

ILP-assisted de novo drug design

Rama Kaalia¹ · Ashwin Srinivasan² · Amit Kumar¹ ·
Indira Ghosh¹

Received: 10 December 2014 / Accepted: 23 February 2016 / Published online: 16 March 2016
© The Author(s) 2016

Abstract De novo design of drugs uses the three-dimensional structure of a target protein (often called the receptor) to design molecules (or ligands) that could bind to the receptor and hence inhibit its functioning. Thus, unlike a ligand-based approach, this form of drug design does not require prior knowledge of inhibitors. In this paper, the three-dimensional structure of a receptor is used indirectly, in the form of molecular interaction fields of the receptor and small molecules (or probes). In addition, we also use domain-specific constraints encoding basic geometric and pharmacological requirements imposed by the target. Interaction energies of one or more targets with a set of probes are used to identify three-dimensional constraints that occur in many—preferably all—targets. In a graph-theoretic sense, the constraints are (small, fixed-size) cliques in graphs with labelled vertices representing probe-specific points of high interaction energy, and edges between a pair of vertices are labelled by the three-dimensional distance between the corresponding points of interaction. Our interest is in the discovery of frequent cliques that satisfy domain-specific constraints. In the paper, the discovery of such patterns is done using an Inductive Logic Programming (ILP) engine. The case for the use of ILP stems primarily from the explicit ways of incorporating domain-constraints, but any other technique capable of discovering frequent cliques from data can be used with some additional effort. The frequent cliques discovered are used to hypothesize pharmacophore-like structures on potential ligands. We test the utility of this approach by conducting a case study on the discovery of anti-malarials. Specifically, we test the approach on proteins belonging to the class of aspartic proteases. We are particularly interested in plasmepsin II, which is an enzyme in the haemoglobin degradation pathway of *Plasmodium falciparum*. We assess the pharmacophore-like constraints using: (a) a database of known inhibitors and non-inhibitors of aspartic proteases; and (b) a database of decoys that are physico-chemically similar to the aspartic proteases. Our results suggest that the

Editors: Jesse Davis and Jan Ramon.

✉ Ashwin Srinivasan
Ashwin@goa.bits-pilani.ac.in

¹ School of Computational and Integrative Sciences, Jawaharlal Nehru University, New Delhi, India

² Department of Computer Science and Information Systems, BITS, Pilani, Goa, India

approach could be used to obtain pharmacophores with good precision and recall for aspartic proteases.

Keywords De novo drug design · Inductive Logic Programming · Pharmacophores · Frequent cliques

1 Introduction

Ideally, drug development commences with the knowledge that disrupting the functioning of a specific protein or enzyme will yield some positive medicinal effect. It is known, for example, that the enzyme plasmepsin II is a protein involved in the degradation of haemoglobin in the parasite *Plasmodium falciparum*. This parasite is the major cause of malaria, one of the world’s worst diseases. (There are over 200 million cases of malaria worldwide (WHO 2014)). Inhibiting plasmepsin II disrupts the degradation of haemoglobin and kills the parasite (Silva et al. 1996) and consequently, the disease. Plasmepsin II thus constitutes the primary target for anti-malarial drug development. Other than Plasmepsin II, there are multiple proteins (Plasmepsin I, IV and V) belonging to the same protein family (aspartic proteases) which are crucial for survival of the parasite in human red blood cells (Berry 1997; Wyatt and Berry 2002; Sedwick 2014). These proteins constitute a group of potential targets for drug designing. In such cases, a type of drug is needed that can act against these multiple targets.

The chemistry of drug-development is a cyclic process (see Fig. 1) that attempts to find potential drugs by identifying small molecules that act as “keys” that fit into protein “locks” (in the example just above, plasmepsin II constitutes the lock, and a molecule that inhibits its activity is a key). Depending on whether the structure of the lock or key—or at least of their important parts—is known, there are two different routes to drug-design. If important

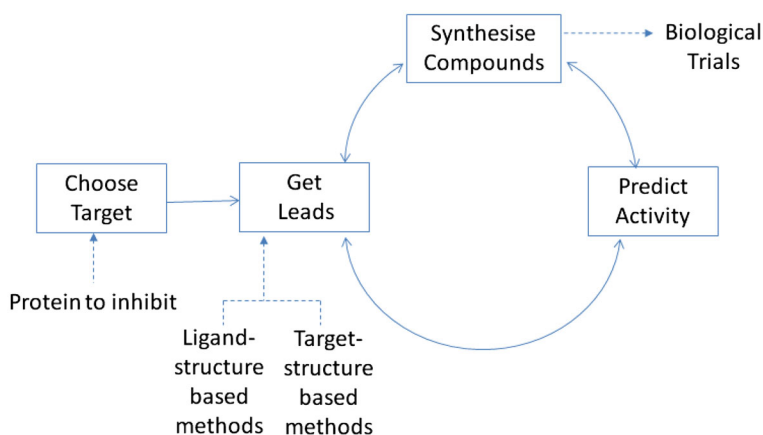


Fig. 1 A simplified view of the chemistry involved in the drug-design process. Ligands are small molecules that can bind to the active site (this is usually found by empirical testing), and leads are small molecules that interact—usually sub-optimally—with the target. Leads usually have properties that could be optimised to yield potential drugs. Previously, ILP has been used to assist ligand-based identification of leads by constructing pharmacophores (3-d constraints on the location of some functional groups on the ligand). The focus of this paper is to use 3-d information about the target to identify new drug-like molecules

structural features needed for a key (a potential drug) are known, then these features can be used to search databases of known compounds.¹ On the other hand, if we know the structure of the target (the lock), then we are able to design new molecules directly to complement the target's structure. This second approach is commonly referred to as *de novo* design of drugs.

Of the two approaches, *de novo* design is the more direct, and attractive. According to the sc-PDB database, out of about 100,000 structures listed in Protein Data Bank (PDB), only about 3500 unique proteins have the ability to bind with high affinity to a known set of drug compounds (Desaphy et al. 2014). Moreover, infectious diseases like malaria and tuberculosis are known continually to develop resistance against traditional compounds (Wongsrichanalai et al. 2010), leaving progressively little scope for using similar drug compounds as leads. Any *de novo* approach that does not require prior knowledge of inhibitors and directly uses the three-dimensional structure of a target protein (the receptor) to design molecules that could bind to the receptor is therefore of significant interest. This paper is concerned with the use of Inductive Logic Programming to assist in the development of such an approach.²

We will rely on deducing properties of the active site using molecular interaction fields (MIFs) of targets with small organic molecules called *probes*. An MIF, ideally, is a continuous surface, that at each point in three-dimensional space gives the interaction energy of a target with the probe. In the method proposed here, given a target, (discretised approximations of) the MIF of the target with a number of probes are obtained. Our goal will be to identify favourable interaction patterns from the MIFs of one or more probes that occur repeatedly in a series of related targets. We pose this as a problem of identifying frequent cliques in a complete graph in which vertices are MIF interaction points and edges are labelled with distances between points. The occurrence of a clique in the MIFs of more than one target requires all edge-distances to be checked with some tolerance. In this paper, we will identify such frequently occurring “elastic” cliques using an ILP engine. The cliques identified specify constraints on high-energy interaction points for the target and each probe. This allows us to do two things. First, we are able to conjecture residue-level information about the target site. Second, we are able to use the constraints to specify a pharmacophore-like constraint that can be used either to assist in the construction of new molecules, or to search existing databases. In fact, it will generally be insufficient to look simply at high-energy interactions of probes with targets, and we also incorporate some well-understood domain specific constraints that follow from geometric and pharmacological requirements of the target site. These can be naturally incorporated as inputs to the ILP engine. We first summarise the principal contributions of the paper. To the chemistry of drug-design, this paper's contributions are as follows:

- This is the first method that can perform multi-target, multi-probe drug design using molecular interaction fields, at the same time incorporating target-specific and probe-specific constraints in a flexible manner. The multi-target approach results in a consensus active-site approach that cannot be achieved with a single-target approach.
- To the best of our knowledge, this is the first report of pharmacophores to aid antimalarial research, obtained using a series of related aspartic proteases. The screens show that we are able to identify known inhibitors with good precision and recall.

¹ The features of importance are obtained by identifying compounds that are known to interact with the target protein. Three-dimensional constraints common to compounds found to interact strongly with the protein form the basis of these structural features. Elsewhere (Finn et al. 1998; Hirst et al. 1994a, b; King et al. 1992; Marchand-Geneste et al. 2002), it has been shown that Inductive Logic Programming can assist ligand-based drug design by identifying these 3-d constraints, or pharmacophores, from the structure of small molecules with known activity.

² A third alternative, of using receptor-and-ligand pairs with known structure is not considered here, although there is a natural application of ILP techniques here as well.

- There are many ligand-based approaches in field of drug design. These require prior information about inhibitors but there are very few receptor-based methods that use the structure of proteins to generate pharmacophores. Even those that do often require the protein structure along with a bound ligand (co-crystallised, or docked using a computer simulation). Here we construct pharmacophores without the need for inhibitors. This is of special relevance with the emergence of drug-resistant parasites for diseases like malaria. In such cases, the use of historical databases of inhibitors is of limited value.

Based on the long experience of one of the authors in the pharmaceutical industry³, and a recent publication in the chemical literature (Kaaliala et al. 2015) we have good reasons to believe that these are significant contributions to the area of drug-design. To the field of ILP, this paper's contributions are as follows:

- The paper constitutes an application of ILP to a real-world problem of significant scientific and industrial interest. It also moves forward significantly the use of ILP in drug-design from a ligand-based approach to a receptor-based approach. This is the first demonstration of the use of ILP for mapping the active-site in receptor-based drug design.
- We demonstrate on a real problem, the principal feature of an ILP-based approach that differentiates it from many other forms of machine-learning, namely: the incorporation of diverse aspects of domain-expertise as background (prior) knowledge. Our results also show performance and computational benefits of incorporating domain constraints when conducting a resource-bounded search for solutions.
- On a more technical note, the application also demonstrates the use of an explicitly defined “refinement operator” in the search to take into incrementally extend frequent cliques to larger ones. To the best of our knowledge, no ILP applications have employed refinement operators in this manner.

The work presented here is a substantial extension of the work in Kaaliala et al. (2015). The principal differences are these: (a) Here, the focus is on the use of ILP for the problem. To this end, we have included algorithmic descriptions of the procedures used to find frequent cliques; (b) In Kaaliala et al. (2015), only maximally-specific pharmacophores are considered. Here we provide a method of extending this to more general pharmacophores using the notion of quasi-cliques; (c) The assessment in Kaaliala et al. (2015) are largely of a qualitative nature, focusing on the chemistry implied by the results. This paper contains quantitative assessments of results in a manner familiar to researchers in machine learning; (d) We have included comparisons against a Baseline variant that allows us to assess the role of domain-knowledge; and (e) We have created an additional set of decoys and present results on this set, that allows us to compare against a random choice approach.

The rest of the paper is organised as follows. In Sect. 2 we describe the use of molecular interaction fields (MIFs) to characterise targets. High-energy points in the MIFs from multiple probes are treated as vertices in a graph. Section 2.1 describes the identification of frequent cliques in such graphs. We would like the cliques identified to be meaningful for drug design. The use of an ILP engine provided with domain-specific constraints to find meaningful cliques in MIF-graphs is described in Sect. 3. Deriving pharmacophores using the cliques found is described in Sect. 3.4. The application of the approach to the discovery of antimalarials is in Sect. 4. Related work is in Sect. 5, and Sect. 6 concludes the paper. The Appendices contain some relevant graph-theoretic terms and concepts; and a brief description of the chemical rationale underlying the domain-constraints used in the paper.

³ I.G. was at Astra-Zeneca.

2 Drug design using cliques from MIF surfaces

A molecular interaction field, or MIF, denotes the potential energy variations arising from the interaction of a target molecule with an atom or small group of atoms called a *probe*. The steps involved in computing MIFs are described in detail in Cruciani (2006). The main ideas are these: (1) the target is usually taken to be a rigid structure, or in one of small number of alternative shapes or conformations; (2) The atomic coordinates (in three-dimensions) of the target are known, either experimentally, or theoretically from energy-minimisation simulations; (3) MIF values are to be computed at points on a rectilinear grid structure, within which the target is placed; (4) The electrostatic potential is computed at each point on the probe. For each point, this is the amount of work required to bring the probe from infinity to that point. If the probe is a unit positive charge, then the reader will recognise that the MIF is just the electrostatic potential of the target at that point. The energy calculations consist of four different components, accounting for Van der Waals interactions, electrostatic interactions, directional interactions due to hydrogen bonds, and interactions arising from displacement of water molecules. The details are unimportant here, but broadly speaking, the net result is a set of grid positions where the target interacts favourably with the probe, and those where it does not.

A 1-dimensional example is shown in Fig. 2a–d [from Kaalia et al. (2015): in reality, the MIF is a three-dimensional surface]. In Fig. 2a, we focus on points of high-energy, denoting favourable interactions with two probes. These are shown here as the grid points A–G, obtained using probe-specific thresholds on the energy of interaction. The energy levels corresponding to the threshold are shown as broken lines: grid points with interaction

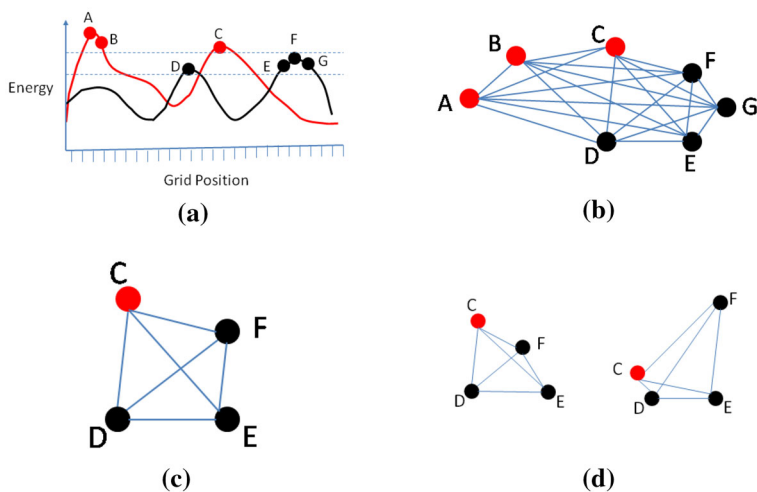


Fig. 2 A 1-d example of using MIFs to characterise energy interactions between targets and probes. **a** The energy variation at pre-specified x-values (“grid points”) between a (hypothetical) target and two probes (the energy of interaction from one probe is shown here in red and the other in black). The red circles are the high-energy points for the red-probe, and the black circles are the high-energy points for the black probe. What constitutes a “high-energy” point is determined by probe-specific thresholds on energy-levels, shown by dotted horizontal lines; **b** the high-energy points A–G as vertices in a complete graph. The edge-labels are distances between the points; **c** a subgraph of the high-energy interaction graph. Since the graph in **b** is complete, it is a clique, and any subgraph of it is necessarily also a clique; **d** topologically equivalent cliques, arising from stretching or contracting distances (Color figure online)

energy above the threshold (A–C for one of the probes, and D,E,F,G for the other) are the ones of interest. Figure 2b is a graph-based representation of how grid-points with high-interaction energy are spatially related to each other. In the graph, vertices denote points of high interaction energy and edges are labelled by the (Euclidean) distance between the points. This graph is a complete graph, or a clique (that is, every pair of vertices has an edge between them). Given a set of related targets, we are interested in finding the largest such clique that is common to all (or most) of them. Thus, what we want is to identify maximal cliques that are most frequent, or “frequent maximal-cliques” for short. Some graph-theoretic terms and concepts of relevance to this paper are reproduced in “Appendix 1”.

Although related, no two targets would be identical, which gives rise to the following difficulties:

1. A maximal clique from any one target (such as the one in Fig. 2b) would not necessarily occur in the MIF of a different target. We will therefore often be looking for common subgraphs across target MIFs (like Fig. 2c). However not all common subgraphs may be useful for drug-design.
2. Positions of equivalent vertices may not be at exactly the same positions in the graphs obtained from targets. Checking for the occurrence of subgraphs would therefore have to employ notions of topological equivalence that allows for shrinking and stretching. However, to prevent completely dissimilar cliques from being judged as being equivalent (like the ones in Fig. 2d), We therefore want to place some restrictions on the amount of such deformations.

We account for both these issues by stating our problem as follows. Given a series of targets, we want to find the largest, most frequently occurring cliques, that satisfy certain topological and domain-specific constraints. The former is to account for variations in targets and the latter is to allow meaningful drug design.

2.1 Discovering frequent MIF cliques

The clique detection problem normally refers to the task of determining whether a graph has a subset of vertices such that every pair of vertices in the subset are connected by an edge (that is, the subset represents a complete subgraph of the original graph). The problem extends naturally to multiple graphs, with the task of determining whether the same clique occurs in all (or many) of the graphs. A straightforward approach for MIF-graphs, based on the Apriori algorithm (Agrawal and Srikant 1994) is shown in Algorithm 1.

Here are some observations related to this procedure:

Pruning For frequent clique finding to be effective we need: (a) the generation of cliques to be efficient; and (b) efficient testing to see if a clique occurs in a set of graphs. Concerning (a), Since all the MIF-graphs are complete graphs, all vertices in G are connected. So, in the worst case, $O(N^k)$ subgraphs will have to be examined. The procedure in Algorithm 1 exploits the *Downward Closure Property* (see “Appendix 1”). This is used in Algorithm 1 to examine possibly fewer cliques than the worst-case. Concerning (b), the general problem of finding whether a subgraph is contained in a supergraph is the subgraph isomorphism problem, which is computationally hard (Michael and David 1979). We have some advantage in that our vertices are points in a metric space, and edges are labelled with distances. We will however require a form of error-tolerant matching, as described next.

```

1 Algorithm mif_cliques
2 |  $T, k_{max}$ 
   Input: MIF-graphs  $T$  of related targets; a maximum  $k_{max}$  ( $> 1$ ) on the size of acceptable cliques.
   Output: A set containing frequent cliques of the size  $k$  in the  $T$  graphs ( $k \leq k_{max}$ ), s.t. there are no
           frequent cliques of size  $k + 1$  in the  $T$  graphs
3 Let  $G = select\_graph(T)$ 
4 Let  $H_2$  be all frequent 2-cliques of  $G$ 
5 increment  $k$ 
6  $T' = T - \{G\}$ 
7 while ( $k \leq k_{max}$ ) do
8 |  $H_k = extend\_cliques(G, H_{k-1}, T')$ 
9 | if ( $H_k = \emptyset$ ) then
10 | | return  $H_{k-1}$ 
11 | increment  $k$ 
12 return  $\mathcal{H}_{k_{max}}$ 

```

Algorithm 1: Apriori-like MIF-clique finding. *select_graph* selects one of the T MIF-graphs represented as a set of (labeled) vertices and edges G . *extend_cliques* extends a set of frequent cliques of some size $k - 1$ to frequent cliques of size k . Here, by frequent clique C , we mean that the clique C in G also occurs in the $T - \{G\}$ MIF-graphs.

Approximate matching

In practice, when judging the occurrence of a clique (subgraph) in a MIF-graph (the supergraph), we will not be able to obtain a perfect match of a grid point in the MIF with the position of a vertex in the clique. Instead, we require the grid point to be within some ϵ -radius of the position of the vertex in the clique. The matter is thus somewhat like finding the occurrence of a subgraph in which the vertices are connected by elastic bands (or springs). In this paper, we will employ a logical encoding that will allow a usual logic-programming theorem prover to decide whether a clique occurs in a graph. The backtracking-based approach used by logic-programming systems may not always be the most efficient, but it suits the use of ILP (described below). Further, although we require cliques to occur in the MIF-graphs of all targets, this can be easily changed to cliques occurring in some minimum number of targets (a “support” threshold in the literature of frequent pattern-mining).

Incremental search

The function *select_graph* returns one of the MIF-graphs. If the clique sought has to occur in all target MIFs, then it does not matter, conceptually speaking, which of the target MIF-graphs is selected. However, some MIF-graphs may be smaller than others, and this can make a computational difference. The function *extend_cliques* extends existing frequent cliques in H_{k-1} by adding a vertex, and checks that the resulting clique occurs in the remaining $T - \{G\}$ MIFs (in the approximate sense just described).

There are several frequent subgraph mining (FSM) methods that have been developed based on the Apriori approach that are essentially no different to the procedure *mif_cliques* (see Jiang et al. 2013). In principle, any of these could be used to find frequent cliques in graphical representations of MIFs (with some modifications restricting subgraphs to cliques, and isomorphism checking to handle distance-based edge labels, as described). There are some practical difficulties however. The MIF from each probe can have many points of high-energy interaction, and the problem is compounded further by the use of multiple probes.

Table 1 Examples of constraints used to reduce the complexity of clique-finding

Probe-specific	Target-specific	Generic
(a) Min. energy of interaction	(a) Focus on a specific target protein	(a) Distance tolerances
(b) Min. occurrences in a clique	(b) Active-site size on focused target	(b) Max. vertices in cliques
(c) Max. occurrences in a clique	(c) Use amino-acid on focused target as an “anchor”	(c) Max. cliques examined

As a result, MIF-based graphs can have several hundred vertices, representing high-energy interaction points with probes. Even if the size of the clique we seek is fixed to some k , since the graphs are complete there is no straightforward algorithmic reduction in complexity possible. Some computational gains may be possible however with the use of probe, target and clique-specific constraints. Examples of the constraints we use in this paper are shown in Table 1.

Constraints like these can reduce the complexity of clique-finding, provided we have an approach that uses them directly when constructing acceptable cliques. It is possible to develop a special-purpose FSM that has these properties: for example, a specialised form of *extend_cliques* in Algorithm 1 can be used that includes such constraints. The difficulty is that this requires re-programming the function each time the domain-constraints are altered.

In this paper, we use an Inductive Logic Programming (ILP) approach to implement frequent clique finding with domain-constraints. Our principle motivation to do so is that ILP engines are general-purpose programs that can be specialised by the inclusion of domain-specific background knowledge. We believe this provides a scalable approach to the problem of inclusion of domain-knowledge into the basic clique finder in Algorithm 1.

3 Domain-specific MIF clique finding with ILP

We do not describe general details of an ILP system here: the reader is referred to a good general survey like Muggleton (1994), and to any good text on logic programming for representations of the statements here. For our purposes, it is sufficient to know that although we will be presenting logical statements in a form of English, the ILP system uses the Prolog language to represent the statements.

3.1 Representation

We start with MIF-graphs. The MIF interaction of multiple probes with a target will be represented by a logical statement of the form shown in the example below.

Example 1 Clausal representation of a MIF-graph:

Protein A is a potential target if:

*protein A has a hydroxyl probe peak at B, and
protein A has a hydroxyl probe peak at C, and
protein A has a hydroxyl probe peak at D, and
...
(and so on), and*

*protein A has a carbonyl probe peak at D, and
protein A has a carbonyl probe peak at J, and*

protein A has a carbonyl probe peak at R2, and
 ...
(and so on), and

protein A has an amide probe peak at C, and
protein A has an amide probe peak at P, and
protein A has an amide probe peak at K3, and
protein A has an amide probe peak at G, and

(and so on), and

the distance between B and D is $d_1 \pm \epsilon_1$ Angstroms, and
the distance between C and B is $d_2 \pm \epsilon_2$ Angstroms, and
the distance between C and D is $d_3 \pm \epsilon_3$ Angstroms, and
 ...
(and so on), and
the distance between J and B is $d_k \pm \epsilon_k$ Angstroms, and
the distance between J and R2 is $d_{k+1} \pm \epsilon_{k+1}$ Angstroms,
the distance between B and R2 is $d_{k+2} \pm \epsilon_{k+2}$, and
 ...
(and so on)

In this paper, this will be the representation adopted for each of the T MIF-graphs. Readers familiar with the ILP literature on inverse entailment (Muggleton 1995) will recognise this as the most-specific clause given: (a) a target protein; (b) its MIFs with multiple probes; (c) background knowledge for determining peaks in MIFs; and (d) allowing the computation of the (Euclidean) distance between pairs of points. A moment's reflection will convince the reader that this clause encodes the complete graph of the kind shown in Fig. 2b, generalised to three-dimensions (jumping ahead, it is the clausal-representation of the set \perp in Algorithm 3). The distances d_i and tolerances ϵ_i are numbers, and any clique of interest is a logical statement that can be obtained from most-specific statement. For example, a clique of the kind in Fig. 2c could be represented by a clause.

Example 2 Clausal representation of a clique:

Protein A is a potential target if:
protein A has a hydroxyl probe peak at B, and
protein A has a hydroxyl probe peak at C, and
protein A has a hydroxyl probe peak at D, and
protein A has a carbonyl probe peak at J, and
the distance between B and D is 6.3 ± 0.7 Angstroms, and
the distance between C and B is 3.6 ± 0.7 Angstroms, and
the distance between C and D is 9.4 ± 0.7 Angstroms, and
the distance between J and B is 5.9 ± 0.7 Angstroms, and
the distance between J and C is 9.3 ± 0.7 Angstroms,
the distance between J and D is 4.2 ± 0.7 Angstroms.

It is evident that this represents a clique with 4 vertices. In ILP terms, this is a clause that logically entails the most-specific clause, and it is the task of the ILP system to find such clauses, usually by employing some form of combinatorial search (like a branch-and-bound search).

3.2 Constraints

The computational complexity of the task should now be evident: given a most specific clause encoding a MIF graph with N vertices, the ILP engine is looking to choose k vertices from

the N , and obtain values for the corresponding pairwise distances and tolerances such that the resulting logical statement is true for all or most targets provided. In the worst-case, for an ILP engine employing a branch-and-bound search, this is $O(N^k k^2)$. By inclusion of domain-constraints, we can try to lower N , remove some edges in the graph, and avoid exploring all of the search space. For an ILP engine, these constraints form part of the background knowledge B that is provided as input to the system. We describe the constraints provided for experiments in this paper (“Appendix 2” describes the chemical intuition underlying some of these constraints).

3.2.1 Probe-specific constraints

These relate to the probes being used:

- *Energy* In Fig. 2a, thresholds on the energy levels are used to identify points of high-energy interaction between a target and a probe. Since each point corresponds to a vertex in the MIF graph, it is evident that changing the threshold will alter the number of vertices in the graph.
- *Occurrences* All cliques have constraints on the minimum and maximum number of vertices from a hydroxyl, amide, and carbonyl probes. This is not independent of knowledge of the active-sites of the target. This restricts the number of possible cliques that can be examined.

3.2.2 Target-specific constraints

These relate to information about the targets:

- *Focus* Although of less relevance if the cliques being sought occur in all MIF-graphs, it may be important to ensure that frequent cliques occur in the MIF-graph of at least one specific target protein. For example, for the case study in this paper, the target plasmepsin II is of special importance, and we want frequent cliques to occur in the MIF-graph of that protein.
- *Active-site size* The vertices identified by the clique will eventually be translated into pharmacophoric constraints on potential ligands. These are usually small molecules. It makes little chemical sense therefore to identify cliques with large inter-vertex distance (the area covered by the active site may be exceeded). Ligands are also more flexible than their protein targets. To allow for some conformational movement, it is advisable not to have vertices too close to each other (Van der Waals forces may develop when the points are too close, which we would like to avoid). Together, these translate to minimum and maximum constraints on distance between vertices in the MIF-graph.⁴ Edges that do not satisfy these constraints are removed from the MIF graph. The MIF-graphs are then no longer complete graphs (that is, not all vertices are connected to each other).
- *Anchor* For the case study in this paper, we want to ensure that each clique has at least one hydroxyl vertex that is not far from an aspartic acid residue. This is because the application we consider involves aspartic proteases. In particular, we will be using the locations of aspartic acid residues in plasmepsin II.

⁴ We can easily make these probe-specific. That is, minimum or maximum distances or both could be between pairs of vertices of particular probe types (between OH probe-points, OH–N probe-points, N–N probe-points and so on). We have not employed this facility here.

3.2.3 Generic constraints

The following constraints are generic to the clique-finder:

- *Tolerances* Without specific values for inter-vertex distances (the d 's in the examples earlier), it is not meaningful to test whether a clique does or does not occur in the MIF-graph of a target. Also, since not all targets interact in the same way to probes, a clique rarely occurs precisely in the same way in all targets. So unless tolerances are allowed on inter-vertex distances (the $\pm\epsilon$'s in the examples earlier), almost all cliques will be infrequent. Distances and tolerances have to be estimated from target MIFs. Only those distances that satisfy the distance constraints described are of interest. We would expect the tolerance value to be small (no more than an Angstrom or so), in order that the design of ligands is unaffected.
- *Clique-size* We have already stated the worst-case complexity of finding a k -vertex clique in a graph with N nodes ($O(N^k k^2)$). This bounds k to some fixed value.
- *Number of cliques.* For large graphs, the exponent k does not always reduce the complexity to manageable levels. For tractability, it will often be necessary to bound the total number of cliques examined. This will clearly lose any property of completeness for the clique-finder.

Broadly speaking, ILP engines allow constraints of these kinds to be encoded in one of two main ways: either as definitions in the background knowledge that control the specification of the search-space; or as the definition of statements that control the enumeration of elements in the search-space. We illustrate each of these in turn.

First, constraints can be encoded directly as part of the background knowledge. For example, the locations of probe peaks can be computed by constraining MIF energies to be above some probe-specific energy thresholds.

Example 3 Computing high-energy peaks in MIF surfaces:

*Protein A has a P probe peak at L if
the MIF energy of protein A at L has magnitude E and
the minimum energy required for probe P is E_{min} and
 $E_{min} \leq E$*

Distance-computations can also be defined in the background knowledge, and can be constrained to check ranges:

Example 4 Constrained distance computation:

Generic definition for computing Euclidean distances:

*The distance between points P1 and P2 is $D \text{ \AA}$ if
the coordinates of P1 are $(X1, Y1, Z1)$ and
the coordinates of P2 are $(X2, Y2, Z2)$ and
 D is the square-root of $(X1 - X2)^2 + (Y1 - Y2)^2 + (Z1 - Z2)^2$ and
the minimum distance allowed is D_{min} and
the maximum distance allowed is D_{max} and
 $D_{min} \leq D \leq D_{max}$*

Background knowledge definitions can also contain statements establishing values of parameters required.

Example 5 Parameter values:

The probes allowed are: n, o, and oh

The minimum distance between probe n and probe n is 3 \AA

...

The minimum occurrences of probe n is 1

The maximum occurrences of probe n is 3

... (and so on)

The second way in which an ILP system allows the incorporation of constraints is in the form of specific kinds of statements that are checked during the ILP engine's search for hypotheses. For example, here is a statement that checks if a clique (recall that this is encoded by the ILP engine as a first-order definite clause) fails the constraint of the distance from a hydroxyl peak to an ASP dyad location in plasmepsin II. If the check fails, then the clique is discarded ("pruned"):

Example 6 Search constraint:

Clause C can be pruned if
it is not true that: Clause C has a hydroxyl probe peak near an ASP dyad

Clause C has a hydroxyl probe peak near an ASP dyad if
There exists a target protein T and
Clause C has a hydroxyl probe peak at $P1$ and
There is an ASP dyad location for T at $P2$ and
The distance between points $P1$ and $P2$ is $D \text{ \AA}$ and
The maximum ASP distance allowed is D_{max} and
 $D \leq D_{max}$

The specific ILP engine we use also allows one additional way in which enumeration of the search-space can be controlled. This is in the form of explicit specification of "refinement operators" that allow the ILP engine to enumerate the search-space much more efficiently. Normally, ILP systems designed to construct and use a most-specific clause usually search the space by starting from a trivial statement ("All proteins are potential targets") and progressively adding conditions, one at a time, from the most specific clause. This is not well-suited to the representation we have shown earlier, which requires condition-triples to be added for every pair of vertices in a clique (one condition for each of the vertices, and one for the pairwise distance between them).⁵ We use the facility of being able to specify a refinement operator to change this form of default behaviour. This allows us to enumerate complete cliques that satisfy constraints like these:

1. All clauses encoding cliques of size n are refinements of clauses encoding frequent cliques of size $n - 1$ (if they exist); and
2. Extending a clause encoding a clique of size n to a clique of size $n + 1$ involves adding all conditions from a most-specific clause \perp representing a new vertex and all pairwise distances of the new vertex to vertices in the existing clique.

Here is an example of a refinement operator that implements these conditions (\square represents the empty clause):

Example 7 Clausal definition of a refinement operator:

Clause \square can be refined to Clause C if

⁵ We could, of course, have used a different representation, in which single statements encode pairs of energy-peaks along with their distances in a single condition. This representation is harder to understand, and we do not pursue it further here.

There exists a most-specific clause \perp and
The set of probes allowed is P and
Clause C contains a clique of size N and
There exists a frequent clause C' and
 C' encodes a graph (V', E') and
 C' contains a clique of size $N - 1$ and
 v is a new vertex of type $p \in P$ and
 E is the set of edges between v and vertices in C' and
Clause C encodes the graph $(V' \cup \{v\}, E' \cup E)$ and
Clause C subsumes \perp and
it is not true that Clause C can be pruned

(This can be generalised to add more than one vertex at-a-time, and more statements would be needed for boot-strapping the search, with no prior frequent clauses.)

At this point the reader may be concerned that specifying such a refinement operator may be as hard as writing a special-purpose frequent clique finder. Writing refinement operators is indeed not a straightforward business. Fortunately, this has to be done once, and as is evident, is domain-independent. Variations in domain-specific constraints are expected from one problem to another. These remain somewhat easier to write and modify.⁶ Of course, we do not have to specify a refinement operator for the ILP engine to function—their use here is purely for efficiency. The usual general-to-specific or specific-to-general operators used by most ILP engines should also yield the same results, with more computational (but less programming) effort.

3.3 Implementation

Without getting into the minutiae of the implementation, it is sufficient to think of the ILP engine as implementing the simplified procedure in Algorithm 2. In our case, we require that this procedure correctly returns frequent cliques of some size k (we cannot guarantee completeness unless the resource-bound n_{max} is large enough). This will minimally require that the *refine* function is able to extend a clique of size k to a set of cliques, each of size $k + 1$. The ILP-based frequent clique-finder is then used in a straightforward manner to find frequent cliques of size at most k_{max} (see Algorithm 3), which is the requirement of the *mif_clique* procedure in Algorithm 1.⁷

The procedure in Table 1 is the MIF-graph version of the ILP-based clique finder for ligands described in Finn et al. (1998). We can also distinguish a special case of the procedure With $B = \emptyset$. In this case we obtain an unconstrained frequent-clique finder in MIF-graphs. This can be seen as the MIF-graph equivalent of the frequent clique-finder for ligands described in Podolyan and Karypis (2009). This latter procedure will be used in experiments below as a baseline for comparing the performance of the domain-specific MIF-clique finder.

3.4 Pharmacophores from cliques

Any of the (frequent) cliques found by the ILP engine can be used to act as a template for a pharmacophore. Recall that a pharmacophore is a collection of chemical properties—called the pharmacophore’s features—such as hydrogen bond donor, hydrogen bond acceptor, elec-

⁶ In this paper we do not look at automatic translation of domain-constraints into ILP-understandable logical statements. For the kinds of constraints here, there does not appear to be any significant difficulty in doing this.

⁷ The repeated computation effort of refining clauses of size k when considering clauses of size $k + 1, k + 2, \dots$ can be avoided. We do not go into these details here.

```

1 Algorithm ilp_find_clauses
2 |  $B, E, c_{max}, n_{max}$ 
   Input: Background knowledge including domain-constraints  $B$ ; a set  $E$  of instances; an upper bound
            $c_{max}$  ( $c_{max} > 0$ ) on the size of the clauses examined; and a bound  $n_{max}$  ( $n_{max} > 0$ ) on the
           number of clauses examined.
   Output: A set of clauses of size at most  $c_{max}$  s.t. each clause in the set is “good”
3 Let  $e \in E$ 
4  $\perp = \text{most\_specific\_clause}(e, B)$ 
5  $i = 1$ 
6  $n = 0$ 
7  $Clauses_i = \{\square\}$ 
8  $Good_i = \emptyset$ 
9 while ( $Clauses_i \neq \emptyset$ ) do
10 | Select  $Clause$  from  $Clauses_i$ 
11 |  $NewClauses = \text{refine}(Clause, \perp, B)$ 
12 | increment  $n$ 
13 | if ( $n > n_{max}$ ) then
14 | | return  $Good_i$ 
15 |  $New = \{C : C \in NewClauses \text{ and } size(C) \leq c_{max} \text{ and } \neg \text{prune}(C, B)\}$ 
16 |  $NewGood = \{C : C \in New \text{ and } \text{good}(C, E, B)\}$ 
17 | Store each clause  $C \in NewGood$  as a good clause of size  $size(C)$ 
18 |  $Good_{i+1} = Good_i \cup NewGood$ 
19 |  $Clauses_{i+1} = (Clauses_i - \{Clause\}) \cup New$ 
20 | increment  $i$ 
21 return  $Good_i$ 

```

Algorithm 2: A simplified ILP implementation that uses a refinement operator to proceed from the empty clause \square to clauses upto some maximum size. The implementation uses the following functions: *most_specific_clause*, that the “most specific clause” in the sense described in Muggleton (1995); *size*, that returns some measure of the size of a clause; *prune*, that is *TRUE* if a clause should be removed from further consideration; *refine*, that returns a set of refinements for a clause; and *good* that is *TRUE* for any clause that is an acceptable answer. Here, if the refinement operator is defined to return cliques and “good” is defined as a frequent clause, then the ILP engine returns frequent cliques. The refinement operator can be further configured to construct frequent cliques of size k using cliques of size $k - 1$ stored in Step 17.

trostatic and hydrophobic interactions complementary to a target’s active site; along with three-dimensional constraints on distances between the features. There is a straightforward translation of the cliques obtained into pharmacophores. The vertices (carbonyl, hydroxyl and amide) are first translated into pharmacophore features of donors and acceptors (carbonyls map to acceptors, hydroxyls and amides to donors). The distance between the vertices translates to the distances between the corresponding donors and acceptors in the pharmacophore.⁸

It is worth reiterating at this point that in ligand-based approaches, a pharmacophore is derived from the structure of one or more known inhibitors of the target. Here, cliques are obtained using the target’s interaction with the probes. The resulting pharmacophores are based on the structure of a family of receptors, which is an important conceptual difference. Ligand-based approaches are also subject to the “excluded-volume problem”. The problem

⁸ The lock and key model assumes that active site is flexible and it changes its shape when an inhibitor binds to the active site. The flexibility of the active site is taken into account by introducing a tolerance of 1 Å on inter-feature distances in pharmacophores. For the case study in this paper, this translation results in pharmacophore queries in the form of SMARTS (Bone et al. 1999) patterns. These queries can then be used to directly screen a chemical database. We do not elaborate further on this here.

```

1 Algorithm ilp_find_mif_cliques
2 |  $B, T, k_{max}, n_{max}$ 
   Input: Background knowledge including domain-constraints  $B$ ; MIF-graphs  $T$  of related targets
           encoded using suitable logical predicates in  $B$ ; a maximum  $k_{max}$  ( $> 1$ ) on the size of
           acceptable cliques; and a maximum  $n_{max}$  on the number of cliques examined of any size
   Output: A set containing frequent cliques of some size  $k$  ( $k \leq k_{max}$ ), s.t. there are no frequent cliques
           of size  $k + 1$ 
3  $k = 2$ 
4  $H_{k-1} = \emptyset$ 
5 while ( $k \leq k_{max}$ ) do
6 |    $c_{max} = \lceil 1 + k + (k)(k - 1)/2 \rceil$ 
7 |    $H_k = ilp\_find\_clauses(B, T, c_{max}, n_{max})$ 
8 |   if ( $H_k = \emptyset$ ) then
9 |     | return  $H_{k-1}$ 
10 |   increment  $k$ 
11 return  $H_{k_{max}}$ 

```

Algorithm 3: Using the ILP-based good clause-finder to find MIF-cliques. This assumes that the *ilp_find_clause* procedure has a refinement operator that can enumerate clauses encoding cliques and a function that can decide that a clique (clause) is good when it is frequent in the T graphs.

stems from the fact that a ligand atom cannot occupy a space already taken up by a protein atom. This information is not usually available when generating pharmacophores purely based on ligand-structure. Here, since the MIF surfaces are not calculated for grid points lying inside the protein, the volume excluded by the target is automatically taken into account.

3.4.1 Generalising cliques

It is evident that a frequent clique yields a pharmacophore with maximal distance constraints on features. Pharmacophores with the same number of features, but fewer constraints can be obtained as quasi-cliques from the subgraph lattice of a frequent clique (see “Appendix 1” for the meaning of these terms). An example of this lattice is in Fig. 3. With the clausal representation adopted here, this lattice is part of the clause subsumption lattice familiar to ILP practitioners (dropping an edge amounts to dropping a literal in the clausal representation of the graph). Adopting the convention used in the ILP literature, we will call pharmacophores derived from cliques at higher levels in the lattice as being more general than those from a lower level. In this paper, we will be looking at pharmacophores derived from frequent cliques (Level 0) and from some kinds of quasi-cliques at Level 1 only (the details are in Sect. 4.4).

We now describe an application of the approach to the design of antimalarials.

4 Case study: discovering antimalarials

We investigate the ILP-based method of determining multi-probe MIF-cliques using a series of proteins known to form targets for malaria. Specifically, we have structures of six proteins related to the receptor plasmepsin II; and MIF data from the interaction of the six proteins with hydroxyl, amide and carbonyl probes. Plasmepsin II is involved in the haemoglobin degradation pathway of *Plasmodium falciparum*, the parasite most commonly involved in malarial deaths: plasmepsin II is a known target for anti-malarial drugs against this parasite. Our aim is to discover frequent multi-probe MIF cliques using the ILP-based procedure

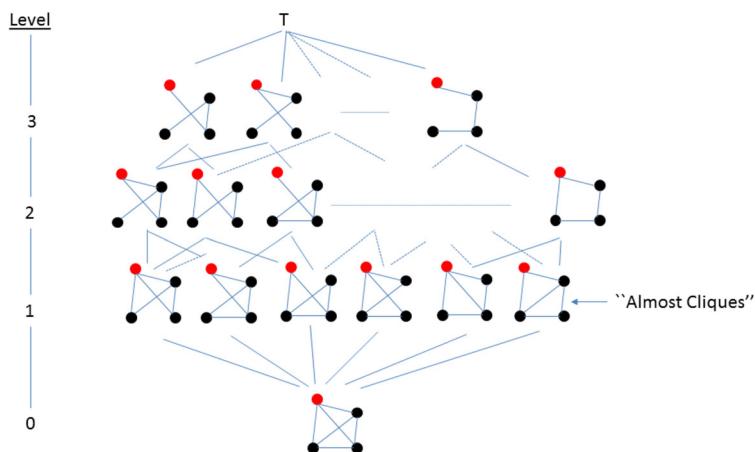


Fig. 3 The subgraph lattice obtained from a clique. The clique is from Fig. 2. Vertices in the lattice are restricted to connected subgraphs of the clique. Using the terminology in “Appendix 1”, each vertex is a quasi-clique. The term “almost clique” is used to denote a subgraph at Level 1. That is, these are the quasi-cliques resulting from dropping a single edge from the clique (Color figure online)

just described; translate these into pharmacophore-like constraints on potential ligands; and evaluate these constraints quantitatively.

To the best of our knowledge, *de novo* design of this kind has not been attempted with this series of target proteins.

4.1 Data

4.1.1 Training

Training data are obtained from six proteins targets for antimalarials (specifically, plasmodial aspartic proteases). These are: Cathepsin D, Pepsin, Plasmepsin Vivax, Plasmepsin I, plasmepsin II and plasmepsin IV. Crystal structures of five of these are known in advance. The structure of plasmepsin I is based on a model. Three probes will be used: amide (N), carbonyl (O) and hydroxyl (OH).

The program computing the MIF surface usually employs a much finer-grained grid than is necessary for our purposes. We pre-process the data to obtain a coarser characterisation of the MIF surface by clustering together groups of grid-points that do not show significant variations in energy. The energy for such clusters are taken to be the average energy of all points in the cluster, and the centre of the cluster is the coordinate-mean of the points in the cluster. By “significantly different” to points in an existing cluster, we mean that the energy of a grid-point is at least 2 standard deviations away from the average energy of the cluster.

The result is for each target and probe a set of (mean) grid-points, each associated with a (mean) energy value. The table below shows the numbers of points in the averaged MIF field for each of the targets and probes:

Target	Probe		
	N	O	OH
Cathepsin	209	221	228
Pepsin	159	101	160
Plasmepsin Vivax	189	149	193
Plasmepsin I	228	232	258
Plasmepsin II	179	174	193
Plasmepsin IV	188	158	213

On average, the clustering reduces the number of grid points by a factor of 10 for each protein-probe combination. That is, the number of grid points before clustering are in the order of 1000s. Target-specific MIFs therefore contain a description of the mean MIF-values for each target-probe combination. That is, the coarse-grained MIF surface for Cathepsin will consist of $209 + 221 + 228 = 658$ points, for Pepsin, 420 points and so on. For any target, each point is a potential vertex in the MIF-graph for that target.

4.1.2 Testing

Our interest here is in two different kinds of populations of molecules. First, the antimalarial campaign by pharmaceutical companies like GlaxoSmithKline and Novartis has resulted in a large set of chemicals (ChEMBL: [Gaulton et al. 2012](#)) for which bioactivities against several plasmoidal aspartic proteases are known. These are used to identify sets of positives and negatives. The subset of positives is composed of 568 chemicals which are exclusive inhibitors of aspartic proteases (plasmepsin II, cathepsin D and pepsin); and set of negatives contains 626 chemicals that do not target aspartic proteases. These include non-protease inhibitors (393; mostly kinase, phosphodiesterase, phosphatases, reductases and cytochrome inhibitors) and protease inhibitors (233; but excluding inhibitors of aspartic proteases like plasmepsins, cathepsin D and pepsin). The details are below:

Targets	Positives	Negatives
PlasII	236	0
CathD	243	0
PlasII+CathD	89	0
Protease	0	233
Non-Protease	0	393
[6pt] Total	568	626

The chemicals in these two subsets do not overlap, and their inhibition constants like K_i or IC_{50} are known. To use active inhibitors for screening by pharmacophores all the inhibitors of the selected target proteins whose inhibition constant (K_i or IC_{50}) is less than $50 \mu\text{M}$ are selected. Finally, ligand flexibility is considered by generating conformers for each molecule. As usual, a pharmacophore “hits” a small molecule if it occurs in at least one conformer of the molecule.

Secondly, we are interested in a set of “decoy” compounds. These are small molecules that are physico-chemically similar to the aspartic protease inhibitors (positives). Decoy compounds are extracted from the ZINC database (<http://zinc.docking.org/>). The compounds are similar to the positives in their physico-chemical properties (like molecular weight, HBA, HBD, logP, number of rotational bonds and so on: slight differences are permitted, like ± 3 for HBA and HBD; ± 30 Da for molecular weight and so on), but are structurally dissimilar (Tanimoto coefficient < 0.7). At least 20 decoys for every positive example is found, resulting in a total 11878 decoy compounds.⁹

In all cases, about 250 low-energy conformations are generated for each chemical in the positives, negatives and decoys to cover the space of possible conformations of the small molecules. In principle, a good pharmacophore model should be able to identify more of the positives and fewer of the negatives.

4.2 Background knowledge

The (coarse-grained) MIF-surface for a target is encoded using the predicate:

$$Has_Energy(t, p, l, e)$$

that is true if the (mean) interaction energy of target t with probe p at (cluster) location l (locations are specified by three-dimensional coordinates) is e . The principal predicates defined using this are:

$$Has_Probe_Peak(t, p, l)$$

and

$$Dist(l_1, l_2, d, \epsilon).$$

The first predicate encodes the locations l at which the interaction energy of target t with probe p exceeds the lower bound on energy for the probe. The second predicate is true if the distance between locations $l_{1,2}$ is $d \pm \epsilon$. Both these predicates incorporate domain-specific constraints for energy thresholds, distance constraints and tolerances. For the ILP engine used here these are communicated through user-defined parameters.

As described earlier, the background knowledge also includes definitions of conditions under which a clause (or clique) should be removed from the search. This is done using the predicate:

$$Prune(c)$$

that is true for clauses c that should be pruned from the search. These are used to remove cliques in which: (a) vertices contain probes that fall below a lower bound on the number of occurrences of the probe; (b) vertices contain probes that exceed an upper bound on the number of occurrences of the probe; and (c) there is no hydroxyl peak in plasmepsin II within 5 Å of an ASP dyad.

⁹ The machine learning literature would call the decoys “near-misses”, but the term decoy is standard in the literature on drug-design. Decoys are near-misses in a physico-chemical sense, but not necessarily in the classification sense. That is, it is not known for sure whether or not the set of decoys do or do not bind to the aspartic proteases. The decoys have a low Tanimoto similarity to the positives, but as will be seen later in the paper, there are many pairs of positives which also show such low Tanimoto similarities. The sets of positives and negatives are fairly small (due to limited experimental validation), and the decoy data can be used to obtain a realistic estimate of false positives over a large chemical space.

In addition to the problem-specific predicates above, it is possible to configure the ILP engine to enumerate cliques using a refinement operator. For a given clique size k , we use a refinement operator that incrementally constructs cliques of size k from frequent cliques of size $k - 1$ (see Sect. 2.1). On the first iteration, the refinement operator constructs cliques *ab initio*, since no frequent cliques have been determined.

We can also ensure that some of these conditions tested by the *Prune* predicate do not arise by ensuring that the refinement operators do not generate such clauses. This would make the definition of these operators somewhat specific to MIF-graphs.

4.3 Algorithms and machines

The principal programs used in this paper were: GRID (Goodford 1985) for generating the molecular interaction fields for target-probe pairs; and the procedure in Table 1. The ILP engine used by the procedure is Aleph (Srinivasan 1999). We distinguish two variants of the MIF-clique finder: one that uses the domain-constraints described above, and a second that does not. The latter will act as a baseline for judging both the utility of the background knowledge, as an application of standard frequent clique-finder to the problem.

Generation of conformers for the molecules in the test set were done using the OpenBabel program (O'Boyle et al. 2011). Pharmacophore-based searching of the dataset is carried out using a standard chemical development kit package (CDK: Steinbeck et al. 2003) where the pharmacophores features are first converted into SMARTS patterns (Desaphy et al. 2014). Chemicals for the decoy dataset were identified using the DecoyFinder program (Cereto-Massaquet al. 2012).

4.4 Methods

We adopt the following method to test the *ilp_mif_cliques* procedure in Table 1:

1. With and without background knowledge:
 - (a) For each malaria target T and each probe P obtain the MIF values at pre-defined grid points.
 - (b) Find the set of cliques for clique-sizes at most k_{max} using *ilp_mif_cliques*.
 - (c) Convert the cliques found into pharmacophores.
 - (d) Obtain the performance of the pharmacophores quantitatively, using the test datasets D1 (positives and negatives) and D2 (positives and decoys).
2. Compare the quantitative performance of the pharmacophores with and without background knowledge.

For simplicity, we will call the frequent-clique finder with domain constraints as the “Domain-Specific” variant; and the frequent-clique finder without background knowledge as the “Baseline” variant. The comparison in the last step serves the following purposes. First, we are able to obtain an assessment of the utility of the background constraints. Secondly, we can view this as a comparison against a procedure used in the literature for ligand-based frequent clique finding (Podolyan and Karypis 2009), adapted to the problem of finding frequent cliques in MIF-graphs. The Baseline variant also extends the results that would be obtained by the commercial program FLAP (Cross et al. 2010), that finds up to 4-vertex cliques (not necessarily frequent) using target structure.

The following details are relevant:

- We use the following probe- and target-specific constraints in experiments: (a) Minimum distance; (b) Maximum distance; (c) Minimum and maximum occurrences; (d) Energy thresholds; and (e) Distance to anchor residue. The values used here are: Maximum distance: 10 Å; Minimum distance: 3 Å; Minimum occurrences: 1 (all probes), Maximum occurrences: 2 (all probes); Energy thresholds: -3.5 (N), -2.2 (O), -3.0 (OH); and Anchor distance: 5 Å. For more details on some of these constraints, see “Appendix 2”. The following values were used for generic constraints: Distance tolerance: 0.70 Å; Max. clique size: 8; Max. number of cliques: 100,000 (for each clique size). In addition, we impose a time-limit of 2 days for finding frequent cliques of any given size. This is to keep runtimes within reasonable limits.
- Language constraints provided to the ILP engine are in the form of “mode” declarations that specify argument types and input–output roles of arguments for the two main predicates described above, namely *Has_Probe_Peak* and *Dist*. We refer the reader to Muggleton (1995) for a description of mode declarations.
- Quantitative assessments will be based on the 2×2 confusion matrix resulting from the use of pharmacophores to identify aspartic proteases. We will examine performance on 2 datasets: (D1) the set of positives and negatives; and (D2) the set of positives and decoys. As usual, counts will be obtained for: positives that are predicted as inhibitors (true positives, or TP); positives that are not predicted as inhibitors (false negatives, or FN); negatives (or decoys) that are predicted as inhibitors (false positives, or FP); and negatives (or decoys) that are predicted as non-inhibitors (true negatives, or TN).¹⁰ For convenience, we will call $P = TP + FN$ and $N = TN + FP$. The usual measure that balances correct prediction and errors is the F_β -score:

$$F_\beta = \frac{(1 + \beta^2) \cdot TP}{(1 + \beta^2) \cdot TP + \beta^2 \cdot FN + FP}$$

We are primarily interested in two kinds of pharmacophores: those with high precision and those with high recall. The former are useful for obtaining structure-activity information from the hits, and the latter are useful as screening for potential ligands. The former will have high F -values for $\beta < 1$ and the latter will have high F -values for $\beta > 1$. We will use $\beta = 0.5$ and $\beta = 2$ to assess the pharmacophores tested. On dataset D2, it is of interest in the area of drug-design to calculate the Enrichment Factor (EF) which measures the performance of the pharmacophores as a model for targeting inhibitors, over a random choice targeting model. This is defined as:

$$EF = \frac{TP/(TP + FP)}{P/(P + N)}$$

Readers will recognise this as the lift, or gain in precision, obtained by using the pharmacophores on dataset D2. An EF value close to 1 indicates no gain in performance over a random predictor.

- We expect pharmacophores obtained from frequent cliques to be highly specific (that is, have high precision and perhaps low recall). Using the introduced terminology, we will call these *maximally-specific* pharmacophores. For more general pharmacophores, we also consider quasi-cliques at Level 1. At this level, we restrict ourselves to quasi-cliques obtained by dropping an edge that may be chemically redundant. For example, if the

¹⁰ The counts are for the pharmacophore ensemble, and not average counts of individual members of the ensemble. That is, we have a hit if any member of the ensemble predicts a molecule as an inhibitor, and a miss if no member of the ensemble predicts the molecule as an inhibitor.

frequent clique contains an edge between a pair of amides and a pair of hydroxyls then one of these may be chemically redundant (both amides and hydroxyls are hydrogen-bond donors, and there may be no need to restrict the location of both). We will call the resulting pharmacophores as *slightly-general* pharmacophores. Generalised pharmacophores are assessed in the same way as the maximally specific pharmacophores.

4.5 Results

Summaries of the performance of pharmacophores from the Domain-Specific and Baseline variants on dataset D1 (positives and negatives) are in Table 2. Summaries of the graphs on the training data, and the time taken to find frequent cliques are in Table 3. Recall that each variant actually identifies k -vertex frequent cliques, which are then converted into k -feature maximally-specific pharmacophores. Slightly general forms of these are obtained by examining the quasi-cliques arising from dropping any one constraint that may be chemically redundant. For example, the largest sized frequent cliques found by the Domain-Specific variant have 5-vertices. These translate to 5-featured pharmacophores. “Slightly-general” pharmacophores are obtained by dropping 1 of the 10 edges in the 5-vertex cliques found. The Domain-Specific variant has predominantly one type of clique ($N_2-O_1-OH_2$: see Table 4), and the tabulations are the performance of that type. Two kinds of slightly general pharmacophores result by removing the $N-N$ edge or the $OH-OH$ edge, both N and OH are hydrogen bond donors, and chemically deemed possibly redundant.

At first glance, it would appear erroneous that the Baseline graph in Table 3(a) does not contain all pairwise distances (that is, $(546 \times 545)/2 \approx 150,000$ edges). This is because a target’s interaction points with distinct probes may not all be distinct (that is, the 3-d positions of vertices are not always distinct). This means some pairwise distances will be 0; and others may be duplicated. Zero-distance edges are eliminated, being chemically meaningless; and duplicate edges are represented only once.

For the Domain-Specific variant, as expected, the maximally-specific cliques resulted in the highest precision and lowest recall. Some of the low recall may also result from the limited active chemical space spanned by the chemicals used in the positive examples. This chemical space involves inhibitors for aspartic proteases like plasmepsin, pepsin and cathepsin D which is far less than the chemical space for the negative examples (containing inhibitors for non proteases and other proteases). Eliminating an edge allows us to take into account the flexibility induced when a protein binds to a ligand. Chemical considerations suggest that edges between the two donor sites (N or OH) are likely candidates for such an elimination: this is confirmed by the results, both of which increase the number of hits (at the cost of precision, of course). These results suggest that these slightly general pharmacophores may be useful for virtual screening (especially the one resulting from dropping the $OH-OH$ edge, which greatly increases recall, without too great a loss in precision).

The Baseline variant clearly results in pharmacophores that have no value either for structure-activity prediction or for screening. Even the smallest MIF-graph that is considered by the Baseline is substantially larger than the Domain-Specific one, and this is reflected in significant increases in the time required to find frequent cliques. Recall that the Baseline is in effect, a resource-bounded version of the frequent clique-finder used in Podolyan and Karypis (2009) with ligands. These results appear to suggest that a direct adaptation of a ligand-based frequent clique finder will not work well: probably because MIF-graphs are substantially larger than ligand-graphs. The caveat is that the frequent clique-finder used is a resource-bounded one (for any size of clique, no more than n_{max} 100,000 cliques and no more than 2 days of search time). It is possible therefore that given more resources, a better

Table 2 Performance of pharmacophores on the test-dataset D1 (positives and negatives)

Pharmacophore features	Pharmacophore specificity	Performance				P	R	$F_{0.5}$	F_1	F_2
		TP	FP	FN	TN					
(a) Domain-Specific										
5	Maximally-specific	111	14	457	612	0.89	0.20	0.52	0.32	0.23
5	Slightly-general ($N-N$)	133	54	435	572	0.71	0.23	0.51	0.35	0.27
5	Slightly-general ($OH-OH$)	270	75	298	551	0.78	0.48	0.69	0.59	0.52
(b) Baseline										
7	Maximally-specific	0	0	568	626	-	0.00	0.00	0.00	0.00
7	Slightly-general ($N-N$)	0	0	568	626	-	0.00	0.00	0.00	0.00
7	Slightly-general ($OH-OH$)	0	0	568	626	-	0.00	0.00	0.00	0.00
6	Maximally-specific	0	0	568	626	-	0.00	0.00	0.00	0.00
6	Slightly-general ($N-N$)	0	0	568	626	-	0.00	0.00	0.00	0.00
6	Slightly-general ($OH-OH$)	0	0	568	626	-	0.00	0.00	0.00	0.00

P and R denote precision and recall respectively ($P = TP/(TP + FP)$ and $R = TP/(TP + FN)$). The F -measures summarise the performance on the test-data. The pharmacophore with high $F_{0.5}$ value denotes higher precision, and is useful for extracting structure-activity relations from the hits. Pharmacophores with higher F_2 values are better for virtual screening. For the Domain-Specific variant, the largest sized frequent cliques found were of size 5, and overwhelmingly of a single type ($N_2 - O_1 - OH_2$). For the Baseline variant, the largest sized frequent cliques were of size 7, and contained a range of types. For the Baseline, we have tabulated the performance of both 7-vertex and 6-vertex cliques: it is evident that performance does not improve with the smaller sized cliques

Table 3 Empirical summaries of frequent-clique finding on the training data

Variant	Smallest MIF-graph				
	Vertices	Edges			
(a)					
Domain-Specific	147	2,403			
Baseline	546	100,128			
Variant	Time for finding frequent cliques				
	4-vertex	5-vertex	6-vertex	7-vertex	8-vertex
(b)					
Domain-specific	34.1 min	2.1 h	2.1 h	–	–
Baseline	1.7h	1.5 days	>2 days	>4 days	>4.3 days

For the Domain-Specific variant, only interactions above a certain energy threshold can appear as vertices in a MIF-graph. In addition, constraints on distances in sparser MIF-graphs with the Domain-Specific variant. Since we want cliques that are present in all MIF-graphs, it is sufficient for the frequent-clique finder to obtain candidate cliques from the smallest of these graphs. The size of this smallest graph is tabulated in (a). The time taken to find frequent cliques of any given size is restricted to a maximum of 2 days (see the notes in the Sect. 4.4). This restriction is required to ensure some measure of tractability with the Baseline approach. Both Domain-Specific and Baseline variants use the Downward Closure property to ensure that frequent cliques of size $k + 1$ are extensions of frequent cliques of size k . For the Domain-Specific variant there are no frequent 6-vertex cliques, making it unnecessary to consider 7- and 8-vertex cliques

Table 4 Pharmacophores obtained from frequent cliques found by the domain-specific variant

Features	Type	Count
5	$N_2-O_1-OH_2$	2031
	$N_2-O_2-OH_1$	2
	$N_1-O_2-OH_2$	1

performance may result. Nevertheless, other things being equal, it appears that the Domain-Specific variant yields substantially better pharmacophores, in substantially less time, and we will henceforth not consider the Baseline variant. In summary, the results in Tables 2 and 3 provide evidence for the following:

- The performance of the Domain-Specific variant is substantially better than the Baseline, and models are found substantially faster with domain- and generic-constraints than without; and
- Good precision, recall, and F-values are obtained with the Domain-Specific variant on test data consisting of positives (compounds that only target aspartic proteases) and negatives (compounds that do not target aspartic proteases).

Table 5 shows the Enrichment Factors obtained with the Domain-Specific variant on dataset D2 (positives and decoys). It is evident that good enrichment factors are obtained with the Domain-Specific variant. The tabulations suggest that the gain in precision may be about 10–12 times higher than the expected precision of a random choice targeting model.

The reader may be concerned that the large number of frequent cliques (here about 2000 for the Domain-Specific variant) will result in diminished comprehensibility. This is correct in the first instance, but we are able to alleviate this concern somewhat by clustering the

Table 5 Performance of the Domain-Specific variant on the set of positives and decoys

Pharmacophore	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>TN</i>	P_{Pharm}	P_{Av}	$EF = \frac{P_{Pharm}}{P_{Av}}$
Maximally-specific	111	73	457	11,805	0.60	0.05	12
Slightly-general (<i>N–N</i>)	133	154	435	11,724	0.46	0.05	9
Slightly-general (<i>OH–OH</i>)	270	237	298	11,641	0.53	0.05	10

Here P_{Pharm} denotes the precision obtained using the pharmacophores ($=TP/(TP + FP)$) and P_{Av} is the expected precision without any model (568/11,878). EF denotes the Enrichment Factor, or gain in precision, or lift from using the pharmacophores as a mechanism for targeting aspartic protease inhibitors

maximally-specific cliques (see Kaalia et al. 2015). The clustering suggests that there are probably only about 10 different clusters of cliques of type $N_2-O_1-OH_2$ found by the ILP engine, with most of the cliques contributing to the hits falling in the same cluster. This suggests that there may be a small number of truly different and active pharmacophores that need to be considered. The analysis in Kaalia et al. (2015) presents a case of why the main type of clique ($N_2-O_1-OH_2$) makes sense chemically, and we refer the reader to that paper for chemical-comprehensibility of the results.

When assessing the body of results presented thus far, an entirely reasonable question to ask is this: how well can we discriminate between positives and negatives, based simply on molecular similarity? Figure 4 shows the intra- and inter- subset similarity of molecules in the positive and negative examples in dataset D1. Similarity is calculated using the pairwise Tanimoto coefficient from Daylight fingerprints of the compounds.¹¹ Figure 4 shows that there is a significant overlap between the bulk of the positive and the negative examples (a subset of positives can be identified with precision with Tanimoto coefficients above 0.4 or so, but recall will suffer). This is also evident from the scatterplot in Fig. 5 generated using multi-dimensional scaling using the Tanimoto similarity: the small set of separable positives is in the upper left-hand corner.

4.5.1 Note on the precision-recall tradeoff

There are no defined standards for precision and recall values of pharmacophore-based screening of ligand databases. For example, a study on ligand-based pharmacophore models for the discovery of 17β -Hydroxysteroid Dehydroxygenase 2 inhibitors (Vuorinen et al. 2014) reported precision rates of 0.24–0.50, and recall rates of 0.40–0.50. Screening by receptor-based pharmacophore models designed for HIV-1 protease inhibitors (Fisher and Gner 2002) reported precision of about 0.10–0.4; and recall values of 0.007–0.05. This suggests that the values obtained here are quite respectable.¹²

¹¹ See: <http://www.daylight.com/dayhtml/doc/theory/theory.finger.html>. Daylight fingerprints are basically binary strings of 0 and 1 denoting the presence/absence of certain structural features. For a pair of binary strings S_A and S_B , let A denote the set of non-zero bits in S_A ; B denote the set of non-zero bits in S_B ; and C denote the set of common non-zero bits in S_A and S_B . Then the Tanimoto similarity between S_A and S_B is the ratio $|C|/|A \cup B|$. This is simply the Jacquard index.

¹² Precision with the decoys may seem low. Decoys are near-misses, and therefore, precision is expected to be lower on dataset D2 than on D1. In fact, D2 represents quite a difficult test for the method, since for every positive, there are 20 near misses included in the decoy dataset. Under the circumstances, our chemical expertise suggests that the precision obtained is good (for every 3 positives, 2 decoys of the corresponding 60 near-misses are classified as positive).

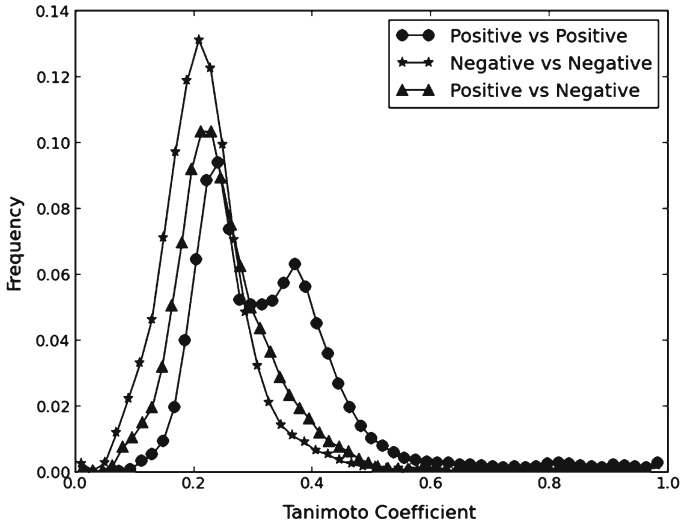


Fig. 4 Comparison of similarity (Tanimoto coefficient) between the sets of positive and negative examples

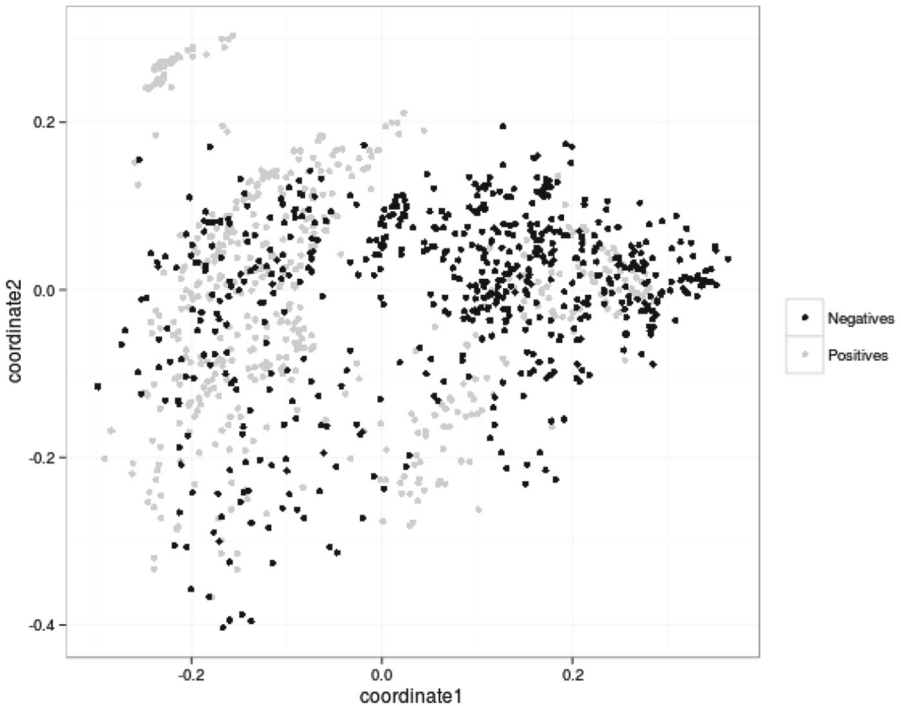


Fig. 5 A visualisation of the positive and negative examples that maps the Tanimoto similarities onto a two-dimensional space using multi-dimensional scaling

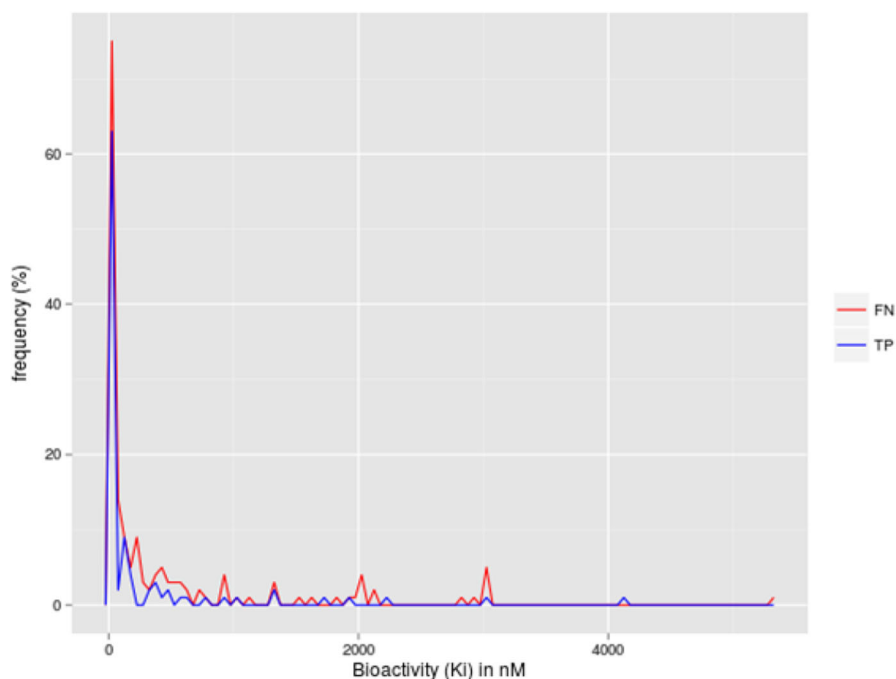


Fig. 6 Frequency distribution of true-positives and false-negatives. Across a wide range of bioactivity, there are roughly as many true-positives and false-negatives, suggesting that the models used do not show any preference for any particular bioactivity range

Generalising pharmacophores in the manner we have described—dropping distance constraints based on chemical principles—is one way of improving recall at the cost of precision. There are other ways of achieving this, although results may be less predictable. We have investigated elsewhere (Kaaliala et al. 2015) the effect of changing the distance-tolerance between pharmacophore features from 10 Å as used here to 20 Å. In the first instance, it may be thought that relaxing the distance tolerance should allow more general pharmacophores (and hence increase recall). In fact, the results in Kaaliala et al. (2015) do not bear this out, probably because the target’s structure precludes overly-large pharmacophores. It may be more productive to conduct instead a heuristic search through the subgraph lattice described in Sect. 3.4.1, guided for example, by F -values, or by the activities of the false-negatives. False negatives may be acceptable if they are restricted to poor inhibitors: the generalisation technique in this paper is agnostic to activities of true-positives and false-negatives (see Fig. 6).

5 Related work

De novo drug-design is an active area of research, with techniques ranging from the purely biophysical to the purely computational. We focus first on related work that has used machine-learning in some manner. It has been our intention here to extend the ligand-based discovery of pharmacophore cliques in Finn et al. (1998) to target-based discovery of MIF cliques. Both the approach of Finn *et al.* and the paper here use ILP as the basic engine for clique discovery. In the former, background knowledge contains the location of hydrogen bond

donors, acceptors, and the location of zinc-sites on ligands. No probe- or target-specific constraints of the kind described here are employed to restrict the search space. Independently of the work in [Finn et al. \(1998\)](#), [Podolyan and Karypis \(2009\)](#) also approach the problem of pharmacophore detection as clique identification. They do not use a general-purpose machine learning program, but a special-purpose program that incrementally constructs larger cliques from smaller, frequent cliques. The Baseline variant that we have constructed here is a direct adaptation of this ligand-based method to the problem of frequent clique-finding in MIF graphs.

There are several other methods that concentrate purely on ligand-based pharmacophore modeling such as HipHop ([Barnum et al. 1996](#)), HypoGen ([Li et al. 2000](#)), DISCO ([Martin 2000](#)), Catalyst ([Hecker et al. 2002](#)) and PHASE ([Dixon 2006](#)). These cannot be re-used for receptor-based models: we have been able to use essentially the same ILP program used for ligand-based models in [Finn et al. \(1998\)](#) to construct receptor-based models here. pharmacophore generation methods do exist such as LigandScout ([Langer and Wolber 2005](#); [Lai and Chen 2006](#); [Schuller et al. 2006](#)). These require either protein structure and at least one known ligand for identifying pharmacophoric features or domain knowledge in the form of residue based interactions. None of these use MIFs to identify pharmacophoric features, or the kind of domain-knowledge we have used here.

Perhaps the most directly relevant computational approach to this paper is FLAP (“Fingerprints for ligands and proteins”: [Cross et al. 2010](#)). As in this paper, FLAP uses MIF values at grid-points calculated by the GRID program. FLAP refers to proprietary software that contains, amongst other useful procedures for assisting receptor-based drug design, programs that find 3- or 4-vertex cliques from the MIF surface obtained from some part of the active site of a single target and a probe. There are some difficulties with using FLAP for the problem described in this paper: cliques found here are of size 5; we use MIFs from all of the active site; we can use MIFs from multiple targets and probes at once; and we have ensured that the result conforms to various domain-specific intuitions. We are not aware of any immediate way to extend FLAP to account for these differences. Nevertheless, there may not be any conceptual issues preventing the development of a generalised form of FLAP that has these facilities.

Turning to more general work in the area of machine learning, the research that is clearly relevant is that on frequent subgraph mining (FSM: see [Jiang et al. 2013](#)). It is clear that an FSM program can be adapted to the task of finding cliques in MIF-graphs. It is less apparent what modifications would be needed to incorporate flexible methods of incorporating biological and chemical constraints into subgraph mining.

6 Conclusions

This study explores an ILP-assisted approach for de novo drug design in which an ensemble of pharmacophores is designed for a class of target proteins, without any prior knowledge of ligands. The approach here is to first identify frequently-occurring cliques in a graphical representation of energy surfaces. The surfaces are obtained from the energies of interaction of the proteins with some specific kinds of chemicals (or probes). Points of high energy in the surfaces are treated as vertices in a graph, with edges weighted by the inter-vertex distance. Cliques that occur in the energy-graphs of all (or many) proteins form the basis for pharmacophores that can be used to identify or describe potential ligands. The case for using ILP to find cliques rests primarily on its ability to incorporate domain-knowledge directly into

the clique-detection process. We have shown here how different kinds of domain constraints like distances between energy peaks, probe-specific energy thresholds, and pharmacological constraints imposed by the target proteins can be encoded as background knowledge provided to an ILP engine that is configured to find frequent cliques in a set of graphs. The result is a general-purpose method that can be specialised to any class of proteins and any number of probes (what will change are some of the domain-constraints—the ILP engine and the algorithm it uses remains the same). To the best of our knowledge, there have been no machine-learning or graph-based methods that have been shown to perform this form of multi-target, multi-probe de novo drug design that takes such a variety of domain-knowledge into account. Of course, there is no necessary requirement for multiple probes, and the technique we have used here can work without modification with multiple targets and a single probe. The notion of a frequent clique with a single target becomes trivial: the approach here will find all cliques occurring in the target, but if this is to be manageable, then we would need more background knowledge than we have here.

We have presented an extensive case study concerned with identifying inhibitors for aspartic proteases, which is of special interest in the design of antimalarials. Again, this appears to be a novel contribution to the specific problem of antimalarial targeting. The pharmacophores obtained from the ILP-discovered cliques were tested on a database of known inhibitors and a set of decoys. The results show that: (a) The ILP-constructed patterns can be used to identify aspartic protease inhibitors with reasonably high precision and recall; (b) The role of domain-constraints is extremely important in being able to find such patterns; and (c) The models found are unlikely to be chance patterns, based on the low hit-rate on decoys.

In this era of Big Data, it may seem a period-piecer to consider problems where data are from the interactions of 6 proteins with 3 probes. There are two points that are worth emphasizing here. First, there are many problems where “small data” are the norm [see, for example, [Hand et al. \(1994\)](#) for an entire book of such problems]. This is quite often so in the life sciences, where generating a data instance (for example, the crystal structure of the target site of a protein) involves painstaking experimental work. For the antimalarial problem considered here, for example, we are unlikely to have many more crystal structures than what is available at present. Secondly, the number of data instances can be quite misleading. Each data instance here is a complex object exhibiting a rich internal structure, that entails a very large space of possible patterns. Under such circumstances, the role of domain knowledge to rule out chance-patterns (over-fitting) is substantial, as has been clearly demonstrated here with the discovery of patterns that are able to discriminate both positives from negatives; and positives from decoys. If the importance of domain knowledge for dealing with Small Data problems of this kind is accepted, then it follows naturally that we will need discovery engines that can use such knowledge with little or no re-programming. It is our contention that ILP engines, which explicitly accept background knowledge as input are a natural choice in such cases.

There are a number of ways in which the work described here could be extended. The specific application to antimalarials can be extended usefully in two different ways. First, rather than focus on a general class of proteins like aspartic proteases, we can focus on just a specific target like plasmepsin II. Second, there is significant interest in selective pharmacophores that discriminate one set of proteins from another (plasmodial versus human, for example). Both these tasks are well within the capabilities of the kind of ILP engine used here, and provide opportunities to enrich the background knowledge further. Target-specific knowledge in the form of what is already known about plasmepsin II, for example, could constrain the search for cliques. Looking beyond the specific application presented here, it is naturally of interest to demonstrate the techniques applicability to other classes of proteins

and for other diseases. We intend to compare this against the receptor-based work on HIV-1 protease inhibitors reported in Fisher and Gner (2002).

The current implementation is a collection of programs that communicate results through the straightforward mechanism of text files. The natural next step is to integrate these into a single platform, with suitable background knowledge libraries and user-interfaces to allow the approach to be used by domain specialists with little or no ILP knowledge. On the algorithmic front, worst case results for clique-finding are not promising. Nevertheless, there may be better ways to find frequent cliques than the general-to-specific approach we have used here. This is especially the case if we want cliques that occur in all target graphs. For example, it is evident that using the least-general-generalisation (lgg) of most-specific cliques from each target, provided it is well-defined, would immediately yield a maximal clique. Difficulties may arise in computing this lgg, but there may be compact ways of representing multiple edges between a pair of vertices (using intervals of distances, rather than an edge for each distance, for example). The subgraph lattice induced by a frequent clique introduces an interesting direction to proceed when we have structures of receptors and ligands. Once a frequent clique has been obtained from the MIF-graphs of receptors, we have relied here upon chemical intuition to generalise the maximally-specific pharmacophore. If the structure of ligands are also available, then we can search the subgraph lattice to find the pharmacophores that best identify the inhibitors. Again, if this search can be guided by domain-knowledge, then we would expect ILP engines to be a reasonable choice. An altogether different approach would be to convert the entire problem of specifying constraints and finding cliques into some form of continuous optimisation problem for which there are good methods of solution. This may yield a more efficient route than ILP to solving the problem of pattern finding in MIF-graphs for de novo drug design.

Acknowledgments During some of this work, AS held a Ramanujan Fellowship from the DST, Government of India; and was at the IIIT-Delhi. He is also a Visiting Professor at the Department of Computer Science, University of Oxford; and a Professorial Fellow at the School of CSE, UNSW. RK is supported by the BINC Fellowship from the DBT, Government of India. AK acknowledges the DEITY, Government of India for fellowship support. The authors thank the Center of Excellence, Department of Biotechnology (DBT) and DST-PURSE, at the SCIS for supporting the computational work.

Appendix 1: Some useful graph-theoretic concepts

- Graph* A graph G is a tuple (V, E) in which V denotes a set of vertices and E is the set of edges between vertices. For a pair of vertices $v_{i,j}$ in V , we will say that there is an edge between v_i and v_j if either $(v_i, v_j) \in E$ or $(v_j, v_i) \in E$. (correctly, for the problem here, the vertices are labelled, and edges are weighted by the distance between the vertices).
- Connected graph* There is a *path* between a pair of vertices $v_{i,j}$ in a graph (V, E) if there is an edge between v_i and v_j ; or there an edge between v_i and some $v_k \in V$ s.t there is a path between v_k and v_j . A graph is *connected* if there a path between every pair of nodes in V . A *complete* graph is a graph in which every there is an edge between every pair of vertices $v_{i,j} \in V$.
- Subgraph* A subgraph S of $G = (V, E)$ is a graph (V', E') s.t. $V' \subseteq V$ and $E' \subseteq E$. G is also said to be a supergraph of S .
- Clique* A clique C occurs in a graph G if it is a complete subgraph of G . A *maximum clique* in a graph is the largest sized clique in the graph. A *maximal clique* in a graph G is clique in G that cannot be extended by addition of any additional

vertices in G . All maximum cliques are maximal, but the converse is only true for some kinds of graphs (it is evident that complete graphs are in this category).

Quasi-clique For $0 \leq \gamma \leq 1$ a quasi-clique $C_\gamma = (V, E)$ of G generalises the notion of a clique to that of an approximately complete subgraph of G . Specifically, if $|V| = n$, C_γ is a quasi-clique if $|E| \geq \gamma \frac{n(n-1)}{2}$. With $\gamma = 1$ all quasi-cliques are the same as cliques. With $\gamma = 1 - \frac{2}{n(n-1)}$ quasi-cliques have just one edge short of being a clique. We will call these *almost cliques*.

Frequent subgraph A subgraph is a frequent clique in the graphs G_1, G_2, \dots, G_n if it occurs in at least k of the n graphs (here, we will take $k = n$); A *frequent maximum-clique* C in a graph G , given graphs G_1, G_2, \dots, G_n is a maximum clique in G that is frequent in the graphs G_1, G_2, \dots, G_n ; A *frequent maximal-clique* C in graph G , given graphs G_1, G_2, \dots, G_n is maximal clique in G that is frequent in the graphs G_1, G_2, \dots, G_n ; and A frequent maximum-clique is also a frequent maximal-clique. For complete graphs, the converse also holds.

Downward closure If a graph is frequent then all its subgraphs are frequent. The converse need not be true. But the contrapositive follows logically, namely: if a subgraph is infrequent, then its supergraphs are infrequent.

Subgraph lattice We will restrict ourselves here to connected graphs. A lattice of connected subgraphs induced by a connected graph G is a (directed) graph L . Vertices in L are connected subgraphs of G . We will also include a special graph $\top = (\emptyset, \emptyset)$. It is evident that \top is a subgraph of any graph. The join of graphs $g_1 = (V_1, E_1)$ and $g_2 = (V_2, E_2)$ is $\text{lub}(g_1, g_2) = (V_1 \cup V_2, E_1 \cup E_2)$ if this is connected, or \top otherwise. The meet of g_1 and g_2 is $\text{glb}(g_1, g_2) = (V_1 \cap V_2, E_1 \cap E_2)$ if this is connected, or \top otherwise. Vertex g_1 is a parent of vertex g_2 in L if g_2 is a subgraph of g_1 and g_1 and g_2 differ by exactly one edge. We will call G the maximally specific graph and \top the maximally general graph. The levels of the lattice can be numbered in order of increasing generality, with Level 0 denoting G , Level 1 denoting the subgraphs obtained by dropping one edge in G , and so on. If G is a clique, then the vertices of L are quasi-cliques, and the subgraphs of G at Level 1 are almost cliques.

Appendix 2: Chemistry underlying the domain-constraints

The target proteins whose MIF grid points are considered for clique identification have a specific 3d structure and specific binding sites. The probes [hydroxyl(OH), amide(N) and carbonyl(O)] used for calculating energies at grid points have specific atomic interactions due to the different atoms involved. The vertices of the cliques are eventually translated into pharmacophoric features that have to be used for searching potential ligands in a database of small molecules. The domain-specific information used here allows us to pose some probe-specific, target-specific constraints, and generic constraints. The chemical rationale for the important ones are as follows:

Minimum distance Two atoms of a chemical cannot come closer than the sum of their van der Waal's radii (short contacts: [Wongsrichanalai et al. 2010](#)). The minimum reasonable value for this is 3 Å ([Bondi 1964](#)).

<i>Maximum distance</i>	The maximum inter-vertex distance for cliques is taken as 10 Å. This is the distance between the hydroxyl groups present in the active site of aspartic proteases and plays an important role in their catalytic activity (Silva et al. 1996; Sleebbs et al. 2014; WHO 2014).
<i>Minimum occurrences</i>	The minimum occurrence for each probe is taken as at least 1 so that every clique has all three pharmacophoric features (OH, N, O). Most of the chemicals available in searchable chemical database normally have such a composition.
<i>Energy thresholds</i>	As there were more than 100 grid values for every target's MIF per probe, there is a possibility of a million cliques. To concentrate the domain-specific acceptable cliques, the equivalent cliques were calculated in case of every probe-specific MIF for a range of energy values as cut off (eg. -3.5 to -4.5 Kcal/mol for hydroxyl probe as maximum number of cliques exist within this range). The lowest energy value (highest negative, where the interaction is strongest) at which the clique is present in all six target proteins is taken as the energy threshold for respective probe. The energy threshold values thus calculated for amide, carbonyl and hydroxyl probe are -4.5 , -2.2 and -3.0 Kcal/mol.
<i>Anchor distance</i>	The vertices within 5 Å of the anchor residue (Asp214) are selected as this is known from active site structure and enzyme function related knowledge from target proteins (human and plasmodial aspartic proteases; Silva et al. 1996).
<i>Distance tolerance</i>	A distance tolerance of 0.70 Å was taken to allow some flexibility during calculation of equivalent cliques across multiple target proteins. This value was selected by analysis of number of cliques over a range of different tolerance cut-off values for all three probes (Gil et al. 2011).
<i>Maximum size</i>	Cliques were restricted to a maximum of 8 vertices. Besides good computational reasons for imposing a bound on the clique-size, our interest is in deriving pharmacophores for screening (recall a clique with k -vertices results in a k -feature pharmacophore). Pharmacophores with 3 to 7 features are generally efficient and most commonly used in various studies of drug designing (Dame et al. 1994; Kumar and Ghosh 2007; Nagarajaram et al. 2005).

References

- Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, & C. Zaniolo (Eds.), *VLDB'94, proceedings of 20th international conference on very large data bases, September 12–15, 1994, Santiago de Chile, Chile* (pp. 487–499). Burlington: Morgan Kaufmann.
- Barnum, D., Greene, J., Smellie, A., & Sprague, P. (1996). Identification of common functional configurations among molecules. *Journal of Chemical Information and Computer Sciences*, 36(3), 563–571.
- Berry, C. (1997). New targets for antimalarial therapy: The plasmepsins, malaria parasite aspartic proteinases. *Biochemical Education*, 25, 191–194.
- Bondi, A. (1964). van der Waals volumes and radii. *The Journal of Physical Chemistry*, 68(3), 441–451.
- Bone, R. G. A., Firth, M. A., & Sykes, R. A. (1999). SMILES extensions for pattern matching and molecular transformations: Applications in chemoinformatics. *Journal of Chemical Information and Computer Sciences*, 39(5), 846–860.
- Cereto-Massaqué, A., Guasch, L., Valls, C., Mulero, M., Pujadas, G., & Garcia-Vallve, S. (2012). DecoyFinder: An easy-to-use python GUI application for building target-specific decoy sets. *Bioinformatics*, 28(12), 1661–16621.

- Cross, S., Baroni, M., Carosati, E., Benedetti, P., & Clementi, S. (2010). FLAP: GRID molecular interaction fields in virtual screening. Validation using the DUD data set. *Journal of Chemical Information and Modeling*, 50(8), 1442–1450.
- Cruciani, G. (Ed.). (2006). *Molecular interaction fields: Applications in drug discovery and ADME prediction*. Weinheim: Wiley-VCH.
- Dame, J. B., Reddy, G. R., Yowell, C. A., Dunn, B. M., Kay, J., & Berry, C. (1994). Sequence, expression and modeled structure of an aspartic proteinase from the human malaria parasite *Plasmodium falciparum*. *Molecular and Biochemical Parasitology*, 64(2), 177–190.
- Desaphy, J., Bret, G., Rognan, D., & Kellenberger, E. (2014). sc-PDB: A 3D-database of ligandable binding sites—10 years on. *Nucleic Acids Research*, 42, 928–928.
- Dixon, S. L., Smondyrev, A. M., Knoll, E. H., Rao, S. N., Shaw, D. E., & Friesner, R. A. (2006). PHASE: a new engine for pharmacophore perception, 3D QSAR model development, and 3D database screening: 1. Methodology and preliminary results. *Journal of Computer-Aided Molecular Design*, 20(10–11), 647–671.
- Finn, P., Muggleton, S., Page, D., & Srinivasan, A. (1998). Pharmacophore discovery using the inductive logic programming system Progol. *Machine Learning*, 30, 241–270.
- Fisher, L. S., & Gner, O. F. (2002). Seeking novel leads through structure-based pharmacophore design. *Journal of the Brazilian Chemical Society*, 13(6), 777–787.
- Gaulton, A., Louisa, J. B., Bento, A. P., Chambers, J., Davies, M., Hersey, A., et al. (2012). ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1), D1100–D1107.
- Gil, L. A., Valiente, P. A., Pascutti, P. G., & Pons, T. (2011). Computational perspectives into plasmepsins structure function relationship: Implications to inhibitors design. *Tropical Medicine*, 2011, 657483.
- Goodford, P. J. (1985). A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of Medicinal Chemistry*, 28, 849–857.
- Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., & Ostrowski, E. (Eds.). (1994). *A handbook of small data sets*. London: Chapman and Hall.
- Hecker, E. A., Doraiswamy, C., Andrea, T. A., & Diller, D. J. (2002). Use of catalyst pharmacophore models for screening of large combinatorial libraries. *Journal of Chemical Information and Computer Sciences*, 42(5), 1204–1211.
- Hirst, J. D., King, R. D., & Sternberg, M. J. E. (1994a). Quantitative structure-activity relationships by neural networks and inductive logic programming. II. The inhibition of dihydrofolate reductase by pyrimidines. *Journal of Computer-Aided Molecular Design*, 8, 405–420.
- Hirst, J. D., King, R. D., & Sternberg, M. J. E. (1994b). Quantitative structure-activity relationships by neural networks and inductive logic programming. II. The inhibition of dihydrofolate reductase by triazines. *Journal of Computer-Aided Molecular Design*, 8, 421–432.
- Jiang, C., Coenen, F., & Zito, M. (2013). A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28, 75–105, 3.
- Kaalia, R., Kumar, A., Srinivasan, A., & Ghosh, I. (2015). An ab initio method for designing multi-target specific pharmacophores using complementary interaction field of aspartic proteases. *Molecular Informatics*, 34(6–7), 380–393.
- King, R. D., Muggleton, S. H., & Sternberg, M. J. E. (1992). Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proceedings of the National Academy of Sciences*, 89(23), 11322–11326.
- Kumar, A., & Ghosh, I. (2007). Mapping selectivity and specificity of active site of plasmepsins from *Plasmodium falciparum* using molecular interaction field approach. *Protein & Peptide Letters*, 14, 569–674.
- Lai, L., & Chen, J. (2006). Pocket v.2: Further developments on receptor-based pharmacophore modeling. *Journal of Chemical Information and Modeling*, 46(6), 2684–2691.
- Langer, T., & Wolber, G. (2005). LigandScout: 3-D pharmacophores derived from protein-bound ligands and their use as virtual screening filters. *Journal of Chemical Information and Modeling*, 45(1), 160–169.
- Li, H., Sutter, J., & Hoffmann, R. (2000). HypoGen: An automated system for generating 3D predictive pharmacophore models. In O. F. Guner (Ed.), *Pharmacophore perception, development, and use in drug design* (Vol. 2, pp. 171–189). San Diego: International University Line.
- Marchand-Geneste, N., Watson, K. A., Alsberg, B., & King, R. D. (2002). A new approach to pharmacophore mapping and QSAR analysis using Inductive Logic Programming. Application to thermolysin inhibitors and glycogen phosphorylase B inhibitors. *Journal of Medicinal Chemistry*, 45, 399–409. (with corrections in Vol 46, pg. 653).
- Martin, Y. C. (2000). DISCO: What we did right and what we missed. In O. F. Guner (Ed.), *Pharmacophore perception, development, and use in drug design* (Vol. 2, pp. 49–68). San Diego: International University Line.

- Michael, R. G., & David, S. J. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York, NY: W. H. Freeman & Co.
- Muggleton, S. (1994). Inductive Logic Programming: Derivations, successes and shortcomings. *SIGART Bulletin*, 5(1), 5–11.
- Muggleton, S. (1995). Inverse entailment and Progol. *New Generation Computing*, 13, 245–286.
- Nagarajaram, H. A., Iengar, P., & Ghosh, I. (2005). In silico approach to antimalarial drug discovery: Design of selective inhibitor to *Plasmodium falciparum* aspartic proteases. *Frontiers in Biophysics*, 6, 168–181.
- O’Boyle, N. M., Banck, M., James, A., Morley, C., Vandermeersch, T., & Hutchison, G. R. (2011). Open Babel: An open chemical toolbox. *Journal of Chemical Information and Modeling*, 3(33), 1–14.
- Podolyan, T., & Karypis, G. (2009). Common pharmacophore identification using frequent clique detection algorithm. *Journal of Chemical Information and Computer Sciences*, 49(1), 13–21.
- Schuller, A., Fechner, U., Renner, S., Franke, L., Weber, L., & Schneider, G. (2006). A pseudo-ligand approach to virtual screening. *Combinatorial Chemistry & High Throughput Screening*, 9(5), 359–364.
- Sedwick, C. (2014). Plasmepsin V, a secret weapon against Malaria. *PLoS Biology*, 12(7), e1001898.
- Silva, A. M., Lee, A. Y., Gulnik, S. V., Maier, P., Collins, J., Bhat, T. N., et al. (1996). Structure and inhibition of plasmepsin II, a hemoglobin-degrading enzyme from *Plasmodium falciparum*. *Proceedings of the National Academy of Sciences*, 93(19), 10034–10039.
- Sleebbs, B. E., Lopaticki, S., Marapana, D. S., O’Neill, M. T., Rajasekaran, P., Gazdik, M., et al. (2014). Inhibition of Plasmepsin V activity demonstrates its essential role in protein export, PfEMP1 display, and survival of malaria parasites. *PLoS Biology*, 12(7), e1001897.
- Srinivasan, A. (1999). The Aleph Manual. <http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/>.
- Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann, E., & Willighagen, E. (2003). The Chemistry Development Kit (CDK): An open-source Java library for chemo- and bioinformatics. *Journal of Chemical Information and Computer Sciences*, 2(43), 493–500.
- Vuorinen, A., Engeli, R., Meyer, A., Bachmann, F., Griesser, U. J., Schuster, D., et al. (2014). Ligand-based pharmacophore modeling and virtual screening for the discovery of novel 17-hydroxysteroid dehydrogenase 2 inhibitors. *Journal of Medicinal Chemistry*, 57(14), 5995–6007.
- WHO (2014) Malaria Fact Sheet No. 94. <http://www.who.int/mediacentre/factsheets/fs094/en/>.
- Wongsrichanalai, C., Varma, J. K., Juliano, J. J., Kimerling, M. E., & MacArthur, J. R. (2010). Extensive drug resistance in malaria and tuberculosis. *Emerging Infection Disease*, 16(8), 1063–1067.
- Wyatt, D. M., & Berry, C. (2002). Activity and inhibition of plasmepsin IV, a new aspartic proteinase from the malaria parasite, *Plasmodium falciparum*. *FEBS Letters*, 513(2), 159–162.