Computer Assisted Peptide Design and Optimization with Topology Preserving Neural Networks

Jörg D. Wichard[1], Sebastian Bandholtz[2], Carsten Grötzinger[2] and Ronald Kühne[1]

[1] FMP Berlin, Robert-Rössle-Str. 10, D-13125 Berlin, Germany,
wichard@fmp-berlin.de,
WWW home page: http://www.fmp-berlin.de
[2] Charité, Department of Hepatology and Gastroenterology,
Augustenburger Platz 1, D-13353 Berlin, Germany,
WWW home page: http://www.charite.de

Abstract. We propose a non-standard neural network called TPNN which offers the direct mapping from a peptide sequence to a property of interest in order to model the quantitative structure activity relation. The peptide sequence serves as a template for the network topology. The building blocks of the network are single cells which correspond one-to-one to the amino acids of the peptide. The network training is based on gradient descent techniques, which rely on the efficient calculation of the gradient by back-propagation. The TPNN together with a GA-based exploration of the combinatorial peptide space is a new method for peptide design and optimization. We demonstrate the feasibility of this method in the drug discovery process.

1 Introduction

An important task in modern drug discovery is to understand the quantitative structure activity relation (QSAR). QSAR problems can be divided into a coding and learning part. The learning part could be solved with standard machine learning tools. Artificial neural networks are commonly used in this context as nonlinear regression models that correlate the biological activities with the physiochemical or structural properties of those chemical compounds that were tested in a specific assay.
The most important part in QSAR analysis is the identification of molecular descriptors which encode the essential properties of the compounds under investigation. Alternative approaches of the classical machine-learning-based QSAR circumvent the problem of computing and selecting a representative set of molecular descriptors. Therefore the molecules are considered as structured data - represented as graphs - wherein each atom is a node and each bond is an edge. This is the main concept of the *Molecular Graph Network* [1–3] and of the *Graph Machines* [4] which translate a chemical structure into a graph that works as topology-template for the connections of a neural network.
In this work, we follow the idea of translating the chemical structure of a compound directly into the topology of a learning machine. Our strategy is focused on peptides which are chains of amino acids. Each cell in the network corresponds one-to-one to an amino acid in the peptide. Hence the amino acid sequence of a peptide determines the topology of the network. We call this architecture *Topology Preserving Neural Network*

(TPNN) and we propose a learning strategy that adapts the weights of the cells with respect to the the assay data. The adapted cells are used to build models for the QSAR of new *virtual* peptides in order to optimize the desired property *in silico*. We explore the high dimensional space of all possible peptides with a genetic algorithm wherein the output of TPNN-model defines the fitness function. Only the top ranking *in silico* peptides are selected for synthesis and *in vitro* testing in the assay.
The fully connected TPNN is described in the next section and the training and regularization follows in section 3. The use of TPNN models in peptide design is reported in section 4 and first results are presented in section 5.

2 Network Representation of Peptides

Peptides are short linear polymers built from amino acids that are linked with an amide bond. The 20 proteinogenic amino acids are the most important ones and they are the building blocks of almost all proteins in nature. The string representation of the peptide $\boldsymbol{S}$ is called the *peptide sequence* and it is given by the order in which the amino acid lie in the chain. We further assume that the peptides are composed of amino acids from a pool of M different individuals called the *alphabet*. In a TPNN each amino acid from the alphabet is represented as a particular cell with individual weights that are adjusted during the network training. The internal weight of the cell is φ, the inputs from the neighboring cells are connected with the weights $\omega_{-N,\dots,N}$ and the feedback is controlled by ω_0. The weights are combined in the weight vector $\boldsymbol{\omega}$. The cells are connected to form a chain as shown in figure 1 with an one-to-one correspondence between the amino acids in the alphabet that build the peptides and the cells in the network. The TPNN is iterated several times which governs the system dynamics. The internal states of the network are denoted in the state vector $\boldsymbol{y}$ with respect to their order. The state of the i-th TPNN cell y_i^t evolves for iterations $t = 0, \dots, T-1$ according to

$$\begin{aligned} x_i^{t+1} &= \varphi_i + \boldsymbol{y}^t \boldsymbol{\omega_i} \\ y_i^{t+1} &= \sigma(x_i^{t+1}), \end{aligned} \tag{1}$$

wherein the activation function $\sigma(x)$ is a hyperbolic tangent with an additional linear term that leads to non-vanishing derivatives of the training error

$$\sigma(x) = \tanh(x) + \lambda x \quad \text{with } \lambda \ll 1. \tag{2}$$

The number of iterations T is set to be the average length of the sequences under investigation. The final output of the network is simply the sum over all internal states y_i^T after the final iteration.

3 Training TPNN Models

The training procedure of a TPNN is a combination of stochastic gradient descend and back propagation with several improvements that make the training of the shared weights feasible. The true gradient is approximated by the gradient of the loss function

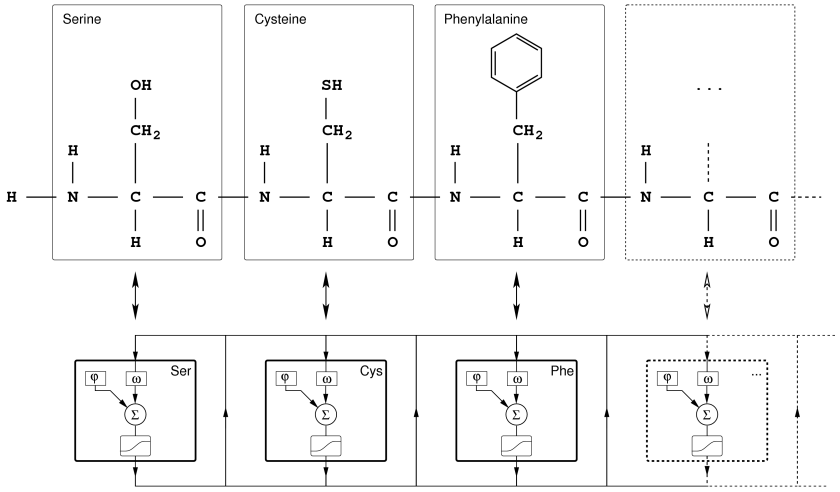

Fig. 1. This is an example of the translation process from a peptide starting with the amino acids Ser-Cys-Phe into a fully connected TPNN. Note the one-to-one correspondence between the amino acids of the peptide and the elementary cells of the TPNN.

which is evaluated on single training samples. The network weights are adjusted by an amount proportional to this approximate gradient. A training sample consists of two parts: The first part is the peptide sequence $\boldsymbol{S}$ that is a composition of the M possible amino acids taken from the alphabet. The second part is the measured activity that could be a continuous value or a class label, for example the classes *active*, *weak-active* or *non-active*. Let's assume that we have a collection of K training samples $\{\boldsymbol{S}_n, a_n\}_{n=1,\ldots,K}$ and the weights $\boldsymbol{\omega}^j = (\varphi^j, \omega^j_{-N}, \ldots, \omega^j_N)_{j=1,\ldots,M}$ of the individual cells that correspond to the M different amino acids in the alphabet are organized in the weight vector $\Omega = (\boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^M)$. Let $f(\boldsymbol{S}_i, \Omega)$ denote the output of the TPNN for a given sequence $\boldsymbol{S}_i$ with respect to the network weights Ω. This output value has to be compared to the training label a_i by means of a *loss function*. The loss function measures the deviation of the TPNN output from the desired value a_i. In optimization usually a quadratic loss function is used, basically due to the simplicity of the resulting derivatives. We propose the use of an ϵ–insensitive squared loss function

$$\lambda_\epsilon(\xi) := \begin{cases} 0 & : \xi \leq \epsilon \\ (\xi - \epsilon)^2 & : \xi > \epsilon. \end{cases} \tag{3}$$

The output of the TPNN has zero loss and gradient if it lies inside the ϵ–margin of the desired output. This forces the training algorithm to focus on the training samples that are not properly explained by the current model rather than adjusting the network weights by gradient steps of already correctly learned samples. We choose an ϵ that is close to the mean of the single measurement variances.

Thus the training error $E(\Omega, \epsilon)$ is simply the loss averaged over the entire training set

$$E(\Omega, \epsilon) := \sum_{i=1}^{K} \lambda_\epsilon(a_i - f(\boldsymbol{S}_i, \Omega)). \tag{4}$$

Furthermore we need regularization in order to prevent overfitting. Weight decay is a regularization method that penalizes large weights in the network and forces the insignificant weights to converge to zero. This results in a model with a minimum number of free parameters, according to the principle of Occam's razor also known as the law of parsimony. It tells us to prefer the simplest of all equally good models. The weight decay penalty term is defined as

$$P(\Omega) = \gamma \sum_{i=1}^{N} \frac{\omega_i^2}{1 + \omega_i^2}, \tag{5}$$

where Ω denotes the weight vector of the TPNN and the regularization parameter $\gamma = 0.001$ is small. The penalty term is added to the training error and contributes to the gradient.

3.1 Stochastic Gradient Descent

Training a learning machine is put to effect by minimizing the training error as defined in equ. 4 with respect to the network weights Ω. The method of training a TPNN is based on *stochastic gradient descend*. The gradient of the entire training error from equ. 4 and the penalty term from equ. 5 is a sum of terms of the form

$$\frac{\partial}{\partial \Omega} \left[\lambda_\epsilon(a_i - f(\boldsymbol{S}_i, \Omega)) + P(\Omega) \right]. \tag{6}$$

The stochastic gradient descent performs a series of very small consecutive steps, determining each step direction from the gradient of an individual term only. After each step, the new parameter set Ω is re–inserted into the loss function before the next gradient is computed. This defines an update rule for the parameters of the form

$$\Omega_i = \Omega_{i-1} - \delta\Omega_i, \tag{7}$$

with $i = 1 \ldots K$, wherein K is the number of training samples. The update $\delta\Omega_i$ depends on the i-th training sample only and is given by

$$\delta\Omega_i = \mu \frac{\partial}{\partial \Omega} \left[\lambda_\epsilon(a_i - f(\boldsymbol{S}_i, \Omega_{i-1})) + P(\Omega_{i-1}) \right]. \tag{8}$$

We calculate the update $\delta\Omega_i$ with the standard error back-propagation technique as it is used for the common feed-forward multilayer perceptron [5].
The parameter μ controls the stepsize of the gradient descend. The initial step size is already small (around $\mu = 0.01$) and it is decreased after each training epoch with a constant factor. This is necessary to achieve a slow convergence of the weights. Note that in each training step only a few selected values of the entire weight vector Ω are adjusted, namely the ones that correspond to amino acids that appear in the sequence of the training sample.

3.2 Building TPNN-Ensembles

A common way to improve the performance of neural networks in regression or classification tasks is ensemble building [6]. It is well known, that neural network ensembles perform better in terms of generalization than single models would do [7, 8]. An ensemble of TPNNs consists of several single TPNN models that are trained on randomly chosen subsets of the training data and the training starts with random weight initializations. This ensures the diversity of the resulting models which is the key issue in ensemble building. To compute the output of the ensemble for one input sequence, the output variables of all TPNNs belonging to the ensemble are averaged. We build ensembles with 20-30 individual trained TPNN models.

4 Computer Assisted Peptide Design with TPNN Models

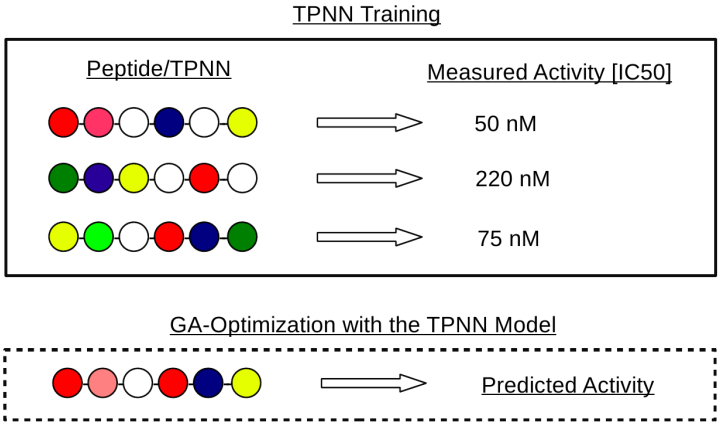

Fig. 2. The example shows how the TPNN model works in computer assisted peptide design: The model learns the properties of the peptides in the training set and the adopted cells build *virtual* peptides that are evaluated in the GA-optimization.

The main objective in building a TPNN model is to recover the fundamental characteristics of the structure activity relation. Therefore a start population of peptides is selected. The sequence strings of the peptides together with the measurements from the biological assay deliver the data for TPNN training as described in section 3. Training the network means adapting the weights of the cells that correspond to the amino acids of the peptides in the training set. The adapted cells work as building blocks of new *virtual* peptides which are generated by rearranging the order of the cells and calculating the output of the network according to equ. 1. The resulting TPNN-model defines the fitness function in a genetic algorithm (GA) that is generating new suggestions for peptide synthesis based on the learned structure activity relation. The reason for the GA

approach is the huge dimension of the "peptide space" that we have to explore. In this study we investigate 9-mer peptides from an alphabet that consists of the 20 natural amino acids and 15 non-natural D-amino-acids. We include D-amino acids in order to increase the metabolic stability of the peptides. This leads to combinatorial library of $35^9 \approx 7.8 \times 10^{13}$ possible peptides.
The GA represents each chromosome as amino acid sequence following the building block theory introduced by Holland [9] and Goldberg [10]. We perform mutation and 2-point crossover on the sequence level. The start population consists of 2000 randomly generated 9-mer peptides and evolves over 5000 generations. We prefer to use *elitist selection* by keeping the best performing individuals of the population unchanged. The new GA-based peptide suggestions are synthesized and tested in the biological assay and the results are included in the next TPNN training cycle. This process is repeated several times and improves the peptides in each round with respect to the desired properties.

5 Results

The first target to test our approach was a human G-protein-coupled receptor (GPCR) from the rhodopsin family. G-protein-coupled receptors are a protein family of transmembrane receptors that modulate several vital physiological events and comprise one of the largest families in the human genome with more than 800 identified sequences [11]. The involvement in many biological processes has the consequence that GPCRs play a key role in several pathological conditions, which has led to GPCRs being the target of up to 40% of today's marketed drugs [12, 13]. Nevertheless very little is known about the structure and structure-function relationship of this important target family because up to now there are only three mammalian GPCR crystal structures published[3]. The lack of structural knowledge is the reason why the common structure based modelling techniques cannot be used without difficulties. The advantage of the TPNN approach is that it makes no assumption about the structural features of the drug target. This method works without any explicit structural information.
The goal of our approach was the development of a 9-mer peptide with high activity and metabolic stability. The activity of the peptides was measured in a functional Cellux-assay on Ca^{2+} mobilization using stably transfected HEK293tet cells expressing the human GPCR. We applied eleven different concentrations of the peptides to obtain concentration response curves for EC_{50} calculations. All EC_{50} values are results of 3 experiments made in duplicates. The metabolic stability was measured via reverse phase HPLC with a ZORBAX Eclipse XDB-C18, $4,6 \times 150$ mm, 5 μm column. For that the peptides had been incubated in 25% human serum at 37°C. Samples were analyzed at different time points to determine the half life (in minutes) of the peptides in the human serum.
We started with a random population of 29 peptides followed by three optimization cycles with 30-50 peptides each. The results with respect to the EC_{50} values of the activity and the metabolic stability of the compounds are shown in figure 3.

[3] The GPCR crystal structures so far are the inactive conformation of bovine-rhodopsin [14, 15], the human β_2-adrenergic receptor [16] and the human A2A adenosine receptor [17].

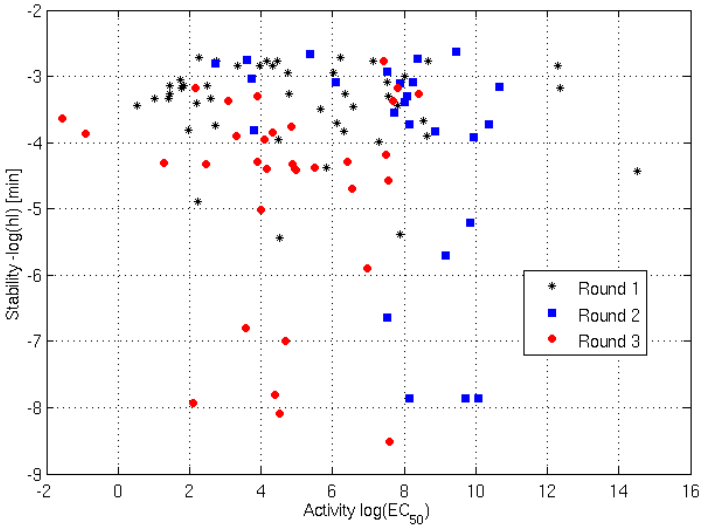

Fig. 3. This figure shows the metabolic stability (negative logarithm of the half life in minutes) versus the log(EC_{50}) values of the activity for the first three optimization rounds. The trend points to the lower left corner of the diagram, i.e. peptides that combine activity with high metabolic stability.

6 Conclusion

We introduced a new topology preserving cellular neural network that operates on single cells as building blocks with a one-to-one correspondence to the amino acids of the peptide. The TPNN mimics the chain structure of a peptide and translates a chemical structure directly into the topology of a learning machine. This overcomes the obstacle of designing and computing molecular descriptors for the QSAR. Furthermore the TPNN does not rely on the availability of structural information about the drug target.
The concept of TPNN together with the GA-based exploration of the combinatorial peptide space is the core concept of a novel peptide optimization process in drug discovery. The feasibility of the design approach was demonstrated for the construction and optimization of peptidic GPCR ligands in an iterative process of 3 design cycles of computer assisted optimization with respect to the activity and the metabolic stability. Synthesis and experimental fitness determination of less than 160 different compounds from a virtual combinatorial library of more than 7.8×10^{13} peptides were necessary to achieve this goal.
This is work in process. We plan to run further optimization cycles in order to improve activity and metabolic stability of the compounds. We cannot disclose the target and the peptide sequences before the investigation is finished and the patent situation is clarified.

Acknowledgment

The authors would like to thank the members of the Molecular Modelling Group at FMP Berlin and the members of the AG Tumortargeting at Charité Berlin.

References

1. M. Ogorzałek, C. Merkwirth, J. Wichard, Pattern recognition using finite-iteration cellular systems, in: Proceedings of the 9th International Workshop on Cellular Neural Networks and Their Applications, 2005, pp. 57–60.
2. C. Merkwirth, T. Lengauer, Automatic generation of complementary descriptors with molecular graph networks, Journal of Chemical Information and Modeling 45 (5) (2005) 1159–1168.
3. C. Merkwirth, M. Ogorzałek, Applying CNN to cheminformatics, in: Proceedings of the ISCAS, 2007, pp. 2918–2921.
4. A. Goulon, T. Picot, A. Duprat, G. Dreyfus, Predicting activities without computing descriptors: Graph machines for QSAR, SAR and QSAR in Environmental Research 18 (2007) 141–153.
5. D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
6. L. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. on Pattern Analysis and Machine Intelligence 12 (10) (1990) 993–1001.
7. S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural Computation 4 (1992) 1–58.
8. M. P. Perrone, L. N. Cooper, When networks disagree: Ensemble methods for hybrid neural networks, in: R. J. Mammone (Ed.), Neural Networks for Speech and Image Processing, Chapman-Hall, 1993, pp. 126–142.
9. J. H. Holland, Adaptation in natural and artificial systems, MIT Press, Cambridge, MA, USA, 1975.
10. D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, Boston, MA, USA, 1989.
11. R. Fredriksson, M. Lagerström, L. Lundin, H. Schiöth, The G-protein-coupled receptors in the human genome form five main families. Phylogenetic analysis, paralogon groups, and fingerprints, Mol. Pharmacol. 63 (2003) 1256–1272.
12. A. Hopkins, C. Groom, The druggable genome, Nature Reviews Drug Discovery 1 (9) (2002) 727–730.
13. D. Filmore, It's a GPCR world, Modern Drug Discovery 7 (11) (2004) 24–28.
14. K. Palczewski, T. Kumasaka, T. Hori, C. A. Behnke, H. Motoshima, B. Fox, I. L. Trong, D. Teller, T. Okada, R. Stenkamp, M. Yamamoto, M. Miyano, Crystal structure of rhodopsin: A G protein-coupled receptor, Science 289 (5480) (2000) 739–745.
15. D. Teller, T. Okada, C. Behnke, K. Palczewski, R. Stenkamp, Advances in determination of a high-resolution three-dimensional structure of rhodopsin, a model of G-protein-coupled receptors, Biochemistry 40 (2001) 7761–7772.
16. S. Rasmussen, H. Choi, D. Rosenbaum, T. Kobilka, F. Thian, P. Edwards, M. Burghammer, V. Ratnala, R. Sanishvili, R. Fischetti, G. Schertler, W. Weis, B. Kobilka, Crystal structure of the human β-2 adrenergic G-protein-coupled receptor, Nature 450 (2007) 383–387.
17. V. Jaakola, M. Griffith, M. Hanson, V. Cherezov, E. Chien, J. Lane, A. IJzerman, R. Stevens, The 2.6 Angstrom Crystal Structure of a Human A2A Adenosine Receptor Bound to an Antagonist, Science 322 (5905) (2008) 1211–1217.