Strategies for the development of a peptide computer

Hubert Hug* and Rainer Schuler#

*Universitäts-Kinderklinik Ulm, Prittwitzstrasse 43, D-89075 Ulm, Germany,
#Abteilung Theoretische Informatik, Universität Ulm, D-89069 Ulm, Germany

Corresponding author:
Hubert Hug, Universitäts-Kinderklinik Ulm, Forschungslabore, Prittwitzstrasse 43, D-89075 Ulm, Germany
Tel: 49-(0)731-503-3250
Fax: 49-(0)731-502-6765
E-mail: hubert.hug@medizin.uni-ulm.de

Abstract

Motivation: We devise a computational model using protein-protein interactions.

Results: Peptide-antibody interactions can be used to perform a large number of small logical operations in parallel. We show e.g. how a sequence of operations can be used to compare the number of occurrences of an element in two sets and how to estimate the number of occurrences of an element in a set. Similar to DNA-computing, these techniques could in principle be extended to solve instances of NP-complete problems. We give as an example a procedure to solve instances of the satisfiability problem.

Contact: hubert.hug@medizin.uni-ulm.de
schuler@informatik.uni-ulm.de

Introduction

DNA can be used to solve computational problems by DNA hybridization (Adleman, 1994; Guarnieri et al., 1996; Ouyang et al., 1997; Liu et al., 2000; Faulhammer et al., 2000). In a similar way antibodies which specifically recognize peptide sequences can be used for calculation. One peptide can contain more than one recognition site (epitope) for the same or different antibodies. In the proposed model, peptides represent the search space of a given problem and antibodies are used to select certain subsets. Similar to a DNA-computer, parallel interactions between peptide sequences and antibodies should make it possible to solve huge problems.

The advantages of such a peptide computer in comparison to a DNA-computer are that at each position 20 different building blocks instead of 4 are possible. The binding of an antibody to its epitope is very specific. Different antibodies that recognize different regions or structures of the same epitope are possible. The binding affinity of different antibodies recognizing the same epitope is different which leads to additional flexibility (Hashida et al., 1995; Shreder, 2000). In comparison, a DNA strand has only one possible reverse complement of the same length that binds to it. Due to faint differences in the melting temperature of oligonucleotides of similar sequence and a relatively high error rate, only the perfect complementary oligonucleotide can be used to bind to its target, which restricts the use of a DNA-computer significantly (Frutos et al., 1997; Aoi et al., 1999).

Nature uses nucleic acids to store information and it uses proteins to carry out orders of the genes. In addition to base pairing, computing with nucleic acids can exploit splicing mechanisms, cutting and religation of DNA (Winfree, 1996; Rothemund, 1996; Laun and Reddy, 1999), DNA hairpin formation (Sakamoto et al., 2000) or RNA editing (Landweber, 1999). The interactions between proteins and the enzymatic mechanisms are not as clearly defined as with nucleic acids and they are far more complex. Therefore, the use of proteins for calculations is expected to reach beyond our imaginations.

The mathematical problems considered here concern the number of occurrences of an element. The first problem we address is the comparison of the amount of a single element in different sets. A similar approach using DNA chips has already been used to determine differences in mRNA expression of fibroblasts after treatment with serum (Iyer et al., 1999). Similar applications at the protein level are currently not possible since proteins bind to a variety of targets (DNA, RNA, membranes, other proteins, small molecules) with different mechanisms (hydrophobic interactions, Van der Waals interactions, hydrogen bonds, electrostatic interactions). Therefore, the application of peptide computers are in specific areas.

As a second application we show how peptide-antibody interactions can be used to estimate the number of occurences of an element in a set. The number of operations (and the number of different antibodies necessary) growth logarithmically in the size of the set. Even though approximate counting is a simple problem for a peptide computer, the number of operations compared to a classical computer is exponentially small. As a final example, demonstrating the power of this model, a procedure solving instances of the NP-complete satisfiability problem is given.

Abstract model

With the hybridization of DNA molecules without additional enzymatic modifications only a yes or no decision (binary) is possible. This is used to solve computational problems and can in fact be applied to all problems in the class NP (Lipton, 1995; Rooß and Wagner, 1996). In comparison, the epitopes of peptides are recognizable by more than one antibody and with a different affinity. Therefore, logical gates can be encoded and more efficient calculations become possible.

The resulting set of peptides and antibodies should satisfy the following properties similar to those defined for DNA computers (Adleman, 1994).

(1) Under easily achievable conditions (Temperature, pH value, salt concentration) each antibody binds reliably to its peptide (epitope).

(2) Under easily achievable conditions (Temperature, pH value, salt concentration) each antibody reliably dissociates from its peptide. This can be achieved by using a second antibody with a higher affinity for the same epitope that competitively removes the first antibody or by using an excess of the free epitope itself. Antibodies can be generally dissociated by changing the pH value.
If necessary, all antibodies bound to the epitopes become covalently attached to their epitopes.

(3) Under neither of the conditions above does any antibody bind to another peptide (epitope).

Comparing the quantity of an element in two sets

We consider as a first example the following problem. Given are two (multi) sets containing elements of a finite set (e. g. proteins or antibodies of certain types). We are comparing the absolute number of a chosen element in the two sets. In particular, we want to verify whether this number differs in the two sets significantly.

Let X denote the element we are looking for. G and H denote two sets containing antibodies of certain types. In these sets X is the only antibody to the epitope for X. We prepare peptides with the following properties. The peptides contain two (disjoint) binding sites for the element (antibody) X and furthermore, binding sites for additional elements (antibodies) A1, A2 and B. Antibodies A1, A2 and B are not present in the two sets. The positions of the binding sites are shown in Figure 1.

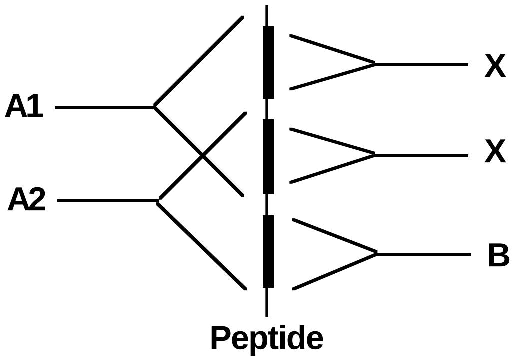

Figure 1: Binding sites of the antibodies X, A1, A2 and B to the corresponding epitopes. The epitopes are shown by thick lines on the peptide (thin line). The length of an epitope is about 10 amino acids. The whole length of the peptide is between 30 to 40 amino acids.

The binding affinities of the antibodies are crucial for the calculation. A1, A2 and X recognize overlapping epitopes (Figure 1) implying that at each time point only one antibody can be present at overlapping epitopes. The epitope that is recognized by antibody B must be part of the epitope recognized by antibody A2. It is required, that the affinity of B to its target epitope is bigger than that of A2 and that the affinity of A2 to its target epitope is bigger than that of X. The binding affinity of A1 must be lower than that of X. Antibody A1 can only bind, when both binding sites for antibody X are free. The affinity of the upper epitope for X is bigger than the affinity of the lower epitope, i. e. one amino acid in the epitope is replaced by a similar amino acid.

Generally, three major steps are involved. First, the element X of the first set is bound to one of the two possible binding sites for X on the peptide. Second, the element X of the second set is bound to the other binding site for X. Third, a set containing labeled elements of X is used to detect any free binding site for X. Detection is by fluorescence.

In practice, the following steps are necessary (Figure 1):

(1) The antibody A2 is added. Antibody A2 blocks the second binding site of X.

(2) The first set G is added. The elements X of the first set are bound to the first binding site of X.

(3) The antibody B is added. Since the affinity of antibody B to its epitope is higher than the affinity of A2, antibody B will remove antibody A2. The second binding site for X will be free.

(4) The antibody A1 is added. Since the affinity of antibody A1 for the binding site is lower than that of antibody X, A1 will bind only to those peptide positions where antibody X is not present at both binding sites.

(5) Cross-linker is added. All bound antibodies will be covalently attached to the peptide via a cross-linker like e. g. ethylene glycol *bis*(succinimidylsuccinate) (Pierce).

(6) The second set H is added. The elements X of the second set are bound to the second binding site of X. Bound antibodies X will be covalently attached to the peptide via a cross-linker.

(7) Labeled antibody X is added. Labeled antibody X will bind to all remaining free binding sites for antibody X. The fluorochrome could be fluorescein isothiocyanate (Calbiochem) with an excitation wavelength between 450 to 490 nm and an emission wavelength of 520 to 560 nm or Cy3 or Cy5 (Molecular Probes).

(8) Detect label. If any labeled antibody X is bound, its fluorescence will be measured.

If fluorescence is detected, then the first set G contains more elements of X than the second set H. Otherwise the second set contains at least as many elements of X as the first set.

However, due to errors in the chemical reactions involved, the procedure may lead to an incorrect result. Under the assumption that the number of elements differs significantly, this procedure yields the correct result with a high probability. This probability can be amplified by repeating the whole procedure several times.

Note that it is not necessary to know the amounts of the antibodies of the sets for this kind of computation. The result basically yields one bit of information, namely whether the first set contains more elements of X than the second set. In principle, the procedure can be extended to compare more than one element simultaneously.

Estimating the number of an element in a set

In this example we show how to estimate the number of occurrences of an antibody in a set. Let G denote the set and X denote a single antibody. Let n be a number such that 2^n is an upper bound for the number of antibodies X in G (e. g.: for n = 20 the number of antibodies X is bounded by approximately 1 million). We prepare different peptides with different epitopes E_1, E_2, ... E_n in different amounts. Each peptide contains an epitope for the antibody X and an overlapping epitope for some E_k ($1 \leq k \leq n$). Antibodies A_k ($1 \leq k \leq n$) bind to epitopes E_k. The number of peptides with epitope E_k is approximately 2^k (e. g. between $(1\text{-}c) * 2^k$ and $(1+c) * 2^k$ for some constant $c < 0.25$). The binding affinity of antibody A_k for the epitope E_k is stronger than that of antibody X for its overlapping epitope. Furthermore we need a labeled antibody Y which binds to the epitope or an overlapping epitope for antibody X but with a weaker affinity. The concentrations of the antibodies A_k and peptides E_k are determined by standard methods (Pierce).

For the calculation the following steps are necessary:

(1) The set G is added to peptides E_n in the defined quantity (2^n preptides). The antibodies X of set G bind to their epitopes.

(2) Labeled antibody Y is added. Labeled antibody Y will bind to all remaining free binding sites for antibody X.

(3) Detect label. If no label is detected, the number of X is at least 2^n. In this case the procedure is stopped.

(4) Let k = n - 1.

(5) The antibody A_k (2^{k+1} antibodies) and peptide E_k (2^k preptides) are added. Antibody A_k will bind to all of its epitopes. The antibody X will be replaced and will bind to any free binding sites for X on the peptides E_k. All peptides E_{k+1} to E_n are blocked for binding by antibodies X or Y.

(6) Labeled antibody Y is added. Labeled antibody Y will bind to all remaining free binding sites for antibody X on peptides E_k.

(7) Detect label. If no label is detected, the number of X is at least 2^k. In this case the procedure is stopped.

(8) Let k = k - 1. If k > 0 continue at step (5).

If no labeled antibody Y is detected after k iterations, all binding sites for antibody X were blocked. Therefore, the number of antibodies X in G is between 2^k and 2^{k+1}. If label is detected after n iterations, no antibody X is present in set G.

Extension to NP complete problems

The ideas of the previous sections can be extended to solve e. g. the satisfiability problem, an NP complete search problem in boolean logic (Cook, 1971). Let F be a formula in conjunctive normal form, i. e. F consists of clauses C_1 to C_k, where each clause contains a finite number of variables and negated variables. An assignment X attributes a truth-value (true or false) to every variable. A clause becomes true if it contains a variable with a truth-value true or a negated variable with a truth-value false. F is true under assignment X if all clauses C_1 to C_k are true under assignment X.

A formula F is satisfiable if it is true under some assignment to the variables. The satisfiablility problem is, given a formula F, to decide whether F is satisfiable and in this case to find an assignment such that F is true. Let n denote the number of variables used in F. Recall that $N = 2^n$ assignments are possible.

Consider the following (easy) example. Let F consist of the three clauses C_1, C_2 and C_3, where C_1 contains variable v_1 and $\neg v_2$ ($\neg v_2$ is the negated variable v_2), C_2 contains $\neg v_2$, and C_3 contains v_1 and v_2. Every assignment gives a truth-value true or false to variables v_1 and v_2. Let X_1 = (false, false), X_2 = (false, true), X_3 = (true, false) and X_4 = (true, true) denote the four possible assignments. Here, the first truth-value is assigned to v_1 and the second to v_2. Note that F is true under X_3.

For each assignment X we prepare a peptide and different antibodies binding to different overlapping epitopes as shown in Fig. 2. Each peptide, i. e. each assignment X, has an epitope specific for an antibody A. The epitopes for antibodies B and C consist of the same sequences in each peptide, respectively. Antibody A recognizes all three epitopes (Fig. 2).

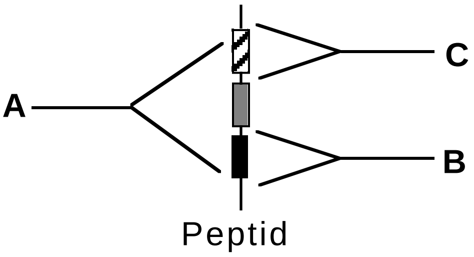

Figure 2: Binding sites of the antibodies A, B and C to their corresponding epitopes. The antibody B and the antibody C are the same for each assignment. The antibody A is specific for each assignment. Antibodies A recognize an epitope which includes a specific part (gray) and the epitopes for B (black) and C (hatched). The binding affinities are C > A > B. The order of the epitopes on the peptide is arbitrary.

For each assignment X we use a specific antibody A. In our example, four antibodies A_1 to A_4 are needed. In addition to its specific epitope antibody A recognizes the epitopes for antibodies B and C. The binding affinities are decreasing from C to A and from A to B. Note that B and C can be present at the same time, whereas A blocks binding of B, and C blocks binding of A.

To test the satisfiability of a formula F (C_1 to C_k) we prepare sets G_1 to G_k of partial solutions as follows: G_i contains antibodies A if C_i is true under the corresponding assignment X. In our example C_1 is true under assignment X_1, X_3 and X_4. C_2 is true under assignment X_1 and X_3. C_3 is true under assignment X_2, X_3 and X_4. Therefore, we prepare $G_1 = \{A_1, A_3, A_4\}$, $G_2 = \{A_1, A_3\}$ and $G_3 = \{A_2, A_3, A_4\}$.

In practice, the following steps are necessary:

(1) Let m = k.
(2) The antibody set G_k is added. The antibodies A of G_k bind to their epitopes.
(3) Antibodies B are added. Antibodies B bind to all free binding sites for B.
(4) Antibodies C are added. Antibody C binds to all of its epitopes, since it has the highest binding affinity. Note that all antibodies A of set G_k are removed, whereas antibodies B remain bound to their epitope.
(5) Antibodies C are removed by adding epitope C in excess.
(6) All remaining antibodies are covalently attached to their epitopes.
(7) Let m = m – 1. If m > 0 go to (2).
(8) Add labeled antibodies A or B.

(9) Fluorescence is detected. If the peptides are bound to an addressed chip the solution can be immediately read. Otherwise only the satisfiability of F can be detected, if fluorescence is present.

In our example, in the first iteration of the loop (step 2 to 7 above) antibody B binds covalently to all peptides prepared for assignment X_1. This prevents binding of antibodies A_1 in further iterations. After the third iteration, B is covalently bound to all peptides except peptides prepared for X_3.

Similar protocols are possible for other NP-complete search problems.

Discussion

A putative model of using peptides as the search space and antibodies as probes to solve computational problems has been presented.

For all described calculations peptides can be in an aqueous solution or bound to a chip surface. A platform for peptide binding could be XNA on Gold^{TM} (Hybaid), but other methods of building peptide chips are also possible (Livache et al., 1998). The secondary structure of a peptide when bound to a chip may change; and therefore, an antibody that recognizes the peptide in solution may not recognize it anymore when it is bound to a chip and vice versa.

A variety of cross-linkers, e. g. ethylene glycol *bis*(succinimidylsuccinate), from Pierce could be tested for binding the antibodies covalently to the peptides. Some of the cross-linkers are cleavable and therefore, the peptide chips may be re-usable. Cleavable photoreactive cross-linkers like sulfosuccinimidyl 2-(7-azido-4-methyl-coumarin-3-acetamido)ethyl-1,3-dithiopropionate (Pierce) could also be tested. Under conditions not yet defined, cross-linking may be omitted.

More than the 20 amino acids that are used in the living world can be used for the peptide computer, e. g. the L-amino acids and amino acids which are not encoded by the genetic code. Therefore, the number of possible peptide sequences increases dramatically.

Sensitive methods for the detection of single fluorescent molecules have been described (Eigen and Rigler, 1994). Instead of attaching a fluorochrome to the antibody, cDNAs coding for the antibodies could be ligated cDNAs coding for members of the green fluorescent protein family (Clontech) and expressed as fusion proteins, which are by itself fluorescent.

The antibodies binding with the postulated affinities to their peptide epitopes can be obtained by screening phage display libraries (de Bruin et al., 1999). Antibody expressing phages (Morphosys, Cambridge Antibody Technology) could be used directly in the assay. This has the advantage that antibody expressing phages contain the DNA encoding the antibody in the same particle. The DNA could be extracted and amplified by the polymerase chain reaction (PCR). The PCR product could be used for hybridization of an addressed chip with DNA as described (Liu et al., 2000). In addition, single-chain variable antibody fragments, in which the heavy and light chain variable domains are covalently connected, could be used (Weber-Bornhauser et al., 1998). These fragments may be the first choice for the application of a peptide computer since they are smaller by retaining the specificity.

Current limitations for the peptide computers are the laborious laboratory work in obtaining the monoclonal antibodies. Antibodies recognizing different epitopes of a protein are necessary for the construction of protein chips to detect qualitative and quantitative changes in protein expression patterns of cell lines and tissues. These antibodies will then be available for the peptide computer. Thus, in the near future it should be possible to use the peptide computer to solve small mathematical problems under conditions where the components are in an aqueous solution. The read-out would be in a standard fluorometer (Hybaid).

References

Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science,* **266**, 1021-1024.

Aoi, Y., Yoshinobu, T., Tanizawa, K., Kinoshita, K., & Iwasaki, H. (1999). Ligation errors in DNA computing. *BioSystems,* **52**, 181-187.

Cook, S. A. (1971). The complexity of theorem-proving procedures. *Conference records of Third Annunal ACM symposium on theory of computing 1971.* pp. 151-158, Shaker Heights, Ohio

de Bruin, R., Spelt, K., Mol, J., Koes, R., & Quattrocchio, F. (1999). Selection of high-affinity phage antibodies from phage display libraries. *Nat. Biotechnol.,* **17**, 397-399.

Eigen, M. & Rigler, R. (1994). Sorting single molecules: Application to diagnostics and evolutionary biotechnology. *Proc. Natl. Acad. Sci. USA,* **91**, 5740-5747.

Faulhammer, D., Cukras, A. R., Lipton, R. J., & Landweber, L. F. (2000). Molecular computation: RNA solutions to chess problems. *Proc. Natl. Acad. Sci. USA,* **97**, 1385-1389.

Frutos, A. G., Liu, Q., Thiel, A. J., Sanner, A. M., Condon, A. E., Smith, L. M., & Corn, R. M. (1997). Demonstration of a word design strategy for DNA computing on surfaces. *Nucl. Acids Res.,* **25**, 4748-4757.

Guarnieri, F., Fliss, M., & Bancroft, C. (1996). Making DNA add. *Science,* **273**, 220-223.

Hashida, S., Hashinaka, K., & Ishikawa, E. (1995). Ultrasensitive enzyme immunoassay. *Biotechnol. Annu. Rev.,* **1**, 403-451.

Iyer, V. R., Eisen, M. B., Ross, D. T., Schuler, G., Moore, T., Lee, J. C. F., Trent, J. M., Staudt, L. M., Hudson, J. Jr., Boguski, M. S., Lashkari, D., Shalon, D., Botstein, D., & Brown, P. O. (1999). The transcriptional program in the response of human fibroblasts to serum. *Science,* **283**, 83-87.

Landweber, L.F. (1999). RNA based computing: some examples from RNA catalysis and RNA editing. In: L. F. Landweber & E. B. Baum, (eds), *DNA based computers II: Proceedings of a DIMACS workshop 1996.* pp. 181-189, Amer. Math. Soc.

Laun, E. & Reddy, K.J. (1999). Wet splicing systems. In: H. Rubin & D.H. Wood, (eds), *DNA based computers III: Proceedings of a DIMACS workshop 1997.* pp. 73-83, Amer. Math. Soc.

Lipton, R. J. (1995). DNA solution of hard computational problems. *Science,* **268**, 542-545.

Liu, Q., Wang, L., Frutos, A. G., Condon, A. E., Corn, R. M., & Smith, L. M. (2000). DNA computing on surfaces. *Nature,* **403**, 175-179.

Livache, T., Bazin, H., Caillat, P., & Roget, A. (1998). Electroconducting polymers for the construction of DNA or peptide arrays on silicon chips. *Biosens. Bioelectron.,* **13**, 629-634.

Ouyang, Q., Kaplan, P. D., Liu, S., & Libchaber, A. (1997). DNA solution of the maximal clique problem. *Science,* **278**, 446-449.

Rooß, D. & Wagner, K. W. (1996). On the power of DNA-computing. *Information and Computation,* **131**, 95-109.

Rothemund, P.W.K. (1996). A DNA and restriction enzyme implementation of turing machines. In: R. J. Lipton & E. B. Baum, (eds), *DNA based computers: Proceedings of a DIMACS workshop 1995*. pp. 75-119, Amer. Math. Soc.

Sakamoto, K., Gouzu, H., Komiya, K., Kiga, D., Yokoyama, S., Yokomori, T., & Hagiya, M. (2000). Molecular computation by DNA hairpin formation. *Science,* **288**, 1223-1226.

Shreder, K. (2000). Synthetic haptens as probes of antibody response and immunorecognition. *Methods,* **20**, 372-379.

Weber-Bornhauser, S., Eggenberger, J., Jelesarov, I., Bernard, A., Berger, C., & Bosshard, H. R. (1998). Thermodynamics and kinetics of the reaction of a single-chain antibody fragment (scFv) with the leucine zipper domain of transcription factor GCN4. *Biochemistry,* **37**, 13011-13020.

Winfree, E. (1996). On the computational power of DNA annealing and ligation. In: R. J. Lipton & E. B. Baum, (eds), *DNA based computers: Proceedings of a DIMACS workshop 1995*. pp. 199-221, Amer. Math. Soc.

Acknowledgements

We thank the anonymous referees for helpful comments improving the paper.