Bioinformatics, 32(15), 2016, 2289–2296
doi: 10.1093/bioinformatics/btw133
Advance Access Publication Date: 11 March 2016
Original Paper

OXFORD

Structural bioinformatics

PinaColada: peptide–inhibitor ant colony ad-hoc design algorithm

Daniel Zaidman* and Haim J. Wolfson*

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel

*To whom correspondence should be addressed.

Associate Editor: Burkhard Rost

Received on August 30, 2015; revised on February 16, 2016; accepted on March 4, 2016

Abstract

Motivation: Design of protein–protein interaction (PPI) inhibitors is a major challenge in Structural Bioinformatics. Peptides, especially short ones (5–15 amino acid long), are natural candidates for inhibition of protein–protein complexes due to several attractive features such as high structural compatibility with the protein binding site (mimicking the surface of one of the proteins), small size and the ability to form strong hotspot binding connections with the protein surface. Efficient rational peptide design is still a major challenge in computer aided drug design, due to the huge space of possible sequences, which is exponential in the length of the peptide, and the high flexibility of peptide conformations.

Results: In this article we present PinaColada, a novel computational method for the design of peptide inhibitors for protein–protein interactions. We employ a version of the ant colony optimization heuristic, which is used to explore the exponential space (20^n) of length n peptide sequences, in combination with our fast robotics motivated PepCrawler algorithm, which explores the conformational space for each candidate sequence. PinaColada is being run in parallel, on a DELL PowerEdge 2.8 GHZ computer with 20 cores and 256 GB memory, and takes up to 24 h to design a peptide of 5-15 amino acids length.

Availability and implementation: An online server available at: http://bioinfo3d.cs.tau.ac.il/PinaColada/.

Contact: danielza@post.tau.ac.il; wolfson@tau.ac.il

1 Introduction

Protein–protein interactions (PPIs) play a crucial role in many cellular processes and pathways. Thus, development of small molecules that affect PPIs has become an intensive research field in Structural Bioinformatics and Computer Aided Drug Design. The task is to discover small molecules that will block a specific PPI, without harming the function (e.g. catalytic activity) of the target proteins (Arkin and Wells, 2004; Arkin and Whitty, 2009). Some of those desired inhibitors include small-molecule inhibitors and short peptide inhibitors. An interaction between two proteins usually consists of a large interface without a well characterized binding pocket (Conte *et al.*, 1999; Fletcher and Hamilton, 2006). Peptides could be good starting points for new leads in rational design of inhibitory drugs by mimicking part of the interacting surface of one of the proteins (Mochly-Rosen and Qvit, 2010; Monfregola *et al.*, 2009; Naider and Anglister, 2009; Nieddu and Pasa, 2007; Parthasarathi *et al.*, 2009). Peptides, both natural and non-natural, have been used to inhibit PPIs. Some experimental studies have investigated inhibitory peptides as competitive docking partners (Hancock *et al.*, 2013). Some studies found inhibitory peptides for various kinds of interactions, including membrane protein interactions (Caputo *et al.*, 2008), protein–DNA binding (Deng *et al.*, 2004), signaling pathways (Wang *et al.*, 2008), etc. These studies have been conducted both in vitro and in vivo. Many biological studies that try to find potent inhibitory peptides, used screening of a random peptide library to explore the potential sequence space. Such a method could be potentially improved, by exploring a pool of peptides which have already been in-silico validated, thus significantly decreasing the search space. This emphasizes the need for a rational algorithm to design a pool of in-silico validated suggestions thus reducing

© The Author 2016. Published by Oxford University Press. All rights reserved. For Permissions, please e-mail: journals.permissions@oup.com

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

significantly the search space of in-vitro peptide design. A common method to design inhibitory peptides is to extract a short linear fragment from one of the proteins in a given PPI complex (Barr *et al.*, 2002; Hashemzadeh *et al.*, 2008; Hayouka *et al.*, 2010; Laudet *et al.*, 2007; Phan *et al.*, 2010). We also use this approach for the derivation of an initial template for our design algorithm. Another validation for this approach comes from studies showing that, the interaction energy of many key processes and pathways is dominated by protein–peptide interactions (London *et al.*, 2010b). Moreover, the binding of intrinsically disordered proteins is partly or mainly peptide mediated (Babu *et al.*, 2011). However, many trials show that non-mutated peptides extracted from the interface do not achieve high enough binding affinity themselves. In some case studies randomly selected mutations improved the binding energy of linear derived peptides. In other cases, peptides were stapled either to make a cyclic peptide, or to ensure that the peptide keeps a helical shape (Yong *et al.*, 2013). Nevertheless, evidence suggests that linear derived peptides are a good target-specific starting point for peptide design (London *et al.*, 2010a).

In the recent years, there have been major developments in the related field of protein–peptide docking. Promising algorithms for peptide binding site prediction, which play a significant role in peptide design have been developed. These include algorithms for peptide binding site prediction, such as PepSite2 (Trabuco *et al.*, 2012), algorithms for local docking and refinement such as PepCrawler (Donsky and Wolfson, 2011), FlexPepDock (London *et al.*, 2011) and algorithms for ab-initio peptide docking such as FlexPepDock ab-initio (Raveh *et al.*, 2011), GalaxyDock (Lee *et al.*, 2015) and pepATTRACT (Schindler *et al.*, 2015).

There have also been several developments in knowledge based computational design of peptides. However, most of them deal with target-specific design, such as inhibitors for the well studied PDZ, SH2, SH3 domains which have simple binding sequence motifs such as PxxP, the NOTCH signaling pathway (Moellering *et al.*, 2009), MHC class II molecules (Lin *et al.*, 2008), or T-cell epitopes (Honeyman *et al.*, 1998). A comprehensive review by Vanhee *et al.* (2011) encompasses a lot of the variability in this field.

In this article, we present PinaColada – a novel computational method for inhibitory peptide design. Given an input protein–protein complex, our algorithm extracts a suitable initial peptide from one of the proteins, with high affinity to bind to the other. Then, it transforms the initial peptide through a sequence of mutations which are optimized by the ant-colony method (Dorigo and Gambardella, 1997). The goal is to find the highest affinity inhibitory peptide sequences for the given PPI. We deal with the problem of predicting the peptide conformation and free energy of the interaction with the receptor protein, by applying the PepCrawler algorithm (Donsky and Wolfson, 2011), which is a fast refinement (local docking) algorithm for peptide–protein interactions. We also let the user specify biological constraints for the resulting peptides. Such constraints may consider solubility, hydrophobicity, helicality and other more specific constraints. The output of the algorithm is a list of suggested peptide sequences to bind the target and inhibit the interaction. A webserver, which enables users to apply PinaColada has been set up at: http://bioinfo3d.cs.tau.ac.il/PinaColada/.

2 Methods

2.1 Ant-colony optimization

The optimization step of our method is based on the Ant Colony Optimization algorithm, introduced by Dorigo and Gambardella (1997), which has been applied for various tasks, including the traveling salesman problem, protein folding (Hu *et al.*, 2008; Nardelli *et al.*, 2013; Shmygelska *et al.*, 2002), classification (Martens *et al.*, 2007), etc. Ant-colony optimization is one example of the more general set of swarm algorithms.

In nature, ants try to find the shortest path to a food source by spreading pheromones. This is called stigmergic communication. Ants which choose better paths, will return faster and more pheromones will accumulate on their path. The newly arriving ants choose their path based on the pheromone levels on the ground. Thus, they soon converge to the shortest path, or a very good local minimum. The ants behavior is depicted in Figure 1.

The main idea of ant-colony optimization is having many distributed artificial 'ant' processes, which try to complete a given task in a pseudo-random fashion. It also keeps a pheromone network (weighted graph), which the ants update after their life cycle is finished. Ant-colony optimization (ACO) fits our problem very well due to its nature of combining good partial solutions to better global solutions as well as due to its inherent balance between exploration (searching the space), and exploitation (drilling down good solutions to minimize them as much as possible). This balance is explained later on. The algorithm is executed in parallel for M ants (in our experiments $M = 40$).

2.2 Pheromone networks

Generally, a pheromone network is represented by a weighted graph which the ants traverse. After an ant finished traversing a path of this graph, it deposits pheromones (by adding to the weight of the edge) to all the edges it has gone through. In our case, the peptide sequence of length n design task is represented by two pheromone networks. The first network represents the absolute positions of amino acids (a.k.a. Absolute Network) i.e. the location of a residue in the peptide sequence (see Fig. 2) with a START node and n columns of 20 (a.a. type) nodes each. Thus the peptide sequence can be represented as: $P = \{(i,k) | 1 \leq i \leq n, 1 \leq k \leq 20\}$ (i, k) is representing the choice of amino acid k in position i, regardless of the amino acid choice in previous positions.

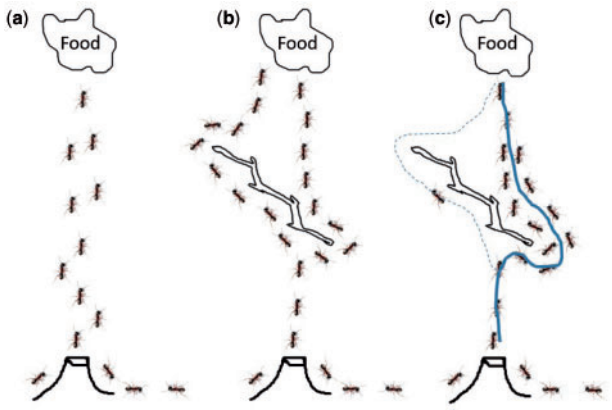

Fig. 1. Ants behavior in nature – stigmergic communication. Stigmergy is a mechanism of indirect coordination between agents or actions. The principle is that the trace left in the environment by an action stimulates the performance of a subsequent action, by the same or a different agent. In that way, subsequent actions tend to reinforce and build on each other, leading to the spontaneous emergence of coherent, apparently systematic activity. In the figure: (**a**) ants find their way from the nest to the food source; (**b**) the ants are presented with an obstacle and initially explore many directions; (**c**) after a while, the ants converge into the shortest path to the food

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

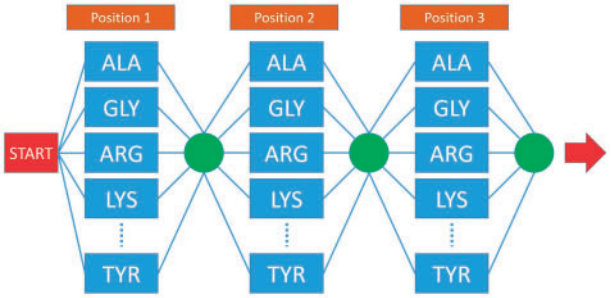

Fig. 2. Absolute network. It includes a START node and n columns of 20 nodes each (one node per amino acid, per position). This network includes a single node between 2 adjacent positions, and edges to and from this node to all the nodes in the preceding and next columns. These nodes' role is to separate the pheromone of choosing each positions and make them independent of each other (as opposed to the second network). Each column represents a position that the ant has to go through, and choose a residue for that position. When an ant gets to the n'th column, it finishes its journey. Formally: $E = \{(i,k)|1 \leq i \leq n, 1 \leq k \leq 20\}$, (i, k) is representing the transition to amino acid k, in position i, regardless of the previous positions

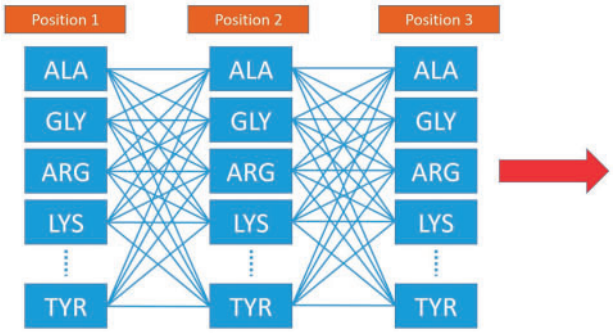

Fig. 3. Relative network. It includes n columns with 20 nodes each (like the absolute network). It also includes an edge between each node, to all 20 nodes in the next column. That way, favorable pairs of adjacent residues could be enhanced. In total, there are 400 edges between 2 adjacent positions. Formally: $E = \{(i,j,k)|2 \leq i \leq n, 1 \leq j \leq 20, 1 \leq k \leq 20\}$, (i, j, k) is representing the transition from amino acid j in position i-1 to amino acid k in position i[TQ3]

The second network represents the relative positions of pairs of adjacent residues (a.k.a. Relative Network), thus including in the optimization the cooperative effect of adjacent residues. In this second network (Fig. 3) there are n columns of 20 (a.a.type) nodes each and 400 edges between 2 adjacent positions, one for each pair of residues. Specifically,

$$E = \{(i,j,k)|2 \leq i \leq n, 1 \leq j \leq 20, 1 \leq k \leq 20\}$$

where (i, j, k) is representing the transition from amino acid j in position i - 1 to amino acid k in position i.

When an ant needs to choose its appropriate peptide sequence, it traverses a linear path on the 2 networks (graphs) simultaneously. It chooses exactly one node (amino acid) at each step of its path, until it arrives to the nth column of both networks. The choice of the next node is (probabilistically) guided by the amount of pheromone deposited on the edges of the networks. This gives the new round of ants a bias in favor of edges that 'successful' ants chose. This probability for the ants path choice is explained in detail in the next section.

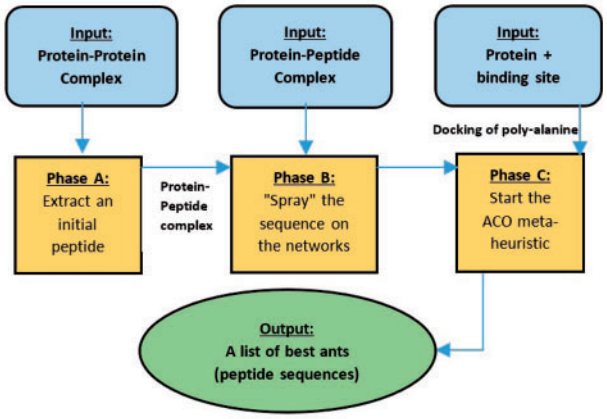

Fig. 4. PinaColada general scheme. The algorithm has 3 modes of action. (1) Protein–Peptide. In this mode, the algorithm receives as input a protein–peptide complex structure, and designs better binding peptides for the same binding site as the input peptide. (2) Protein–Protein. In this mode, it receives a protein–protein complex, extracts a linear peptide from the ligand binding site (discussed in the next section), and uses this peptide as the initial template for the design. (**3**) Protein-Binding site. In this mode, it receives only one protein and binding site information, and performs the design in an ab-initio fashion

2.3 Outline of the PinaColada Algorithm

PinaColada is an algorithm for the design of novel peptide inhibitors for protein–protein interactions. PinaColada has three modes of action.

1. Protein–Peptide. In this mode, the input is a protein–peptide complex structure, and the algorithm designs better binding peptides for the original peptide's binding site.
2. Protein–Protein. In this mode, where the input is a protein–protein complex, the algorithm extracts a linear peptide from the ligand binding site (discussed in the next section), and uses this peptide as the initial template for the design.
3. Protein-Binding site. In this mode, the input is the receptor protein, and binding site information, and the design is done in an ab-initio fashion. This mode can be used in cases when only one of the protein structures is available. First, the PatchDock algorithm (Duhovny *et al.*, 2002) is applied to dock an extended poly-alanine peptide into the receptor binding site and then the poly-alanine sequence is subsequently mutated to design the required peptide.

In all the three modes, after having an initial protein–peptide template structure, the initial peptide is mutated by applying the ant colony optimization framework. Each ant chooses a peptide sequence, which is then modeled, and its binding energy is calculated. Each ant then deposits pheromones onto the pheromone networks (explained in detail below) according to her traversed paths energetic score. The ants run in parallel, and the communication between them is through the pheromones, directing the ants towards better sequences. Finally, the k (default = 50) best ants have their sequences printed, with the corresponding energy and other parameters to form the output file of peptide sequences.

2.4 Preprocessing

2.4.1 Extracting a peptide

At the beginning of the algorithm, we trim a linear peptide from the interacting proteins (the mode of monomer + binding site is discussed later). We let the user specify the length of the desired peptide

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

(as well as biological constraints to be discussed later). Then we try all linear peptides of that length, which have at least one residue in the interface of the two proteins. For each one, we calculate the energy function (explained below in Section 2.7) of its binding to the specified protein binding site. The peptide with the lowest energy is chosen as the initial template peptide.

2.4.2 'Spray' initial pheromones

The networks are initialized with 0.1 (standard use of ant colony), to make the probability calculation valid (if not, there might be division by 0). Then, the networks are updated with the initial peptide. It means that each edge of the networks that is compatible with the initial peptide sequence gets an initial amount of pheromones. This procedure allows the ants an external guidance, in favor of discovering new and better solutions, which are partially based on the extracted peptide.

2.5 Ant life cycle

1. Choice of path.
2. Modeling the peptide.
3. Pulling the peptide away from the binding site.
4. Conformation exploration by PepCrawler.
5. Scoring and sorting of the ants.
6. Update of pheromone networks.

2.5.1 Choice of path

Each ant chooses each step of its route, according to two distributions based on pheromone levels on the edges of the two networks. These are the distributions for the ant's choice for the i'th amino acid of the peptide. For the absolute (a.a. Pr_i^a):

$$Pr_i^a(k) = \frac{(\mathrm{Net}_{ik}^a)^\alpha}{\sum_{l=1}^{20}(\mathrm{Net}_{il}^a)^\alpha}$$

where Net_{ij}^a is the pheromone value on the edge that connects to the jth node on the ith column on the absolute network. For the relative network (a.a. $Pr_{i,\mathrm{prev}=j}^r$):

$$Pr_{i,\mathrm{prev}=j}^r(k) = \frac{(\mathrm{Net}_{ijk}^r)^\beta}{\sum_{l=1}^{20}(\mathrm{Net}_{ijl}^r)^\beta}$$

High values of α and β enforce the probability of paths which are richer in pheromones. While we would like to give strong preference to such paths, we still do not want to eliminate the possibility of an ant to follow a weaker path. After testing various values of these parameters, we set α to 4, and β to 1. However, we saw that due to the highly explorative nature of the algorithm, changing these parameters (between 1 and 5) does not change the best results dramatically. Each ant chooses its path according to a mixed distribution which depends on the two distributions depicted above:

$$Pr_{i,\mathrm{prev}=j}^{\mathrm{mix}}(k) = R * Pr_i^a(k) + (1-R) * Pr_{i,\mathrm{prev}=j}^r(k)$$

R is the ratio factor between the absolute and the relative networks. After testing various values R was set to 0.8. That means we get better results when the ants' choice is much more biased towards the absolute position of the amino acids than towards the relative pairs of neighboring amino acids.

2.5.2 Modeling the peptide

After an ant chooses its path according to the pheromone derived distribution, the chosen sequence is modeled. We do that by superimposing the template amino acids, upon the peptide template backbone, while maintaining the dihedral angles of the template backbone.

2.5.3 Pulling the peptide away from the binding site

The peptide is translated from the protein, in 1.5 Å steps, until there are no steric clashes. This is done for two reasons. First, to avoid steric clashes between the protein and the peptide. Second, to redock the peptide into its binding site, using the PepCrawler local docking-refinement algorithm.

2.5.4 Conformation exploration

In this stage we apply the PepCrawler algorithm (fast robotics motivated peptide–protein local docking) to explore the conformational space of the peptide and to choose the lowest binding energy conformation. We also employ PepCrawler, to compute the energy funnel slope for the chosen conformation: the energy funnel is a dense binding-energy/RMSD plot between all tested peptide conformations, and the final predicted conformation (Zhang *et al.*, 1999). The funnel score is the slope of this plot. We use this score due to the fact that the peptide's binding-energy itself is often not enough to predict binding-affinity (Kastritis and Bonvin, 2010). This score helps evaluating the affinity of the binding solution. When this value is higher, there is a higher chance that the peptide will achieve the final predicted conformation. According to Donsky and Wolfson (2011) a funnel slope above 5 indicates a high probability to bind.

2.5.5 Scoring and sorting of the ants

The M ants are ranked according to their binding energy and (binding) funnel slope. Formally, Ant[i] is compared to Best[j], by evaluating:

$$\Delta E = \mathrm{Ant}^E[i] - \mathrm{Best}^E[j]$$

$$\Delta F = \mathrm{Ant}^F[i] - \mathrm{Best}^F[j]$$

$$R_{\max} = \min\{\Delta F * 0.05, 0.15\}$$

where $\mathrm{Ant}^E[i], \mathrm{Ant}^F[i]$ are the Energy and the funnel score of the ith ant of a round, and $\mathrm{Best}^E[j], \mathrm{Best}^F[j]$ are the Energy and the funnel score of the jth best ant overall. Then, the condition for Ant[i] to become Best[j] is: $\frac{\Delta E}{\mathrm{Best}^E[j]} < R_{\max}$ *and* ($\mathrm{Best}^F[j] \geq 5$ and $\mathrm{Ant}^F[i] < 5$)

It means that we require energetic improvement, while not allowing the funnel slope score to be worse than 5 (which is considered not likely to bind). When this happens, Best[j]'s rank goes down into Best[$j+1$]. Each ant receives a share of the global pheromone allowance based on their rank. Special pheromone bonuses are given to the best ant of a round, to the best 10 ants and to ants with a very good funnel score. The above mentioned sorting rules are heuristic and are used mainly to distinguish between ants with close energy scores, but different funnel scores.

2.5.6 Update of pheromone networks

The pheromone share an ant received is used to update the pheromone quantity of the edges it traversed. This results in a higher pheromone value on edges, which have been traversed by more ants which 'designed' better binding energy peptides. These are the

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

update rules for the absolute/relative networks appropriately: $\mathrm{Net}^a_{ik} = \mathrm{Net}^a_{ik} + \sum_{A\in(\text{ants with } k \text{ in position } i \text{ of their path})}\Delta(A)$

$$\mathrm{Net}^r_{ijk} = \mathrm{Net}^r_{ijk} + \sum_{A\in(\text{ants with } k \text{ in pos } i, \text{ and } j \text{ in pos } i-1)}\Delta(A)$$

2.6 Between rounds

2.6.1 Update of k-best ants list

After each round, we update the list of k-best ants. This is done for two reasons. First, in order to award the best scoring ants with a pheromone bonus in updating the pheromone networks. Second, the final best ants list (after the last round of ants), is the output for the user. Each ant in this final list represents one suggestion for an inhibitory peptide sequence.

2.6.2 Pheromone evaporation

As used in the original version of ant-colony optimization (Dorigo and Gambardella, 1997), evaporation also mimics real ants communication. The pheromones evaporate over time, thus decreasing a little the high pheromone levels on chosen paths (after some rounds). This step is crucial in avoiding local minima, by having the ants 'reassure' the best ways, and also allows them to investigate other directions with less pheromones, without getting stuck in few initial good paths. This is one of the reasons of the balance between exploration and exploitation mentioned above. Without the evaporation, the exploitation could take over, and we could miss many good, albeit less obvious solutions.

2.6.3 Exploration–exploitation factor

In each step of the ants path, the ant flips a biased coin before choosing the next amino acid. The coin is another feature in favor of a balance between exploration and exploitation (in addition to the pheromone evaporation discussed above). If the coin falls on exploration, the ant chooses its path according to the pheromone derived probability function mentioned above. If it falls on exploitation, it chooses the node which is connected by an edge with the highest pheromone value. Specifically,

$$\mathrm{PathExploit}(i, \mathrm{prev} = j) = \mathrm{argmax}_k\{Pr^{\mathrm{mix}}_{i,\mathrm{prev}=j}(k)\}$$

That makes the overall probability function:

$$Pr^{\mathrm{overall}}_{i,\mathrm{prev}=j}(k) = Pr(\mathrm{Explore}) * Pr^{\mathrm{mix}}_{i,\mathrm{prev}=j}(k) + Pr(\mathrm{Exploit}) * \mathrm{PathExploit}(i, \mathrm{prev} = j)$$

We chose $Pr(\mathrm{Explore}) = 0.75$, which makes $Pr(\mathrm{Exploit}) = 0.25$ for the factor initialization. After each round (but only for the first half of the rounds), this factor is updated according to the ants' result. If the best ant of the round has not surpassed the globally best ant, exploitation is increased by 0.05. If the best ant of the round is the globally best ant, exploitation is decreased by 0.05 (in favor of exploration). This is also done to ensure the exploration-exploitation balance.

2.7 Further details

2.7.1 Energy function

PepCrawler uses the binding energy function score of our previously developed FireDock algorithm (Andrusier *et al.*, 2007). It combines several energy terms: desolvation energy (ACE potential), VdW interactions (attractive, repulsive), partial electrostatics, hydrogen and disulfide bonds, π–π interactions, cation–π interactions and aliphatic interactions. For full description, we refer to equation no. 17 of the FireDock paper (Andrusier *et al.*, 2007). The input peptide undergoes large conformational changes during the PepCrawler simulation. Therefore its self energy is computed, which includes the probability of side-chains rotamers to appear according to the rotamer library, the rVdW forces between the peptide's atoms and penalties for very low Ramachandran potential torsion angles. The final complex energy of a conformation is the sum of the binding energy and the self energy of the peptide.

2.7.2 Monomer + binding site mode

The algorithm has another mode of action which is more in the ab-initio spirit. In cases where a crystal structure of the protein complex is not available PinaColada accepts just one of the proteins (monomer) and information about its binding site residues. Then, it applies PatchDock (Duhovny *et al.*, 2002) to dock an initial template of an extended poly-alanine peptide (of a specified length) to the binding site. From the top 10 results, the docking solution with the largest binding interface is chosen. From here on, we use the same outline as previously described except for updating the pheromone network with the initial peptide sequence (because we do not want the poly-alanine to affect the sequence choice).

2.7.3 Biological constraints

We let the user of the algorithm specify various kinds of biologically driven constraints. He could choose default constraints, like high chance of solubility (by limiting hydrophobicity and restricting the hydrophobic/charged residue ratio), high helix propensity and a constraint that prevents gel-making peptides. Other than that, the user can specify target-specific constraints, like conservation of some peptide positions, forcing positions to be from a pre-decided group of amino acids, keep different ratios between groups of amino acids and limit the number of other amino acid groups. The constraints then affect the ants, which could only choose peptides that comply with the rules. However, they are still choosing according to the pheromone levels on the networks edges.

2.8 Output file

The output file returned to the user includes the energy score, energy funnel slope, sequence and predicted interface size (number of residues) of all the suggested peptides. The number of results is a user defined parameter (default = 50). Those peptides could than be tested in vitro for inhibition of the PPI.

3 Results

To qualitatively assess our method, we compared it to a computational study by Smith and Kortemme (Smith and Kortemme, 2010) (later addressed as SK) as well as evaluated its ability to predict the FGDF binding motif.

Smith and Kortemme explored the sequence space of peptides recognized by PDZ domains. They used known complexes from the PDB which included a peptide (or protein) bound to a PDZ domain. Then, they extracted peptides of length 5 from these complexes. SK used the design module of the Rosetta backrub method (Lauck *et al.*, 2010) to predict additional length 5 peptide sequences which bind to the corresponding PDZ domains. Then, SK compared their results with phage display experiments which yielded a good agreement, namely the sequence profiles of peptides that bind to the same domains looked similar in many cases. We run PinaColada on nine out of the PDZ domains from that study (those that were received from X-ray crystallography). Due to the random fashion of our

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

PDZ domain	CASK-1					DLG1-2					DLG1-3				
PDB code	1KWA					2I0L					2I0I				
Phage Display	F	F	F	D	V	R	E	T	F	V	R	E	S	S	V
PinaColada	S	Y	I	E	I	R	E	T	Q	V	R	E	T	Q	V
SK	D	F	E	D	F	R	D	T	E	V	R	E	T	T	I
PDZ domain	DLG2-3					DLG4-3					DVL2-1				
PDB code	2HE2					1TP5					1L6O				
Phage Display	K	E	T	S	V	K	E	T	R	V	F	Y	G	W	F
PinaColada	H	E	T	S	V	K	E	T	W	V	L	L	D	T	V
SK	H	E	T	H	V	R	E	R	T	V	L	K	D	T	L
PDZ domain	MPDZ-10					MPDZ-12					SNTA1-1				
PDB code	2OPG					2IWP					1QAV				
Phage Display	R	I	S	D	V	F	G	T	W	V	R	E	T	R	L
PinaColada	K	V	T	F	L	S	C	I	S	V	L	E	L	T	F
SK	K	R	R	F	L	F	Y	T	W	V	R	E	T	T	F

Fig. 5. This table depicts the most frequently appearing amino acid in each position, calculated over the 100 peptides with best computational energy, as well as the most frequent appearing amino acid in the SK study and in the phage display

	PinaColada	SK
Overall number of hits:	17/45	18/45
Peptides with 4 hits:	3/9	0/9
Peptides with 3 hits:	1/9	4/9
Peptides with 2 hits:	0/9	3/9
Peptides with 1 hit:	2/9	0/9
Peptides with 0 hits:	3/9	2/9

Fig. 6. We defined a hit if our (or SK) most frequent amino acid at a certain position is the same as the most frequent amino acid in the phage display. This table consists of the overall number of hits, and also the number of hits per peptide

algorithm, we ran it five times (independently) for each peptide. We sorted the results of the 5 runs according to their computational energy. Then, we considered the most frequently appearing amino acid in each position, calculated over the 100 peptides with best computational energy. We compared it with SK's most frequent amino acids and with the phage display. In Figure 5 we can see the results of SK, PinaColada and the phage display for the nine peptides. We defined a hit if our (or SK) most frequent amino acid at a certain position is the same as the most frequent amino acid in the phage display. In Figure 6 we present the summery of total hits and also the number of hits per peptide. Remarkably, even for non-hits, we often got the same most frequent acid as SK. In 24 positions out of 45 (53.3%) we got the same amino acid (either hit or non-hit) as SK, validating a good correlation between the two methods.

PinaColada was further tested whether it could detect peptide binding motifs which are abundant in protein–peptide interactions in nature. We tested it upon a recent crystal structure of the NTF2-like domain of G3BP (human Ras GTPase SH3 Binding Protein) which binds to a peptide containing a FGDF (PHE-GLY-ASP-PHE) motif (Kristensen, 2015). We started PinaColada without any initial pheromones, which means no initial knowledge about the peptide sequence, in order to see if PinaColada would design peptides which include the motif or a part of it. Indeed, we saw that in almost all the results, the two PHE residues were placed correctly in the high

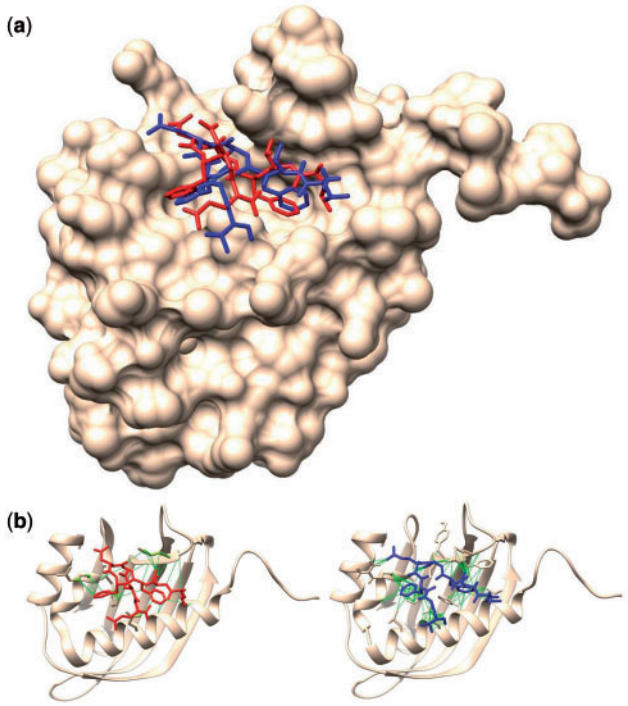

Fig. 7. (**a**) The original peptide (red) and the second ranked result of PinaColada (blue). One could see that the aromatic rings of the PHE residues of the binding motif are located very close in both models. Moreover, the rings are located in two very significant pockets of the protein interface. (**b**) On the left, in red, are the contacts (atoms in the peptide which are closer than 0.4 Å to an atom from the receptor) of the original peptide with the receptor. There are 43 contacts, out of which 23 are from the two PHE residues. On the right, in blue, are the contact for the second ranked result of PinaColada. There are 129 contacts, out of which 69 are from the two PHE residues (Color version of this figure is available at *Bioinformatics* online.)

scoring peptides, including the first ranked result. In the second position, the most common amino acid in the results was CYS which differs from GLY only by two atoms. In the second ranked result, there is GLU in the third position. According to McInerney (2015) ASP could likely be substituted by GLU and USP10 (amphibian and fish proteins) contain GLU in that position. Figure 7 depicts the native complex (from the PDB) and PinaColada's second result. One could see that the aromatic rings of the PHE residues are located very close in the two peptides. Furthermore, in the complex taken from the PDB, there are 43 contacts (atoms in the peptide which are closer than 0.4 Å to an atom from the receptor), out of which 23 are from the two PHE residues. In the first design of PinaColada there were 104 contacts, out of which 59 from the two PHE residues, and in the second design 69 PHE contacts out of 129 contacts. Moreover, the computational energy of the two best ranked designs of PinaColada were -66.93 and -68.57 as compared to -56.09 for the native one, and their funnel score were 9.20 and 8.34 as compared to 4.06 for the native one.

Figure 8 presents the behavior of the energy and funnel scores of a typical run of PinaColada. We could see that the energy reaches a minimum at round 24 out of 40, but the average energy of a round of ants, continues to decrease even thereafter. This is important because it means that even after achieving the minimum of a certain round, the ants are still improving on average, and thus the reported list of k top performing ants might still change. We could also see, that the average funnel score increases gradually as the algorithm

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

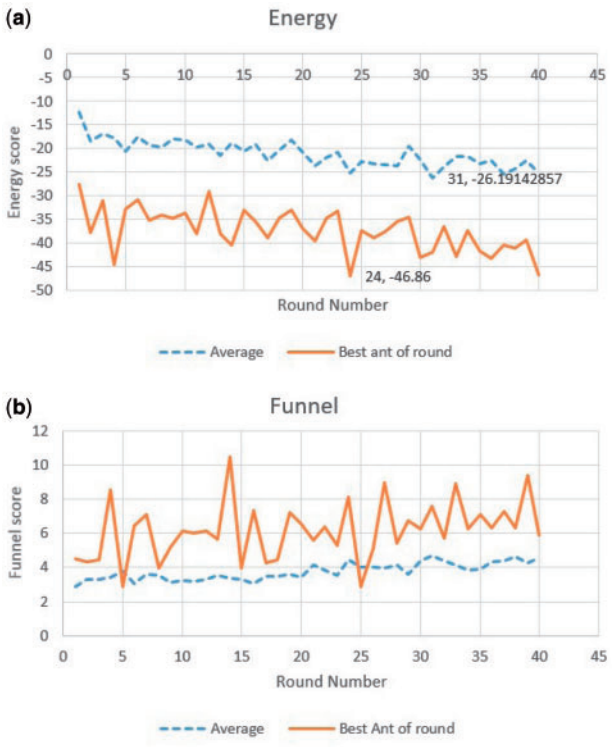

Fig. 8. (**a**) Binding-energy as a function of the round-number. (**b**) Funnel-score as a function of the round-number. In both graphs, the blue dashed line represents the average score among all the ants in a specific round, while the constant brown line represents the score of the best ant of each round. We can see that there is a strong correlation between the local maxima of the best-ant function score and the local minima of the best-ant energy score

proceeds. The funnel score of the best ant of a round fluctuates, due to the fact that we look for the energy minimum, and not necessarily the optimal funnel. Furthermore, we see that the funnel score of the best ant of a round has several local maxima (with the highest one at rounds 14, 27 and 39). Most of the highest funnel scores of best ants, correspond to local minima in the best ant energy score.

At the moment, we do not have yet in-vitro validation of the results obtained by our method. However, the energy and funnel calculation of PepCrawler were tested on the PeptiDB dataset (London *et al.*, 2010b). This dataset includes numerous examples of protein–peptide complexes from the PDB, while not containing many redundancies. This served as a validation of the funnel score as a measure of peptide binding prediction. The average funnel score of all 103 input complexes was 8.41, and only 13 complexes had a funnel score lower than 5. The experiments on the PeptiDB and a decoy benchmark have shown clear correlation between the funnel steepness score and binding affinity.

Running time: We have measured PinaColada's performance, on a DELL PowerEdge 2.8 GHZ computer with 20 cores and 256 GB memory, running the ants of each round in parallel, while the different rounds run sequentially. In our experiments, we used 35 ants in each round, for up to 40 rounds (afterward, there is no significant improvement in the results). Each round lasts between 10 and 20 min, depending on the protein and peptide size and other parameters. The whole run could take up to 24 h to design a peptide with length of between 5 and 15 amino acids. The running time could be further reduced by using several multi-core computers, and increasing the number of ants in each round, while decreasing the number of rounds.

4 Discussion and future work

As opposed to protein domains, which are often stable and folded, peptides are usually unfolded in their free state, and are very flexible (Ho and Dill, 2006). Moreover, they lose a large amount of configurational entropy upon binding (Killian *et al.*, 2009). This is one of the reasons short peptides sometimes bind more strongly than longer ones. However, in this study the energy function does not take into account entropy loss and the transitions from the completely unfolded state to a near folded solution. It would be an interesting challenge to include this kind of information in the sequence exploration task. Nonetheless, in this study, we set the length of the peptide at the beginning of the algorithm, thus comparing only peptides of the same length and avoiding the above mentioned problem.

Another issue we would like to approach is the separation of the two tasks, e.g. the exploration of the conformational space and the sequence space. In this study, we took advantage of the rapid PepCrawler method which deals only with conformational exploration, which allowed us to explore many sequences with the ACO. It could be interesting to think of a way to handle the two tasks simultaneously but still rapidly in order to increase the efficiency.

Furthermore, due to the random nature of the method, the results could be improved by starting the algorithms with several random seeds in parallel, and choosing the best solutions. This could help avoiding local minima, and be closer to the real minimum over all sequences. Also, in cases where several peptides are already known to bind the receptor, initializing the pheromone networks with some or all of them could significantly improve the guidance of the ants towards good solutions.

We also plan to check the results of PinaColada in-vitro. We are currently collaborating with several biological 'wet-labs' on the design of novel peptides and the testing of PinaColada's results in-vitro.

5 Summary and conclusions

We have presented a generic algorithm for inhibitory peptide design, while efficiently exploring both the sequence space and the conformational space. We used the ant-colony meta-heuristic approach in order to search the peptide sequence space efficiently, and to find the best peptides to inhibit a given interaction. PinaColada exploits the advantages of the fast PepCrawler method for conformational exploration, in order to test in silico many different peptide sequences in a parallel fashion. PinaColada is being run in parallel, on a DELL PowerEdge 2.8 GHZ computer with 20 cores and 256 GB memory, and takes up to 24 h for a full run, to design a peptide of 5–15 amino acids length.

To conclude, we believe that our method could be used by biologists in order to minimize the search space of inhibitory peptide. This can replace the use of a random library of peptides, which is both expensive and covers only a part of the vast sequence space (especially for longer peptides). The method could be used for a variety of different targets, as long as at least one crystal structure of the interacting proteins exists. With the new developments in the field of peptide conformational prediction, computational peptide design will start to play a more prominent role in Computer Aided Drug Design.

Acknowledgments

We thank Nir Ben-Tal, Nir London and the anonymous reviewers for their thoughtful remarks, which helped to improve this manuscript.

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

Funding

This research was supported by the Israel Science Foundation (grant No. 1112/12), the I-CORE program of the Budgeting and Planning Committee and the Israel Science Foundation (center No. 1775/12) and by the Hermann Minkowski Minerva Geometry Center.

Conflict of Interest: none declared.

References

Andrusier,N. *et al.* (2007) FireDock: Fast interaction refinement in molecular docking. *Proteins*, **69**, 139–159.

Arkin,M. and Wells,J. (2004) Small-molecule inhibitors of protein–protein interactions: progressing towards the dream. *Nat. Rev. Drug Discov.*, **3**, 301–317.

Arkin,M.R. and Whitty,A. (2009) The road less traveled: modulating signal transduction enzymes by inhibiting their protein–protein interactions. *Curr. Opin. Chem.*, **13**, 284–290.

Babu,M. *et al.* (2011) Intrinsically disordered proteins: regulation and disease. *Curr. Opin. Struct. Biol.*, **21**, 1–9.

Barr,R.K. *et al.* (2002) Identification of the critical features of a small peptide inhibitor of JNK activity. *J. Biol. Chem.*, **277**, 10987–10997.

Caputo,G.A. *et al.* (2008) Computationally designed peptide inhibitors of protein–protein interactions in membranes. *Biochemistry*, **47**, 8600–8606.

Conte,L. *et al.* (1999) The atomic structure of protein–protein recognition sites. *Mol. Biol.*, **285**, 2177–2198.

Deng,S.J. *et al.* (2004) Identification of peptides that inhibit the DNA binding, trans-activator, and DNA replication functions of the human papillomavirus type 11 E2 protein. *J. Virol.*, **78**, 2637–2641.

Donsky,E. and Wolfson,H.J. (2011) PepCrawler: a fast RRT-based algorithm for high-resolution refinement and binding affinity estimation of peptide inhibitors. *Bioinformatics*, **27**, 2836–2842.

Dorigo,M. and Gambardella,L.M. (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.*, **1**, 53–66.

Duhovny,D. *et al.* (2002) Efficient unbound docking of rigid molecules. In: Gusfield *et al.* (ed.) Proceedings of the 2'nd Workshop on Algorithms in Bioinformatics(WABI) Rome, *Italy*, Lecture Notes in Computer Science 2452, pp. 185–200.

Fletcher,S. and Hamilton,A.D. (2006) Targeting protein–protein interactions by rational design: mimicry of protein surfaces. *J. R. Soc. Interface*, **3**, 215–233.

Hancock,R. *et al.* (2013) Peptide inhibitors of the Keap1-Nrf2 protein–protein interaction with improved binding and cellular activity. *Org. Biomol. Chem.*, **11**, 3553–3557.

Hashemzadeh,M. *et al.* (2008) Chemical structures and mode of action of intravenous glycoprotein iib/iiia receptor blockers: a review. *Exp. Clin. Cardiol.*, **13**, 192–197.

Hayouka,Z. *et al.* (2010) Mechanism of action of the HIV-1 integrase inhibitory peptide LEDGF 361-370. *Biochem. Biophys. Res. Commun.*, **394**, 260–265.

Ho,B.K. and Dill,K.A. (2006) Folding very short peptides using molecular dynamics. *Plos*, **2**, e27.

Honeyman,MC. *et al.* (1998) Neural network-based prediction of candidate T-cell epitopes. *Nat. Biotechnol.*, **16**, 966–969.

Hu,X.M. *et al.* (2008) Protein folding in hydrophobic-polar lattice model: a flexible ant- colony optimization approach. *Protein Pept. Lett.*, **15**, 469–477.

Kastritis,P. and Bonvin,A. (2010) Are scoring functions in protein–protein docking ready to predict interactomes? clues from a novel binding affinity benchmark. *J. Proteome Res.*, **9**, 2216–2225.

Killian,B. *et al.* (2009) Configurational entropy in protein–peptide binding: computational study Of Tsg101 ubiquitin E2 variant domain with an HIV-derived PTAP nonapeptide. *J. Mol. Biol.*, **389**, 315–335.

Kristensen,O. (2015) Crystal structure of the G3BP2 NTF2-like domain in complex with a canonical FGDF motif peptide. *Biochem. Biophys. Res. Commun.*, **467**, 53–57.

Lauck,F. *et al.* (2010) RosettaBackrub – a web server for flexible backbone protein structure modeling and design. *Nucleic Acids Res.*, **38**, 569–575.

Laudet,B. *et al.* (2007) Structure-based design of small peptide inhibitors of protein kinase CK2 subunit interaction. *Biochem J.*, **408**, 363–373.

Lee,H. *et al.* (2015) GalaxyPepDock: a protein–peptide docking tool based on interaction similarity and energy optimization. *Nucleic Acids Res.* doi:10.1093/nar/gkv495.

Lin,H.H. *et al.* (2008) Evaluation of MHC-ii peptide binding prediction servers: applications for vaccine research. *BMC Bioinform*, **9**, (Suppl. 12), s22.

London,N. *et al.* (2010a) Can self-inhibitory peptides be derived from the interfaces of globular protein–protein interactions? *Proteins*, **78**, 3140–3149.

London,N. *et al.* (2010b) The structural basis of peptide–protein binding strategies. *Structure*, **18**, 188–199.

London,N. *et al.* (2011) Rosetta FlexPepDock web server – high resolution modeling of peptide–protein interactions. *Nucleic Acids Res.*, **39**, W249–W253. doi: 10.1093/nar/gkr431.

Martens,D. *et al.* (2007) Classification with ant colony optimization. *IEEE Trans. Evol. Comput.*, **11**, 651–665.

McInerney,G. (2015) FGDF motif regulation of stress granule formation. *DNA Cell Biol.*, **34**, 557–560.

Mochly-Rosen,D. and Qvit,N. (2010) Peptide inhibitors of protein–protein interactions: from rational design to the clinic. *Chimica Oggi*, **28**, 14–16.

Moellering,R. *et al.* (2009) Direct inhibition of the NOTCH transcription factor complex. *Nature*, **462**, 182–188.

Monfregola,L. *et al.* (2009) A SPR strategy for high-throughput ligand screenings based on synthetic peptides mimicking a selected subdomain of the target protein: a proof of concept on HER2 receptor. *Bioorg. Med. Chem.*, **17**, 7015–7020.

Naider,F. and Anglister,J. (2009) Peptides in the treatment of AIDS. *Curr. Opin. Struct. Biol.*, **19**, 473–482.

Nardelli,M. *et al.* (2013). Cross-lattice behavior of general aco folding for proteins in the hp model. In: Proc. of ACM SAC 2013, pp. 1320–1327.

Nieddu,E. and Pasa,S. (2007) Interfering with protein–protein contact: molecular interaction maps and peptide modulators. *Curr. Top. Med. Chem.*, **7**, 21–32.

Parthasarathi,L. *et al.* (2009) Approved drug mimics of short peptide ligands from protein interaction motifs. *J. Chem. Inf. Model.*, **48**, 1943–1948.

Phan,J. *et al.* (2010) Structure-based design of high affinity peptides inhibiting the interaction of p53 with MDM2 and MDMX. *J Biol Chem*, **285**, 2174–2183.

Raveh,B. *et al.* (2011) Rosetta FlexPepDock ab-initio: simultaneous folding, docking and refinement of peptides onto their receptors. *PLoS ONE*, **6**, e18934.

Schindler,C.E. *et al.* (2015) Fully blind peptide–protein docking with pepATTRACT. *Structure*, **S0969-2126**. doi:10.1016/j.str.2015.05.021.

Shmygelska,A. *et al.* (2002) An ant colony algorithm for the 2D HP protein folding problem. In: *Proceedings of the 3rd International Workshop on Ant Algorithms/ANTS 2002*, Lecture Notes in Computer Science, vol. 2463, pp. 40–52.

Smith,C.A. and Kortemme,T. (2010) Structure – based prediction of the peptide sequence space recognized by natural and synthetic PDZ domains. *J. Mol. Biol.*, **402**, 460–474.

Trabuco,LG. *et al.* (2012) Pepsite: prediction of peptide-binding sites from protein surfaces. *Nucleic Acids Res.*, **40**, W423–W427.

Vanhee,P. *et al.* (2011) Computational design of peptide ligands. *Trends Biotechnol.*, **29**, 231–239.

Wang,Y. *et al.* (2008) Identification of peptides that inhibit regulator of G protein signaling 4 function. *Pharmacology*, **82**, 97–104.

Yong,C. *et al.* (2013) Stapled helical peptide drug development: a potent dual inhibitor of MDM2 and MDMX for p53-dependent cancer therapy. *PNAS*, **110**, 3445–3454.

Zhang,C. *et al.* (1999) Protein–protein recognition: exploring the energy funnels near the binding sites. *Proteins*, **34**, 255–267.

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016