Automated identification of protein-ligand interaction features using Inductive Logic Programming: a hexose binding case study

Jose C A Santos1*, Houssam Nassif2, David Page2, Stephen H Muggleton1 and Michael J E Sternberg3

Abstract

Background: There is a need for automated methods to learn general features of the interactions of a ligand class with its diverse set of protein receptors. An appropriate machine learning approach is Inductive Logic Programming (ILP), which automatically generates comprehensible rules in addition to prediction. The development of ILP systems which can learn rules of the complexity required for studies on protein structure remains a challenge. In this work we use a new ILP system, ProGolem, and demonstrate its performance on learning features of hexose-protein interactions.

Results: The rules induced by ProGolem detect interactions mediated by aromatics and by planar-polar residues, in addition to less common features such as the aromatic sandwich. The rules also reveal a previously unreported dependency for residues CYS and LEU. They also specify interactions involving aromatic and hydrogen bonding residues. This paper shows that Inductive Logic Programming implemented in ProGolem can derive rules giving structural features of protein/ligand interactions. Several of these rules are consistent with descriptions in the literature.

Conclusions: In addition to confirming literature results, ProGolem’s model has a 10-fold cross-validated predictive accuracy that is superior, at the 95% confidence level, to another ILP system previously used to study protein/hexose interactions and is comparable with state-of-the-art statistical learners.

Background

Elucidating unifying features of protein-ligand interactions in systems showing a diversity of interaction modes remains a challenging problem, often requiring extensive human intervention. In this work we present an automated general approach to identify these features using Inductive Logic Programming (ILP). We apply ILP to study the factors relevant to protein-hexose binding.

Hexoses are 6-carbon monosaccharides involved in numerous biochemical processes, including energy release and carbohydrate synthesis [1]. Several non-homologous protein families bind hexoses using a diverse set of protein-ligand interactions. Several research groups have used computational techniques to model and analyze hexose- and sugar-protein interactions, often employing extensive visualization and empirical methods [2-5].

*Correspondence: jcs06@doc.ic.ac.uk
1 Computational Bioinformatics Laboratory, Department of Computer Science, Imperial College London, London, SW7 2BZ, UK
Full list of author information is available at the end of the article

Some techniques use surface and binding site similarities to search for matching functional sites in other proteins [6,7]. Others apply machine learning algorithms to construct sugar-specific classifiers [8,9]. Such classifiers can be combined with programs that detect protein surface-pockets of a given size [10,11] to discriminate potential binding-sites.

Recently Nassif et al. [12] used the ILP system, Aleph [13], to study hexose binding. A powerful feature of ILP is that, in addition to prediction, it automatically learns rules which can be readily understood. It has been successfully applied to predict and model various medical [14,15] and biological datasets [16,17]. However, the complexity and size of the hypothesis space often presents computational challenges in search time which limit both the insight and the predictive power of the rules found.

Recognizing the limitations of Aleph and other current ILP systems, Muggleton et al. [18] developed ProGolem to facilitate the learning of long, complex rules. Such rules are common in molecular biology and we propose...
that a sophisticated ILP system such as ProGolem is a promising approach to automatically learn these rules from molecular data.

The present work extends previous hexose prediction work in several ways. First we supplement the background knowledge with both atomic and amino-acid information. Second, we bias the hypothesis space to reduce the search space and increase the likelihood of generating meaningful rules. Third, we employ the newly-developed ProGolem, which has been shown to learn better than Aleph in highly non-determinate domains such as this hexose-binding application. Finally, we explore several approaches to curb the limitations of the recall bound, the maximum number of times a predicate may succeed, in ILP systems.

The combined usage of an extended background knowledge, a better biased search, and the ILP system ProGolem allowed the discovery of more accurate and insightful rules explaining the stereochemistry of hexose binding. Automatically finding these stereochemical rules and providing their explanation is the main contribution of this paper. While some of the rules ProGolem found were already known from the literature, other rules, namely one that specifies a dependency over residues CYS and LEU, have never been reported but are plausible and require further investigation.

Predicting whether an actual protein binds an hexose is of secondary importance to us. Nevertheless, the predictive accuracy of our approach is competitive to statistical learners such as Support Vector Machines and superior to the logic-based approach Aleph applied to study hexose/protein interactions [12].

Table 1 The positive dataset, composed of 80 non-redundant protein-hexose binding sites

| Hexose | PDB ID | Ligand | PDB ID | Ligand |
|--------|--------|--------|--------|--------|
| Glucose | 1BDG   | GLC-501| 1ISY   | GLC-1471|
|         | 1EX1   | GLC-617| 1JOY   | GLC-1601|
|         | 1GJW   | GLC-701| 1JG9   | GLC-2000|
|         | 1GW   | GLC-1371| 1KW   | GLC-653|
|         | 1HSU   | GLC-998| 1KME   | GLC-501|
|         | 1TZ    | GLC-1381| 1MMU | GLC-1|
|         | 1HIZ   | GLC-1382| 1NF5 | GLC-125|
|         | 1HKC   | GLC-915| 1NSZ   | GLC-1400|
|         | 1HSJ   | GLC-671| 1PB   | GLC-405|
|         | 1ISY   | GLC-1461| 1SSM | AGC-1001|
|         | 1SZ   | GLC-1001| 1SSZ | BGC-2001|
|         | 1UZS   | GLC-1| 1UA4   | GLC-1457|
|         | 1V2B   | AGC-1203| 1WOQ | GLC-290|
|         | 12BD   | GLC-901| 2BP   | GLC-337|
|         | 2BW   | GLC-602| 2BVW  | GLC-603|
|         | 2F2E   | AGC-401|        |        |
| Galactose | 1AXZ  | GLA-401| 1MUQ   | GAL-301|
|         | 1DW   | GAL-1400| 1NS0   | GAL-1400|
|         | 1DR    | GAL-1104| 1NS2   | GAL-1400|
|         | 1DZQ   | GAL-502| 1NS8   | GAL-1400|
|         | 1EUU   | GAL-2| 1NSM   | GAL-1400|
|         | 1ISZ   | GAL-461| 1NSU   | GAL-1400|
|         | 1ISZ   | GAL-471| 1NSX   | GAL-1400|
|         | 1JZ7   | GAL-2001| 1KO    | GLB-901|
|         | 1KWK   | GAL-701| 1OQL   | GAL-265|
|         | 1L7K   | GAL-500| 1OQL   | GAL-267|
|         | 1L7T   | GAL-104| 1PHE   | GAL-1|
|         | 1R4   | GAL-1101| 1SSD  | GAL-704|
|         | 1S5E   | GAL-751| 1SSF  | GAL-104|
|         | 1S5O   | GAL-500| 1TLG   | GAL-1|
|         | 1US    | GAL-1501| 1UGW  | GAL-200|
|         | 1XC6   | GAL-9011| 1ZHJ  | GAL-1|
|         | 2GAL   | GAL-998|        |        |
| Mannose  | 1BQP   | MAN-402| 1KZB   | MAN-1501|
|         | 1KLF   | MAN-1500| 1KZC  | MAN-1001|
|         | 1KX1   | MAN-20| 1KE   | MAN-1001|
|         | 1KZA   | MAN-1001| 1OF3  | MAN-503|
|         | 1OUR   | MAN-301| 1QMO   | MAN-302|
|         | 1U4J   | MAN-1008| 1U4J  | MAN-1009|

The table lists the protein’s PDB ID and the hexose ligand considered.

Dataset

For ease of comparison, we use the same dataset and cross-validation folds described in Nassif et al. [12]. They obtained the positive examples from the Protein Data Bank (PDB) [19] by selecting proteins with coordinates for bound ligands from the most common hexoses: galactose, glucose and mannose [20]. Theoretical structures and files older than PDB format 2.1 were ignored. Glycosylated sites and redundant structures (at most 30% overall sequence identity using PISCES [21]) were also ignored. The positive subset consisting of 80 protein-hexose binding sites (33 galactose, 35 glucose and 12 mannose) is presented in Table 1.

Their negative dataset consists of 80 PDB examples: 22 binding sites that bind hexose-like ligands (e.g. hexose or fructose derivatives, 6-carbon molecules, and molecules similar in shape to hexoses), 27 other-ligand binding sites and 31 non-binding sites. The non-binding sites are surface pockets that look like binding sites but are not known to bind any ligand. The negative dataset is presented in Table 2 (non-hexose binding sites) and Table 3 (non-binding sites).
**Table 2** Protein binding-sites that bind non-hexose ligands

| PDB ID | Cavity center | Ligand | PDB ID | Cavity center | Ligand |
|--------|---------------|--------|--------|---------------|--------|
|        | Hexose-like ligands |        |        | Other ligands |        |
| I1AU   | 4320, 4323    | BEZ-1  | 1A17   | 6074, 6077    | IPH-1  |
| I1AWB  | 4175, 4178    | IPD-2  | 1DBN   | pyranose ring | GAL-102|
| IEOB   | 3532, 3536    | DHB-999| 1F9G   | 5792, 5785, 5786 | ASC-950|
| 1G0H   | 4045, 4048    | IPD-292| 1JU4   | 4356, 4359    | BEZ-1  |
| 1LBX   | 3941, 3944    | IPD-295| 1LBY   | 3944, 3939, 3941 | F6P-295|
| 1LIU   | 15441, 15436, 15438 | FBP-880 | 1MOR   | pyranose ring | G6P-609|
| 1NCW   | 3406, 3409    | BEZ-601| 1PSD   | pyranose ring | G1P-658|
| 1T1O   | 4366, 4361, 4363 | F6P-1001 | 1UOF   | pyranose ring | G6P-900|
| 1UKB   | 2144, 2147    | BEZ-1300| 1X9I   | pyranose ring | G6Q-600|
| 1Y9G   | 4124, 4116, 4117 | FRU-801 | 2B0C   | pyranose ring | G1P-496|
| 2SB3   | 3941, 3944    | IPH-401| 4PBG   | pyranose ring | BGP-469|
|        | Other ligands |        |        |                |        |
| 11AS   | 5132          | ASN-1  | 11G5   | 1672, 1675    | MES-3  |
| 1A0J   | 6985          | BEN-246| 1A42   | 2054, 2055    | BZO-555|
| 1ASO   | 4939, 4940    | FIP-270| 1AS3   | 2016, 2017    | IGP-300|
| 1AA1   | 4472, 4474    | 3PG-477| 1A1N   | 6074, 6079    | AAN-1  |
| 1A5S   | 3276, 3281    | PLA-415| 1AL8   | 2652          | FMN-360|
| 1B8A   | 7224          | ATP-500| 1BO5   | 7811          | GOL-601|
| 1BOB   | 2566          | ACO-400| 1DO9   | 7246          | PAL-1311|
| 1EIQY  | 3831          | ATP-380| 1IOL   | 2674, 2675    | EST-400|
| 1JTV   | 2136, 2137    | TES-500| 1KF6   | 16674, 16675  | OAA-702|
| 1RTK   | 3787, 3784    | GBS-300| 1TJ4   | 1947          | SUC-1  |
| 1TVO   | 2857          | FRZ-1001| 1UK6   | 2142          | PPI-1300|
| 1W8N   | 4573, 4585    | DAN-1649| 1ZYU   | 1284, 1286    | SKM-401|
| 2D7S   | 3787          | GLU-1008| 2GAM   | 11955         | NGA-502|
| 3PCB   | 3421, 3424    | 3HB-550|        |                |        |

The table lists the protein's PDB ID, the ligand considered and the specified cavity center. 22 ligands are similar to hexoses in shape and/or size. The cavity center is the centroid of the reported PDB atom numbers.

The data also specify the center for each of the resulting 160 examples. The binding-site center is computed as the hexose pyranose ring centroid for the positive examples, and as the ligand or empty pocket centroid for the negative ones. The hexose pyranose-ring atoms are located up to 2.9 Å away from the ring’s centroid. Since some atomic interactions can be important up to 7 Å [22], we consider the binding-site as all protein atoms present within a 10 Å radius sphere around the binding center. All other atoms are discarded.

**Inductive logic programming**

ILP is a machine learning approach that generates a hypothesis composed of a set of logical if-then rules that explains a given dataset [23]. ILP has three major advantages over other machine learning and data mining techniques. First, it allows an easy interaction between humans and computers by using background knowledge to guide the search. Second, it returns results in an easy-to-understand if-then format. Finally, ILP can easily operate on relational databases, as relational databases are naturally expressed as relations in first-order logic.

Most leading ILP approaches start by a saturation step, randomly selecting a positive example for which they construct the bottom clause: the most specific hypothesis that explains the example. This most specific clause is the rule formed by the conjunction of all features (called predicates or literals) pertaining to the chosen example. In the reduction step, ILP generalizes this rule (called clause or hypothesis) to include other positive examples using one of two basic induction methods, generalization or specialization.
Table 3 Non-bindingsites negative dataset, composed of random surface pockets that do not bind any ligand

| PDB ID | Cavity center | PDB ID | Cavity center | PDB ID | Cavity center |
|--------|---------------|--------|---------------|--------|---------------|
| 1A04   | 1424,2671     | 1A0I   | 1689,799      | 1A22   | 2927          |
| 1AA7   | 579           | 1AF7   | 631,1492      | 1AM2   | 1277          |
| 1A0R   | 154,1663      | 1ATG   | 1751          | 1C3G   | 630,888       |
| 1C3P   | 1089,1576     | 1DXJ   | 867,1498      | 1EVT   | 2149,2229     |
| 1F12   | 1493          | 1KLM   | 4373,4113     | 1KWP   | 1212          |
| 1QZ7   | 3592,2509     | 1YQZ   | 4458,4269     | 1YVB   | 1546,1814     |
| 1ZT9   | 1056,1188     | 2A1K   | 2758,3345     | 2AUP   | 2246          |
| 2BG9   | 14076,8076    | 2C9Q   | 777           | 2CL3   | 123,948       |
| 2DN2   | 749,1006      | 2F1K   | 316,642       | 2G50   | 26265,31672   |
| 2G69   | 248,378       | 2GRK   | 369,380       | 2GSE   | 337,10618     |
| 2GSH   | 6260          |        |               |        |               |

The table lists the protein’s PDB ID and the specified cavity center, computed as the centroid of the reported PDB atom numbers.

Aleph, using a specialization approach, starts with the most general hypothesis, “all sites are hexose-binding sites”, calling all examples positives. It then refines this hypothesis by repeatedly adding the literal from the bottom-clause that best improves the hypothesis score. The new rule will be more specific, covering only a subset of the examples previously covered.

ProGolem, in contrast, uses a generalization search. Starting with the bottom clause, it successively drops a minimal set of literals to allow coverage of one additional positive example. By dropping this set of literals the clause becomes more general, and will cover a superset of the examples previously covered.

Both ProGolem and Aleph stop hypothesis refinement when the hypothesis score stops improving. A rule scores well if it covers many positive and few negative examples. If the rule passes a certain performance threshold, it is added to the theory, and all the positive examples it covers are removed. The cycle of saturation and reduction continues on the remaining examples. When all positive examples are covered or no new rules can be found, the ILP system outputs its theory, the set of the best rules found so far. Then, in the testing stage, a new instance is classified as positive (i.e. hexose-binding) if it is covered by any of the theory rules, otherwise it is labeled as negative.

The newly developed ProGolem is more than a specific-to-general version of Aleph. Two additional features set it apart. Aleph adopts a local theory construction method, incrementally adding a new rule to its theory after each reduction cycle. This method depends on the ordering of the positive examples, and it is possible that the best rules are not generated. This situation may occur if these better rules would be generated by examples that were removed by previous sub-optimal rules. By contrast, ProGolem implements a global theory construction approach, which ensures that the theory is only constructed after all rules have been generated. ProGolem repeatedly adds to the theory the rule that best improves the global theory score. The global-theory-construction feature of ProGolem is especially useful in this application.

The second feature makes ProGolem specifically suitable for our application. In ILP an example can have multiple instances from the same predicate. For example, a binding site has multiple atoms. If a predicate has more than one possible solution, it is called non-determinate. Hence the site has_atom predicate is non-determinate. Our hexose dataset is highly non-determinate.

When evaluating a clause, Aleph will proceed literal by literal from left to right. This is the standard Selective Linear Definite (SLD) resolution [24], which is the only option in most ILP systems. However, ProGolem has to evaluate longer clauses than Aleph due to its specific-to-general hypothesis search. SLD-resolution is too slow to compute the coverage of such long clauses. To cope with this problem ProGolem supports the usage of different resolution engines, including the smallest variable domain resolution, which enumerates the possible values a variable in a literal may take and, during clause evaluation, chooses at each moment the variable with the smallest domain [25]. This clause evaluation engine is better suited to our problem than SLD-resolution, drastically reducing the runtime per evaluated clause.

Background knowledge

The background knowledge is the set of features, facts and rules known a priori. This is given to the ILP system as a basis for learning and constructing the classification rules. The piece of background knowledge central to our task is the binding site representation. Table ?? is an excerpt of the background knowledge for protein 1BDG. The center_coords predicate specifies the binding-site center coordinates, which is the pyranose ring centroid of the
bound glucose in this structure. The has\textunderscore aminoacid predicate specifies each amino acid present within the protein binding site, listing its unique identifier and name. The has\textunderscore atom predicate details the residue atoms, specifying the PDB atom name and its coordinates. By extracting the coordinates of the center and the various atoms, we compute their respective distances. We set a tolerance of 0.5 Å on distances between atoms, a sensible error margin in a hexose binding site [26].

In addition to these facts, ILP allows for a higher level of expressiveness within its background knowledge: human coded rules. Using the facts of Table 4, and the Euclidean distances between atoms that are derived from this data, we can now define the predicates atom\textunderscore to\textunderscore center\textunderscore dist and atom\textunderscore to\textunderscore atom\textunderscore dist. These predicates respectively compute the distance between an atom and the binding site center, and between two atoms. We also define a diff\textunderscore aminoacid predicate which allows expressing that two amino acids are different. This may be relevant when there are multiple amino acids of the same type and each amino acid needs to be individually identified.

Hypothesis space

We experiment with multiple hypothesis spaces. Similarly to Nassif et al. [12], we first exclude residue information and limit the background to the atoms and their 3D coordinates. In this atom\textunderscore only representation, the binding site is a sphere of radius 10 Å containing atoms in space, for which distances can be computed (see Table 5).

The distances between atoms are computed by having a dist literal in the ILP background knowledge, allowing ILP systems to express the 3D conformation of the binding site. However, the number of possible distances grows quadratically with the number of atoms considered, resulting in an exponential growth of the bottom clause. Starting its generalization search from the bottom clause, ProGolem learning time is highly sensitive to its length. To keep learning tractable, both ProGolem and to a lesser extent Aleph, require a bound on the maximum number of solutions a given predicate may return, called the recall (not to be confused with the statistical measure of the same name, also called sensitivity). In practice this recall bound limits the hypothesis search space by forcing that only the first recall solutions of a literal be considered in the bottom clause.

By relying on the ordering of the atoms and residues in the background knowledge, having a bound on the recall of a predicate is an approach subject to data idiosyncrasies as the ordering of the background knowledge predicates is arbitrary and only the first recall are considered. In this work we explore two alternative approaches of organizing the background knowledge to curb the limitations of having a recall bound. The first approach, randomized recall, considers all solutions first, out of which it randomly picks a number equal to recall; rather than the first recall atoms in the binding-site data representation. This is achieved by either altering the internal recall routine, as we did, or equivalently, by randomizing the order of the atoms and residues in the background knowledge.

The second approach is domain\textunderscore dependent. Using Random Forests to measure feature importance [27,28], Nassif et al. [9] show that atoms closest to the binding center have higher discriminative power. Closest atoms are more likely to determine whether or not the binding site binds hexose, as compared to more distant atoms. The domain\textunderscore dependent approach orders the atoms and residues in the background knowledge by their distance from the binding site center. For instance, in this approach, the distance literal will attempt to match the recall atoms closest to the binding center.

Another contribution of this work is the re\textunderscore modeling of the problem representation and a better bias to the hypothesis space. We propose two major improvements to the atomic representation. First is the inclusion of residues using the has\textunderscore aminoacid predicate. The second is imposing that atoms cannot appear dangling in a hypothesis. A residue has to be introduced into a rule first,
before atom \( \text{to} \) atom dist and atom \( \text{to} \) center dist predicates compute its atomic distances. We thus only compute distances between atoms of residues already in a rule. In this amino acid representation, the binding site sphere is composed of amino acids, which in turn contain atoms (Table 4).

By first dealing with residues instead of atoms, the binding site sphere now contains a smaller number of elements. In addition, in the amino acid representation we can express the distance between two atoms using only one literal, atom \( \text{to} \) atom dist. A rule can contain up to recall residues, and for each atom of a given residue we measure its distance to recall atoms of each one of the included residues. In contrast, in the atom-only representation we need three literals to express a distance, two has atom and one dist. A rule can contain only recall atoms, and each atom can only detect recall other atoms in the feature space. Thus, for the same recall bound, the amino acid representation considers both more features and generates more informative clauses than the atom-only representation.

Table 6 is an example of an amino acid representation hypothesis, in raw Prolog format as induced by ProGolem. A Prolog clause follows a Head:Body structure. The head is verified (i.e. is true) if the body (a conjunction of literals) holds true. The uppercase letters in the clause, in this case A, B, C and D, are logical variables and represent a certain entity. Lowercase strings, string within quotes (e.g. leu, cys, ‘N’, ‘OD2’, and ‘C’) and numbers are constants representing themselves. The variable A in this clause represents a protein, variables B, C and D represent amino acids.

**Methods**

All materials (i.e. dataset, ILP systems and scripts) to reproduce these experiments are available at http://www.doc.ic.ac.uk/~jcs06/Hexose.

**ILP settings**

We apply two ILP systems, Aleph and ProGolem, with both atom-only and amino acid representations, and use YAP 6.0.6 as the Prolog compiler [29]. To ensure a fair comparison, we use the same settings for both ILP algorithms whenever possible. We set the recall bound to 7, and the maximum number of negatives a hypothesis may cover to 5. We evaluate clauses according to the usual scoring function in ILP, compression: positive examples covered - negative examples covered - clause length. For instance, for the same clause length, say 6 literals, a clause covering 37 positives and 4 negatives has a better score \((37 − 4 − 6 = 27)\) than one covering 30 positives and no negatives \((30 − 0 − 6 = 24)\).

We use ProGolem with its global theory construction and smallest variable domain resolution. In Aleph, we set the number of nodes to be explored when searching for an acceptable clause to 5000. The clause length in Aleph, i.e. the maximum number of literals allowed in a hypothesis, was set to match the clause length that ProGolem generates (5 for atom-only, 6 for amino acid). If the same clause length was used in both representations, the predictive accuracies of Aleph would be lower. In ProGolem, the user does not need to specify the maximum clause length of a rule, as the hypothesis search is from specific to general.

**Homology and cross-validation**

Our dataset consists of 160 binding sites, belonging to 152 unique proteins (8 of the hexose-binding proteins have two distinct binding sites). These 152 proteins belong to a total of 122 CATH [30] superfamilies. In order to guarantee that rules are not being learned from homologous proteins, the correct procedure is that each cross-validation fold does not contain proteins whose superfamilies (i.e. homologues) also occur in other folds.

Unfortunately, with this particular dataset, it is impossible to construct cross-validation folds that verify this non-sharing superfamily constraint. This is because the binding site may span multiple chains, each belonging to a different superfamily. Moreover, a single chain may be subdivided into domains, each belonging to different CATH superfamilies. Thus, if binding site A belongs to superfamilies sf1 and sf2, B to sf2 and sf3, and C to sf3 and sf4, the binding sites A, B and C must be in the same cross-validation fold. With our dataset, this constraint would result in a single cross-validation fold containing 48 binding sites (34 positives, 14 negatives) out of 160, creating a significant imbalance between folds.

Given this impossibility, and in order for our results to be comparable with previous work, we performed a 10-fold cross-validation using the same folds as [12]. Since the number of hexose binding proteins is limited, the dataset proteins share a low sequence identity (\(\leq 30\%\)), and the main goal of this paper is to provide insight into the hexose-binding discriminating process rather than the predictive accuracy of the classifiers, we consider our methodology acceptable. Each fold consists of 8 positive...
and 8 negative examples. Following standard machine learning technique, where for small datasets a large number of folds is needed, we train ProGolem on 9 folds (144 examples) and test on the remaining fold (16 examples), repeating this process 10 times such that each fold is used for testing once [31]. We use the testing fold results to compute the relevant statistics. When comparing two approaches or algorithms on the 10 folds, we consistently use a two-tailed paired t-test at the 95% confidence level.

**Results and discussion**

It is important to note that the main aim of this work is to discover rules describing the stereo-chemistry of protein-hexose binding. Although there is empirical evidence suggesting that many hexose dockings are not accompanied by substantial protein conformational changes [26], we do not aim to predict the binding sites of new hexose-binding proteins, as we would not know in advance the coordinates of the hexose ligand. Nevertheless, we use 10-folds cross-validated predictive accuracies as a measure to demonstrate the robustness of the rules.

**Altering ProGolem recall**

Simply relying on the given order of the background knowledge introduces placement bias. Both randomizing recall selection, and incorporating domain knowledge by ordering the atoms according to their distance to the binding site center, significantly improves accuracy when compared to the given PDB ordering (p-values of 0.026 and 0.021, respectively). This showcases the importance of domain knowledge, whereas clever manipulations based on prior knowledge will have better results compared to default settings. We also argue that randomizing recall selection should be used as default since it avoids data idiosyncrasies.

As explained previously, an important parameter when running ILP systems, is the recall bound, which imposes a bound on the maximum number of solutions a given Prolog predicate may return. Since, for performance reasons, the recall setting has to be limited to a relatively low value, we started by performing experiments to determine how to best order the atom and residues to get the most out of a limited recall.

We considered three schemes. The first considers the atoms of the protein according to the order of their occurrence in the PDB file, which follows the order of the primary sequence. The second scheme randomizes the order of the atoms in the background knowledge. The third scheme, domain-dependent, orders the atoms by their distance to the binding-site center.

Using the atom-only representation, the three approaches respectively yield an accuracy of 59.4%, 68.8% and 74.4% (Table 7). Sorting the binding-site atoms according to their distance from the binding center outperforms randomizing them, which in turn outperforms using their PDB sequence order. We therefore adopt the domain-dependent approach to organize the atoms and residues in the background knowledge in all our subsequent runs, involving both ILP algorithms (ProGolem and Aleph) and both data representations (atom-only and amino acid).

**ProGolem performance**

Table 8 shows the 10-fold cross-validation predictive accuracies for Aleph and ProGolem with the atom-only and amino acid representations. We also compare our results to a state-of-the-art approach, which uses Random Forests [27] for feature selection, and Support Vector Machines (SVM) [32] for classification. Internal validation selects the best Random Forests and SVM parameters for each training fold before predicting the testing fold. Note that SVM is a statistical classifier requiring a constant-length feature vector as input. This necessitates a different problem representation than the one used with ILP. Essentially we divide the binding site in concentric spherical layers, and for each we compute atomic and residue properties. We also add various atomic features namely hydrophobicity, charge and hydrogen-bonding. Refer to Nassif et al. [9] for method and representation details.

From Table 8 we notice that ProGolem performs better using the enhanced amino acid representation rather than atom-only (p-value = 0.029). However, the amino acid representation yields no statistically significant improvement in Aleph (p-value = 0.39). A possible explanation as to why ProGolem takes advantage of the amino acid representation more than Aleph is the myopia effect [33].

| Fold | Primary sequence | Randomized | Domain-dependent |
|------|-----------------|------------|------------------|
| 1    | 43.8%           | 56.3%      | 87.5%            |
| 2    | 62.5%           | 93.6%      | 78.5%            |
| 3    | 81.3%           | 87.5%      | 87.5%            |
| 4    | 56.3%           | 50.0%      | 43.8%            |
| 5    | 68.8%           | 68.8%      | 81.3%            |
| 6    | 37.5%           | 56.3%      | 81.3%            |
| 7    | 56.3%           | 62.5%      | 75.0%            |
| 8    | 68.8%           | 68.8%      | 81.3%            |
| 9    | 62.5%           | 81.3%      | 62.5%            |
| 10   | 56.3%           | 62.5%      | 68.8%            |
| Mean | 59.4%           | 68.8%      | 74.8%            |
| Std Dev | 1.6%         | 1.4%       | 1.3%             |
The myopia effect occurs because general-to-specific ILP systems, like Aleph, indirectly assume literals are conditionally independent given the target class. They refine the working hypothesis by adding one literal at a time, the one that maximizes a fitness function. If literals have a strong conditional dependency, any selected literal will roughly have the same score. Thus multiple literals need to be added before Aleph can determine which set is optimal. If the literals are highly non-determinate, as is our case, a significant portion of the search resources is wasted searching very similar hypotheses, which results in a poorer chance of finding good theories.

ProGolem outperforms Aleph for both representations (Table 8). The differences in their predictive accuracies are statistically significant for both atom-only (p-value = 0.043) and amino acid (p-value = 0.004) representations, the latter being significant even at the 99% confidence level. This discrepancy is in part explained by ProGolem’s global theory construction, which only constructs the final theory after all hypotheses have been generated rather than incrementally, on a per-example basis, as Aleph does.

Finally, we compare ILP to SVM. Despite amino acid ProGolem having a higher average accuracy and a lower standard deviation than SVM, the difference is not statistically significant (p-value = 0.52). More surprisingly, SVM does not significantly outperform amino acid Aleph (p-value = 0.057). SVM significantly outperforms both Aleph (p-value = 0.005) and ProGolem (p-value = 0.025) in the atom-only representation.

### Table 8 10-folds cross-validation predictive accuracies for Aleph, ProGolem and SVM

| Fold | Aleph 1 | ProG. 1 | Aleph 2 | ProG. 2 | SVM |
|------|---------|---------|---------|---------|-----|
| 1    | 50.0%   | 75.0%   | 56.3%   | 75.0%   | 81.3% |
| 2    | 68.8%   | 81.3%   | 68.8%   | 81.3%   | 87.5% |
| 3    | 62.5%   | 68.8%   | 68.8%   | 93.8%   | 87.5% |
| 4    | 50.0%   | 56.3%   | 68.8%   | 75.0%   | 75.0% |
| 5    | 75.0%   | 81.3%   | 56.3%   | 81.3%   | 75.0% |
| 6    | 68.8%   | 87.5%   | 81.3%   | 87.5%   | 87.5% |
| 7    | 75.0%   | 81.3%   | 75.0%   | 81.3%   | 93.8% |
| 8    | 93.8%   | 81.3%   | 75.0%   | 93.8%   | 87.5% |
| 9    | 68.8%   | 75.0%   | 75.0%   | 81.3%   | 75.0% |
| 10   | 56.3%   | 56.3%   | 87.5%   | 81.3%   | 62.5% |

Mean 66.9% 74.4% 71.3% 83.2% 81.3%

The f besides Aleph and ProGolem stands for the atom-only representation and the f for the amino acid representation. SVM uses a different representation (see text).

### Insight from rules

In this section we present the English translation and the biological explanation for some of the most relevant rules found by ProGolem using the amino acid representation. Although these particular rules were generated from the whole data set, the rules found on each cross-validation fold are similar; the themes associated to hexose-binding are consistently identified by ProGolem on each fold. The actual positive and negative examples covered by each rule are presented in Table 9.

According to ProGolem, a site is hexose-binding if:

1. It contains two different ASN residues and an ASP residue whose CG atom is 5.4 ± 0.5 Å away from the binding center.
   [Positives covered = 37, Negatives covered = 4]

2. It contains an ASN whose N and C atoms are 2.4 ± 0.5 Å apart, and a GLU whose CB and CG atoms are 8.0 ± 0.5 Å and 6.9 ± 0.5 Å away from the binding center, respectively.
   [Positives covered = 24, Negatives covered = 0]

3. It contains an ASN residue whose N atom is 8.2 ± 0.5 Å away from the binding center, and an ASN residue whose N and ND2 atoms are 4.1 ± 0.5 Å apart and whose N and O atoms are 3.6 ± 0.5 Å apart.
   [Positives covered = 30, Negatives covered = 0]

4. It contains a TRP residue whose CB atom is 7.1 ± 0.5 Å away from the binding center, and whose N and CD1 atoms are 4.0 ± 0.5 Å apart.
   [Positives covered = 14, Negatives covered = 0]

5. It contains a TYR residue whose CB and OH atoms are 5.6 ± 0.5 Å apart, a HIS residue whose ND1 atom is 8.9 ± 0.5 Å away from the binding center, and a TYR residue whose O atom is 9.8 ± 0.5 Å away from the binding center.
   [Positives covered = 6, Negatives covered = 0]

6. It contains CYS and LEU residues, and an ASP residue whose N and OD2 atoms are 4.6 ± 0.5 Å apart, and whose C atom is 7.6 ± 0.5 Å away from the binding center.
   [Positives covered = 18, Negatives covered = 0]

Note that the binding center is the hexose pyranose ring centroid, and that up to the first 4 Å of the distance between a binding-site atom and the binding center are occupied by the docked hexose. In addition, hydrogen atoms are generally not included in PDB files. Thus the presence of an atom may be a surrogate for its hydrogen involved in a hydrogen-bond.

The first rule requires the presence of an ASP and two ASN. Previously, Rao et al. [34] highlighted the importance of both residues in hexose binding. Studying the lectin protein super-family, they report that the 3D positions of binding-site ASP and ASN residues are conserved.
Table 9 Positive and negative examples covered by each reported ProGolem rule

| Rule | Positive examples                                                                 | Negative examples          |
|------|-----------------------------------------------------------------------------------|---------------------------|
| 1    | 37: 1BDG, 1BQP, 1DZQ, 1HKC, 1HSJ\text{?}, 1ISY, 1ISZ, 1ISZ\text{?}, 1JOY, 1JG9, 1JZ7, 1KLF, 1MMU, 1MUQ, 1NSU, 1NSX, 1NSZ, 1OKO, 1OP3, 1OQL, 1OQL\text{?}, 1OQ4, 1P1E, 1P33, 1SSM, 1SZ2, 1SZ2\text{?}, 1TLG, 1U2S, 1U4J, 1U4J\text{?}, 1U44, 1UAS, 1WQQ, 2BQP, 2BVW, 2BVW\text{?} | 4: 1AWB, 1W8N, 2B0C, 2B32 |
| 2    | 24: 1DJR, 1EUU, 1HIZ, 1HSJ, 1HSJ\text{?}, 1KWK, 1KX1, 1KZA, 1KZB, 1KZC, 1KZC\text{?}, 1L7K, 1MUQ, 1NS8, 1NSM, 1NSU, 1NSX, 1NSZ, 1PWB, 1SSD, 1SSS, 1SO0, 1TLG, 1X6 | 0 |
| 3    | 30: 1DIW, 1DJR, 1EUU, 1HIZ, 1ISZ, 1KX1, 1KZA, 1KZB, 1KZC, 1KZC\text{?}, 1L7K, 1L7I, 1NS0, 1NS8, 1NSM, 1NSU, 1NSX, 1NSZ, 1OKO, 1OQL\text{?}, 1OQ4, 1P1B, 1POMQ, 1SSD, 1SSS, 1SSS, 1X6F, 1XO0, 1X2S, 2GAL | 0 |
| 4    | 7: 1HSJ, 1HSJ\text{?}, 1KME, 1RYD, 1SSM, 1TLG, 1UGW | 0 |
| 5    | 6: 1HIZ\text{?}, 1KWK, 1POMQ, 1U4J, 1U44\text{?}, 1X6 | 0 |
| 6    | 18: 1ISY, 1ISZ\text{?}, 1ISZ\text{?}, 1NS8, 1NSM, 1PF0, 1OKO, 1OQL, 1P1E, 1R47, 1SO0, 1SZ2, 1V1Z\text{?}, 1TLG, 1U4J, 1U44, 1UAS, 2BVW, 2BVW\text{?} | 0 |

This holds despite lectins binding various types of hexoses and exhibiting different sugar-binding specificities.

That same rule requires the ASP CG atom to be 5.4 Å away from the centroid of the hexose pyranose ring. The pyranose radius itself being 3 Å, the ASP actually interfaces the docked hexose. Binding-site interface residues are key for hexose recognition and binding [9], especially planar polar residues that establish a network of hydrogen bonds with the various hydroxyl groups of the docked hexose [35]. Quiocho and Vyas [36] report that the most common planar polar amino acids involved in hexose binding are mainly ASP and ASN, followed by GLU. ProGolem detects the role of GLU in the second rule.

The second rule also implies a specific conformation with a triangular distance between GLU’s CB and CG atoms, and the binding center. Sujatha and Balaji [26] report that spatial disposition of protein-galactose interacting atoms is not conserved per se, but is conserved with respect to the docking position of the ligand. Similarly, ProGolem often specifies the distance of an atom with respect to the centroid of the hexose.

An additional advantage of inducing rules using ILP is the straightforward reverse-engineering to find the particular proteins, residues and atoms covered by a given rule. This is achieved by executing the ILP rule in a Prolog interpreter. As an example, Figure 1 visualizes the second rule with protein 1HIZ, a xylanase. The hexose ligand is depicted with its backbone in light pink. The two amino acids involved in the rule, a glutamic acid and an asparagin, have a white backbone. The relevant distances are shown.

In addition to specifying the distance from the binding center, ProGolem can detect specific amino acid stereo-chemical dispositions. The third rule determines a particular ASN conformation, specifying the distances between backbone N and O atoms, and the side chain ND2 atom.

![Figure 1 Visualization of the second ProGolem rule instantiated with protein 1HIZ (covered by the rule). The hexose is the glucose molecule to the left, with a pink backbone. To the right, with a white backbone, are the amino acids ASN and GLU, in closer contact with the hexose. The dotted black lines highlight the distances between the atoms in the amino acids and the center of the hexose.](image-url)
The various spatial dispositions of the different rules need further investigation to compare them with known 3D hexose-binding-site conformations.

The aromatic residues (TRP most frequently, TYR, PHE, and to a lesser extent HIS) provide a stacking platform for the hexose to dock on [36]. The hexose pyranose ring forms a planar apolar hydrophobic side that stacks, through hydrophobic and van der Waals interactions, over the aromatic residues planar apolar hydrophobic side chain ring [37]. Similarly, the ProGolem fourth rule requires the presence of TRP in a particular stereochemical conformation.

The fifth rule requires the presence of one or two TYR, and a HIS. This rule is thus describing a conformational representation of two or three aromatic residues around the binding-site center. It is interesting that this low-coverage rule may indeed be capturing the infrequent sandwich interaction, whereby two or more aromatic residues engage both faces of a hexose pyranose ring [38].

The last rule specifies CYS and LEU residues. Both have negative interface propensity measures (see below) and do not form hydrogen bonds with hexoses [39]. To quantify the disposition amino acids to be in contact with the docked sugar, Taroni et al. [39] devised an interface propensity measure, defined as the logarithm of the ratio between a surface residue frequency at the sugar binding site, and the average frequency of any surface residue at the binding site. They compute and report the sugar-interface propensity measure for the 20 common amino acids. A residue with a negative propensity measure does not favor the sugar binding-site region since it is present there less frequently than average.

This rule covers 18 positive examples and no negative examples, and clearly specifies the presence of CYS and LEU as a discriminatory factor for hexose-binding site recognition. This dependency over LEU and CYS is not previously identified in literature and merits further attention.

Conclusion

Inductive Logic Programming (ILP) is a leading technique to mine accurate and comprehensible rules. The newly developed ILP system ProGolem is well suited for complex non-determinate problems as often found in biological datasets. In our hexose-binding application, its predictive accuracy is significantly better than previous approaches, while showing a clear insight of the underlying discrimination process.

ProGolem was able to infer different aspects of the established biochemical information about hexose-binding, namely the presence of a docking aromatic residue, the importance of interface atoms, and the hydrogen-bonding activity of planar-polar residues (ASN, ASP, GLU). ProGolem also detected the less common aromatic sandwich interaction.

In addition, ProGolem reveals an important previously unreported finding: a dependency over residues CYS and LEU. It also specifies stereo configurations involving aromatic and hydrogen bonding residues. The newly reported relationship and 3D conformations require further investigation.

Finally, we recommend randomizing the recall selection by default and have implemented this option in ProGolem. We also note that incorporating domain-dependent knowledge in parameter settings is likely to lead to the best results.

Competing interests

Professors Muggleton and Sternberg are Directors and Shareholders in Equinox Pharma Ltd, a company which commercializes Inductive Logic Programming in the areas of bioinformatics and chemoinformatics. The remainder authors have no competing interests.

Authors’ contributions

JCAS developed the ProGolem ILP system under the supervision of SHM. ProGolem’s theoretical foundations were laid out by SHM. HM created the hexose dataset and performed the Support Vector Machines evaluation. HN and Page did the initial experiments with Aleph. JCAS undertook the current experimental evaluation between ProGolem and Aleph. JCAS, HN and MJES wrote the paper. All authors read and approved the final manuscript.

Acknowledgements

This work was partially supported by the US National Institute of Health grant R01CA127379-01. Jose Santos thanks the Wellcome Trust for funding his Ph.D. grant 0807/12/06/2. Stephen Muggleton thanks the Royal Academy of Engineering and Microsoft for funding his Research Chair. We thank Dr. Suhail Islam for his help with Figure 1 and Professor Kurt Dickramer for comments on the rules. We are also indebted to two anonymous reviewers for helpful suggestions.

Author details

1Computational Bioinformatics Laboratory, Department of Computer Science, Imperial College London, London, SW7 2BZ, UK. 2Department of Computer Sciences, Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, Madison, WI-53706, USA. 3Centre for Bioinformatics, Department of Life Sciences, Imperial College London, London, SW7 2AZ, UK.

Received: 8 September 2011 Accepted: 15 June 2012

Published: 11 July 2012

References

1. Solomon E, Berg L, Martin DW: Biology. 8th edition. Belmont, CA: Brooks Cole, 2007.
2. Shionyu-Mitsuyama C, Shirai T, Ishida H, Yamane T: An empirical approach for structure-based prediction of carbohydrate-binding sites on proteins. Protein Eng 2004, 17(7):467–478.
3. Sujatha MS, Saindhav YU, Balaji PV: Energetics of galactose- and glucose-aromatic amino acid interactions: implications for binding in galactose-specific proteins. Protein Sci 2004, 13(9):2502–2514.
4. Chakrabarti R, Kilbanov AM, Friesner RA: Computational prediction of native protein ligand-binding and enzyme active site sequences. Proc Nat Acad Sci USA 2005, 102(8):10153–10158.
5. Doxey AC, Cheng Z, Moffatt BA, McCorkley BJ: Structural motif screening reveals a novel, conserved carbohydrate-binding surface in the pathogenesis-related protein PR-5d. BMC Struct Biol 2010, 10:23.
6. Gold ND, Jackson RM: Fold independent structural comparisons of protein-ligand binding sites for exploring functional relationships. J Mol Biol 2006, 353(5):1112–1124.
7. Cipriano G, Wesenberg G, Grim T, Jnr GN, Gleicher M: GRAPE: GRaphical Abstracted Protein Explorer. Nucleic Acids Res 2010, 38:W959–W961.
8. Malik A, Ahmad S: Sequence and structural features of carbohydrate binding in proteins and assessment of predictability using a neural network. BMC Struct Biol 2007, 7:1.

9. Nassif H, Al-Ali H, Khan S, Keirouz W: Prediction of protein-glucose binding sites using support vector machines. Proteins 2009, 77(2):119–122.

10. Kawahata T: Detection of multi-scale pockets on protein surfaces using mathematical morphology. Proteins 2010, 78(5):1193–1211.

11. Wong GY, Leung FH: Predicting Protein-Ligand Binding Site with Support Vector Machine. In Proceedings of the IEEE Congress on Evolutionary Computation. Barcelona, Spain; 2010:1–5.

12. Nassif H, Al-Ali H, Khan S, Keirouz W, Page D: An inductive logic programming approach to validate hexose biochemical knowledge. In Proceedings of the 19th International Conference on ILP. Leuven, Belgium; 2009:149–165.

13. Snirinasa A: The Aleph Manual. 4th edition. 2007. [http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/aleph.html].

14. Snirinasa A, King RD, Muggleton SH: Carcinogenesis predictions using ILP. In Proceedings of the 7th International Workshop on Inductive Logic Programming. Prague, Czech Republic; 1997:273–287.

15. Dutra I, Nassif H, Page D, Shavlik J, Strigel R, Wu Y, EM, E: Burnside E: Integrating machine learning and physician knowledge to improve the accuracy of breast biopsy. In American Medical Informatics Association (AMIA’11) Symposium Proceedings. Washington, DC; 2011:349–355.

16. Finn P, Muggleton S, Page D, Snirinasa A: Pharmacophore discovery using the inductive logic programming system PROGOL. Machine Learning 1998, 30(2–3):241–270.

17. Szaboova A, Kuzelka O, Zelezny F, Tolar J: Prediction of DNA-binding proteins from structural features. In Proceedings of the 4th International Conference on Machine Learning in Systems Biology. Edinburgh; 2010:273–287.

18. Muggleton S, Santos J, Tamaddoni-Nezhad A: ProGolem: a system based on relative minimal generalisation. In Proceedings of the 19th International Conference on ILP. Leuven, Belgium: Springer; 2009:131–148.

19. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: The protein data bank. Nucleic Acids Res 2000, 28:235–242.

20. Fox MA, Whitesell JK: Organic Chemistry. 3rd edition. Boston, MA: Jones & Bartlett Publishers; 2004.

21. Wang G, Dunbrack RL: PISCES: a protein sequence culling server. Bioinformatics 2003, 19(12):1589–1591.

22. Yamashita MM, Wesson L, Eisenmur G, Eisenberg D: Where metal ions bind in proteins. Proceedings of the National Academy of Sciences USA 1990, 87(15):5648–5652.

23. Mitchell TM: Machine Learning. Singapore: McGraw-Hill International Editions; 1997.

24. Kowalski RA, Kuehner D: Linear resolution with selection function. Am Intell 1971, 2(3):227–260.

25. Santos J, Muggleton S: Subsumer: A Prolog theta-subsumption engine. In Technical communications of the International Conference on Logic Programming. Edinburgh, Scotland; 2000:172–181.

26. Sujatha MS, Balaji PV: Identification of common structural features of binding sites in galactose-specific proteins. Proteins 2004, 55:44–65.

27. Breiman L: Random forests. Machine Learning 2001, 45:5–32.

28. Diaz-Uriarte R, de Andres S: Gene selection and classification of microarray data using random forest. BMC Bioinf 2006, 7:3.

29. Santos Costa V: The life of a logic programming system. In Proceedings of the 24th International Conference on Logic Programming. Edited by de la Banda MG, Pontelli E, Udine I. Springer-Verlag; 2008:1–6.

30. Oregano C, Michie A, Jones S, Jones D, Swindells M: CATH—a hierarchical classification of protein domain structures. Structure 1997, 5:1093–1108.

31. Jain AK, Duin RPW, Mao J: Statistical pattern recognition: a review. IEEE Trans Pattern Analysis Machine Intelligence 2000, 22:4–37.

32. Vapnik VN: Statistical Learning Theory. New York: John Wiley & Sons; 1998.

33. Kononenko I, Mravec E, Robnik-Sikonja M: Overcoming the myopia of inductive learning algorithms with RELIEFF. Appl Intell 1997, 7:389–55.

34. Rao VSR, Larr K, Qasba PK: Architecture of the sugar binding sites in carbohydrate binding proteins—a computer modeling study. Int J Biol Macromol 1998, 23(4–5):295–307.

35. Zhang Y, Swaminathan GJ, Deshpande A, Boix E, Natesh R, Xie Z, Acharya KR, Brew K: Roles of individual enzyme-substrate interactions by alpha-1,3-galactosyltransferase in catalysis and specificity. Biochemistry 2003, 42(16):13512–13521.

36. Quochio FA, Vyas NK: Atomic interactions between proteins/enzymes and carbohydrates. In Bioorganic Chemistry: Carbohydrates. Edited by Hecht SM. New York: Oxford University Press; 1999:441–457.

37. Screen J, Stanca-Kaposta EC, Gambill DP, Liu B, Macleod NA, Smedley LC, Davis BG, Simons JP: IR-spectral signatures of aromatic-sugar complexes: probing carbohydrate–protein interactions. Angew Chem Int Ed 2007, 46:3644–3648.

38. Boraston AB, Bolam DN, Gilbert HJ, Davies GJ: Carbohydrate-binding modules: fine-tuning polysaccharide recognition. Biochem J 2004, 382:709–716.

39. Taroni C, Jones S, Thornton JM: Analysis and prediction of carbohydrate binding sites. Proteins 2000, 39(2):89–98.

Cite this article as: Santos et al.: Automated identification of protein-ligand interaction features using Inductive Logic Programming: a hexose binding case study. BMC Bioinformatics 2012 13:162.

http://www.biomedcentral.com/1471-2105/13/162

Submit your next manuscript to BioMed Central and take full advantage of:

• Convenient online submission
• Thorough peer review
• No space constraints or color figure charges
• Immediate publication on acceptance
• Inclusion in PubMed, CAS, Scopus and Google Scholar
• Research which is freely available for redistribution

Submit your manuscript at www.biomedcentral.com/submit