

Adaptive Peptide Design

Gisbert Schneider*, Yen-Chu Lin, Christian P. Koch, Max Pillong, Anna M. Perna, Michael Reutlinger, and Jan A. Hiss

Abstract: Computer algorithms help in the identification and optimization of peptides with desired structure and function. We provide an overview of the current focus of our research group in this field, highlighting innovative methods for peptide representation and *de novo* peptide generation. Our evolutionary molecular design cycle contains structure–activity relationship modeling by machine–learning methods, virtual peptide generation, activity prediction, peptide syntheses, as well as biophysical and biochemical activity determination. Such interplay between computer-assisted peptide generation and scoring with real laboratory experiments enables rapid feedback throughout the design cycle so that adaptive optimization can take place. Selected practical applications are reviewed including the design of new immunomodulatory MHC-I binding peptides and antimicrobial peptides.

Keywords: Antimicrobial peptide · *De novo* design · Epitope · MHC-I · Machine learning · Membrane interaction · Molecular similarity

1. Introduction

Peptides are currently experiencing a renaissance as tool compounds and lead structures in pharmaceutical research and chemical biology,^[1] specifically through combinations of computational, chemical, and biological approaches.^[2,3] From the 20 standard amino acids one can construct 20^x possible sequences of length x . While solid-phase synthesis and *in vitro* activity testing allow several thousand peptides to be analyzed at a time, exhaustive peptide libraries become prohibitively impracticable with growing peptide length. Access to up to approximately 10^{15} sequences can be achieved by phage display methods.^[4,5] This elegant experimental approach has been shown to be applicable in a broad variety of studies and deliver peptides with desired properties, but it also suffers from limitations. For example, very hydrophobic and aggregating sequences, as well as peptides that are lethal to the host bacteria used for phage production will not be found by phage display. Computer-assisted *de novo* peptide design offers a possibility to avoid such pitfalls and work

with peptides that are otherwise difficult to handle. It also offers an alternative when time and resources are limited. When a design hypothesis is available, *e.g.* a structure–activity relationship model, pharmacophore model or receptor structure, then computer-based peptide generation may be employed to efficiently find solutions for combinatorial amino acid sequence optimization. This methodology is grounded on a predictive model of peptide activity (the objective or ‘fitness’ function), and a robust optimization method for virtually moving in peptide sequence space towards regions of high fitness (Fig. 1a).^[6] Here, we present an overview of the peptide design concepts that have been developed in our group, highlighting computational algorithms and selected practical applications focusing on immunomodulatory (major histocompatibility protein I, MHC-I ligands) and antimicrobial peptide (AMP) design.

2. Peptide Evolution *in silico*

The general idea of computational peptide design is to use search algorithms to select a subset of peptides with predicted desirable properties from the pool of all peptides (sequence space) with a given length N . The number of all theoretically possible subsets p of size k is $N!/(k!(N-k)!)$. Evidently, it is impossible to perform exhaustive computational subset sampling even for small peptide pools. Without *any* prior knowledge about the structure of the macromolecular target, potential structure–activity relationships or known reference peptides that could be used as design templates, one is limited to performing some kind of random guessing. *De novo* design can sometimes provide a solution, so that new actives are found with limited effort, and unnecessary synthesize-and-test cycles are avoided. A key component is to perform an *adaptive*

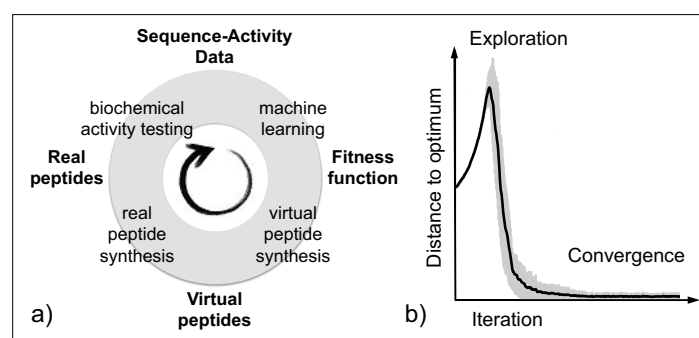


Fig. 1. a) Peptide design cycle consisting of computational model building and sequence assembly, and practical peptide synthesis and testing. Iterating through the cycle allows for swift peptide optimization. b) Typical course of a peptide optimization process. After a phase of broad sequence sampling (exploration phase) the process converges toward a local optimum (adapted from ref. [7]).

*Correspondence: Prof. Prof. adj. Dr. G. Schneider
Eidgenössische Technische Hochschule
Department of Chemistry and Applied Biosciences
Institute of Pharmaceutical Sciences
CH-8093 Zürich
Tel.: +41 44 633 74 38
E-mail: gisbert.schneider@pharma.ethz.ch

local search for candidate sequences, *i.e.* progress through stages of *exploration* and *exploitation*. Typically, such an adaptive search first examines a broad variety of different peptide sequences before homing in on local optima in the fitness landscape (Fig. 1b). Nature-inspired algorithms have repeatedly been shown to excel in this regard and provide practical – though not necessarily optimal – solutions to the task of peptide design.^[8]

2.1 Peptide Representation and Activity Prediction

According to Rechenberg,^[9] *strong causality* between the objects that are subject to design and their fitness (function, property, activity) is required for any systematic technical optimization. This notion was independently rephrased in the context of quantitative structure–activity relationships (QSAR) and molecular design as the *Chemical Similarity Principle*.^[10] Without some kind of function-related order of the molecular building blocks that are used for compound construction, one would perform a ‘random’ search for optimal products. In other words, a *context-sensitive* molecular representation is needed so that small structural variations between molecular building blocks result in only small changes of the predicted and the measured activity or property. As an illustration of this fundamental concept, we have computed a ‘similarity wheel’ for the 20 most common proteinogenic amino acids from their physicochemical properties (Fig. 2). We represented each residue by 19 principal component scores (PPCA descriptor),^[11] which we obtained by principal component analysis^[12] of 544 amino acid properties (AAindex database v9.1).^[13] Using these 19-dimensional vector representations of the 20 amino acids, we generated a self-organizing map (SOM)^[14] consisting of 20 ‘neurons’ (data clusters) arranged as a ring. The SOM algorithm performs a topology-conserving data projection so that high-dimensional data can be visually inspected. In our example, the relative similarity of the 20 amino acids in terms of their characteristic physicochemical features is mirrored in their relative ordering on the ring formed by the SOM. Neighboring residues are most similar to each other, while opposing residues are most dissimilar in terms of physicochemical properties. The ordering of amino acids shown in Fig. 2 is comparable to the original residue grouping proposed by Taylor,^[15] and related reduced alphabets that were shown to bear biological significance in sequence alignments.^[16] It is important to realize that such a general grouping does not automatically result in strong causality in every biological or chemical context. Depending on the spe-

cific system under investigation, two residues may be swapped with or without having an effect on molecular function. A prominent example is given by the positively ionizable amino acids arginine and lysine, which are neighbors on the similarity wheel. In the context of protein targeting signals, however, one may not always substitute lysine by arginine without influencing the targeting capability and func-

tion of the signal peptide.^[17] To the best of our current understanding, there is no globally applicable concept of molecular similarity.^[18,19] Lacking better knowledge, molecular similarity could thus be considered as a context-dependent property.^[20,21] We typically employ two types of peptide representations as basis for computational model building:

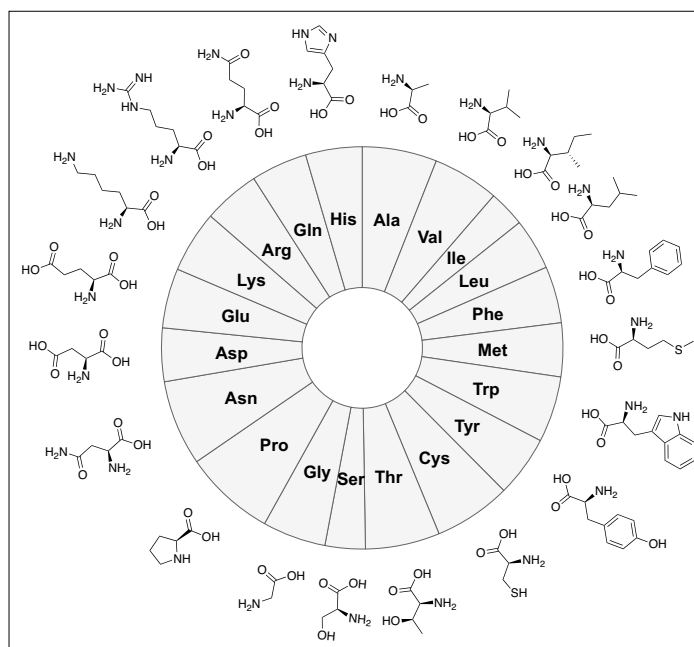


Fig. 2. Ring-forming self-organizing map (SOM) that mirrors physicochemical similarities of the 20 standard amino acids. Neighboring residues are more similar to each other than pairs of residues spaced further apart along the wheel. The size of a field is proportional to its distance to the two neighboring residues.

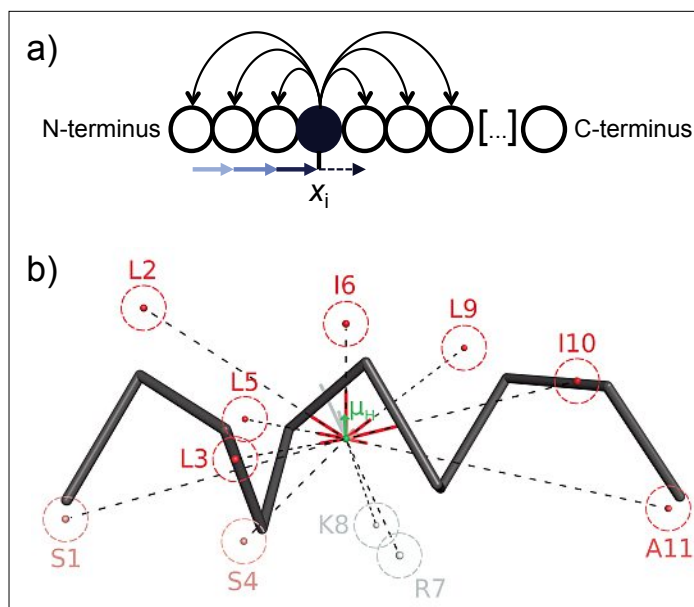


Fig. 3. Peptide representation by property cross-correlation (a), and hydrophobic moment (b). In a) a peptide sequence is shown as a string of beads. By computationally moving from the N- to the C-terminus (indicated by blue arrows), property x at residue position i is put into context with its neighboring residues up to a predefined correlation distance (here $D = 3$). In b) the computed hydrophobic moment vector μ_H (green) is shown for the helical peptide SLLSLIRKLI A featuring a partially amphiphilic distribution of residue hydrophobicity (gray to red shading indicates increasing hydrophobicity). The global hydrophobic moment vector results from summing up the individual contributions by the residue side-chains (dashed lines).

i) Global properties (e.g. total hydrophobicity, charge, residue composition), and

ii) Local properties (e.g. secondary structure, position-specific properties, residue bulkiness).

The hydrophobic moment μ_H is an example of a global conformation-sensitive descriptor (Eqn. (1), Fig. 3a).^[22] The length of this vector is the signed numerical hydrophobicity associated with the type of amino acid side chain, and its direction is defined by the orientation of the amino acid side chains. A large value of μ_H indicates that the peptide structure is amphiphilic perpendicular to its main axis.^[23] This molecular representation has been shown to be relevant for peptide bioactivity modeling, especially membrane-peptide interaction.^[24,25]

$$\mu_H = \sum_{p=1}^R s_p \cdot H_p \quad (1)$$

where H_i is the hydrophobicity value of the residue at position p , and s_p is a unit vector pointing from the alpha carbon atom of the p^{th} residue to the center of the residue's side chain. Numerous variations of this concept have been conceived over the years and have also been used as a direct design principle for amphiphilic peptide structures.

Fourier-transform and correlation-based feature representation has been shown to be particularly useful in capturing periodic property patterns and locally correlated residue positions.^[26] The concept of correlation-based feature extraction is depicted in Fig. 3a. Amino acid sequences are computationally scanned using the sliding-window technique, which places each residue position in turn into a context of the neighboring residues. The maximal correlation distance is defined by the size of the sliding window. For each peptide, a so-called *correlation vector* is computed (Eqn. (2)) that allows for straightforward sequence analysis by, e.g. similarity assessment *via* the Euclidian distance metric (Eqn. (3)).

$$C_D = \sum_{p=1}^R \sum_{q=1}^R (x_i \cdot x_j) \quad (2)$$

where D is the correlation distance under consideration, R the number residues within distance D , and x a real-valued amino acid property. The autocorrelation term $x_i \cdot x_j$ may be modified to allow for different properties to be considered (cross-correlation).

$$d(A, B) = \sqrt{\sum_{i=1}^{Dim} (A_i - B_i)^2} \quad (3)$$

where A and B are peptide descriptor vectors of dimension Dim . Using d as a similarity metric, peptides can be clustered and ranked by pairwise comparison of their descriptor vectors.

The ability for a computer program to assess peptide similarity quantitatively is the prerequisite for structure-activity models and activity prediction. Such fitness functions guide fully automated peptide design. As peptide-activity relationships are usually nonlinear, the underlying mathematical model should be able to act as a universal function approximator. Among the many modeling approaches, various types of artificial neural networks^[11] and kernel-based machine learning models^[27] like support vector machines and Gaussian process models have become popular. We often employ cascaded jury models that consist of several classifiers or quantitative predictor functions, each trained with a certain peptide representation (numerical descriptor). These first-stage models are connected to feed a second-stage model producing the final score or fitness value (Fig. 4). This way, errors made by individual first stage models are alleviated and the jury prediction often turns out to be more robust with sustained prediction accuracy.^[28,29]

2.2 Adaptive Optimization

The umbrella term *adaptive optimization* stands for a whole class of computational optimization methods that contain an element of dynamic program recompilation or just-in-time parameter adjustment.^[30,31] We have investigated various nature-inspired optimization algorithms like the *Evolution Strategy*,^[32] *Genetic Algorithms*,^[33] *Ant Algorithms*,^[34] *Particle*

Swarm Algorithms,^[35,36] and *Genetic Programming*^[37] for their ability to find novel peptides with desired properties from scratch.^[38] According to Koza^[37] artificial adaptive systems require several essential elements, specifically

- Structures that undergo adaptation,
- Initial structures (starting solutions),
- Fitness measure that evaluates the structures,
- Operations to modify the structures,
- State (memory) of the system at each stage,
- Method for designating a result,
- Method for terminating the process, and
- Parameters that control the process.

The most relevant aspect of adaptive peptide optimization is the so-called 'memory' because by keeping track of the past progress during an optimization process and learning from previous experience helps the computer algorithm make informed decisions for the next step(s) of the search. In other words, memory turns a blind or random search into rational molecular design. There are many ways to implement memory in a peptide design program, for example in terms of peptide library diversity. This idea has been exemplarily realized in the *Evolution Strategy*: At the beginning of the search process, a population of peptides is initialized with random sequences. Each peptide is evaluated by the fitness function, where fitter peptides will have a higher probability to be selected as parents of the next generation and generate offspring (sequence variations). Genetic operators like sequence mutation and crossover create variations of winning peptides (parents). Such a simulated evolutionary process stops either after a fixed number of generations or when a given fitness threshold is reached. Each new population consists of variations of the fittest individuals of

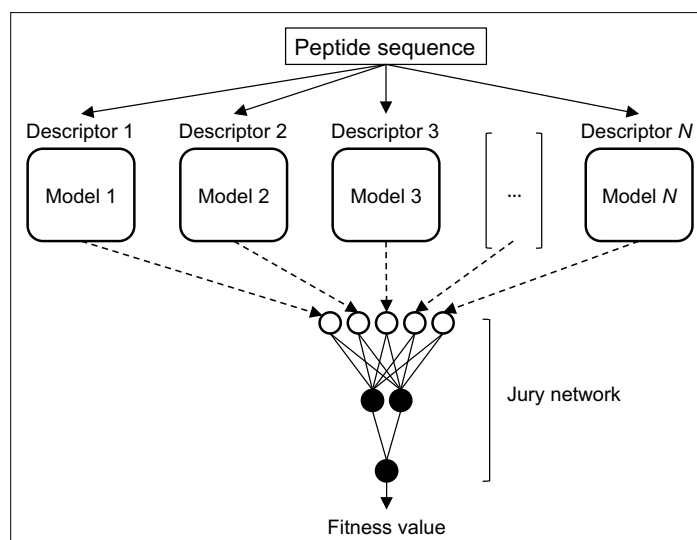


Fig. 4. Schematic of a cascaded jury model for peptide activity prediction.

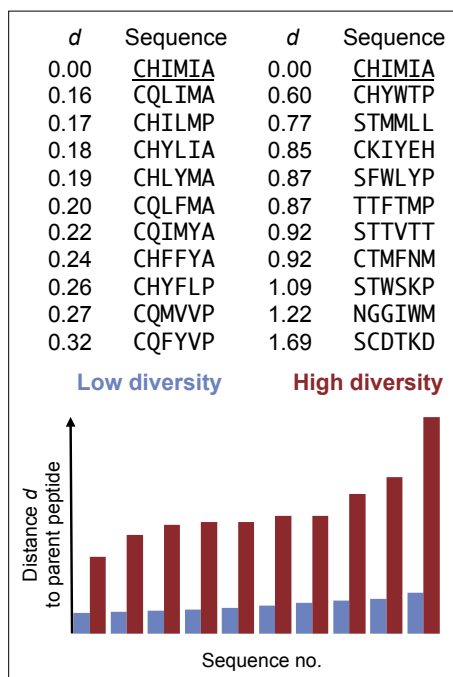


Fig. 5. Simulated evolution of the seed peptide CHIMIA by applying a Gaussian randomization model. d is the Euclidian distance (Eqn. (3)) to the seed peptide. Residues are mutated based on physicochemical similarity. The series of ten peptides shown on the left was generated with low diversity (blue), the right series with high diversity (red), *i.e.* wider spread around the seed peptide. Diversity is an adaptive parameter that automatically adjusts to the local requirements of the fitness landscape.^[40]

the previous generation, so that over time the process adapts to the underlying fitness function and produces solutions to the optimization problem. The adaptive parameter is the diversity of the population in terms of sequence variation.^[39] Fig. 5 presents an example of a virtual evolution of the parent sequence CHIMIA over one generation of offspring. By pursu-

ing this concept of adaptive library design the number of synthesize-and-test cycles can be kept minimal.^[41]

3. Applications

3.1 Antimicrobial Peptides

One of our main projects is to study peptide–membrane interaction. We distinguish between lipid membrane targeting, transition/crossing, and disruption by peptides.^[3] Representatives for each class are signal peptides, cell-penetrating peptides, and antimicrobial peptides (AMPs). We strive for a profound understanding of both the mutual and the discriminative structural features responsible for these delicately tuned functions.^[42] A first step towards the rational design of peptides with the desired membrane activity is to find a suitable molecular representation that allows for unsupervised clustering into the respective activity classes.^[43] One such approach is the SOM, which allows for visual inspection of the distribution of peptides in sequence space. Fig. 6a shows such a nonlinear map, which we obtained for AMPs and bacterial signal peptides. In Fig. 6b, another model of the fitness landscape is presented, based on the same data. Apparently, the descriptor used (here: correlated physicochemical properties) is meaningful for function-related sequence clustering. AMPs are clearly separated from signal peptides. Once such a map is available, one can use it for peptide design by projecting novel sequences onto the map and picking the ones located on favorable activity islands.^[46] The sequence RVKVATTLIIV_{NH2} represents an example of a *de novo* generated AMP peptide, which we identified using the SOM approach, synthesized and tested successfully. At a concentration of 100 μ M this computationally found sequence displayed strong lytic effects on large

unilamellar vesicles with a lipid composition that mimics bacterial membranes (POPE:POPG = 7:3; not shown), and inhibited the growth of *Escherichia coli* (*E. coli*) and *Staphylococcus aureus* (*S. aureus*) strains *in vitro* (Fig. 7).

3.2 MHC-I Binding Peptides

We trained a cascaded jury model on the recognition of peptides that bind to MHC-I allomorph H-2K^b, and employed this prediction system as artificial fitness landscape guiding the search for novel H-2K^b ligands.^[47] For sequence space exploration we relied on an ant colony optimization algorithm, which we adapted to work on peptides. The idea was to have artificial ants (search agents) construct new peptide sequences (octamers) by following a pheromone trail through sequence space. The principle of pheromones was realized by *pseudo*-probabilities for picking up certain amino acid residues for chain elongation (Fig. 8). Overall, the ant algorithm turned out to be excellently suited for combinatorial peptide assembly with hit rates of up to 95% and sustained stabilization potential of MHC-I molecules on the surface of TAP-deficient RMA-S cells. Systematic investigation of the minimal length of peptide ligands revealed a length of five residues for the H-2K^b allomorph, which we confirmed in a newly established thermal denaturation assay.^[29] Computational analysis of viral proteomes

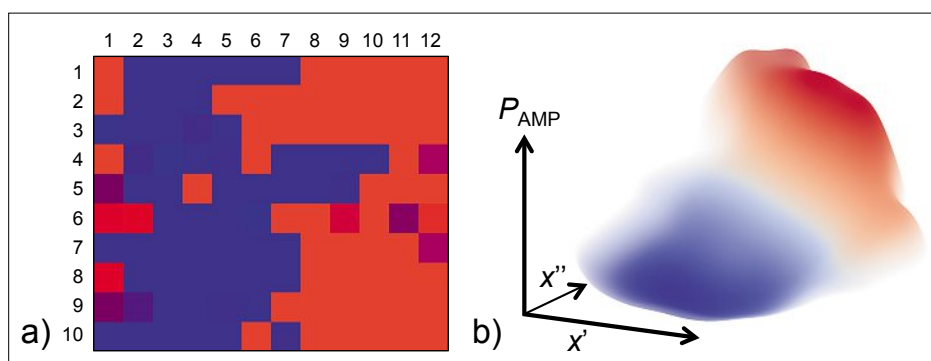


Fig. 6. Fitness landscape models. a) Self-organizing map (SOM) showing the distribution of antimicrobial peptides (AMPs, 'activity island' in red) and bacterial signal peptides (blue). b) Landscape of the same data shown in a), but projected by stochastic neighbor embedding.^[44] We used our LiSARD tool for visualization.^[45] The model computes *pseudo*-probabilities of finding an antimicrobial peptide (AMP) in sequence space. Apparently both landscape models capture important features that distinguish membrane-disrupting AMPs from membrane-targeting signal peptides. Amino acid sequences were encoded by cross-correlated physicochemical properties using a sliding-window of seven residues (maximal correlation distance $D = 3$ residue positions).

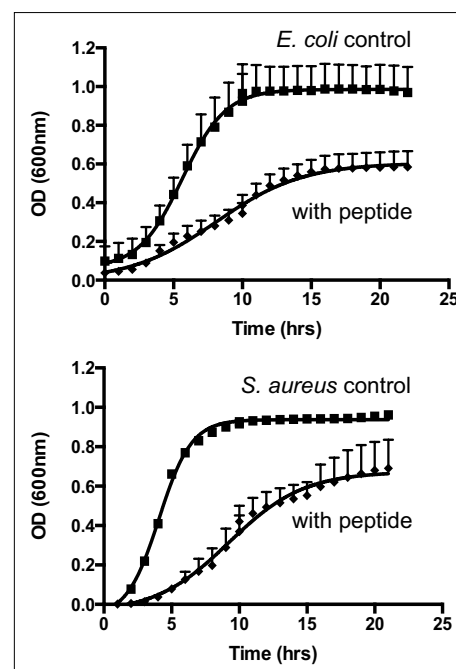


Fig. 7. Growth inhibition of *E. coli* and *S. aureus* over time by a *de novo* designed antimicrobial peptide. GFP-transformed bacteria (9×10^6 cells) were incubated with 100 μ M peptide, and growth was monitored by detecting the intensity of the fluorescence signal at $\lambda = 600$ nm over time.

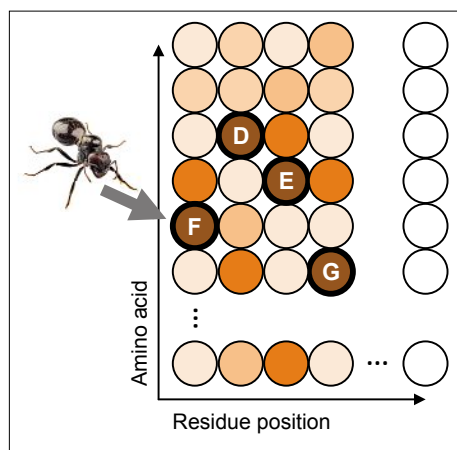


Fig. 8. Principle of ant colony optimization of amino acid sequences. The concept of optimal path finding by some ant species has been adapted to finding optimal peptides in sequence space. The artificial ant assembles new peptides by collecting amino acid residues, starting at residue position 1 (N-terminus) and proceeding through the sequence. The selection is based on a probabilistic sampling scheme that mimics pheromone trails. These pseudo-probabilities are computed by the fitness function employed. In the figure color intensity corresponds to pheromone density. The current peptide sequence is FDEG.

led to the identification of vesicular stomatitis virus derived peptides that induced CD8⁺ T-cell proliferation after viral infection of mice.^[48] These studies demonstrate that advanced machine-learning models in combination with adaptive peptide design algorithms provide a unique access to rationally designed peptides with desired properties and a promising approach towards ‘reverse vaccinology’.^[49]

4. Conclusions and Outlook

Computational peptide design that implements the design cycle shown in Fig. 1a has resulted in several new peptides that exhibit the desired bioactivity. The key question remains how to represent peptides so that meaningful structure–activity relationship models can be deduced from the available data. We have mainly considered static peptide models and sequence patterns as input to machine learning models. However, peptide dynamics and consequently representations that explicitly take into account molecular flexibility, conformational preferences, and environment-sensitive descriptors will likely be needed to make a significant step forward in our modeling capabilities. There is no doubt that laboratory experiments are essential to provide feedback and improve a modeled fitness function. We expect that a tight combination of computational and practical experiments will lead to further

improvements. For example, fully automated robotic peptide synthesis coupled to inline analytics and subsequent microfluidic assay systems might close the design cycle so that computational predictions can be realized and optimized without manual intervention.

Acknowledgements

This study was supported by ETH Zürich, the Swiss National Science Foundation (SNF, grant no. 205321-134783), and the OPO Foundation Zürich. The bacterial growth data presented in Fig. 7 were measured by Peter Thelesklaf and Prof. Dr. Silja Wessler (Paris-Lodron University, Salzburg, Austria). We thank Sarah Haller for technical support.

Received: September 8, 2013

- [1] P. Vlieghe, V. Lisowski, J. Martinez, M. Khrestchatskiy, *Drug Discov. Today* **2010**, *15*, 40.
- [2] J. Audie, C. Boyd, *Curr. Pharm. Des.* **2010**, *16*, 567.
- [3] C. D. Fjell, J. A. Hiss, R. E. W. Hancock, G. Schneider, *Nat. Rev. Drug Discov.* **2011**, *11*, 37.
- [4] I. Rentero, C. Heinis, *Chimia* **2011**, *65*, 843.
- [5] J. Pande, M. M. Szewczyk, A. K. Grover, *Biotechnol. Adv.* **2010**, *28*, 849.
- [6] G. Schneider, W. Schrödl, G. Wallukat, J. Müller, E. Nissen, W. Rönspack, P. Wrede, R. Kunze, *Proc. Natl. Acad. Sci. USA* **1998**, *95*, 12179.
- [7] N. Jäger, J. M. Wisniewskaja, J. A. Hiss, F. O. Losch, P. Walden, P. Wrede, G. Schneider, *Mol. Inf.* **2010**, *29*, 65.
- [8] a) S. Bandholtz, J. Wichard, R. Kühne, C. Grötzingler, *PLoS One* **2012**, *7*, e36948; b) C. D. Fjell, H. Janssen, W. A. Cheung, R. E. Hancock, A. Cherkasov, *Chem. Biol. Drug Des.* **2011**, *77*, 48; c) K. Abe, N. Kobayashi, K. Sode, K. Ikebukuro, *BMC Bioinformatics* **2007**, *8*, 451; d) J. Ziegler, S. Schwarzinger, *J. Comput. Aided Mol. Des.* **2006**, *20*, 47; e) W. Zhang, M. G. Loughran, S. Kanna, K. Yano, K. Ikebukuro, Y. Yokobayashi, R. Kuroda, I. Karube, *Proteins* **2003**, *53*, 193; f) P. Wrede, O. Landt, S. Klages, A. Fatemi, U. Hahn, G. Schneider, *Biochemistry* **1998**, *37*, 3588.
- [9] I. Rechenberg, ‘Optimierung technischer Systeme nach Prinzipien der biologischen Evolution’, Frommann-Holzboog, Stuttgart, **1973**.
- [10] Concepts and Applications of Molecular Similarity, Eds. M. A. Johnson, G. M. Maggiora, Wiley, New York, **1990**.
- [11] G. Schneider, P. Wrede, *Prog. Biophys. Mol. Biol.* **1998**, *70*, 175.
- [12] H. Abdi, L. J. Williams, *WIREs Comput. Stat.* **2010**, *2*, 433.
- [13] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, M. Kanehisa, *Nucleic Acids Res.* **2008**, *36*, 202.
- [14] T. Kohonen, *Biol. Cybern.* **1982**, *43*, 59.
- [15] W. R. Taylor, *J. Theor. Biol.* **1986**, *119*, 205.
- [16] E. L. Peterson, J. Kondev, J. A. Theriot, R. Phillips, *Bioinformatics* **2009**, *25*, 1356.
- [17] R. G. Summers, C. R. Harris, J. R. Knowles, *J. Biol. Chem.* **1989**, *264*, 20082.
- [18] H. Eckert, J. Bajorath, *Drug. Discov. Today* **2007**, *12*, 225.
- [19] J. D. Holliday, E. Kanoulas, N. Malim, P. Willett, *J. Cheminform.* **2011**, *3*, 1.
- [20] G. Schneider, M. Hartenfeller, M. Reutlinger, Y. Tanrikulu, E. Proschak, P. Schneider, *Trends Biotechnol.* **2009**, *27*, 18.
- [21] G. Schneider, S. S. So, ‘Adaptive Systems in Drug Design’, Landes Bioscience, Georgetown, **2003**.
- [22] D. Eisenberg, R. M. Weiss, T. C. Terwilliger, *Nature* **1982**, *299*, 371.
- [23] D. A. Phoenix, F. Harris, O. A. Daman, J. Wallace, *Curr. Protein Pept. Sci.* **2002**, *3*, 201.
- [24] H. D. Thaker, A. Cankaya, R. W. Scott, G. N. Tew, *ACS Med. Chem. Lett.* **2013**, *4*, 481.
- [25] R. W. Scott, W. F. DeGrado, G. N. Tew, *Curr. Opin. Biotechnol.* **2008**, *19*, 620.
- [26] D. D. Jones, *J. Theor. Biol.* **1975**, *50*, 167.
- [27] T. Diethe, M. Girolami, *Neural Comput.* **2013**, *25*, 567.
- [28] A. Givehchi, G. Schneider, *Mol. Divers.* **2005**, *9*, 371.
- [29] C. P. Koch, A. M. Perna, M. Pillong, N. K. Todoroff, P. Wrede, G. Folkers, J. A. Hiss, G. Schneider, *PLoS Comput. Biol.* **2013**, *9*, e1003088.
- [30] J. H. Holland, ‘Adaptation in Natural and Artificial Systems’, University of Michigan Press, Michigan, **1975**.
- [31] A. Schüller, G. Schneider, *J. Chem. Inf. Model.* **2008**, *48*, 1473.
- [32] H. G. Beyer, H. P. Schwefel, *Natural Computing* **2002**, *1*, 3.
- [33] D. E. Goldberg, ‘Genetic Algorithms in Search, Optimization, and Machine Learning’, Addison-Wesley, Reading, **1989**.
- [34] M. Dorigo, T. Stützle, ‘Ant Colony Optimization’, MIT Press, Cambridge, **2004**.
- [35] J. Kennedy, R. Eberhart, *Proc. IEEE Int. Conf. Neural Netw.* **1995**, *4*, 1942.
- [36] M. Hartenfeller, E. Proschak, A. Schüller, G. Schneider, *Chem. Biol. Drug Des.* **2008**, *72*, 16.
- [37] J. R. Koza, ‘Genetic Programming’, MIT Press, Cambridge, **1992**.
- [38] J. A. Hiss, M. Hartenfeller, G. Schneider, *Curr. Pharm. Des.* **2010**, *16*, 1656.
- [39] G. Schneider, P. Wrede, *Biophys. J.* **1994**, *66*, 335.
- [40] a) G. Schneider, J. Schuchhardt, P. Wrede, *Comput. Appl. Biosci.* **1994**, *10*, 635; b) G. Schneider, J. Schuchhardt, P. Wrede, *Biol. Cybern.* **1995**, *73*, 245.
- [41] G. Schneider, A. Schüller, *Methods Mol. Biol.* **2009**, *572*, 135.
- [42] N. Papat, Y. Shai, *Peptides* **2003**, *24*, 1693.
- [43] M. Reutlinger, G. Schneider, *J. Mol. Graphics Modell.* **2012**, *34*, 108.
- [44] G. Hinton, S. Roweis, *Advances in Neural Information Processing Systems (NIPS)* **2002**, *15*, 857.
- [45] M. Reutlinger, W. Guba, R. E. Martin, A. I. Alanine, T. Hoffmann, A. Klenner, J. A. Hiss, P. Schneider, G. Schneider, *Angew. Chem. Int. Ed.* **2011**, *50*, 11633.
- [46] P. Schneider, Y. Tanrikulu, G. Schneider, *Curr. Med. Chem.* **2009**, *16*, 258.
- [47] a) J. A. Hiss, A. Bredenbeck, F. O. Losch, P. Wrede, P. Walden, G. Schneider, *Protein Eng. Des. Sel.* **2007**, *20*, 99; b) J. M. Wisniewska, N. Jäger, A. Freier, F. O. Losch, K. H. Wiesmüller, P. Walden, P. Wrede, G. Schneider, J. A. Hiss, *J. Biomed. Biotechnol.* **2010**, 396847.
- [48] C. P. Koch, A. M. Perna, S. Weissmüller, S. Bauer, M. Pillong, R. B. Baleeiro, M. Reutlinger, G. Folkers, P. Walden, P. Wrede, J. A. Hiss, Z. Waibler, G. Schneider, *ACS Chem. Biol.* **2013**, *8*, 1876.
- [49] R. Rappuoli, *Vaccine* **2001**, *19*, 2688.