Peptide Data Mining: From Virtual Design to Knowledge Extraction

Ignasi Belda[a], Ivan Traus[b], Susana Gordo[a], Teresa Tarragó[a], Sergio Madurga[a], Xavier Llorà[c], and Ernest Giralt[a]

[a]Institut de Recerca Biomédica, Parc Científic de Barcelona, E 08028 Barcelona, Spain.
[b] Conducive Corp. NY 10004, USA.
[c] Illinois Genetic Algorithms Laboratory, Department of General Engineering, University of Illinois at Urbana-Champaign. IL 61801, USA.

{ibelda,sgordo,ttarrago,smadurga,egiralt}@pcb.ub.edu, itraus@conducivecorp.com, xllora@illigal.ge.uiuc.edu

Categories and Subject Descriptors: I.2.4 Knowledge Representation Formalisms and Methods

General Terms: Algorithms.

Keywords: Molecular Design, Genetic-Based Machine Learnig, Data Mining.

1. INTRODUCTION AND METHODS

We recently developed a *de novo* peptide design tool, ENPDA (Evolutionary structure-based *de Novo* Peptide Design Algorithm), which uses evolutionary algorithms to evolve peptides to become good protein ligands [1]. As part of the fitness function of ENPDA, we use the docking program Autodock 3.0.5 [2]. Docking is a computational simulation procedure that estimates structural molecular interactions between a given ligand and a receptor. In addition, binding energies between the two molecules involved in the interaction can be also derived—these estimated binding energies are also known as docking energies.

In a previous study, we designed alternative ways to evaluate our peptides during the initial evolutionary steps. These alternative methods were machine-learning systems trained with thousands of docking experiments obtained upon ENPDA validation. The machine-learning systems can predict the docking energies of hexapeptides with great accuracy. We tested neural networks [3]; support vector machines (SVM's) [4]; the k-nearest neighbor algorithm [5] and a supervised learning system for rule induction [6]—we used the implementation MOLCS-R [7]. Albeit the most accurate machine-learning system trained in previous studies was an SVM, we decided to continue exploring MOLCS-R owing to the similar error rate of the two systems as well as the ease with which knowledge can be extracted from the rules inferred by the latter system. In MOLCS-R, knowledge can be extracted systematically and the mechanisms of which have been studied extensively [8].

MOLCS-R, is an evolutionary system whereby the individuals contained in the population are composed of a set of variable-length rules. Each of the rules is composed of a *condition* and a *target*. MOLCS-R incorporates mechanisms that force a good generalization. The evolutionary algorithm of MOLCS-R tries to minimize two criteria: the prediction error rate, and the number of rules inside the individuals of the population. Individuals with less rules are thus more general. To perform this task, MOLCS-R uses the NSGA-II algorithm [9], a multi-objective optimization technique. In a normal MOLCS-R training run, thousands of rules are built. If we want to analyze the rules obtained, we have to use some rule-quality metrics, typical in data mining methodologies. The techniques used for the present work are the support (or frequency), the confidence (or reliability) [10], and clustering. We have thus implemented a program that takes a very large set of rules inferred by MOLCS-R as input and displays only those rules exceeding user-determined support and confidence indexes. Regarding the set of rules displayed, we cluster the rules to obtain a prototype set. In this process, also known as *knowledge extraction*, new peptide design knowledge are be obtained from the prototypes.

Intuitively, support is the number of peptides from the initial database that match the conditions of a given rule. Rules with a low support index can only be applied to a restricted number of peptides. For such a reason, we discard this kind of rules, *i.e.*, we want to obtain general knowledge of good peptide ligand design. Therefore, given a rule we add for each peptide of the data-set 0 if none of the attributes of the rule match a given peptide, but if i attributes from the total number of attributes (N) match the rule, a value of $\frac{i}{N}$ is added to the support calculation. The second indicator, confidence, reflects the quality of a rule. The confidence is used to determine how many peptides from those that match the conditions of a given rule have a docking energy similar to the rule target value. To compute this value, we pay attention to those peptides matching the rule being evaluated. We then try to measure the distance between the rule's target and the docking energy of the peptides that match the rule. The confidence is an averaged summation of the confidences of all peptides matching the rule. The average is done by taking into account how many rule's attributes match each peptide involved in the calculation. All of the rules that do not meet user-defined minima for support and confidence are automatically removed. In addition to these two indicators, the user can also specify a target energy maximum of a rule. Hence, all rules with a target energy higher than that stated by the user are removed. The user thus obtains only those rules that describe peptides presenting

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

the best docking energy values for binding with the target protein.

Therefore, rules were clustered using the EM algorithm implemented in the data mining software Weka [11]. And the prototype, or centroid, of each cluster—*i.e.,* the properties shared by rules of the same cluster—was computed. The most important amino acids for the molecular recognition could then be determined from the prototype. In order to obtain the prototype, we developed a scoring system where, for each attribute of each rule of each cluster, we add 1 if there is `1`, we add 0 if there is a `#`, and we subtract 1 if there is a `0`. Therefore, the values of all of the attributes of all the prototypes are normalized between -1 and 1. The most important amino acids are those with a score higher than +0.5, whereas those with scores lower than -0.5 are not recommended. The amino acids scoring between -0.5 and +0.5 are considered neutral.

2. RESULTS

Since the goal of the proposed methodology is to obtain knowledge to design new peptides rather than analyze existing peptides, the data-set used in the validation contained a list of random hexapeptides and their docking energies to a defined protein: prolyl oligopeptidase (POP) [12], a protein related to neuropathologies such as as schizophrenia. 5,742 hexapeptide sequences were generated randomly. Only seven amino acids were use in the random peptides: arginine, triptophan, serine, glutamic acid, alanine, proline and isoleucine. Docking energies were estimated in the same manner than explained in [1].

MOLCS-R was run over the data-set following a 10-fold cross-validation strategy. 10-fold cross-validation consists of splitting the data-set into ten sets. Therefore, the algorithm is run 10 times, each time using a different set as the test set and using the other nine sets as the training set. The results on the test set are then averaged over the ten runs. Hence, the rules extracted from MOLCS-R are the rules of the entire population at the last generation of each run for all runs of the 10-fold cross validation—the following steps are performed in conjunction to all the rules obtained in each entire 10-fold cross validation experiment. In the evolutionary algorithm of MOLCS-R were used 250 individuals and 1,000 generations.

3,437 rules describing ligands of POP were obtained. Therefore, we took only those rules with a support higher than 0.75, a confidence higher than 0.9, and the value of the target higher than 0.9, thus obtaining 15 clusters. In the Table 1 is represented the knowledge extracted from the data set. The table should have as many rows as clusters were obtained. However no significant knowledge was found inside one of the clusters, hence only fourteen are shown. The columns denote the amino acid position. Each cell contains either recommended amino acids (up) or those that are not recommended (down). The amino acids that do not appear in any square are neutral.

Some peptides were further evaluated by combining extracted knowledge with either a random number generator or information on ligand design contributed by experts (results not shown). The docking energy values obtained for these new peptides agreed with expectations; the peptides derived with the random number generator in combination with the derived rules had lower docking energy values, on average, than those of the initial data-set. Whereas the peptides proposed by experts aided by the automatically inferred knowledge, had the lowest binding energy values.

1st	2nd	3rd	4th	5th	6th
S	E			S	
		R		A	R
W	A	E		E	A S P
A S	I W	S	E R		W E
A		W	P		
R	I		I	R	I
	E	E	S		P
R		I		I S	I
		S			
	I R				I
W	I R P	A	I S		P
	A E S	I	R	R	A W R
		W E R	A S	I	R S
I R	R		W	W R	E P
					P
R	I R	I R	I		I R
	W P	A		I S	I P
I S		W E R			R S
W				P	
I R	I	I R	I R	I R	I R
R	W		W R	W E	A R P
E	A I S P	A E S	I	A	I E
I E S		A		I	
	I W S P	W	R	A	A I E R
I S	R S		I		A
	A I	I W P	A W E	S P	P
	W P				E

Table 1: Rules within the prototypes of 14 clusters found inside the best rules inferred.

3. REFERENCES

[1] I. Belda, S. Madurga, X. Llorà, M. Martinell, T. Tarragó, M. G. Piqueras, E. Nicols, and E. Giralt. *Journal of Computer-Aided Molecular-Design*, 19:585–601, 2005.

[2] G. Morris, D. Goodsell, R. Halliday, R. Huey, R. Belew, and A. Olson. *Journal of Computational Chemistry*, 19:1639–1662, 1998.

[3] M. I. Jordan and C. M. Bishop. *ACM Computing Surveys*, 28(1):73–75, 1996.

[4] V. Vapnik. *Statistical learning theory*. Wiley, New York, NY, 1998.

[5] C. E. Rasmussen. *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression*. PhD thesis, Department of Computer Science, University of Toronto, 1996.

[6] K.A. De Jong, W.M. Spears, and D.F. Gordon. *Machine Learning*, 13:161–188, 1993.

[7] I. Traus. A Study of Pittsburgh Classifier Systems based on Multi-Objective Optimization Techniques. Master thesis. Universitat Ramon Llul. Barcelona, 2003.

[8] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, editors. *Interpretability Issues in Fuzzy Modeling*. Springer-Verlag, 2003.

[9] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. In *Proceedings of the PPSN VI*, pages 849–858. Springer, 2000.

[10] R. Rastogi and K. Shim. Mining optimized support rules for numeric attributes. *Information Systems*, 26(6):425–444, 2001.

[11] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000.

[12] B. Vogelstein, D. Lane, and A. J. Levine. *Nature*, 408:307–310, 2000.