Discovering Typical Transcription-Factors Patterns in Gene Expression Levels of Mouse Embryonic Stem Cells by Instance-Based Classifiers

Francesco Gagliardi and Claudia Angelini

Istituto per le Applicazioni del Calcolo 'Mauro Picone' — CNR,
via Pietro Castellino, 111 — 80131 Napoli, Italy
fnc.ggl@gmail.com, claudia.angelini@cnr.it

Abstract. The development of high-throughput technology in genome sequencing provide a large amount of raw data to study the regulatory functions of transcription factors (*TFs*) on gene expression. It is possible to realize a classifier system in which the gene expression level, under a certain condition, is regarded as the response variable and features related to *TFs* are taken as predictive variables. In this paper we consider the families of *Instance-Based (IB)* classifiers, and in particular the *Prototype exemplar learning classifier* (*PEL-C*), because IB-classifiers can infer a mixture of representative instances, which can be used to discover the typical epigenetic patterns of transcription factors which explain the gene expression levels. We consider, as case study, the gene regulatory system in mouse embryonic stem cells (*ESCs*). Experimental results show IB-classifier systems can be effectively used for quantitative modelling of gene expression levels because more than *50%* of variation in gene expression can be explained using binding signals of *12* TFs; moreover the *PEL-C* identifies nine typical patterns of transcription factors activation that provide new insights to understand the gene expression machinery of mouse *ESCs*.

Keywords: Knowledge Discovery, Instance-Based Learning, High-throughput Sequencing, ChIP-Seq, RNA-Seq.

1 Introduction

High-throughput genome-sequencing technologies are now routinely being applied to a wide range of important topics in biology and medicine, often allowing researchers to address important biological questions that were not possible before [1] [2]. The recent development of RNA sequencing (RNA-Seq) technology holds the promise to provide more accurate gene expression measurements than traditional microarray. Meanwhile, chromatin immunoprecipitation (ChIP) coupled with sequencing technologies (ChIP-Seq) have been developed to identify whole-genome localization of protein–DNA binding sites [3][4]. Applications of techniques belongs to fields of Machine Learning (ML) [5][6] and Knowledge Discovery in Databases (KDD) [7] can be useful to make an integrative analysis of these data and providing us an insightful view of genome functions. Predictive modeling (such as the training of a

A. Petrosino, L. Maddalena, P. Pala (Eds.): ICIAP 2013 Workshops, LNCS 8158, pp. 381–388, 2013.

classifier system) is a machine learning strategy to predict an outcome from one or more variables (predictors). In the study of gene regulation, a classifier system can be constructed in which the gene expression levels under a certain condition is regarded as the response variable and various features related to transcription factors (TFs) are taken as the variables. A classifier system can have two mayor purposes: to predict class of new observations and to extract knowledge [7] from past experiences. In this work we choice to use instance-based (IB) classifiers in order to extract the typical patterns and internal structure in data obtained from sequencing of RNA (RNA-Seq) and of chromatin immunoprecipitation (ChIP-Seq) regarding the gene-expression regulatory-system in mouse embryonic stem cells (ESCs).

2 Instance-Based Classifier Systems

Instance-based (*IB*) classifier systems [5] [6] constitute a family of classifiers which main distinctive characteristic is to use the instances themselves as classes representation. *IB* classification relies on the similarity between the new observation to be classified and instances chosen as representative of the learnt class. Within this family we can identify two sub-families [6]: the first is based on prototype methods and the second on nearest-neighbours. The prototype methods build representative instances as centroids of classes or sub-classes, often using iterative clustering algorithms; conversely the nearest-neighbours methods use exemplars filtered from dataset as representative instances. The *IB* based classifiers and in particular the *k-Nearest Neighbour Classifier* (*k-NNC*) achieve such performances to be used in real-life problems, but they are not used in situations where an explanation of the output of the classifier is useful, because it is commonly assumed that *"instances do not really «describe» the patterns in data"* [5, p.79].

Some recent developments show that some hybrid *IB* classifiers, such as the *Prototype exemplar learning classifier* (*PEL-C*) [8] and the *Total recognition by adaptive classification experiments* (*T.R.A.C.E.*) [9], which generalize both prototype-based and exemplar-based classifiers, can be used to discover the "typicality structure" of learnt category, detecting how the class is decomposable in subclasses and their typicality grade within the class. The representative instances obtained from these classifiers are composed of a mixture of instances varying from prototypical ones to atypical ones, which form the so-called *"gradient of typicality"* [10].

We focus on the learning algorithm introduced in [9] (see Algorithm 1 in the following) because it has particular formal characteristics which are explained in detail in [9; sect. 3]. In particular it is possible to demonstrate (theorem 3.2 in [9]) that the representative-instance set inferred by this learning algorithm can vary from that of the Nearest Prototype Classifier (*NPC*), which is completely based on prototypes, to one of the Nearest Neighbour Classifier (*NNC*), which is completely based on exemplars according to the number of learning iterations and to the particular dataset.

We present in the following the learning algorithm[1]; we indicate *TS* as the training set, *RI* as the representative instances set and C_k as the items of the *k-th* class:

[1] A similar version of this algorithm can be found in [9, p.481].

Algorithm 1. Learning algorithm of hybrid instance-based classifiers

```
1. Initialize RI with the barycentres of the classes C_k
2. WHILE NOT (Termination Condition)
%%(Find a new candidate representative instance)
   2.1 Calculate the distances between every instance of TS and every
       instance of RI
   2.2 Among the misclassified instances of TS, find the instance which
       is the farthest from the nearest instance of RI belonging to its
       own class. Call it X and assume that it belongs to the class C_k
   2.3 Add X to RI.
%%(Update RI)
   2.4 Consider only instances of RI and TS belonging to C_k Call them as
       RI_k and TS_k, respectively
   2.5 Update the positions of RI using the k-means clustering algorithm
       applied only to TS_k with starting conditions RI_k:
      2.5.1 Apply the NN-rule to the items of TS_k respect to the RI_k
      2.5.2 Iteratively re-calculate the locations of instances of RI_k
            by updating the barycentres calculated respect to the
            subclasses determined with the NN-rule.
3. END
```

The behaviour of these classifier systems can vary from the one of the *NPC* to the one of *NNC* in an adaptive way and according to the chosen termination condition and to the particular classification problem. In intermediate cases the number and the kind of the representative instances is dynamically determined as a combination of prototypes, exemplars and representative instances of an intermediate abstraction level. We call these classifiers "hybrid" to refer to the type of representative instances set which can be inferred and it does not refer to a kind of classifier obtainable with a simple joining of classifiers *NPC* and *NNC*. In general, we can think about different possible termination conditions for the Algorithm 1, such as the following:

- *Training accuracy*. The accuracy percentage in classification of the training set is fixed and it can be equal to or less than *100%*. In the case it is set to *100%*, the system is forced to classify correctly all the training set, and the obtained classifier is the one proposed by Nieddu and Patrizi [9] and it is known as *T.R.A.C.E.*
- *Predictive accuracy*. The system can estimate its own performance on new instances by using a technique of cross validation [5, p.149] as varying the number of iterations. Therefore, the system is able to find the minimum number of iterations to obtain the maximum capability of generalizing (predictive accuracy on new instances). This termination condition is the one used by the *Prototype exemplar learning classifier* (*PEL-C*) [8], which usually [8][10] infers a number of representative instances definitely lower than both the *T.R.A.C.E.* and the *k-NNC.*

3 The Case-Study in Mouse Embryonic Stem Cells

We consider the gene regulatory system in mouse embryonic stem cells (ESCs) as investigated firstly in [3]. The study of this type of cellular line is fundamental in

molecular biology because ESCs can maintain self-renewal and pluripotency, i.e., having the ability to differentiate to any adult cell type. The considered genes are extracted from the *NCBI Reference Sequence database* (RefSeq) which is an open access, annotated and curated collection of publicly available nucleotide sequences (http://www.ncbi.nlm.nih.gov/refseq/). In order to predict gene expression, we use the ChIP-Seq data of *12* Transcription Factors (TFs): E2f1, Mycn, Zfx, Myc, Klf4, Tcfcp2l1, Esrrb, Nanog, Oct4, Sox2, Stat3, Smad1. The gene expression values of the sequencing data were calculated by the RPKM definition (the number of reads per kilobase of exon region per million mapped reads) based on a RNA-sequencing data. The reference genome used for *Mus musculus* is the *UCSC mm8* (http://genome.ucsc.edu/).

3.1 The Dataset

Following Ouyanga *et al.* [3] we consider 18936 RefSeq genes for the mouse and for each of them we define a feature vectors composed of *12* attributes (one for each TF) called *TF Association Strength* (TFAS); TFAS are computed using a weighted summation of TF binding peaks where those with higher reads intensity or location proximity to the transcription start site (TSS) were given higher weights.

Formally, the association strength of TF_j on gene *i* is a weighted sum of intensities of all of the peaks of TF_j:

$$a_{ij} = \sum_k g_k e^{-d_k/d_0}$$

where: g_k is the intensity (number of reads) of the *k-th* binding peak of the TF_j, d_k is the distance (number of nucleotides) between the TSS of gene *i* and the *k-th* binding peak, and d_0 is a constant, posed equal to 500 bps for E2f1 and 5000 bps for other TFs because E2f1 tends to be closer to TSSs [3]. Gene-expression classes are defined in the following *2* step procedure. First, we apply a logarithmic rescaling to the raw expression values (RPKM) based on mapped mRNA sequencing data for mouse ESCs; as usual, to avoid taking the logarithm of zero, a small positive constant is added; formally:

$$\mathrm{Log}_{10}(\mathrm{RPKM} + 0.01)$$

Then we apply a 5-classes equal-width binning, to transform the continuous variable of expression level into a categorical one with 5-values. Each class represents an intervals of equal size. The five classes are labelled as following: C_1 – '*Very Low*', C_2 – '*Low*', C_3 – '*Middle*', C_4 – '*High*', C_5 – '*Very High*'.

Summarizing, the used dataset[2] has *18936* rows (one for each gene), *12* features (one for each transcription factor) considered as predictive variables and 5 classes (one for each gene-expression level) regarded as the response variable to be predicted.

[2] The original version of the dataset is available as supplementary material of [3]; the dataset used in this work is available on request.

4 Experimental Results and Discussions

4.1 Experimental Procedure

We use an experimental procedure, to evaluate the classifier systems, composed of 3 steps: training, validation, and testing. In the training phase we train the classifier systems on data and in the validation phase we estimate how good the classifiers has been trained to select the best performing parameters, such as the value of *k* for the *k-NNC* or the number of learning iterations for the *PEL-C*. Because accuracy computed on the data used for training or also for validation are optimistically biased to predict real classification capabilities of the systems[3], in the testing phase we compute the accuracy of the previously tuned classifiers on a different data.

According to this procedure the dataset is divided in a sample-dataset, used for training and validation, and a test-dataset, used for testing. The sample-dataset is iteratively split in the training set and validation according to a cross-validation procedure. The sample is composed of 1000 genes randomly selected from the entire dataset with no stratified sampling; so we have a sample set which have $N >> p$ (N is 2 order of magnitude higher than $p = 12$).

We carried out different cross-validation runs applied to sample dataset for every classifier system considered: *NPC*, *NNC*, *k-NNC*, *T.R.A.C.E.* and *PEL-C*. Each run on sample set was prepared by using the leave-one-out procedure as a cross-validation technique.

4.2 Experimental Results

We show here the comparison of the experimental results obtained applying the different classifier systems to the above problem of classification.

The *k-NNC* is applied to sample-dataset by varying *k* between *1* and *25*. The best accuracy is obtained for *k=16*. In order to analyze the behaviour of the *PEL-C* the iterative learning algorithm (see Algorithm 1) is applied to sample-dataset by varying the number of iterations between *1* and *16*. The *PEL-C* obtains the maximum accuracy on the test set after *5* iterations and finds *9* representative instances, while *457* iterations are needed so that the learning algorithm converge to the stop condition of *T.R.A.C.E*, and it finds *461* representative instances. To compare classification performance and the kind of class representations obtained by different *IB* classifiers we have considered the following indexes: the accuracy on sample-set (computed by leave-one-out), the number of representative instances and the accuracy on the test-set (see Table 1).

We observe that *PEL-C* is outperformed in classification performance only by the *k-NNC*, that has the *k* optimized in order to maximize performances, but *PEL-C* obtains a classes representation composed with only *9* instances against the *1000* of the *k-NNC*. The *NPC* uses only *5* instances, which are all pure prototypes, but its accuracy is lower than *PEL-C* and *k-NNC*. The *k-NNC* and *NNC* use a classes representation entirely composed of exemplars, which are the *1000* instances of the whole sample-dataset.

[3] In fact accuracy on sample-set is also called *resubstitution rate* or also *apparent accuracy.*

Table 1. Performance indexes obtained by different classifier systems

	NPC	**PEL-C**	**T.R.A.C.E.**	**NNC**	**k-NNC** (k_{best}=16)
Representation:	*Prototype-based*	*Hybrid*		*Exemplar-based*	
Accuracy on test set (%)	46.36	54.40	49.29	53.23	63.84
Accuracy on sample set (%)	48.45	52.07	47.31	49.59	59.02
Representative instances	5	9	461	1000	1000

To analyse the behaviour of a classifier system to discriminate among different classes in a multiclass problem is often useful to compute the confusion matrix between actual classes and predicted ones; because the actual classes in our problem is obtained by a binning procedure from a continuous values we show (see Figure 1) the box-plot of log values of actual gene expression levels for each predicted classes by the *PEL-C*.

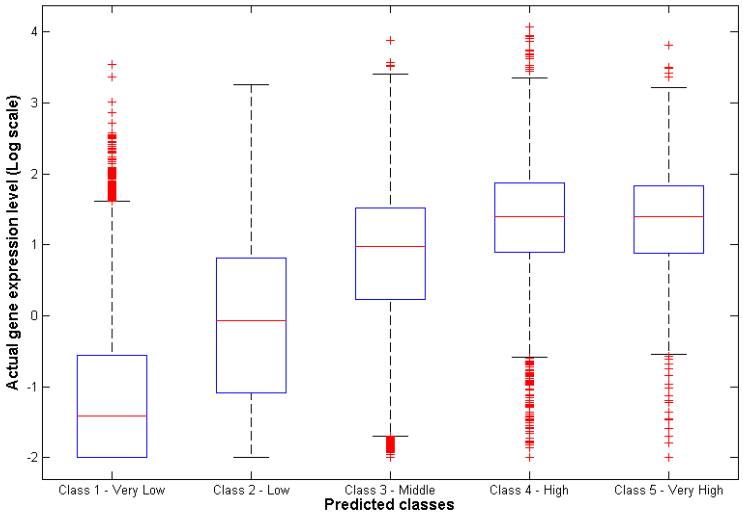


Fig. 1. Actual gene expression values *versus* the gene expression class predicted by *PEL-C*

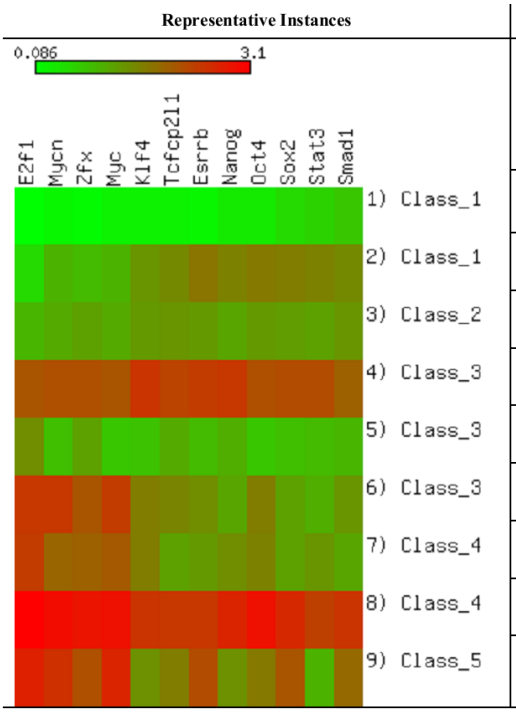
4.3 Knowledge Extraction

Knowledge discovery has been defined as *"a non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns from collections of data"* [7]; its main aim is to reveal some new and useful information from the data.

The most interesting and useful representative instances inferred by *IB* classifiers are the ones obtained by *PEL-C*, which extracts a representation of classes very

concise, regarding the number of instances (only 9 patterns), and expressive, because is composed of a mixture of prototypical instances, with graded abstraction.

In Figure 2 on the left we show the heatmap [11] of typical patterns inferred by *PEL-C*, whereas on the right we reports the representativeness for each pattern inside their own class. This latter index is computed as the ratio between the number of observations assigned to a class using a given representative instance of that class and the total of the instances assigned to that class. The inferred representative instances are the typical patter that explain different level of gene expression. Moreover, these typical patterns is useful to detect how the classes are decomposable in some subclasses and their typicality grade within the own class, in fact we observe that the classes descriptions vary from classes totally based on a prototype (Classes 2 and 5) to classes based on a mixture of prototypes with graded representativeness (Classes 1, 3 and 4).



Number of genes (sample set)	Representativeness (sample set)	Number of genes (test set)	Representativeness (test set)	
300	71.26	5113	69.80	
121	28.74	2212	30.20	
81	100.00	1436	100.00	Full Prototype
117	39.13	2285	42.54	
102	34.12	1687	31.40	
80	26.76	1400	26.06	
79	57.66	1424	53.35	
58	42.34	1245	46.65	
52	100.00	842	100.00	Full Prototype

Fig. 2. The heat map of typical patterns inferred by *PEL-C* and their representativeness

5 Concluding Remarks

We have utilized ChIP-seq and RNA-seq data to explore the relationship between the pattern of *TF* binding activity and gene expression. We show that *IB* classifier systems can be used for quantitative modelling of gene expression levels from binding location data. For the embryonic stem cell, more than *52%* of variation

in gene expression can be explained using binding signals of *12* transcription factors (*TFs*).

As expected only the *TFs* do not explain all the variability of gene expression levels and we should consider other feature definitions criteria or other new features to improve both performances and explanation of gene expression machinery.

Hybrid classifiers as the *PEL-C* are also useful as tool for knowledge discovery providing us the typical patterns of epigenetic factors which explain a considerable part of variability in gene expression. We identify nine typical patterns of transcription factors activation for *5* levels of gene expression (varying from *zero* or *very low* to *very high*). These results provide new potential insights into transcriptional control of gene expression level for embryonic stem cell.

Acknowledgments. This research is supported by "*Italian Flagship Project Epigenomic*" of the Italian Ministry of Education, University and Research and the National Research Council (`http://www.epigen.it/`).

References

1. Soon, W.W., Hariharan, M., Snyder, M.P.: High-throughput sequencing for biology and medicine. Molecular Systems Biology 9, Article number:640 (2013)
2. Hawkins, R.D., Hon, G.C., Ren, B.: Next-generation genomics: an integrative approach. Nature Review Genetics 11(7), 476–486 (2010)
3. Ouyanga, Z., Zhoub, Q., Wongc, W.H.: ChIP-Seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. PNAS 106(51), 21521–21526 (2009)
4. Young, M.D., Willson, T.A., Wakefield, M.J., Trounson, E., Hilton, D.J., Blewitt, M.E., Oshlack, A., Majewski, I.J.: ChIP-seq analysis reveals distinct H3K27me3 profiles that correlate with transcriptional activity. Nucleic Acids Research 39(17), 7415–7427 (2011)
5. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 2nd edn. Morgan Kaufmann, San Francisco (2005)
6. Hastie, T., Tibshirani, R., Friedman, J.: Prototype Methods and Nearest-Neighbors. In: The Elements of Statistical Learning. Data Mining; Inference; and Prediction, 2nd edn., pp. 459–484. Springer, New York (2009)
7. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uturusamy, R.: Advances in Knowledge Discovery and Data Mining. MIT Press, Cambridge (1996)
8. Gagliardi, F.: Instance-based classifiers applied to medical databases: diagnosis and knowledge extraction. Artificial Intelligence in Medicine 52(3), 123–139 (2011)
9. Nieddu, L., Patrizi, G.: Formal methods in pattern recognition: A review. European Journal of Operational Research 120, 459–495 (2000)
10. Gagliardi, F.: Instance-Based Classifiers to Discover the Gradient of Typicality in Data. In: Pirrone, R., Sorbello, F. (eds.) AI*IA 2011. LNCS, vol. 6934, pp. 457–462. Springer, Heidelberg (2011)
11. Pavlidis, P., Noble, W.S.: Matrix2png: A Utility for Visualizing Matrix Data. Bioinformatics 19(2), 295–296 (2003)