform such clustering of hyperplanes. Then, we show how we perform forward step-wise feature selection to extract the most predictive features for each partition found. Furthermore, we explore the dissimilarities between the separated hyperplanes and investigate the possibility of enhancing genome-wide prediction capabilities by means of the clustering performed by the evolutionary procedure.

The remainder of this paper is organized as follows. In Section II we describe related works. Section III is devoted to the discussion of the data used - including how they are retrieved and preprocessed - and the design choices concerning the genetic algorithm. In Section IV we analyze and validate the results of the partitioning procedure in terms of feature rankings, model dissimilarities and possible improvements on the regression task. Finally, conclusions in Section V.

II. Related works

The problem we are addressing is similar to what in the literature is termed as *k-plane clustering*, or more generally as *piece-wise linear affine model fitting*. One of the most common approaches of this kind is the *hinging hyperplane* method [11], which can be considered as a more refined version of the regression tree one, aiming at overcoming the drawbacks of this last, such as convergence to suboptimal solutions. Considered the hinge function as the maximum or minimum of two affine functions, the objective is to approximate the regression function as a sum of these hinge functions. Another possibility is given by the *bounded error approach* [12], in which the objective is to learn a piecewise linear regression function such that, for every point in the training set, the absolute difference between the target value and the predicted value is less than ϵ. This kind of problem is referred as the *maximum feasible sub-system problem*, shown to be NP-hard in [12], where solution is approached by means of several proposed heuristics.

These techniques are mainly designed to fit a supposed non-linear dynamic with piecewise linear functions in order to retain interpretability of the results, but are not well suited to learn possibly discontinuous functions, as it might be the case of our study. In our case, we are not interested in approximating a non-linear dynamic; whereas we aim at learning different linear models in a scenario where dynamics are likely to be overlapped, mixed, and partially lying on sub-dimensional manifolds.

A more suited approach was proposed in [13] and is termed as *k-plane regression*. Its objective is fitting possibly discontinuous functions based on a clustering approach. The main idea is to partition the input points and to learn a linear model for each of those partitions. *K-plane regression* finds a predefined number of hyperplanes such that each point in the training set is close to one of the hyperplanes, and all points in the k-th partition are as closest as possible in the input feature space. That is to say partitions are found by minimizing the following objective function:

$$E(\Theta) = \sum_{k=0}^{K-1} \sum_{i \in \Theta(k)} (t_i - \tilde{\boldsymbol{w}}_k^T \tilde{\boldsymbol{x}}_i)^2 + \gamma \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2^2 \quad (1)$$

where K is the pre-defined number of clusters, $\Theta(k)$ represents the set of samples in cluster k according to Θ - input feature $\tilde{\boldsymbol{x}}_i$ and target value t_i, $(i \in \Theta(k))$, $\boldsymbol{w}_k$ is the weight vector of the least square solution for those points, $\boldsymbol{\mu}$ terms refer to centroids in the feature space and γ is a user defined parameter deciding the relative weight of the two terms in the objective function. The 'tilde' notation is used to indicate the inclusion of the bias term in the regression. The error function $E(\Theta)$ is minimized by an Expectation-Maximization (EM) procedure. The second part of the equation, i.e., the term related to the closeness of the points belonging to the same partition, was introduced in [13] to avoid EM finding suboptimal solutions, and to enforce found partitions not to contain points from disjoint regions of the input feature space.

In the context of transcriptional regulation modeling, possible multi-functionality of epigenetic markers for different gene subsets has been addressed in [14] with a mixture of sparse linear models. The authors focused on four hematopoietic cell types, considering mRNA abundance quantifications as a function of histone mark signals and transcription factor binding affinities. By resorting to a Maximum A Posteriori (MAP) version of the EM algorithm, an ensemble of bayesian elastic nets is fitted on data. Not only the model parameters are estimated, but also the posterior probabilities (or responsibilities) for each observation to belong to a certain model. In so doing, however, genes are not 'sharply' grouped, since a precise partitioning of the gene set into distinct regulative dynamics is not defined. Rather, the expression for a gene is modeled by the weighted sum of the outputs of *all* the fitted linear models.

Differently from [14], in our work we would like, instead, to perform a 'sharp' partitioning, defining clusters of genes sharing a common kind of epigenetic regulative behavior. This is the reason why our method is more similar to the approach of *k-plane regression* proposed in [13], where, however, we explicitly drop the second term of the objective function in Equation 1. Indeed, our prime objective is the separation of overlapped gene expression dynamics and not just the piece-wise linear approximation of a more complex function; in our case, partitions spanning over disjoint regions of the feature space are not necessarily to be avoided. Enforcing closeness between points being partitioned is difficult in our domain, due to the lack of a sensible distance measure in the high-dimensional space of the epigenetic features, and potentially dangerous, since we have no reason to believe gene partitions induce non-intersecting convex hulls in the input space. Furthermore, to cope with the issue of suboptimality in the solution found via the EM procedure, we focused on a genetic clustering approach. This is mainly due to the intrinsic parallel nature of the search-space exploration and the capability of more finely controlling such exploration by tuning the genetic algorithm's hyper-parameters, so to ensure enough 'biological diversity'.

III. Materials and Methods

A. Biological data

To study the relation between epigenetic features - particularly HMs and TF bindings - and cell transcriptional activities, we focused on the interesting K562 immortalized human leukemic cell-line, for which ChIP-seq experiments have been performed for a large amount of TFs, and results are available in the ENCODE repository. To study the cells in their stationary normal status, we disregarded any experiment conducted in response to any chemical treatment, and considered only good quality experiments. The ENCODE project standards define 'audits' metadata to report quality issues. We discarded experiments with at least one of the following audits: "*AUDIT_ERROR: extremely low read depth, extremely low read length*", "*AUDIT_NOT_COMPLIANT: insufficient replicate concordance, missing input control, severe bottlenecking, unreplicated experiment*". Furthermore, we discarded all TF experiments having at least one of the following audits: "*AUDIT_WARNING: insufficient read depth*", "*AUDIT_NOT_COMPLIANT: insufficient read depth*". These quality filters gave ChIP-seq experiment data for 247 different epigenetic markers, i.e., 237 TFs and 10 HMs.

Genes are our units of measurements, represent single observations of epigenetic quantities under study, and correspond to points in the (high-dimensional) feature space. We used the *GENCODE v24* gene annotation, considering only (human) protein coding genes associated with an Entrez Gene ID, i.e., a set of $19,077$ genes. For each of them GENCODE provides the coordinates of all known alternative transcript isoforms.

As gene transcriptional activity data, we considered mature messenger-RNA (mRNA) gene quantifications from ENCODE K562 polyadenylated RNA-seq experiments as output target response values. We averaged the FPKM (fragments per kilobase of exons per million) quantifications of three RNA-seq experiments conducted in similar conditions which showed good internal replicate concordance and high correlations between each other (Pearson correlation coefficient on the natural logarithm of FPKM values greater than 0.92).

B. Data preparation

As far as epigenetic signals are concerned, for each TF and HM we considered processed data in the form of called peaks, whose goodness relies on the way the peak calling procedure has been performed. The ENCODE consortium has defined standards for this procedure, so it was possible to retrieve robust peaks for each feature. We indeed considered "*conservative IDR thresholded peaks*" for TFs and "*replicated peaks*" for HMs. Given the high number of features, using peak data greatly reduces computational and storing requirements without waiving information content; in fact, peak data are refined and partly de-noised as the peak calling procedure also takes into account an input control signal.

To characterize the epigenetic status of a gene for a specific feature (TF or HM), we considered the maximum peak enrichment signal attained by peaks within a symmetric window region of length 10 kbases centered on the gene transcription start site (TSS); this is in accordance with the observation reported in [7] that signals close to a TSS (roughly, in a gene promoter) are the most informative. Each gene usually has different RNA transcript isoforms, each with its own TSS. Among all TSSs of a gene, we chose the one associated with the most expressed gene transcript, according to the average isoform transcript quantifications, provided, together with gene quantifications, by ENCODE RNA-seq experiments.

With the described data, we constructed a dataset that we used to fit statistical models and evolve our genetic algorithm. We refer to such dataset as $\mathcal{D} = (X, T)$, i.e., the ensemble of an input epigenetic signal matrix X and a target gene expression vector T. X is a matrix of dimension $n \times m$, where n is the total number of genes under analysis and m is the number of epigenetic features, with $n = 19,077$ and $m = 247$. The element i,j of the matrix X represents the maximum peak enrichment value found for the epigenetic feature j in the promoter of gene i, with $i \in \{0, \ldots, n-1\}$ and $j \in \{0, \ldots, m-1\}$. The target vector T is a column vector of dimensions $n \times 1$, whose element i is the average FPKM mRNA quantification of gene i, with $i \in \{0, \ldots, n-1\}$.

C. Data preprocessing

Before performing genetic hyperplanes clustering and fitting of linear regression models, on the dataset $\mathcal{D}$ we executed some data-cleaning and transformation procedures.

First, $\mathcal{D}$ was also considered in a 'transformed' version $\tilde{\mathcal{D}} = (X, \tilde{T})$, which is the same dataset as $\mathcal{D}$ but with the target vector T transformed according to: $\tilde{T} = \sqrt{\ln(1+T)}$. The complete modeling pipeline was run on both the two dataset versions. Log-linear regression was already applied in [10]. The further squaring operation on the log-target values was applied because it was observed to help genome-wide least square fitting in terms of enhanced average R^2 scores in k-fold cross validation ($k = 10$).

Second, both $\mathcal{D}$ and $\tilde{\mathcal{D}}$ datasets were purged from outliers, considered as points with both high regression leverage and associated high residual. Considered the leverage threshold l and the residual thresholds $(r, \tilde{r})$, with $r = (r_{lower}, r_{upper})$ and $\tilde{r} = (\tilde{r}_{lower}, \tilde{r}_{upper})$, for $\mathcal{D}$ and $\tilde{\mathcal{D}}$, respectively, k-fold cross-validation ($k = 5$) was performed in fitting a genome-wide linear regression model, and repeated for $N = 10$ times. Each time a gene with leverage higher than l is found to be associated with a residue greater than r (or $\tilde{r}$) in absolute terms, then it is added to the set of outliers. In our study we set l to 0.3, $r = (-255.17, 271.65)$ and $\tilde{r} = (-1.15, 1.18)$. These values correspond to the 1^{st} and 99^{th} percentiles of the residues for the two linear models fitted on the entire datasets $\mathcal{D}$ and $\tilde{\mathcal{D}}$, respectively. This procedure led to the purged datasets $\mathcal{D}_p$ and $\tilde{\mathcal{D}}_p$, where the design matrices and target vectors have n_p and $\tilde{n}_p$ rows, with $n_p = 19,065$ and $\tilde{n}_p = 19,072$.

D. Genetic hyperplanes clustering

The primary objective of the genetic algorithm (GA in the following) we developed is to extract a partitioning of genes such that partition-wise linear fitting minimizes the overall regression error. In the context of genetic algorithms, each individual encodes one possible solution to the optimization problem. In other words, fixed the number of partitions being sought to C, one individual corresponds to a C-way partitioning of the gene set and, thus, we would like to find the individual Θ whose encoded partitioning maximizes the following ***fitness function***:

$$\Phi(\Theta) = -\sum_{c=0}^{C-1} \sum_{i\in\Theta(c)} |t_i - \tilde{\boldsymbol{w}}_c^T \tilde{\boldsymbol{x}}_i| \qquad (2)$$

where $\Theta(c)$ represents the set of genes in partition c according to the partitioning induced by individual Θ, t_i is the element i of the considered target vector, and $y_{i,d} = \tilde{\boldsymbol{w}}_c^T \tilde{\boldsymbol{x}}_i$ is the prediction computed for the input sample $\boldsymbol{x}_i$ by the linear model m_c learned on $\Theta(c)$.

The description of the GA characteristics follows; please refer to Algorithm 1 for the outline of the overall clustering procedure. Several terms describing GAs are borrowed from biology; to avoid annoying clashes with the same terms involved in our application domain, we *emphasize* all those computational terms relating GAs that could generate misunderstandings (e.g., gene in biology vs. *gene* in evolutionary computation).

Encoding: The encoding is the way we describe an individual of the GA, i.e., the way we represent a possible solution to our optimization problem. An individual is classically represented with a *chromosome*, i.e., a string of some objects, e.g., characters, numbers, etc. An element of the string is termed as *gene*, and the values a *gene* can assume are called *alleles*. The encoding that is adopted in this study is very natural and straightforward. Let Θ be an individual describing a partitioning of the dataset $\mathcal{D}_p$, then:

- its *chromosome* consists in a string of length n_p;
- the *gene* at *locus* i in the *chromosome* represents the partition to which gene i in $\mathcal{D}_p$ has been assigned;
- an *allele* a_i associated with *gene* i is such that $a_i \in \{0, \ldots, C-1\}$

Algorithm 1 Hyperplanes Genetic Clustering

```
procedure HypGenClust($\mathcal{D}, C, P, L, \omega, r, o, m_0, m_L, e$)
    $t \leftarrow 0$
    initialize $\mathcal{P}$ with $P$ random individuals reflecting a $C$-way partitioning
    $\Theta^* \leftarrow \arg\max_{\Theta\in\mathcal{P}} \Phi(\Theta, \mathcal{D})$
    while neither $\mathcal{S}_1(t, L)$ nor $\mathcal{S}_2(\omega, \Theta^*)$ is met do
        $\mathcal{P}' \leftarrow \{Elitism(\mathcal{P}, e)\}$
        while $|\mathcal{P}'| < P$ do
            $(i_1, i_2) \leftarrow DetTournamentSelection(\mathcal{P}, r)$
            $\mathcal{O} \leftarrow UniformCrossover((i_1, i_2), o)$
            $m(t) \leftarrow LinearScheduler(m_0, m_L, L, t)$
            $\mathcal{O} \leftarrow UniformSNVLikeMutation(\mathcal{O}, m)$
            $\mathcal{P}' \leftarrow \mathcal{P}' \cup \mathcal{O}$
        $t \leftarrow t + 1$
        $\mathcal{P} \leftarrow \mathcal{P}'$
        $\Theta^* \leftarrow \arg\max_{\Theta\in\mathcal{P}} \Phi(\Theta, \mathcal{D})$
    return $\Theta^*$
```

Selection: Both *Ranked Roulette Wheel Selection* and *Deterministic Tournament Selection* were tested. The latter one showed to be more effective in the sense that, by ensuring greater diversity in the solution population, led to faster convergence of the optimization procedure. *Deterministic Tournament Selection* takes as input $\mathcal{P}$ and r, where $\mathcal{P}$ is the current population and r is the percentage of individuals in the population taking part in a tournament. It randomly samples without replacement two tournaments, i.e., two subsets of $\mathcal{P}$, called $\mathcal{P}_1$ and $\mathcal{P}_2$, of cardinality $\lfloor |\mathcal{P}| \times r \rfloor$. It computes and stores the fitness values of all the individuals in the two tournaments, $\mathcal{F}_1, \mathcal{F}_2$, and takes the individuals with best fitness in $\mathcal{P}_1$ and $\mathcal{P}_2$, say $i_1^1\, i_2^1$. In the case the two selected individuals equal each other, it chooses the individual i_2^2 from $\mathcal{P}_2$, i.e., the one with second best fitness. Those two individuals will be mated. The value for r was chosen as 0.1.

Crossover: *Single Point Crossover* and *Uniform Crossover* strategies were attempted. The latter one revealed to be much more effective, speeding up the convergence of the algorithm. This was probably due, again, to a greater 'diversity'. Some measurements on the average Hamming distance between individuals were taken during the evolution of the GA, and those confirmed this last intuition. This latter strategy led to populations with a greater average Hamming distance between individuals than the former one. *Uniform Crossover* takes as inputs the two individuals being mated $i_1\, i_2$ and the number of descendants to be generated, o. Each new descendant has a *chromosome* built in this way: *allele* at *locus* l, say a_l, can uniformly assume one of the values of the parents, i.e., a_l^1 or a_l^2. In other words, a_l takes value a_l^1 with 0.5 probability and a_l^2 with 0.5 probability. *Uniform Crossover* returns the offspring $\mathcal{O}$ as the set of the o computed descendants. The value for o was set to 1, as it, in principle, gives more population 'diversity', and was shown not to slow down computation, in practice.

Mutation: The choice for the mutation operator arose quite naturally because of the way the encoding has been designed. *Uniform Single-Nucleotide-Variant-like Mutation* takes as input an individual i and a probability of mutation m. For each *gene* in its *chromosome*: with probability $(1 - m)$ the *gene* is kept as is, with probability m its *allele* is uniformly mutated to one of the possible *alleles*, i.e., to one value in the set $A = \{0, \ldots, C-1\}$. It returns the (possibly) mutated individual. The mutation probability m was generally kept low, given the length of the *chromosomes*. In particular, a linear scheduler was used to make the mutation probability decrease as the algorithm was reaching convergence. Given two mutation probability extremes, $m_0, m_L \in [0, 1]$, with $m_0 > m_L$, the mutation probability at time t is given by:

$$m(t) = m_0 + (\frac{m_L - m_0}{L}) \times t, \qquad t \in \{0, \ldots, L\} \qquad (3)$$

where L denotes the maximum time horizon allowed for the convergence of the algorithm. m_0 and m_L values were set to 0.003 and 0.002. Greater values were observed to make the evolution too much noisy.

Elitism: A minimum quantity of elitism was applied just as a safety net in the case too much introduced 'diversity' in the population would possibly bend the evolution away from the best currently found solution. A parameter e controls the proportion of best found solutions that are injected in the new population by overriding the selection and mating process. Even the e value was kept low, as the focus is just to ensure to keep few best overall solutions between subsequent generations. e was set to 0.01 with a population of cardinality $|\mathcal{P}| = 200$ (two elite individuals preserved at each generation).

Population size: The population size was set to $P = |\mathcal{P}| = 200$. This mild value was observed to be enough to avoid the algorithm to rapidly get caught in a poor local optimum, but at the same time was observed to be small enough to allow reasonable computational times.

Stopping criterion: The GA has been equipped with a maximum time horizon value of L generations and a further parameter, ω. The evolution process is iterated in time until one of the two following conditions hold:

$\mathcal{S}_1$: L generations have been performed;

$\mathcal{S}_2$: the fitness function of the best individual, $\Phi(\Theta^*)$ has not increased during the last ω iterations.

The $\mathcal{S}_2$ condition is actually evaluated only when at least ω generations have been performed. At time $\bar{t}, \bar{t} \geq \omega$, we can rewrite the condition as

$$\Phi(\Theta^*; t) \leq \Phi(\Theta^*; t-1)\, \forall t \in \{\bar{t} - \omega + 1, \dots, \bar{t}\}. \quad (4)$$

IV. Results

In this section we report and evaluate the obtained results; these depend on the number of clusters we would like the algorithm to extract, i.e., what we termed C. This parameter is not guessable *a priori*; in our work we ran the algorithm for three different values of C, namely $2, 3, 5$. We ran the GA and analyzed its results for both $\mathcal{D}_p$ and its transformed version $\tilde{\mathcal{D}_p}$. Hyperplanes clustering of $\mathcal{D}_p$ should in principle be an easier task under the hypothesis that mingled, yet different expression dynamics exist. Target transformations might indeed compact these latter ones, and make partitioning harder. Nevertheless, the illustrated target transformation seems to be necessary to reasonably fit linear models to the considered kind of data and obtain acceptable coefficients of determination (R^2).

The whole parameter setting of the GA is the following:

Selection	$DeterministicTournament$
Crossover	$Uniform$
Mutation	$SNV - Like$
Number of partitions:	$C = 2, 3, 5$
Population size:	$P = 200$
Optimization horizon:	$L = 1,600$
Stopping criterion window:	$\omega = L/4$
Tournament size ratio:	$r = 0.1$
Offspring size (for one mating):	$o = 1$
Mutation extremes:	$m_0 = 0.003, m_L = 0.002$
Elitism ratio:	$e = 0.01$

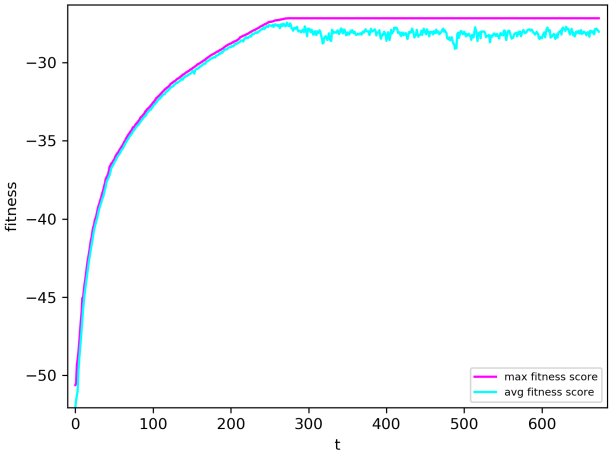


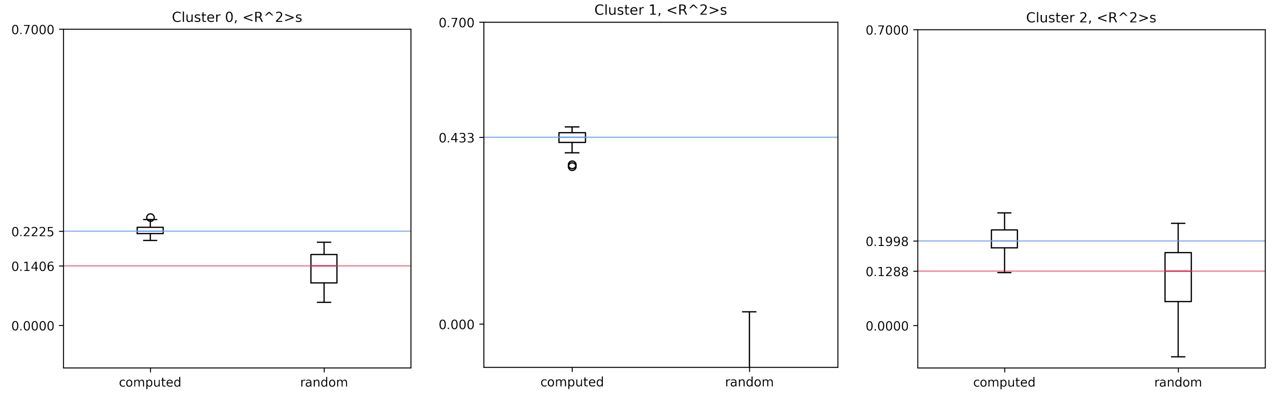
Fig. 1: Trend of the fitness function (refer to Equation 2) for best and average individual in the setting $\mathcal{D}_p, C = 3$ during the evolution process, stopped due to meeting of condition $\mathcal{S}_2$.

For all the experiments, we observed that the optimization procedure ended before reaching the maximum time horizon L. Figure 1 represents the GA evolution process for $C = 3$ on $\mathcal{D}_p$ and its convergence, depicting the trend for the best and average fitness value in the population overtime.

A. Enhanced easiness of linear fitting

Our first approach to assess the validity of the found clusters consisted in studying the average performance of linear models on them. Consider a solution Θ^* inducing a partitioning of the dataset into C clusters. From now on they are termed as 'computed' clusters. For computed cluster $\Theta^*(c)$, with $c \in \{0, \dots, C-1\}$, we *randomly* sampled a subset of $\mathcal{D}_p$ (or $\tilde{\mathcal{D}_p}$) of same cardinality, call it $\mathcal{D}_p^S$, as the 'random' counterpart and then we $5-$folds cross-validated linear models on both $\Theta^*(c)$ and $\mathcal{D}_p^S$, yielding, respectively, two average coefficients of determination $\bar{R^2}^d$ and $\bar{R^2}^S$. For each partition c, this procedure was repeated 20 times, each time storing, separately, the two average performance scores. Such procedure has the precise objective to assess whether the GA has managed to find a partitioning into clusters for which the linear fitting problem is actually easier and more stable (less variance in regression scores) w.r.t. the whole dataset. Such characteristics would suggest a reasonably good data unmingling.

For the $\mathcal{D}_p$ dataset and for all the choices of C, linear regression on clusters has always shown to obtain better scores and to be more robust. For each cluster, the mean of the average coefficients of determination is always higher than the one computed on the randomly sampled subset $\mathcal{D}_p^S$, and their range of variation is always narrower. This behavior is depicted in Figure 2 and quantified in Table I for $C = 3$ in $\mathcal{D}_p$. In this setting, the cluster cardinalities are the following: $|$*cluster 0*$| = 8,393$, $|$*cluster 1*$| = 2,985$, $|$*cluster 2*$| = 7,687$. As far as $\tilde{\mathcal{D}_p}$ dataset is concerned, we observed how one extracted cluster for each choice of C always has an higher



(a) 0: computed vs. random

(b) 1: computed vs. random

(c) 2: computed vs. random

Fig. 2: Boxplotted average R^2s for linear models on computed clusters contrasted with those attained on randomly sampled subsets of same cardinalities (setting $\mathcal{D}_p$, $C = 3$). Medians are elongated in order to more easily read their associated value. Negative scores attained on the random *cluster 1* are probably due to the fact that such model is fitted on a smaller number of (randomly sampled) points, as $|cluster\ 1| = 2,985$.

TABLE I: Mean and standard deviations for average R^2 scores comparisons (as $mean \pm std$). Refer to subsection IV-A.

$\mathcal{D}_p, C = 3$	computed	randomly sampled
$cluster0$	0.22 ± 0.01	0.13 ± 0.04
$cluster1$	0.43 ± 0.02	-0.42 ± 0.33
$cluster2$	0.20 ± 0.04	0.12 ± 0.08

mean for the average coefficients of determination, while the other clusters have means comparable with those of the randomly sampled $\tilde{\mathcal{D}}_p^S$. In any case, the variations in regression scores are always much smaller, confirming the robustness of the partitioning found. Table I reports the mean and standard deviations of the average coefficients of determination for each cluster found in the setting $\mathcal{D}_p, C = 3$.

B. Feature importance

Our second assessment concerns feature importance. We recall the objective of our study is not only partitioning the gene set, but, hopefully, to have such partitions characterized by distinct transcriptional regulation mechanisms, i.e., we would like to have clusters for which different characteristic epigenetic features are mostly predictive w.r.t. other clusters. We hence proceeded to the computation of feature rankings for each of the clusters extracted.

Let Θ^* be a solution inducing a partitioning of the dataset into C clusters. For each cluster $\Theta^*(c), c \in \{0, \ldots, C-1\}$, we performed *step-wise forward feature selection*, and constructed a ranking for the top-10 predictive features according to this criterion. The feature rankings computed on the $\mathcal{D}_p$ dataset generally showed to be different from each other. This was true especially for a lower value of C, i.e., $2, 3$. With $C = 5$ we observed, instead, how rankings encounter larger overlaps, though being generally different. We take as an example $C = 3$, whose clusters top-10 rankings are reported in Table II.

TABLE II: Feature rankings for the setting $\mathcal{D}_p, C = 3$, top-10

$\mathcal{D}_p, C = 3$	$cluster0$	$cluster1$	$cluster2$
1^{st}	H3K79me2	SNRNP70	H3K79me2
2^{nd}	H3K36me3	TAF7	ZZZ3
3^{rd}	GTF2A2	POLR2AphosphoS2	H3K36me3
4^{th}	KAT8	HNRNPUL1	SNRNP70
5^{th}	POLR2AphosphoS2	WDR5	HINFP
6^{th}	AGO1	GTF2F1	BCLAF1
7^{th}	ZNF639	H3K79me2	GTF2F1
8^{th}	CEBPB	POLR2A	ZBTB11
9^{th}	TAF7	MBD2	ZNF24
10^{th}	BRD4	KAT8	TEAD2

For *cluster 1*, transcription factor SNRNP70 was observed to be the most predictive, while the top predictive features in the other clusters are histone modifications. Also, consider how the predictive features WDR5 and HNRNPUL1 for *cluster 1* are absent in the top-10 positions of the other two clusters. *Cluster 0* and *cluster 2* differ, although the two histone modifications H3K79me2 and H3K36me3 are among the top predictive features in both of them. Indeed, most of top-ranked TFs of one cluster are not found in the top-10 ranking of the other one. This suggests that, even though the two mentioned HMs account for top predictive information in these two clusters, they enhance the model accuracy when combined with different sets of transcription factors.

From a biological perspective, *cluster 1* is well characterized by the presence of the leukemia-involved MYC-recruiter transcription factor WDR5 [15] and from the importance assumed by mRNA-binding proteins, i.e., SNRPN70 and HNRNPUL1. The involvement of the former one in the splicing process has been shown in [16]. In *cluster 0*, the top predicting features are directly linked with gene active expression: H3K79me2 and H3K36me3 are activators [17], and GTF2A2 is part of the polymerase complex playing an important role in transcription activation [18]. Finally, *cluster 2* reports, besides the already mentioned histone marks, HINFP, which has been shown to play a role in DNA methylation and transcription repression [19].

As for $\tilde{\mathcal{D}}_p$ dataset, variations in rankings are still observed, but they are much less evident: compacting of the target values does not allow a clear and interpretable hyperplanes clustering.

C. Model diversity

A further analysis is based on the evaluation of model dissimilarity. In the following, model dissimilarity is assessed by comparing slopes and specific weights assigned to features by each of the models. The maximization of the fitness function does not directly include the maximization of the model dissimilarity; hence, if models fitted on different clusters are found to be dissimilar, this is a reasonable confirmation of the goodness of the partitioning.

Evaluation of the model dissimilarity consists in the diversity of the models parameters. For this purpose, we computed the Pearson correlation coefficient between models' weight vectors. Let Θ^* be a solution inducing a partitioning of the dataset into C clusters. For each cluster $\Theta^*(c), c \in \{0, \dots, C-1\}$, we fitted a linear model on it and considered its weight vectors as $\boldsymbol{w}_c$. Call $\mathcal{W}$ the set of the weight vectors: $\mathcal{W} = \{\boldsymbol{w}_c, c \in \{0, \dots, C-1\}\}$. For each pair of weight vectors $(\boldsymbol{w}_{c_1}, \boldsymbol{w}_{c_2}), \boldsymbol{w}_{c_1}, \boldsymbol{w}_{c_2} \in \mathcal{W}$ and $c_1 \neq c_2$, we computed the Pearson correlation coefficient ϱ_{c_1,c_2}. The Pearson correlation coefficient is always bounded between values -1 and $+1$; the closer the value is to $+1$, the more the models describe similar input-response relations. For a value approaching 0, the hyperplanes described by the models tend to be orthogonal. Lastly, the closer the value is to -1, the more the models are anti-correlated, meaning they describe a completely opposite kind of input-response relation.

For dataset $\mathcal{D}_p$ and $C = 2$, the models fitted on the two clusters have a Pearson correlation coefficient $\varrho_{0,1} = 0.34$. For $C = 3$, the two hyperplanes described by *cluster 0* and *cluster 1* are nearly orthogonal as $\varrho_{0,1} = 0.09$. As for the correlation between the other clusters, we have: $\varrho_{1,2} = 0.39$, $\varrho_{0,2} = 0.26$. With $C = 5$, a total number of 10 correlation coefficients are computed. Most of them are near to zero, suggesting that many hyperplanes are nearly orthogonal; consider for instance $\varrho_{0,4} = -0.03$. Consistent results are found for dataset $\tilde{\mathcal{D}}_p$.

A second approach to the evaluation of model dissimilarity consists in the direct comparison of models' weight vectors, at least for the top-predictive features. In order to fairly compare those weights, we performed feature normalization by aligning the min and the max values for each of their value distributions. We then fitted linear models on the normalized datasets and compared the values assumed in different models by the same feature-weights. Great variations were observed among clusters, again conveying the dissimilarity between the models. Table III compares the weights assigned to the top-rank features in *cluster 0* and *cluster 1*, the ones describing the most hyperplanes with the most dissimilar slopes for $C = 3$.

TABLE III: Feature weights for the three most predictive features in *cluster 0* and *cluster 1* in the setting $\mathcal{D}_p, C = 3$.

$\mathcal{D}_p, C = 3$	$cluster0$	$cluster1$
H3K79me2	9.52	51.32
H3K36me3	6.02	13.69
GTF2A2	1.27	40.99
SNRNP70	1.45	276.25
TAF7	-0.29	23.66
POLR2AphosphoS2	4.28	78.61

D. An oracle-based approach

In this last subsection of the result analysis we show how the partition produced by the GA could be used to construct a predictive model for the mRNA abundance quantifications, with enhanced performances w.r.t. a *single* linear model. The setting is the following. Consider dataset $\tilde{\mathcal{D}}_p$ and a train-test split: $\tilde{\mathcal{D}}_p = \tilde{\mathcal{D}}_p^{\,train} \cup \tilde{\mathcal{D}}_p^{\,test}, \tilde{\mathcal{D}}_p^{\,train} \cap \tilde{\mathcal{D}}_p^{\,test} = \{\}$, where the test set was constructed by randomly sampling 10% of genes from the original dataset. The genetic algorithm procedure is launched on $\tilde{\mathcal{D}}_p^{\,train}$ and a solution Θ^* is returned, defining a partition on that train set, i.e., a set $\mathcal{C} = \{\Theta^*(c), c \in \{0, \dots, C-1\}\}$. On each partition in $\mathcal{C}$, a linear model is fitted, generating the ensemble of models $\mathcal{M} = \{M_c, c \in \{0, \dots, C-1\}\}$. Then, the task is to optimally predict genes in $\tilde{\mathcal{D}}_p^{\,test}$ by making use of $\mathcal{M}$, hopefully better than how a genome-wide model would do, that is, a single linear model fitted on the whole $\tilde{\mathcal{D}}_p^{\,train}$. Under the assumption that the partitioning induced by Θ^* is actually meaningful also for genes in the test set, then, given a query point $q \in \tilde{\mathcal{D}}_p^{\,test}$, the steps for predicting its associated response value are the following:

1. assign q to its most representative cluster, named $\bar{c}$
2. predict the response value for q as the output of $M_{\bar{c}}$, i.e., $y_q^{\bar{c}} = \tilde{\boldsymbol{w}}_{\bar{c}}^T \tilde{\boldsymbol{x}}_q$

The most representative cluster is, in principle, the one whose prediction from the associated model is the closest to the real response value t_q associated with q. Unfortunately, step 1. is not a trivial task if the value t_q is not known *a priori*. This is true even for points in the train set. Anyhow, supposing to have an oracle ϖ capable of assigning the best possible cluster to all the query points in $\tilde{\mathcal{D}}_p^{\,test}$, we can accordingly define an upper-bound on the regression performance of any ensemble model based on $\mathcal{M}$. The oracle's decision function takes the following form:

$$f_{\varpi}(\boldsymbol{x_q}; \mathcal{M}) = \tilde{\boldsymbol{w}}_{\bar{c}}^T \tilde{\boldsymbol{x}}_q, \qquad \bar{c} = \underset{c \in \{0,\dots,C-1\}}{\arg\min} |t_q - y_q^c| \tag{5}$$

where we remark how the response t_q is actually hidden, and known only to ϖ.

We evaluated the performance of ϖ in the setting $\tilde{\mathcal{D}}_p, C = 2, 3, 5$. The genome-wide model, termed as γ, scored $R_\gamma^2 = 0.74$ on $\tilde{\mathcal{D}}_p^{\,test}$, where the oracle always scored better coefficient of determinations, with better performance for larger values of C. Scores are $R_\varpi^2 = 0.79$ for $C = 2$, $R_\varpi^2 = 0.83$ for $C = 3$ and $R_\varpi^2 = 0.86$ for $C = 5$.

Note how the scores obtained by the oracle can be used to choose the best value for C *a posteriori*. For instance, one

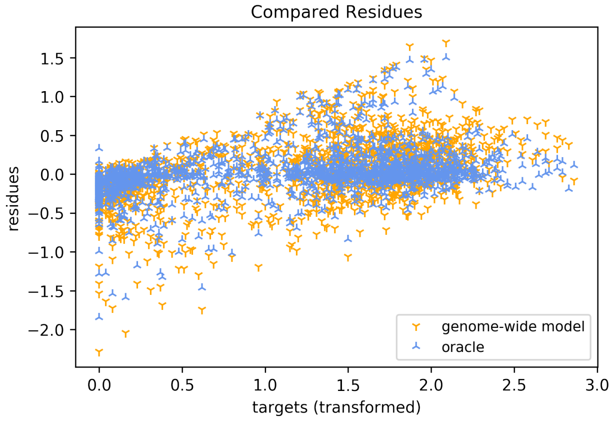


Fig. 3: Residue plots for ϖ and γ on $\tilde{\mathcal{D}_p}^{test}, C = 5$.

can choose the value of C corresponding to an elbow for R^2_{ϖ} scores, averaged over several random test-train splits.

In order to further characterize the behavior of the oracle predictor with respect to the genome-wide model, we compared the residues of both of them for $C = 5$ - the tested choice associated with best score. They are considered as the differences between target and predicted values over all the genes in $\tilde{\mathcal{D}_p}^{test}$, and are reported in Figure 3. Predictions from ϖ were found to be more accurate than those from γ on 92.0% of test genes. It can also be observed how they are relatively less dispersed, as confirmed by a smaller standard deviation: $\sigma_{\mathcal{R}^{\varpi}} = 0.30$ vs. $\sigma_{\mathcal{R}^{\gamma}} = 0.41$.

V. Conclusions

In this work we proposed the application of a genetic algorithm to cluster protein coding genes according to the relation between their epigenetic status and expression. The hyperplanes corresponding to the found partitions revealed to be dissimilar in terms of slope and specific importance of epigenetic marks. The found gene partitioning showed also to be potentially effective in enhancing expression prediction capabilities for unseen genes. The upper bound on regression performance for the ensemble of linear models fitted on each computed cluster has been indeed estimated to be significantly higher that those from a single linear regression model.

Future work will address a more profound biological validation of the results and characterization of the found gene subsets. The procedure will be also applied to measurements conducted on different cell-lines, probing the ability of the algorithm to detect tissue-related dissimilarities in epigenetically-driven regulative dynamics. Further analyses will focus on validating the robustness of the proposed genetic clustering approach w.r.t. classical EM-based procedures applied, for instance, in [13] and [14]. Interesting will also be to extend the proposed algorithm to fitting elastic nets as in [14], rather than simple ordinary least squares models. This is motivated by the so-called ‘grouping effect’ that characterizes the formers, that is, highly correlated variables tend to be either included or excluded from the model in groups [20]. This feature is of interesting application in our domain, in which, from an exploratory perspective, hundreds of - possibly correlated - epigenetic markers are included as monitored variables.

Acknowledgment

This work was supported by the Advanced ERC Grant “Data-Driven Genomic Computing (GeCo)” project (2016-2021), funded by the European Research Council.

References

[1] G. A. Maston, S. K. Evans, and M. R. Green, “Transcriptional regulatory elements in the human genome,” *Annu Rev Genomics Hum Genet*, vol. 7, pp. 29–59, 2006.

[2] P. J. Farnham, “Insights from genomic profiling of transcription factors,” *Nat Rev Genet*, vol. 10, no. 9, pp. 605–616, 2009.

[3] J. M. Vaquerizas, S. K. Kummerfeld, S. A. Teichmann, and N. M. Luscombe, “A census of human transcription factors: function, expression and evolution,” *Nat Rev Genet*, vol. 10, no. 4, pp. 252–263, 2009.

[4] S. Berger, “The complex language of chromatin regulation during transcription,” *Nature*, vol. 447, no. 7143, pp. 407–412, 2007.

[5] S. K. Kurdistani, S. Tavazoie, and M. Grunstein, “Mapping global histone acetylation patterns to gene expression,” *Cell*, vol. 117, no. 6, pp. 721–733, 2004.

[6] ENCODE Project Consortium, “An integrated encyclopedia of DNA elements in the human genome,” *Nature*, vol. 489, pp. 57–74, 2012.

[7] C. Cheng, K.-K. Yan, K. Y. Yip, J. Rozowsky, R. Alexander, C. Shou, and M. Gerstein, “A statistical framework for modeling gene expression using chromatin features and application to modENCODE datasets,” *Genome Biol*, vol. 12, no. 2, p. 15, 2011.

[8] Z. Ouyang, Q. Zhou, and W. H. Wong, “ChIP-seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells,” *Proc Natl Acad Sci*, vol. 106, no. 51, pp. 21521–21526, 2009.

[9] R. Karlić, H.-R. Chung, J. Lasserre, K. Vlahoiviček, and M. Vingron, “Histone modification levels are predictive for gene expression,” *Proc Natl Acad Sci*, vol. 107, no. 7, pp. 2926–2931, 2010.

[10] B. et al., “Predicting expression: the complementary power of histone modification and transcription factor binding data,” *Epigenetics Chromatin*, vol. 7, p. 36, 2014.

[11] L. Breiman, “Hinging hyperplane for regression, classification and function approximation,” *IEEE Trans Inf Theory*, vol. 39, pp. 999–1013, 1993.

[12] E. Amaldi and M. Mattavelli, “The MIN PFS problem and piecewise linear model estimation,” *Discrete Appl Math*, vol. 118, pp. 115–143, 2002.

[13] N. Manwani, “K-plane regression,” *Inf Sci (Ny)*, vol. 292, pp. 39–56, 2015.

[14] T. G. do Rego, H. G. Roider, F. A. T. de Carvalho, and I. G. Costa, “Inferring epigenetic and transcriptional regulation during blood cell development with a mixture of sparse linear models,” *Bioinformatics*, vol. 28, no. 18, pp. 2297–2303, 2012.

[15] L. R. Thomas and W. P. Tansey, “Interaction with WDR5 promotes target gene recognition and tumorigenesis by MYC,” *Mol Cell*, vol. 58, no. 3, pp. 1–13, 2015.

[16] D. J. Adams, L. van der Weyden, A. Mayeda, S. Stamm, B. J. Morris, and J. E. Rasko, “ZNF265 - a novel spliceosomal protein able to induce alternative splicing,” *Cell Biol*, vol. 154, no. 1, pp. 25–32, 2001.

[17] A. J. Bannister and T. Kouzarides, “Regulation of chromatin by histone modifications,” *Cell Res*, vol. 21, no. 3, p. 381395, 2011.

[18] D. J. Mitsiou and H. G. Stunnenberg, “TAC, a TBP-sans-TAFs complex containing the unprocessed TFIIAalphabeta precursor and the TFIIAgamma subunit.,” *Mol Cell*, vol. 6, pp. 527–537, 2000.

[19] M. Sekimata, A. Takahashi, A. Murakami-Sekimata, and Y. Homma, “Involvement of a novel zinc finger protein, MIZF, in transcriptional repression by interacting with a methyl-CpG-binding protein, MBD2,” *J Biol Chem*, vol. 276, no. 46, pp. 42632–42638, 2001.

[20] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *J R Stat Soc, Ser B*, vol. 67, pp. 301–320, 2005.