2011 International Conference on Bioscience, Biochemistry and Bioinformatics
IPCBEE vol.5 (2011) © (2011) IACSIT Press, Singapore

Extracting Knowledge from a Mutation Database Related to Human Monogenic Disease Using Inductive Logic Programming

Tien-Dao Luu[1], Ngoc-Hoan Nguyen[1*], Anne Friedrich[2], Jean Muller[1,3] and Olivier Poch[1]
[1] Institute of Genetics and Molecular and Cellular Biology, Illkirch, France
[2] Department of Genetics, Genomics and Microbiology, University of Strasbourg, Strasbourg, France
[3] Genetic Diagnostics Laboratory, CHU Strasbourg Nouvel Hôpital Civil, Strasbourg, France
[*] Corresponding author: nguyen@igbmc.fr

Abstract— **Understanding the effects of genetic variation on the phenotype of an individual is a major goal of biomedical research, especially for the development of diagnostics and effective therapeutic solutions. In this work, we propose a methodology using Inductive Logic Programming (ILP) to automatically extract knowledge about deleterious/neutral mutations from a multi-relational database, named SM2PH-db. We used 8117 mutations in 805 proteins with known three-dimensional structure in our analysis. After using ILP for learning, we obtained classification rules that can be interpreted by a human expert and that help to improve our understanding of the relationships between physico-chemical and evolutionary features and deleterious mutations. Our experimental results, compared with state-of-the-art methods, show that the proposed approach can be applied to predict the impact of single amino acid replacement on the function of a protein. The rules and the estimated effect of human non-synonymous polymorphisms on the function of a protein are available at http://decrypthon.igbmc.fr/sm2ph/prediction.htm.**

Keywords- Inductive Logic Programming; knowledge discovery and data mining; single nucleotide polymorphisms; genotype-phenotype relationship; SM2PH-db

I. INTRODUCTION

Single nucleotide polymorphisms (SNPs) refer to a genetic change in which one nucleotide is replaced by another one and represent one of the most common forms of human genomic variation. SNPs are highly abundant, stable and distributed throughout the genome [1]. Although SNPs are primarily associated with population diversity and individuality, they can also be linked to the emergence or the predisposition to disease, influencing its severity, its progression or drug sensitivity.

SNPs directly linked to disease emergence are considered as deleterious. These deleterious SNPs occur in both non protein-coding and protein-coding regions. In the first case, the variation will usually affect gene expression by disrupting transcription factor binding sites, splice sites or other functional sites at the transcriptional level. Protein-coding SNPs can be further divided into synonymous and non-synonymous (nsSNPs). nsSNPs, also called missense mutations, result in the alteration of the amino acid sequence of the encoded protein. nsSNPs have been linked to a wide variety of diseases, for example by affecting protein function, by reducing protein solubility or by destabilizing protein structure [2]. These protein alterations can be considered to be the primary molecular phenotype linked to the missense mutation, with a cascade of consequences that finally leads to the emergence of a genetic disease and the associated phenotype. The elucidation of the complex relationships linking genotypic and phenotypic variations is a major challenge in the post-genomic era.

With the huge amount of information now available in various biological databases, including sequences, structures, functions, pathways, together with the data related to their interactions and variations [3], the development of *in silico* analysis tools is now possible to better understand and/or to predict the correlation between a missense mutation and the associated molecular phenotypes.

Several research groups have addressed this topic and have developed tools aimed at predicting the effects of nsSNPs on the function of a protein, with varying degrees of success [4].

Current methods of prediction can be divided into two main categories. The first category encompasses sequence-based methods, generally based on multiple sequence alignments and incorporating different approaches to quantify the conservation of a residue during evolution. This category includes SIFT [5], PANTHER PSEC [6], PMUT [7], PhD-SNP [8], SNAP [9] and LRT [10]. The second category combines both sequence and protein 3D structure data. Although these methods are limited by the availability of structural data, various studies have shown that the inclusion of structural information can improve the performance of prediction methods based only on sequence data [11]. These studies also provide evidence that a majority of nsSNPs has an impact on the structure [12]. The most widely used methods in this category are Polyphen [13], nsSNPAnalyzer [14], SNPs3D [15] and more recently AutoMute [16].

The effectiveness of a prediction method is mainly based on the choice of predictors and on the underlying computational approaches. The latter are numerous and include the use of empirically derived rules [13], Support Vector Machine [8] [15], neural networks [7] [9], random forest [14] [16], Hidden Markov Model [6] or other statistical models [5].

All these methods have their strengths and weaknesses (for review see [4]). While it is not straightforward to compare these methods using the same quality criteria, most

of them seem to perform well for classification purposes. In particular, most of them classify nsSNPs as either deleterious (strong functional effect) or neutral (weak functional effect) with high accuracy. Unfortunately, little explanation concerning the decision computed by these prediction tools is available. PolyPhen provides the rules to predict the effect of nsSNPs on protein function and structure, but these rules are derived empirically. Access to such information is essential in order to understand how genetic alterations affect gene products at the molecular level and subsequently to elucidate the relationships between genotypic and phenotypic variations.

To overcome these limitations, we present a new model, based on Inductive Logic Programming (ILP) to estimate the impact of mutations on protein function in the context of human genetic diseases. ILP has been used recently in bioinformatics studies because of its ability to take into account background knowledge during the learning stage and to work directly with structured data. It has been applied successfully to various bioinformatics problems including breast cancer [17], protein structure prediction [18], gene function prediction [19], protein-protein interaction prediction [20], protein-ligand interaction prediction [21] and microarray data classification [22].

Taking advantage of the ILP approach, we have converted the mutation-oriented relational database SM2PH-db which integrates a large number of human mutations and phenotypes, into a knowledge base, providing a set of rules that can be easily interpreted by the biologist and reused for the prediction of the functional effects of a mutation.

II. MATERIALS AND METHODS

A. SM2PH-db

The data sets used in this study were taken from the relational database SM2PH-db ("from Structural Mutation to Pathology Phenotypes in Human-database", publicly accessible online at http://decrypthon.igbmc.fr/sm2ph). SM2PH-db [23] is designed to facilitate the investigation of these structural and functional impacts of missense mutations with regard to their phenotypic effects in the context of human genetic diseases. It provides access to a wide range of interconnected information related to proteins involved in human monogenic diseases. This information includes: (i) evolutionary background information, (ii) structural views of the protein 3D structure or predicted model, (iii) multi-level characterization of missense mutants, including details of the physico-chemical changes induced by the amino acid modification, information related to the conservation of the mutated residue and its position relative to functional features and in the 3D model.

At the time of writing (August 2010), SM2PH-db includes information related to 27,538 missense mutations, among which 18,434 are considered as disease-causing (deleterious) and 9,104 as non-pathogenic (neutral). The SM2PH-db content is created by a cascade of programs and automatically updated every two months on the Decrypthon Grid [24] to guarantee up-to-date information. SM2PH-db data generation and update have been introduced in our earlier article [23].

B. ILP

ILP [25] combines Machine Learning and Logic Programming. Given a formal encoding of the back-ground knowledge and a set of examples, an ILP system will derive hypotheses which explain all the positive examples and none, or almost none, of the negative examples. In this approach, logic is used as a language to induce hypotheses from the examples and background knowledge. Thus, the result of learning is represented as a logical formula in predicate logic, typically a Prolog program. Briefly, the basic form of the ILP problem is defined as follows.

Given:

- A background knowledge B which is the knowledge available before the learning.
- A finite set of examples E, $E = E^+ \cup E^-$ where E^+ is a nonempty set of positive examples, and E^- is a set of negative examples.

Find: hypotheses H (set of rules), such that:

- All or almost all positive examples $e \in E^+$ are covered by H.
- No or few negative examples $e \in E^-$ are covered by H.

In comparison with other machine learning approaches, ILP has several advantages. Firstly, in data mining, ILP is able to discover knowledge from a multi-relational database consisting of multiple tables. Thus, ILP is also called multi-relational data mining [26]. Secondly, using logic programming allows to encode more general forms of background knowledge such as recursions, functions or quantifiers [27]. Finally, the learned rules, which are based on first-order logic, are comprehensible by humans and computers and can be interpreted without the need for visualization. The following sections present the major steps involved in applying the ILP approach to the SM2PH-db.

C. Problem definition and example construction

The first task involves identifying the problem and translating it into the positive and negative examples. Here, we have limited our study to the task of discriminating mutations linked to known human diseases (deleterious) from those associated with the "polymorphism" term (neutral), in accordance with the nomenclature used in the UniProtKB database [28]. We used the 8,117 mutants with available high quality 3D models in SM2PH-db for the first data set (DS1). Of these, 6480 mutants are associated with human diseases and constitute the positive examples. The remaining 1,637 mutants, associated with the "polymorphism" term, are considered to be neutral and constitute the negative examples.

The predicate for the positive examples is "is_deleterious". For example, a positive example in Prolog syntax is "is_deleterious(m_Q13496_Asn180Lys)". This clause indicates that, in protein Q13496, the replacement of the Asparagine at position 180 by a Lysine is deleterious.

The positive examples are written in a file with a ".f" extension.

The negative examples use the same predicate (is_deleterious) but they are written in difference file (with a ".n" extension).

It should be noted that the class distribution of the tandem mutations in DS1 is imbalanced, i.e., deleterious mutations are overrepresented with respect to neutral mutations. We are thus faced with the so-called "class imbalance problem", which is one of the ten most challenging problems in data mining [29]. Therefore, in order to obtain both imbalanced and balanced training cases, we constructed two additional data sets (DS2 and DS3) by decreasing the number of deleterious mutations through random resampling. DS2 and DS3 contain all neutral mutations (1,637) and respectively, 3,280 (DS2) and 1,640 (DS3) randomly selected deleterious mutations. Detailed statistics of the three data sets are available on our website.

D. Background knowledge construction

The molecular consequences of missense mutations are related to the functional and structural contexts of the affected position, as well as to the physico-chemical characteristics of the substitution [11] [30]. All these types of information are represented in SM2PH-db for the stored missense mutations and they are used as background knowledge in this study. The descriptions of these types of information have been explained in our earlier article [23].

Very recently, [31] noted that "In summary, we believe that researchers should not only look at conservation in their judgment of functional significance of residues in the protein sequence. Correlation patterns between residues clearly provide additional evidence which should not be ignored." To study the effect of neighbouring amino acid residues on a missense mutation, we enhanced the database by including following additional features:

- Neighboring residues. The residues are considered to be neighbors of a position mutation if they occur in a sphere of radius 10A°. For example, Figure 1 shows the neighboring residues of p.Asn180Lys missense mutation in protein Q13496.

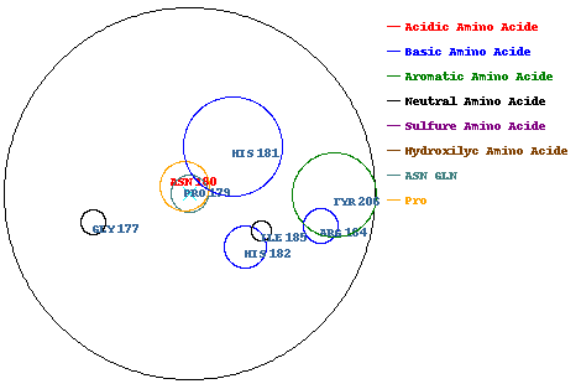

Figure 1. A sphere of radius 10A° with residue Asn180 in the center position (protein Q13496).

- Classification of amino acids. We used the amino acids classification system of Koolman [32] in which, based on the side chain chemical features, the amino acids are divided into aliphatic, acidic, basic, sulfur-containing, aromatic, neutral and imino.

The use of these features associated with physico-chemical, functional, 3D structural and evolutionary features of the missense mutations allows us to discovery hidden knowledge from different points of view of the missense mutations. The completed multi-relational data model in our analysis is shown in Figure 2.

In order to use ILP, we developed SQL scripts to translate the information stored in the SM2PH-db management system (PostgreSQL) into Prolog facts. Table 1 described all predicates derived from our mutation data model. Predicate has a template of the form: p(ModeType, ModeType, ...). Each mode type is either simple or structured. A simple mode type is one of: (1) +ModeType specifying the input, (2) -ModeType specifying the output and (3) #ModeType specifying the constant. A structured mode type is a function f(...), each argument of which is either a simple or structured mode type. In our learning problem, we use only the simple mode type.

E. Selection of ILP system and parameters

Among the different ILP systems available, we chose Aleph[1] with the SWI-Prolog compiler[2] to learn rules from our set of examples because of its popularity, frequent update and flexibility. In addition, Aleph allows customization of all the settings of the learning task. Aleph is also very attractive since it is coded in Prolog, and is thus relatively easy to modify. The Aleph algorithm is based on the classic ILP framework involving five main steps:

- Select an uncovered positive example.
- Find all the Prolog facts which explain this example.
- Combine the facts to generate a clause. Use an evaluation function to estimate the score of the clause on examples.

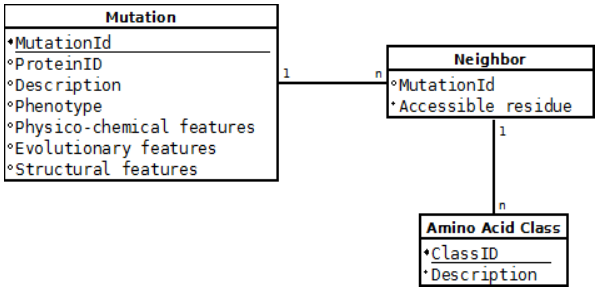

Figure 2. Mutation data model. Each missense mutation has physico-chemical features (size, charge, polarity, hydrophobicity, etc), evolutionary information and 3D structure. And it has one or more than one neighbouring residues, each of which can be classified into one family based on Koolman's method.

[1] http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/
[2] http://www.swi-prolog.org/

TABLE I. PREDICATES USED AS BACKGROUND KNOWLEDGE

Level of information	Predicate	Description
Physico-chemical changes induced by the substitution	modif_size(+mutationid, #modif_size_value)	Size, charge, polarity and hydrophobicity modification
	modif_charge (+mutationid, #modif_charge_value)	
	modif_hydrophobicity(+mutationid, #modif_hydrophobicity_value)	
	modif_polarity(+mutationid, #modif_polarity_value)	
	modif_score(+mutationid, -score)	Distance score between the 2 substituted residues. The higher the score, the less conservative the substitution
	g_p(+mutationid, #gp)	Modification of glycine or proline in the mutation.
Conservation	conservation_class(+mutationid, #conservationclass)	Conservation score of the substituted position in the sampled alignment.
	conservation_wt(+mutationid, -conservationwt)	Percentage of wild type residue in the column of the alignment.
	conservation_mut(+mutationid, -conservationmut)	Percentage of mutated residue in the column of the alignment.
	freq_at_pos(+mutationid, -freqatpos)	Number of known mutations at this position.
	cluster_5res_size(+mutationid, -cluster5ressize)	Grouping the mutations in the cluster if the next mutation is less than 5 residues in sequence.
Localization	secondary_struc(+mutationid, #secondary_struc)	In a secondary structure element? (helix, sheet, not_helix_not_sheet)
Structural data	gain_contact(+mutationid, -gaincontact)	Contacts between: - the wild type residue and its direct 3D neighbors, based on the wild type 3D model - the mutant residue and its direct 3D neighbors, based on the mutant 3D model are computed and compared
	lost_contact(+mutationid, -lostcontact)	
	identical_contact(+mutationid, -identicalcontact)	
	gain_n1_contact(+mutationid, -gainn1contact)	Contacts between: - residues in contact with the wild type residue and their direct 3D neighbors, based on the wild type 3D model - residues in contact with the mutant residue and their direct 3D neighbors, based on the mutant 3D model are computed and compared
	lost_n1_contact(+mutationid, -lostn1contact)	
	identical_n1_contact(+mutationid, -identicaln1contact)	
	wt_accessibility(+mutationid, -wtaccessibility)	Accessibility of the wild type residue.
	mut_accessibility(+mutationid, -mutaccessibility)	Accessibility of the mutant residue.
	cluster3d_10(+mutationid, -cluster3d10)	Grouping the mutations in the 3D cluster of 10A°
	stability (+mutationid, #stabilityvalue)	The change in protein relative stability upon mutation
	reliability_deltag(+mutationid, -reliabilitydeltag)	
Neighbors	neighbor(+mutation, -aa)	Accessible residues surrounding a position of interest (The diameter is 10A°)
	aa_class(+aa, #aaclass)	Classification of amino acids (after Koolman)

- Add the best clause, which has the best score, to the current hypothesis.
- Remove positive examples covered by the best clause.

These steps are iterated until all the positive examples are covered.

There are many parameters of Aleph we can vary in our experiments. First, the parameter *minpos*, indicating the minimum number of positive examples to be covered by an acceptable clause, was set to 5. Second, we needed a larger default search space and we thus set the parameter nodes to 50,000 (default 5,000). The parameter *nodes* defines the maximum number of nodes on the search space to be explored by the algorithm. Finally, the parameter *noise*, defined as the maximum number of negative examples to be covered by an acceptable clause, was set to 0.5% of negative examples. We used default settings for all other parameters.

All experiments were performed on a computer with a 2*AMD Opteron CPU 1.8 GB, 3 GB of RAM and the Ubuntu operating system.

III. RESULTS AND DISCUSSION

A. Novel hypotheses derived from SM2PH-db

All rules learned are available at: http://decrypthon.igbmc.fr/sm2ph/prediction.htm. Figure 3 shows some induced rules obtained from DS1 as presented on the web page. The first column provides a link to the examples covered by each rule. The second column contains the rule identification number. This information is used only to identify the rules in our experiments. The two next columns contain the most important information: the “if” and “then” clauses of the induced rules. The two rightmost columns indicate the number of positive examples (deleterious mutations) and negative examples (neutral mutations) covered by the if-then rule in each row. A filter is available to facilitate the exploration, validation and interpretation of the rules.

To illustrate how to transform ILP rules (expressed as Prolog form) into English sentences, we can consider the fourth rule in Figure 3 (mutation67_97).

	Id	If Statement	Then	Coverage	
				Positive	Negative
		Enter a key word: sub_family_conservatio Submit			
⊞	mutation67_123	conservation_class(A, sub_family_conservation) and freq_at_pos(A, B) and B>=2 and identical_n1_contact(A, C) and C>=20.	deleterious(A)	226 (3.49%)	0 (0.0%)
⊞	mutation67_140	modif_size(A, size_increase) and modif_charge(A, charge_opposite) and conservation_class(A, sub_family_conservation) and identical_n1_contact(A, B) and B>=37.	deleterious(A)	126 (1.94%)	5 (0.31%)
⊞	mutation67_85	conservation_class(A, sub_family_conservation) and conservation_wt(A, B) and B>=51.67 and gain_n1_contact(A, C) and C>=5.	deleterious(A)	111 (1.71%)	3 (0.18%)
⊞	mutation67_97	conservation_class(A, sub_family_conservation) and secondary_struc(A, no_helix_no_sheet) and gain_contact(A, B) and B>=1 and stability(A, decrease).	deleterious(A)	111 (1.71%)	7 (0.43%)

Figure 3. Screenshot of four induced rules obtained from DS1 with noise = 0.5%, minpos = 5, nodes = 50,000. Users can click on icon + to see the covered examples. Key word "sub_family_conservation" is used as a filter in this screenshot.

```
is_deleterious(A) :-
conservation_class(A,sub_family_conservation)
and secondary_struc(A, no_helix_no_sheet)
and gain_contact(A, B) and B>=1
and stability(A, decrease).
```

This rule states that a mutation A is deleterious if:

- The mutated residue belongs to the "sub-family conservation class" [23].
- The residue is found in neither an α-helix, nor a β-sheet.
- The number of contacts gained after point mutation is larger than or equal to 1.
- The stability of the protein after point mutation is decreased.

This rule correctly identified 111 deleterious mutations while misclassifying 7 neutral mutations as deleterious.

In order to facilitate the interpretation of the rules, the individual rules were grouped into rule subfamilies, using the hclust library in R[3] to perform a hierarchical cluster analysis. The hclust library requires similarity measures between individuals. In our case, the similarity between 2 rules was defined as the number of common deleterious mutations covered by these 2 rules (Jaccard similarity coefficient). We paid special attention to the dendrogram (tree diagram) generated from DS1 since DS1 contains all mutants with available high quality 3D models in SM2PH-db. Figure 4 shows a part of the DS1 dendrogram. The whole dendrogram can be found on our website. We performed multiple rule alignment on each subfamily (indicated by red rectangles in the dendrogram). Two interesting rule subfamilies encompassing more than 558 deleterious mutations (8.6% of DS1) were identified, i) the subfamily containing the 4 rules: 67_96, 67_140, 67_58 and 67_210 and ii) the subfamily containing the 5 rules: 67_123, 67_85, 67_97, 67_41 and 67_177. In both subfamilies, the "sub-family conservation" predicate was common and highly predictive for the deleterious state. For example, rule 67_140 (the second rule in Figure 3) covered 126 deleterious mutations. This rule indicates two important factors that characterize a deleterious mutation: first, the mutant residue is larger than the wild type one; and second, the mutated position is relatively conserved during evolution and is classified as a sub-family conservation class. This result confirms previous findings concerning the conservation of the mutated residue and the alteration of the chemical and physical properties of the amino acids in a missense variant having a crucial effect on protein function [33].

B. *Prediction service*

The prediction service is the secondary result of our work by taking advantage of ILP rules obtained at previous steps. Based on the rules learnt by the ILP algorithm described above, a function aimed at estimating nsSNP effects has been incorporated in the SM2PH-db server. It can be accessed via the Prediction link on SM2PH-db web interface, in the main menu. The input form (Figure 5A) allows users to specify the amino acid position and substitution of a given protein to be predicted including: the Uniprot accession number of the protein, the mutation position, the wild type residue and the mutant residue. The wild type residue must correspond to the current protein sequence. The generation of the web interface is programmed in Python.

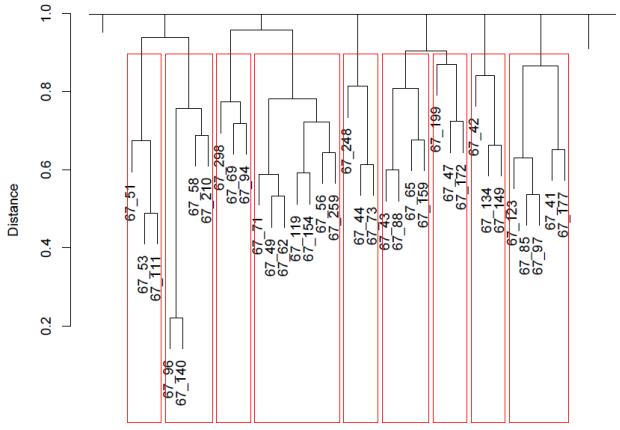

Figure 4. Part of the DS1 dendrogram

[3] http://cran.r-project.org/

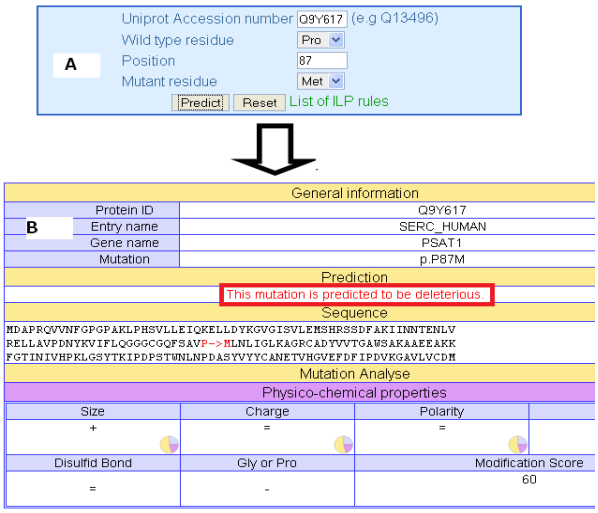

Figure 5. Screenshots of prediction pages. (A) Input form. The mutation predicted in this example is the P87M of Phosphoserine Aminotransferase protein. (B) The output page provides prediction results as well as multi-level (physico-chemical, functional, structural and evolutionary) characterizations of the mutation.

Given the input data, the SM2PH-db system automatically generates a multi-level characterization of the mutant. The process starts with the generation of mutant 3D models. Then, physico-chemical changes and structural modifications induced by the substitution, as well as functional and structural features related to the mutated position are calculated. If a 3D model is available, these values are transferred to the Convert2Prolog module to convert them into Prolog facts which then become the input for the prediction engine.

The prediction engine was built by using the induced rules. Thanks to Prolog, the deductive reasoning process immediately derives a conclusion (deleterious or neutral mutation). Figure 5B shows the output of the system. It displays not only the prediction result (deleterious or neutral mutation), but also descriptions of the modifications induced by the substitution and information related to the conservation of the mutated residue, its position relative to functional features and within the 3D model.

C. *Performance measurement and model evaluation*

In this study, we used sensitivity (Se) and specificity (Sp) to evaluate the performance of our learning system:

$$Se = \frac{TP}{TP + FN} \qquad Sp = \frac{TN}{TN + FP}$$

where True Positives (TP) and True Negatives (TN) are the number of correct predictions of the positive and negative examples, respectively; False Positives (FP) is the number of negative examples incorrectly predicted as positive; and False Negatives (FN) is the number of positive examples incorrectly predicted as negative.

As described earlier, our data sets include both balanced and imbalanced cases and we therefore needed a performance measure which is independent of the distribution of examples between classes. As a consequence, we calculated an additional measure, the geometric mean of accuracies [34] defined as:

$$Gmean = \sqrt{Se * Sp}\ .$$

In order to obtain a stringent evaluation, we conducted a 10-fold cross validation test with the three SM2PH-db datasets to compare our proposed method with SIFT and PolyPhen, the most common and widely used methods that can be acquired, implemented and run locally. Table 3 shows the average sensitivity, specificity and Gmean for 10-fold cross validation on three data sets for each method. SIFT achieves higher specificity than both PolyPhen and the ILP method developed in this manuscript. However, on average 38.43% mutations present in the SM2PH-db could not be predicted due to the lack of a significant number of alignments. The ILP method achieves good sensitivity and specificity on all data sets (imbalanced and balanced cases). The highest sensitivity (87.27%) is achieved for the DS1 data set, but it also has the lowest specificity (51.86%). This can be explained by the imbalanced distribution of deleterious and neutral mutations in this data set. By reducing the number of positive examples, the sensitivity decreases but the specificity increases significantly, as expected. Our method provides a stable performance in the balanced data sets (DS2 and DS3). Average sensitivity and specificity of ILP method are 79.35% and 63.3%, respectively, and average sensitivity and specificity of PolyPhen are 78.83% and 64.27%, respectively.

It should be mentioned that SIFT uses only sequence

TABLE II. COMPARISON OF OUR ILP APPROACH WITH OTHER AVAILABLE APPROACHES.

Data set	**ILP**			**SIFT**				**PolyPhen**		
	Se (%)	Sp (%)	Gmean (%)	Se (%)	Sp (%)	Gmean (%)	Unpred * (%)	Se (%)	Sp (%)	Gmean (%)
DS1	87.27	51.86	67.22	71.72	67.31**	69.48	38.77	78.05	64.27**	70.82
DS2	80.56	62.97	71.13	71.90	67.31	69.57	38.73	77.88	64.27	70.75
DS3	70.23	75.06	72.36	71.59	67.31	69.42	37.78	80.57	64.27	71.96
Average	79.35	63.3	70.24	71.74	67.31	69.49	38.43	78.83	64.27	71.18

* Unpred: The mutations which SIFT cannot predict due to the lack of a significant number of alignments.

** Neutral mutations are similar in all three data sets so that specificity does not change.

information to predict the effects of mutations, while our method and PolyPhen both use a combination of sequence and structure. In addition, even for the latter two methods, the approaches and parameters are very different. Thus, without benchmark data sets, it is hard to evaluate fairly the performances of the different approaches. For instance, the data sets used previously for benchmarking in the SNAP system [9] included the mutagenesis data for E.coli LacI repressor, bacteriophage T4 lysozyme, HIV-1 protease, and Melanocortin-4 receptor. In contrast, our system learns and extracts knowledge from SM2PH-db, a mutation database related to human monogenic disease. These problems of comparison between programs have been noted as general problems in the field of computational biology [35].

IV. Conclusion And Future Work

This study presents a novel application of ILP in the bioinformatics field, namely, the characterization and prediction of the effects of a mutation on protein function and the corresponding human phenotype. Our main goal was to discover knowledge from a mutation database. Using SM2PH-db, a database of mutation and phenotypic data involved in human genetic diseases, we constructed background knowledge and sets of examples (positive and negative examples). The resulting mutation knowledge base contains a set of rules and the important predictors for identifying deleterious mutations. The rules confirmed previous findings concerning the physico-chemical and evolutionary features that characterize a deleterious mutation, such as the importance of the conservation of the mutated residue or the detrimental effects of modification of the amino acid charge, volume and hydrophobicity. Importantly, as almost all mutations can be easily accessed via their associated set of rules, our mutation knowledge base provides useful information for understanding the relationships between the genotypic alteration and the phenotypic features in human diseases. The knowledge discovered by ILP should be helpful for the design of further research experiments. In addition, we have shown that the ILP approach can be effectively used for mutation effect prediction, as illustrated by the performances obtained which are similar to the common and widely used methods.

In the future, we plan to enhance the background knowledge by including more detailed genotypic and phenotypic information and additional data related to functional and physical interactions [36], [37], as well as to structural surface topology descriptions [38]. With richer and more relevant background knowledge, we intend not only to distinguish deleterious from neutral mutations, but also, to discover and extract pertinent relations between phenotype and genotype or gene/protein and genetic disease.

Hopefully, these perspectives will contribute to a more complete elucidation of the chain of events leading from a molecular defect to its pathology.

Acknowledgment

The authors are grateful to Raymond Ripp for his assistance during this work. We would also like to thank Julie Thompson and Nicolas Wicker for a critical reading of the manuscript. The IGBMC, Institut de Génétique et de Biologie Moléculaire et Cellulaire, is acknowledged for assistance.

This work was funded by the Association Française contre les Myopathies (AFM, KBM-14390), the Vietnam Ministry of Education and Training; the Institute National de la Santé et de la Recherche Médicale (INSERM); the Centre National de la Recherche Scientifique (CNRS) and the Université de Strasbourg.

References

[1] B. E. Stranger, M. S. Forrest, M. Dunning, C. E. Ingle, C. Beazley, N. Thorne, R. Redon, C. P. Bird, A. de Grassi, C. Lee, C. Tyler-Smith, N. Carter, S. W. Scherer, S. Tavare, P. Deloukas, M. E. Hurles, and E. T. Dermitzakis, "Relative impact of nucleotide and copy number variation on gene expression phenotypes," Science, vol. 315, pp. 848-53, Feb 9 2007.

[2] D. Chasman and R. M. Adams, "Predicting the functional consequences of non-synonymous single nucleotide polymorphisms: structure-based assessment of amino acid variation," J Mol Biol, vol. 307, pp. 683-706, Mar 23 2001.

[3] M. Y. Galperin and G. R. Cochrane, "Nucleic Acids Research annual Database Issue and the NAR online Molecular Biology Database Collection in 2009," Nucleic Acids Res, vol. 37, pp. D1-4, Jan 2009.

[4] S. V. Tavtigian, M. S. Greenblatt, F. Lesueur, and G. B. Byrnes, "In silico analysis of missense substitutions using sequence-alignment based methods," Hum Mutat, vol. 29, pp. 1327-36, Nov 2008.

[5] P. C. Ng and S. Henikoff, "SIFT: Predicting amino acid changes that affect protein function," Nucleic Acids Res, vol. 31, pp. 3812-4, Jul 1 2003.

[6] P. D. Thomas, M. J. Campbell, A. Kejariwal, H. Mi, B. Karlak, R. Daverman, K. Diemer, A. Muruganujan, and A. Narechania, "PANTHER: a library of protein families and subfamilies indexed by function," Genome Res, vol. 13, pp. 2129-41, Sep 2003.

[7] C. Ferrer-Costa, J. L. Gelpi, L. Zamakola, I. Parraga, X. de la Cruz, and M. Orozco, "PMUT: a web-based tool for the annotation of pathological mutations on proteins," Bioinformatics, vol. 21, pp. 3176-8, Jul 15 2005.

[8] E. Capriotti, R. Calabrese, and R. Casadio, "Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information," Bioinformatics, vol. 22, pp. 2729-34, Nov 15 2006.

[9] Y. Bromberg and B. Rost, "SNAP: predict effect of non-synonymous polymorphisms on function," Nucleic Acids Res, vol. 35, pp. 3823-35, 2007.

[10] S. Chun and J. C. Fay, "Identification of deleterious mutations within three human genomes," Genome Res, vol. 19, pp. 1553-61, Sep 2009.

[11] C. T. Saunders and D. Baker, "Evaluation of structural and evolutionary contributions to deleterious mutation prediction," J Mol Biol, vol. 322, pp. 891-901, Sep 27 2002.

[12] S. Sunyaev, V. Ramensky, and P. Bork, "Towards a structural basis of human non-synonymous single nucleotide polymorphisms," Trends Genet, vol. 16, pp. 198-200, May 2000.

[13] V. Ramensky, P. Bork, and S. Sunyaev, "Human non-synonymous SNPs: server and survey," Nucleic Acids Res, vol. 30, pp. 3894-900, Sep 1 2002.

[14] L. Bao, M. Zhou, and Y. Cui, "nsSNPAnalyzer: identifying disease-associated nonsynonymous single nucleotide polymorphisms," Nucleic Acids Res, vol. 33, pp. W480-2, Jul 1 2005.

[15] P. Yue, E. Melamud, and J. Moult, "SNPs3D: candidate gene and SNP selection for association studies," BMC Bioinformatics, vol. 7, p. 166, 2006.

[16] M. Masso and Vaisman, II, "Accurate prediction of stability changes in protein mutants by combining machine learning with structure based computational mutagenesis," Bioinformatics, vol. 24, pp. 2002-9, Sep 15 2008.

[17] R. W. Woods, L. Oliphant, K. Shinki, D. Page, J. Shavlik, and E. Burnside, "Validation of Results from Knowledge Discovery: Mass Density as a Predictor of Breast Cancer," J Digit Imaging, Sep 16 2009.

[18] A. P. Cootes, S. H. Muggleton, and M. J. Sternberg, "The automatic discovery of structural principles describing protein fold space," J Mol Biol, vol. 330, pp. 839-50, Jul 18 2003.

[19] R. D. King, "Applying inductive logic programming to predicting gene function," AI Mag., vol. 25, pp. 57-68, 2004.

[20] T. P. Nguyen and T. B. Ho, "An integrative domain-based approach to predicting protein-protein interactions," J Bioinform Comput Biol, vol. 6, pp. 1115-1132, Dec 2008.

[21] L. A. Kelley, P. J. Shrimpton, S. H. Muggleton, and M. J. Sternberg, "Discovering rules for protein-ligand specificity using support vector inductive logic programming," Protein Eng Des Sel, vol. 22, pp. 561-7, Sep 2009.

[22] E. Ryeng and B. K. Alsberg, "Microarray data classification using inductive logic programming and gene ontology background information," Journal of Chemometrics, vol. 24, pp. 231-240, 2010.

[23] A. Friedrich, N. Garnier, N. Gagnière, H. Nguyen, L. P. Albou, V. Biancalana, E. Bettler, G. Deléage, O. Lecompte, J. Muller, D. Moras, J. L. Mandel, T. Toursel, L. Moulinier, and O. Poch, "SM2PH-db: an interactive system for the integrated analysis of phenotypic consequences of missense mutations in proteins involved in human genetic diseases," Human Mutation, vol. 31, pp. 127-135, 2010.

[24] N. Bard, R. Bolze, E. Caron, F. Desprez, M. Heymann, A. Friedrich, L. Moulinier, N. H. Nguyen, O. Poch, and T. Toursel, "Decrypthon grid - grid resources dedicated to neuromuscular disorders," Stud Health Technol Inform, vol. 159, pp. 124-33, 2010.

[25] S. Muggleton, "Inductive logic programming," New Generation Computing, vol. 8, pp. 295-318, 1991.

[26] S. Džeroski, "Multi-relational data mining: an introduction," SIGKDD Explor. Newsl., vol. 5, pp. 1-16, 2003.

[27] A. Srinivasan, R. King, and S. Muggleton, "The role of background knowledge: using a problem from chemistry to examine the performance of an ILP program," Oxford University Computing Laboratory, 1999.

[28] Y. L. Yip, M. Famiglietti, A. Gos, P. D. Duek, F. P. David, A. Gateau, and A. Bairoch, "Annotating single amino acid polymorphisms in the UniProt/Swiss-Prot knowledgebase," Hum Mutat, vol. 29, pp. 361-6, Mar 2008.

[29] Q. Yang and X. Wu, "10 challenging problems in data mining research," International Journal of Information Technology & Decision Making, vol. 05, pp. 597-604, 2006.

[30] B. N. Terp, D. N. Cooper, I. T. Christensen, F. S. Jorgensen, P. Bross, N. Gregersen, and M. Krawczak, "Assessing the relative importance of the biophysical properties of amino acid substitutions associated with human genetic disease," Hum Mutat, vol. 20, pp. 98-109, Aug 2002.

[31] A. Kowarsch, A. Fuchs, D. Frishman, and P. Pagel, "Correlated Mutations: A Hallmark of Phenotypic Amino Acid Substitutions," Plos Computational Biology, vol. 6, pp. -, Sep 2010.

[32] J. Koolman and K. Boehm, Colour Atlas of Biochemistry: Thieme, 1996.

[33] J. Thusberg and M. Vihinen, "Pathogenic or not? And if so, then how? Studying the effects of missense mutations using bioinformatics methods," Hum Mutat, vol. 30, pp. 703-14, May 2009.

[34] M. Kubat, R. C. Holte, and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images," Mach. Learn., vol. 30, pp. 195-215, 1998.

[35] S. Veretnik, J. L. Fink, and P. E. Bourne, "Computational biology resources lack persistence and usability," PLoS Comput Biol, vol. 4, p. e1000136, 2008.

[36] L. J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic, P. Bork, and C. von Mering, "STRING 8 - a global view on proteins and their functional interactions in 630 organisms," Nucleic Acids Res, vol. 37, pp. D412-6, Jan 2009.

[37] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thorneycroft, Y. Zhang, R. Apweiler, and H. Hermjakob, "IntAct-open source resource for molecular interaction data," Nucleic Acids Res, vol. 35, pp. D561-5, Jan 2007.

[38] L. P. Albou, B. Schwarz, O. Poch, J. M. Wurtz, and D. Moras, "Defining and characterizing protein surface using alpha shapes," Proteins, vol. 76, pp. 1-12, Jul 2009.