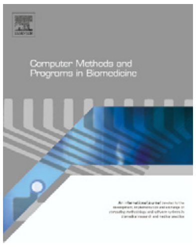

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

A partially supervised classification approach to dominant and recessive human disease gene prediction

Borja Calvo[a,*], *Núria López-Bigas*[b], *Simon J. Furney*[b], *Pedro Larrañaga*[a], *Jose A. Lozano*[a]

[a] *Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV-EHU, Paseo Manuel de Lardizabal 1, E-20018 San Sebastián, Spain*

[b] *Research Unit on Biomedical Informatics, Universitat Pompeu Fabra, Dr. Aiguader 88, E-08003 Barcelona, Spain*

ARTICLE INFO

Article history:
Received 2 October 2006
Received in revised form
30 November 2006
Accepted 8 December 2006

Keywords:
Partially supervised classification
Disease gene prediction
Dominant disease gene
Recessive diseases gene

ABSTRACT

The discovery of the genes involved in genetic diseases is a very important step towards the understanding of the nature of these diseases. In-lab identification is a difficult, time-consuming task, where computational methods can be very useful. In silico identification algorithms can be used as a guide in future studies.

Previous works in this topic have not taken into account that no reliable sets of negative examples are available, as it is not possible to ensure that a given gene is not related to any genetic disease. In this paper, this feature of the nature of the problem is considered, and identification is approached as a partially supervised classification problem.

In addition, we have performed a more specific method to identify disease genes by classifying, for the first time, genes causing dominant and recessive diseases independently. We base this separation on previous results that show that these two types of genes present differences in their sequence properties.

In this paper, we have applied a new model averaging algorithm to the identification of human genes associated with both dominant and recessive Mendelian diseases.

1. Introduction

The identification of genes involved in hereditary diseases is of great importance for the biomedical domain, as this knowledge can lead to improvement in the diagnosis, prognosis or therapy. The process of identifying the genes involved in a particular disease is costly and time-consuming. Several computational methodologies have recently been developed to accelerate this process. Some of these approaches focus on identifying genes likely to be involved in diseases based on sequence properties [1,2]. Other methods focus on the prioritization between positional candidate genes for a particular disease based on function, expression or MEDLINE data [3–8].

Some of the methods used for this kind of predictions usually need positive and negative samples (i.e. a set of genes known to be involved in diseases and a set of genes known not to be involved in diseases). Although the set of positive genes can be generally trusted, producing sets of genes known not to be involved in any disease is not possible. This important feature of the nature of the problem should be considered by the methodology. Classification starting from positive and unlabeled examples is known in the literature as the partially supervised classification problem [9].

[*] *Corresponding author*. Tel.: +34 943018070.
E-mail address: borxa@si.ehu.es (B. Calvo).

0169-2607/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2006.12.003

Recently, it has been demonstrated that disease genes affected by dominant and recessive mutations show significantly different evolutionary profiles [10,11]. For researchers working on the study of a particular disease whose mode of inheritance is known, it would be more effective to use a method specifically built to predict disease genes with that very type of heredity. Thus, a more specific method of disease gene prediction could be built by predicting candidate disease genes causing dominant and recessive mutations separately.

In this paper, we propose a new approach to the disease gene identification problem which takes into account both the absence of reliable negative examples, and the fact that dominant and recessive genes show differential sequence properties. The solution proposed is based on a new algorithm developed to deal with the partially supervised classification problem.

The paper is divided into three different parts. Section 2 introduces the partially supervised classification problem and describes and empirically evaluates our proposed algorithm. In Section 3, the new algorithm is applied to the dominant and recessive disease gene identification, and Section 4 shows the results obtained in the disease gene prediction. Finally, in Section 5 some conclusions and ideas about future work are given.

2. Partially supervised classification

In the supervised classification problem, we start with a given set of elements (usually known as instances or cases) labeled with one class. Each instance is characterized by a set of features and belongs to a given class. Mathematically, each instance is represented by a vector $\boldsymbol{x} \in \Re^n$ of values of random variables and a label $c \in C$. The vector components (the predicting variables) represent the instance's features and the label represents the instance's class. Given a set of labeled examples (called training set), supervised classification algorithms try to induce classification functions g: $\Omega_X \rightarrow \Omega_C$, which, given an instance $\boldsymbol{x}$, predicts its class c by means of $g(\boldsymbol{x})$. In the particular case of binary classifiers (i.e. when C takes two values, 1 and 0, normally referred to as positive and negative), we need training sets containing instances from both positive and negative class. However, in some situations, obtaining examples from one of the classes (typically the negative one) is either difficult or impossible [9–12], as it happens in the identification of disease genes.

Binary classification starting from positive and unlabeled examples has been mainly developed in the text mining domain [9,12–16]. Several algorithms, based on different paradigms, have been proposed. In [9], a solution based on the EM algorithm [17] is presented. An adaptation of the naive Bayes induction algorithm named Positive Naive Bayes (PNB) can be found in Ref. [13]. More details about this algorithm can be found in Section 2.2. Other approaches based on support vector machines are described in Refs. [14–16].

In this paper, we propose a new model averaging procedure named Divergence Convergence Division (DCDiv) that takes as input a set of positive examples and a set of unlabeled instances and returns a classification for the unlabeled cases.

Table 1 – Accuracy, precision, recall and *F* measure definitions

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad \text{precision} = \frac{\text{TP}}{\text{TP+FP}}$$

$$F\ \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \qquad \text{recall} = \frac{\text{TP}}{\text{TP+FN}}$$

	Actual class	Predicted class
True positives (TP)	1	1
False negatives (FN)	1	0
False positives (FP)	0	1
True negatives (TN)	0	0

Accuracy is the fraction of the cases correctly classified, precision is the fraction of the cases labeled as positive (1 in the table) that are actually positive, recall is the fraction of (actual) positive cases that are correctly classified, and the *F* measure is the harmonic mean of precision and recall.

The absence of negative examples makes it infeasible to directly compute classical performance measures, such as accuracy or *F* measure (see Table 1). Due to this reason, in this paper we have tested our algorithm in problems where the absence of negative examples has been simulated. These problems have been built starting from completely labeled real data, and then creating a set of unlabeled data by removing their labels. Taking into account that we know the actual class of all unlabeled classes, we can directly obtain any of the performance measures defined in Table 1.

The rest of this section is organized as follows: Section 2.1 describes the DCDiv algorithm. Then, Section 2.2 empirically compares DCDiv to an adaptation of the PNB algorithm [13] to the case of general discrete variables.

2.1. *DCDiv algorithm*

DCDiv is based on the assumption that the number of positive examples in the set of unlabeled cases (U_0) is lower than the number of negative cases. This assumption, which has already been considered in previous works [9,12,16], makes sense when identifying genes associated with genetic diseases, as we expect that the number of genes related to a genetic disease will be lower than the number of genes not associated with a disease.

Suppose we drew m random samples N_l ($l = 1, \ldots, m$) with replacement from the set of unlabeled instances. If we labeled all the cases in each N_l as negative, these sets could be considered noisy sets of negative examples, being the noise (positive cases in N_l) proportional to the ratio of positive cases in U_0. Now suppose that a set M of m different models $M = \{M_1, \ldots, M_m\}$ were inducted, using the set of positive cases (P_0) as positive examples and each of the N_l ($l = 1, \ldots, m$) as negative examples. Each model $M_l \in M$ could be used to estimate the probability of a given instance being negative ($P_{M_l}(C = 0|\boldsymbol{x})$) or positive ($P_{M_l}(C = 1|\boldsymbol{x})$). Thus, for each model M_l and each case $\boldsymbol{x}$ the following ratio, $F_l(\boldsymbol{x})$, could be obtained:

$$F_l(\boldsymbol{x}) = \frac{P_{M_l}(C = 0|\boldsymbol{x})}{P_{M_l}(C = 1|\boldsymbol{x})}$$

It is important to notice that any classification paradigm that allows us to estimate the former probabilities can be used at this step. When $\boldsymbol{x}$ is a positive instance and it is correctly classified by M_l, the former ratio will be lower than 1 and, in case $\boldsymbol{x}$ is a negative example, its value will be greater than 1. On the other hand, given a case $\boldsymbol{x}$, some of the models in M will classify it correctly, while the rest will misclassify it. If the amount of noise in each N_l set is low, we can expect that most of the models in M will classify $\boldsymbol{x}$ correctly. In this situation, if we compute, for each $\boldsymbol{x}$, the limit of the product

$$\prod_{l=1}^{m} F_l(\boldsymbol{x}) = \prod_{l=1}^{m} \frac{P_{M_l}(C=0|\boldsymbol{x})}{P_{M_l}(C=1|\boldsymbol{x})} \quad (1)$$

when m goes to infinity, we expect that the product will converge to 0 if $\boldsymbol{x}$ is a positive instance, while it will diverge to infinity when $\boldsymbol{x}$ is a negative case. DCDiv takes advantage of the differential behavior of the product for positive and negative examples in order to separate them.

DCDiv works in an iterative way. At each step, a model is built and a new factor is added to the product. Then, the unlabeled cases are partitioned according to their product value by means of a threshold. Those cases with a product value greater than the threshold are considered negative and the remaining, positive. As new terms are added to the product, the differences between its value for positive and negative cases will increase, allowing a better partitioning of the unlabeled cases.

In order to set the threshold, the ‘spy case’ concept introduced in Ref. [9] is used. Spy cases are a set S of positive examples, taken from P_0, which are placed in U_0 and considered unlabeled instances, as shown in Fig. 1. Spy cases are supposed to behave in the same way the positive cases hidden in U_0 do, so they can be used to set the threshold. After this modification, new sets of positive and unlabeled cases are created ($P = P_0/S$ and $U = U_0 \cup S$, respectively). At each iteration the threshold is set in the smallest value such that, for a given fraction of the spy cases (named recall), their product is lower than or equal to the threshold. Fixing the threshold at this value, that fraction (the recall) of the spy cases will be correctly classified as positive cases.

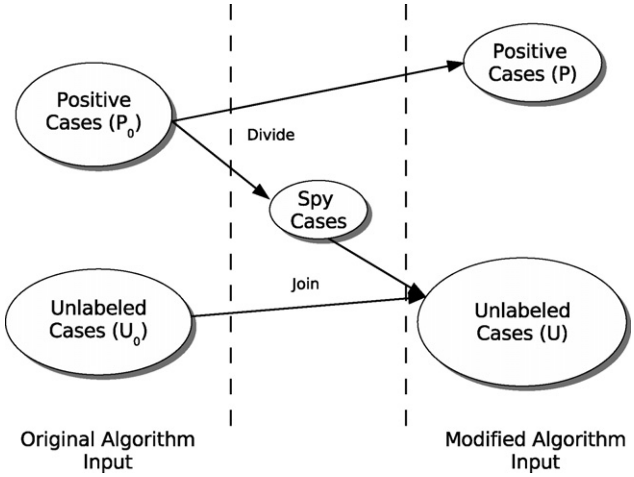

Fig. 1 – Insertion of the spy cases. Scheme of the process of inserting the spy cases into the set of unlabeled instances.

Taking into account that the threshold is actually the product of a positive case (a spy cases), we expect it to converge to zero. We have used this as a stop criterion. The algorithm is halted either when the threshold converges to zero or when a maximum number of iterations is reached. The pseudo-code of the algorithm can be consulted in Fig. 2.

In order to set the recall, the higher the value the better, as it is an estimation of the ratio of positive cases that the algorithm will recover, but a too high recall can lead to the non convergence of the threshold. Thus, we have set it at the highest value that allows the threshold to converge to zero.

The DCDiv algorithm is a model averaging process, where bootstrap samples [18] are drawn from U_0 and models are combined by means of the product in Eq. (1). If we take the product's logarithm

$$\ln\left(\prod_{l=1}^{m} F_l(\boldsymbol{x})\right) = \sum_{l=1}^{m} [\ln(P_{M_l}(C=0|\boldsymbol{x})) - \ln(P_{M_l}(C=1|\boldsymbol{x}))]$$

we have the sum of the logit function of $P(C=0|\boldsymbol{x})$, used in logistic regression. This equation can be rewritten as $\ln(\Pi_{l=1}^{m} F_l(\boldsymbol{x})) = \sum_{l=1}^{m} w_l(\boldsymbol{x})$, where $w_l(\boldsymbol{x}) = \ln(P_{M_l}(C=0|\boldsymbol{x})) - \ln(P_{M_l}(C=1|\boldsymbol{x}))$. We can see this as a weighted voting scheme where the weights depend on the logarithm of the posterior probabilities. This resembles bagging predictors [19], but with some differences. In DCDiv, bootstrap samples are only drawn from the set of unlabeled instances, and the voting is weighted by a function that depends on the logarithm of the posterior probabilities, rather than bagging's simple voting. The main difference with bagging is that, in DCDiv, the threshold is set dynamically at each step, instead of using a prefixed threshold. Model averaging algorithms are useful when applied to unstable algorithms [19]. In this problem, the instability not only comes from the classification paradigm itself, but also from the variability in the amount of noise in N_l.

2.2. *DCDiv experimental evaluation*

Performance evaluation in partially supervised classification is a non-solved problem. The absence of reliable negative examples makes it impossible to estimate measures such as accuracy or F measure (see Table 1), making it unfeasible to compare two or more classifiers in real-life problems. In order to overcome this problem, we have evaluated our algorithm in datasets where the absence of negative examples has been simulated, comparing it to PNB [12,13], one of the state-of-the-art algorithms in partially supervised classification. It must be stressed that these simulated problems have been built starting from real datasets and not from artificially constructed data. We have only simulated the absence of negative examples, not the data itself.

2.2.1. *Adaptation of PNB*

PNB was initially proposed in the text classification domain. In the original paper [13], instances (text documents) were represented as a bag of words. Thus, the equations in Ref. [13] are adapted to this particular way of representing documents. We

```
Inputs
  $U_0$ := set of unlabeled examples
  $P_0$ := set of positive examples
  $r$ := recall
  $s$ := fraction of cases in $P_0$ used as spies
  $Limit$ := maximum number of iterations allowed
Initialization
  Randomly divide $P_0$ into $P$ and $S$, such that the number of cases in $S$ is $s \cdot |P_0|$
  Create $U$ by joining $S$ and $U_0$
  for each case $\boldsymbol{x}$ in $U$
      $prod_0(\boldsymbol{x}) := 1$
  rof
  $l := 1$
Iterations
  do
      Select randomly with replacement $|P_0|$ cases from $U$ to construct $N_l$
      Build a model $M_l$ using $P$ and $N_l$ as positive and negative examples respectively
      for each case $\boldsymbol{x}$ in $U$ such that $\boldsymbol{x} \notin N_l$
          $prod_l(\boldsymbol{x}) = prod_{l-1}(\boldsymbol{x}) \cdot \frac{P_{M_l}(C=0|\boldsymbol{x})}{P_{M_l}(C=1|\boldsymbol{x})}$
      rof
      Set $t$ in the lowest value such that, for $r\%$ of cases in $S$, $prod_l(\boldsymbol{x}) \leq t$
      $l = l + 1$
  while $t > 0$ and $l \leq Limit$
Labelling
  for each case $\boldsymbol{x}$ in $U_0$
      if $prod_l(\boldsymbol{x}) > t$
          $\boldsymbol{x}$ is labelled as negative
      else
          $\boldsymbol{x}$ is labelled as positive
      fi
  rof
```

Fig. 2 – The DCDiv algorithm's pseudo-code.

have reformulated these equations to apply them to the case of general discrete variables.

PNB is based on the naive Bayes algorithm [20]. Lets suppose that each of the predicting variables X_i can take r_i different values and C can take two values, 1 and 0. Given the Bayes rule and under the assumption of conditional independence, we have that, for a given instance **x**:

$$P(C = c|\mathbf{x}) \propto P(C = c)\prod_{i=1}^{n} P(X_i = x_i|C = c)$$

In a naive Bayes model, the parameters needed to define the model are $P(C=1)$, $P(X_i=j|C=1)$ and $P(X_i=j|C=0)$ for all $i=1, \ldots, n$ and $j=1, \ldots, r_i-1$. These parameters are normally estimated from the data by maximum likelihood estimators. In partially supervised classification problems, $P(X_i=j|C=1)$ can be estimated from the positive examples, but neither $P(X_i=j|C=0)$ nor $P(C=1)$ can be obtained from data. However, according to [13], $P(X_i=j|C=0)$ can be expressed as

$$\frac{|U_{ij}| - \hat{P}(X_i = j|C = 1)\hat{P}(C = 1)|U|}{(1 - \hat{P}(C = 1))|U|} \tag{2}$$

where $|U_{ij}|$ represents the number of instances in the set of unlabeled cases where $X_i=j$ and $|U|$ is the total number of instances in the same set. The problem with this estimator is that it can be negative. Denis et al. propose in Ref. [13] to replace the negative values with 0, and then normalize the probabilities. Taking the normalization and the Laplace correction into account we can estimate $P(X_i=j|C=0)$ as

$$\frac{1 + \max(0; R_i(j))(1/Z_i)}{r_i + (1 - \hat{P}(C = 1))|U|}, \quad R_i(j) = |U_{ij}| - \hat{P}(X_i = j|C = 1)\hat{P}(C = 1)|U|,$$

$$Z_i = \sum_{j=1}^{r_i} \max(0; \hat{P}(X_i = j|C = 0)) \tag{3}$$

$P(C=1)$ cannot be estimated from data and, therefore, the user must introduce it as a parameter.

To sum up, positive naive Bayes estimates $P(X_i=j|C=1)$ from positive examples by means of a softened maximum likelihood estimator, $\hat{P}(C = 1)$ is a parameter set by the user, and $P(X_i=j|C=0)$ is estimated using Eq. (3). The implementation of this adaptation of the PNB can be found at the supplementary data web page.[1]

2.2.2. *DCDiv versus PNB*

In order to compare DCDiv and PNB, we have built simulated problems starting from real databases where all the instances are labeled. The process of simulating positive unlabled learning problems begins selecting one class in the database as the positive one. Then, all the instances belonging to this class are labeled as positive examples, and those not belonging to the positive class are labeled as negative. Once all

[1] http://www.sc.ehu.es/ccwbayes/members/borxa/DCDiv/supp_data.html.

the instances are labeled, the set of unlabeled instances is built by joining together some positive and negative instances and removing their labels. The set of positive examples is obtained from the remaining positive instances. Using simulated problems in the comparison not only allows us to evaluate the classification performance, but also to control the ratio of positive cases hidden in the set of unlabeled instances.

In this paper, we have constructed simulations starting from three different databases: *ACCDON*, *Letter Recognition* and *Nursery*. The first one is a database of splice sites, described in Ref. [21], and the other two are databases from the UCI repository [22].

The *ACCDON* database contains sequences of donor and acceptor splice sites. The 2-bp-long constant part of every splice site has been removed. The database consists of six datasets: true acceptor sites, false acceptor sites obtained from coding regions, false acceptor sites obtained from intronic regions, true donor sites, false donor sites obtained from coding regions and false donor sites obtained from intronic regions. Three groups of positive and negative instances have been obtained from acceptor datasets and another three from donor datasets (one containing negatives only from coding regions, another containing negative examples from intronic regions and a last one containing a mixture of coding and intronic sites as negative examples).

Letter Recognition is a set of instances distributed in 26 classes (the letters in the roman alphabet). Three more groups of positive and negative instances have been constructed starting from this database, taking as positive class letters ‘D’, ‘P’ and ‘U’, respectively.

The *Nursery* database contains cases from five classes. A last group of positive and negative examples has been constructed by selecting the *spec_prior* class as the positive one.

Starting from each of these 10 groups of positive and negative instances, 9 different schemes have been defined varying the number of positive cases in U_0 ($|U_p|$) and the cardinality of P_0 ($|P_0|$). The number of negative instances hidden in U_0 was the same in all the schemes (2000 for groups based on *ACCDON* and *Letter Recognition* and 8916 for those based on Nursery).

Both the PNB and the DCDiv algorithms have been applied to the previously described datasets. The PNB algorithm have been run with three values for $P(C=1)$: 0.5, 0.25 and the actual value. The DCDiv algorithm has been applied using tree augmented naive Bayes models (TAN) [23] as base classifier. This classification paradigm is an extension of the naive Bayes, where each variable can have two parents at most: the class and another variable. Ten percent of the positive examples were used as spy cases, the maximum number of iterations was set to 1000 and we considered the threshold to have converged to zero when its value reached the computational zero (which was at 10^{-324}).

The comparison between PNB and DCDiv has been done in terms of accuracy and the F measure, a typical measure in information retrieval problems (see Table 1). The significance of the differences between the results obtained with each algorithm has been tested with the Wilcoxon signed rank test at a confidence level of 99%. An extract of the results obtained in the empirical comparison (those corresponding to datasets based on the database of donor sites where negative examples are a mixture of splice sites obtained from coding and intronic regions) is shown in Table 2. The complete set of results can be consulted at the supplementary web page. Each row in the table represents one of the nine schemes previously defined and shows the average result obtained in 50 different problems built according to the scheme. Table 2 shows that DCDiv outperforms PNB when $\hat{P}(C=1)$ is set to 0.25 in most of the datasets. PNB only outperforms DCDiv when the number of positive cases in U is 500. The reason is that, in these datasets, the actual $P(C=1)$ is 0.2, which is close to the value used in the PNB algorithm (0.25). When the number of positive cases hidden in U_0 is low, no significative differences can be found in the F measure between DCDiv and PNB when the actual value of $P(C=1)$ is used as parameter (i.e. with the best possible result of the PNB). Similar results were obtained with the rest of the datasets based on *ACCDON*. In some of the datasets based on *Letter Recog-*

Table 2 – Comparison results

$\|P_0\|$	$\|U_p\|$	PNB-0.25		DCDiv		PNB-Actual		DCDiv	
		F	Acc	F	Acc	F	Acc	F	Acc
100	100	45.21	88.60	**72.40**	**97.54**	73.01	**97.69**	72.40	97.54
100	300	73.36	90.83	74.63	**94.51**	**83.78**	**95.92**	74.63	94.51
100	500	**87.41**	**94.66**	74.05	91.64	**87.63**	**95.12**	74.05	91.64
200	100	45.42	88.72	**71.84**	**97.48**	73.48	**97.74**	71.84	97.48
200	300	73.36	90.78	**78.28**	**95.07**	**84.15**	**96.01**	78.28	95.07
200	500	**88.22**	**95.00**	77.67	92.56	**88.51**	**95.45**	77.67	92.56
300	100	45.74	88.87	**73.30**	**97.61**	73.99	**97.77**	73.30	97.61
300	300	73.42	90.83	**78.00**	**95.03**	**84.40**	**96.07**	78.00	95.03
300	500	**88.57**	**95.14**	78.53	92.80	**88.49**	**95.45**	78.53	92.80

Extract of the results obtained in the empirical comparison, corresponding to datasets built from the database of donor sites with a mixture of negatives extracted from coding and intronic regions. $|P_0|$ represents the cardinality of P_0, and $|U_p|$, the number of positive examples hidden in U_0. Two values of $\hat{P}(C=1)$ are shown for the PNB algorithm: 0.25 and the actual probability. For both algorithms, the F measure (F) and accuracy (Acc) are reported. Results shown in each row are the average of the 50 problems built using the schemes described in the first two columns. The significance of the differences has been tested with the Wilcoxon signed rank test at a confidence level of 99%. The best values for each dataset where significant differences have been found are in bold text.

nition and *Nursery*, DCDiv even improves the performance of the PNB with the real value. These results suggest that DCDiv is a competitive option in partially supervised classification.

3. Dominant and recessive disease gene prediction

The identification of genes likely to be involved in genetic diseases can be posed as a partially supervised classification problem. Compared to other approaches, this is a more realistic modelization of the problem, as it takes into account the absence of reliable negative examples. Given recent evidence that suggests that genes associated with dominant and recessive diseases show different evolutionary patterns [10,11], we have regarded the prediction of these two kind of genes as separated problems.

Section 3.1 describes the datasets used in the prediction. In Section 3.2, DCDiv is applied to the identification of genes associated with dominant and recessive diseases.

3.1. *Data*

The list of genes involved in hereditary diseases was obtained from the morbid map table in the OMIM database as described in Ref. [2]. This list comprises 1647 genes that have been found to be the causative genes for particular Mendelian diseases when mutated. Disease genes were classified according to the mode of inheritance of the disease they cause using text mining automatic extraction from the clinical synopsis section in the OMIM database and manual curation. A total of 498 disease genes were classified as dominant, 662 as recessive and the rest, as X-linked, chromosomal rearrangements, association for complex traits or unknown mode of inheritance [10].

The list of sequence properties was computed for the longest protein sequence of each gene in the human genome [24]. The properties used for prediction were the conservation score in several eukaryotic genomes (*Pan troglodytes*, *Mus musculus*, *Rattus norvegicus*, *Gallus gallus*, *Fugu rubripes*, *Danio rerio*, *Drosophila melanogaster*, *Anopheles gambiae*, *Caenorhabditis elegans* and *Caenorhabditis briggsae*), the protein length, the gene length, the number of introns and the number of Low Complexity Regions. The conservation score is a measure that gives an estimation of the mutation rate to which the protein has been subject during evolution that is independent of the length of the protein [2]. Protein and gene length, and the number of introns have been previously found to be important factors to determine the probability of a gene to be involved in a disease [25]. Previous reports also associate the presence of Low Complexity Regions with human disease genes [26].

All these properties have been codified as a 15-variable vector. We have created four datasets: *positive_dominant*, *unlabeled_dominant*, *positive_recessive* and *unlabeled_recessive*. The first one contains those instances (genes) that are known to be involved in dominant diseases and the second one contains the rest of the 19,548 genes. The third and fourth datasets are equivalent to the previous ones but in the case of recessive diseases.

3.2. *Essays*

We have carried out the prediction of dominant and recessive diseases genes separately, using the *positive_dominant* and *unlabeled_dominant* datasets for dominant disease gene prediction, and the *positive_recessive* and *unlabeled_recessive* for recessive disease gene prediction. This is as typical approach in information retrieval. We have a large set of unlabeled cases and we want to retrieve those instances belonging to one of the classes.

DCDiv algorithm has been applied to the former datasets using TAN models [23] as base classifier. In both dominant and recessive gene identification, the recall has been fixed at 75%, and 10% of the positive examples have been used as spy cases. The essays have been repeated 50 times. As TAN models are based on discrete variables, the data described in Section 3.1 have been discretized into 10 intervals with equal frequency discretization. The stop criterion used during the essays was a combination of the convergence of the threshold to zero and a limited number of iterations. This limit has been set to 3000 and, as in the experimental evaluation, the threshold was considered to have converged when it reached the computational zero.

4. Results

The result of each repetition is a classification of the unlabeled instances. We do not obtain, for each gene, the probability of being positive or negative. This is a disadvantage if we want to rank the genes according to their probability of being related to genetic diseases. In order to overcome this problem, we have averaged all the classifications by calculating, for each gene, the fraction of repetitions where it was classified as positive. Thus, we have two ratio for each gene: one for dominant mutations and another one for recessive mutations. These ratios can be used to rank all the genes and to select, given a list of genes, which are more likely to cause genetic diseases. The complete classification, a list of more than 19,000 genes containing their name, a brief description and their classifications, is available at the supplementary data web page. In addition to the raw data, in order to improve the accessibility of the results, we have created a web server where predictions can be consulted.[2] The queries can be done according to different criteria, like the gene ID, chromosomic regions, etc.

We can also look at the results more generally, considering the number and the spatial distribution of the genes predicted as causative of genetic disease. In order to obtain an average classification, we need to set a threshold for the previously defined ratio. The higher the threshold, the more restrictive the classification will be (the fewer identified genes we will have). A classical threshold would be 0.5, i.e. an instance is considered positive when it has been classified as positive in at least half of the repetitions. This is equivalent to a simple voting. Setting the threshold at 0.5, 6558 genes were identified as associated with dominant diseases (34.5% of the unlabeled

[2] This service is available at http://bg.imim.es/idgp.

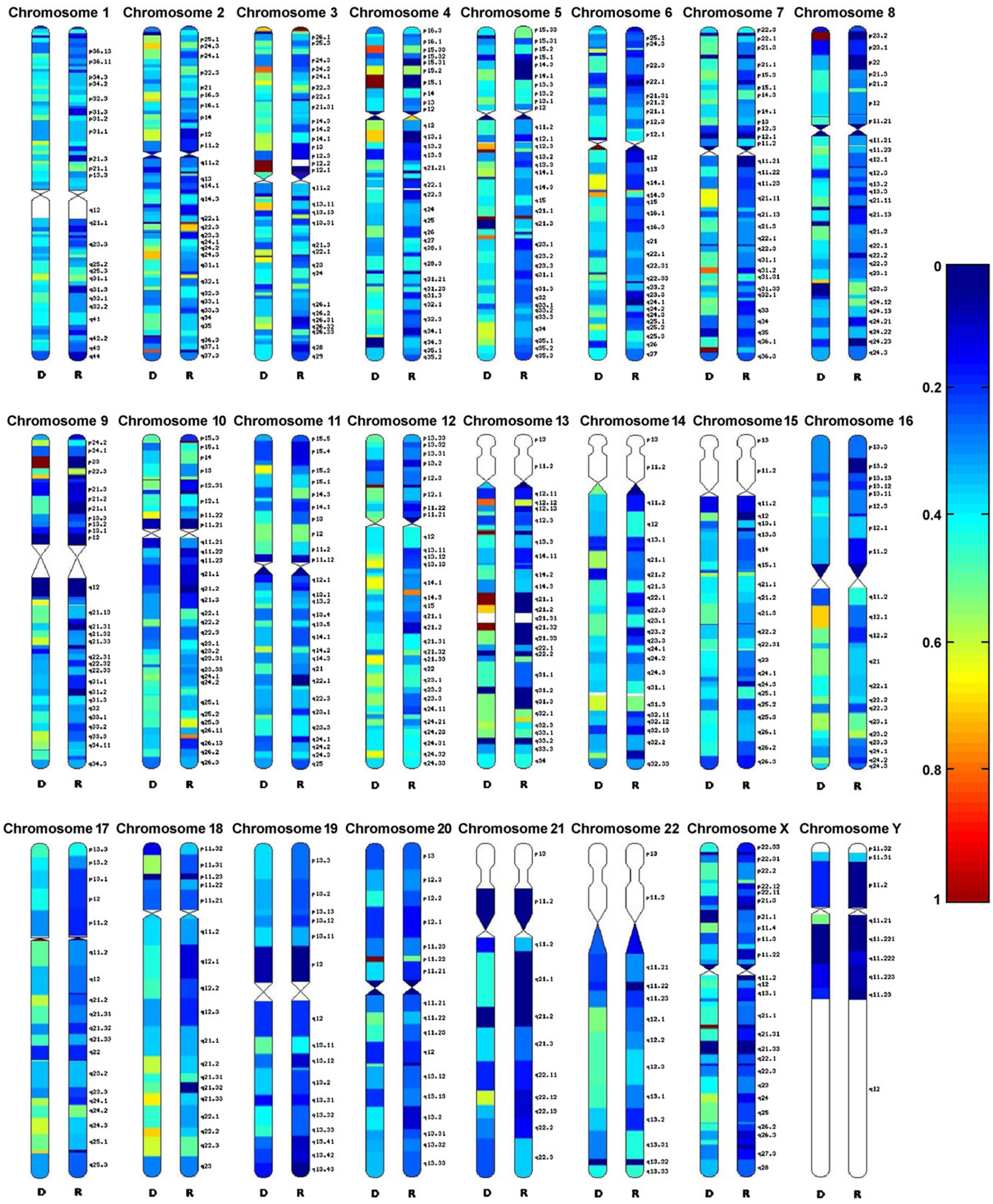

Fig. 3 – Spatial distribution—representation of the fraction of genes that are predicted as related to dominant (D) and recessive (R) disease by chromosome regions. One should take into account that different regions have different numbers of known genes, and this information is not represented in the figure.

genes), and 4582 as associated with recessive genes (24.2% of the unlabeled genes). 1742 (18.5%) of the predicted genes have been labeled as causing both dominant and recessive diseases. The most restrictive threshold would be 1, which would consider a gene as disease-related only if identified in all the repetitions. Considering this threshold, only 2415 genes (12.7%) would be identified as dominant disease-associated and 1880 (9.9%), as recessive disease-associated. Two hundred and eighty eight (7.19%) of the predicted genes have been labeled as causing both dominant and recessive diseases. These results are consistent with the hypothesis made at the beginning of the paper (the number of genes classified as positive is lower than the number of the ones classified as negative). These ratios can be seen as an estimation of the $P(C=1)$ in each set of unlabeled examples. As a way to measure the robustness of the results, we have applied the PNB algorithm using these estimations as input, and we have compared the results obtained with the two algorithms. Depending on the dataset and the threshold considered, the two algorithms asign the same class to between the 72% and the 86% of the instances.

Focusing on the known disease genes (the positive cases), we can see that the number of genes related to dominant diseases is lower than the number of those associated with recessive diseases (498 and 662, respectively). This relation is inverted in the predicted disease genes, where the number of dominant genes predicted is greater than the number of recessive genes. 5.15% of the known positive genes are related to both dominant and recessive diseases.

In addition to the gene classification, we have also represented the spatial distribution of the genes predicted to be associated with disease, both for recessive and dominant mutations. The classification on which this spatial distribution is based is the one obtained by simple voting. For each chromosomic region, we have obtained the ratio between the number of (dominant and recessive) disease genes predicted and the total number of genes in that region (considering only those genes about which we have information). The comparison of the spatial distribution of dominant and recessive genes can be seen in Fig. 3.

5. Conclusions and future work

In this work, we have seen that the prediction of disease genes from sequence properties can be modeled as a partially supervised classification problem. A new algorithm to deal with this problem has been developed and applied to the identification of disease genes. This new algorithm has been empirically compared to one of the state-of-the-art algorithms in partially supervised classification, showing that it gives competitive results in simulated problems.

Recent results suggesting that dominant and recessive disease genes show different sequence properties have been also taken into account, and for the first time, prediction has been specifically done for genes involved in dominant and recessive hereditary diseases. We expect these predictions to be of help for those research groups working on the identification of genes involved in particular diseases, serving as a guide in their study.

In this paper, we have only considered Mendelian disease genes as a first approach to the problem. Taking into account other multigenic diseases in the prediction poses a more complex problem that will be considered for future development of this work.

This kind of approach has been used in other biological problems such as the prediction of functional RNA genes [27,28]. Most probably, many other biological problems can be tackled as partially supervised problems. Indeed, whenever we face a binary classification where obtaining examples of any of the two classes is difficult or impossible, this kind of algorithms would be the right choice. For the future remains the challenge of searching for new applications of partially supervised classification in bioinformatics.

There are several questions related to the algorithm that are still to be developed. A first question is how to use the algorithm's output to obtain a model of the problem. A simple solution would be to use the classification obtained to build a model. A more elaborated approach could be initializing the EM algorithm [17] with DCDiv's output. The inclusion of an adaptation of the recall vs. precision curve in the algorithm, and a theoretical study of the properties of the DCDiv algorithm are also pending tasks.

In this paper, we have used a one against the others approach, as all the algorithms developed in the partially supervised classification context are focused on binary classification. Nevertheless, it would be interesting to explore two (or more) against the other approaches. This would lead us to a generalization of the partially supervised classification problem, where we have m different classes, but examples from only k classes, with $k=2, 3, \ldots, m-1$.

Finally, finding a way to compare algorithms in real-life problems (i.e. in the absence of negative examples) is another task we would like to tackle in the future.

Acknowledgments

This work was supported by the Spanish Ministry of Education and Science (TIN2005-03824) and the Basque Government (ETORTEK and SAIOTEK programs). N.L.-B. is recipient of a Ramon y Cajal contract of the Spanish Ministry of Education and Science (MEC) and acknowledges funding from the International Human Frontier Science Program Organization (HFSPO). S.J.F. acknowledges support from EMBO Short Term Fellowship ASTF 83-2005.

REFERENCES

[1] E.A. Adie, R.R. Adams, K.L. Evans, D.J. Porteous, B.S. Pickard, Speeding disease gene discovery by sequence based candidate prioritization, BMC Bioinform. 6 (2005) 55.

[2] N. López-Bigas, C.A. Ouzounis, Genome-wide identification of genes likely to be involved in human genetic disease, Nucleic Acids Res. 32 (10) (2004) 3108–3114.

[3] M.A. van Driel, K. Cuelenaere, P.P. Kemmeren, J.A. Leunissen, H.G. Brunner, A new web-based data mining tool for the identification of candidate genes for human genetic disorders, Eur. J. Hum. Genet. 11 (2003) 57–63.

[4] M.A. van Driel, K. Cuelenaere, P.P. Kemmeren, J.A. Leunissen, H.G. Brunner, G. Vriend, Geneseeker: extraction and integration of human disease-related information from web-based genetic databases, Nucleic Acids Res. 33 (2005) W758–W761.

[5] C. Pérez-Iratxeta, P. Bork, M.A. Andrade, Association of genes to genetically inherited diseases using data mining, Nat. Genet. 31 (2002) 316–319.

[6] D.G. Silva, C. Schonbach, V. Brusic, L.A. Socha, T. Nagashima, N. Petrovsky, Identification of "pathologs" (disease-related genes) from the riken mouse cdna dataset using human curation plus facts, a new biological information extraction system, BMC Genomics 5 (2004) 28.

[7] N. Tiffin, J.F. Kelso, A.R. Powell, H. Pan, V.B. Bajic, W.A. Hide, Integration of text- and data-mining using ontologies successfully selects disease gene candidates, Nucleic Acids Res. 33 (2005) 1544–1552.

[8] F.S. Turner, D.R. Clutterbuck, C.A. Semple, Pocus: mining genomic sequence annotation to predict disease genes, Genome Biol. 4 (2003) R75.

[9] B. Liu, W.S. Lee, P.S. Yu, X. Li, Partially supervised classification of text documents, in: Proceedings of the Nineteenth International Conference on Machine Learning (ICML-2002), July 2002.

[10] N. López-Bigas, B.B. Blencowe, C.A. Ouzounis, Highly consistent patterns for inherited human diseases at the molecular level, Bioinformatics 22 (3) (2006) 269–277.

[11] S.J. Furney, M. del Mar Albá, N. López-Bigas, Differences in the evolutionary history of disease genes affected by dominant or recessive mutations, BMC Genomics 3 (7) (2006) 165.

[12] F. Denis, A. Laurent, R. Gilleron, M. Tommasi, Text classification and co-training from positive and unlabeled examples, in: Proceedings of the ICML 2003 Workshop: The Continuum from Labeled to Unlabeled Data, 2003, pp. 80–87.

[13] F. Denis, R. Gilleron, M. Tommasi, Text classification from positive and unlabeled examples, in: The Ninth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, 2002.

[14] X. Li, B. Liu, Learning to classify texts using positive and unlabeled data, in: Proceeding of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), August 2003.

[15] B. Liu, Y. Dai, X. Li, W.S. Lee, P.S. Yu, Building text classifiers using positive and unlabeled examples, in: Third IEEE International Conference on Data Mining (ICDM'03), November 2003, p. 179.

[16] H. Yu, C.X. Zhai, J. Han, Text classification from positive and unlabeled documents, in: Proceedings of the Twelfth International Conference on Information and Knowledge Management, ACM Press, November 2003, pp. 232–239.

[17] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B 39 (1977) 1–38.

[18] B. Efron, Estimating the error rate of a prediction rule: improvement on cross-validation, J. Am. Stat. Assoc. 78 (1983) 316–331.

[19] L. Breiman, Bagging predictors, Mach. Learn. 26 (2) (1996) 123–140.

[20] M. Minsky, Steps toward artificial intelligence, Proc. Inst. Radio Eng. 49 (1961) 8–30.

[21] R. Castelo, R. Guigó, Splice site identification by idlBNs, Bioinformatics 4 (20) (2004) I69–I72.

[22] C.L. Blake, C.J. Merz. UCI Repository of Machine Learning Databases, 1998.

[23] N. Friedman, D. Geiger, M. Goldszmit, Bayesian network classifiers, Mach. Learn. 29 (1997) 131–163.

[24] E. Birney, D. Andrews, P. Bevan, M. Caccamo, G. Cameron, Y. Chen, L. Clarke, G. Coates, T. Cox, J. Cuff, V. Curwen, T. Cutts, T. Down, R. Durbin, E. Eyras, X.M. Fernandez-Suarez, P. Gane, B. Gibbins, J. Gilbert, M. Hammond, H. Hotz, V. Iyer, A. Kahari, K. Jekosch, A. Kasprzyk, D. Keefe, S. Keenan, H. Lehvaslaiho, G. McVicker, C. Mel-sopp, P. Meidl, E. Mongin, R. Pettett, S. Potter, G. Proctor, M. Rae, S. Searle, G. Slater, D. Smedley, J. Smith, W. Spooner, A. Stabenau, J. Stalker, R. Storey, A. Ureta-Vidal, C. Woodwark, M. Clamp, T. Hubbard, Ensembl 2004, Nucleic Acids Res. 32 (2004) D468–D470.

[25] N. López-Bigas, B. Audit, C. Ouzounis, G. Parra, R. Guigó, Are splicing mutations the most frecuent cause of hereditary disease? FEBS Lett. 579 (9) (2005) 1900–1903.

[26] S. Karlin, L. Brocchieri, A. Bergman, J. Mrazek, A.J. Gentles, Amino acid runs in eukaryotic proteomes and disease associations, Proc. Natl. Acad. Sci. U.S.A. 99 (2002) 333–338.

[27] C. Wang, C. Ding, R.F. Meraz, S.R. Holbrook, PSoL: A positive sample only learning algorithm for finding non-coding RNA genes, Bioinformatics 22 (21) (2006) 2590–2596.

[28] R.F. Meraz, X. He, C.H.Q. Ding, S.R. Holbrook, Positive sample only learning (PSOL) for predicting RNA genes in *E. coli*, in: Proceedings of the IEEE Computer Society Bioinformatics Conference, August 2004, pp. 535–538.