NeedleHaystack: A program for the rapid recognition of local structures in large sets of atomic coordinates

Andreas Hoppe and Cornelius Frömmel *

Medical Faculty Charité, Humboldt University, Institute of Biochemistry, Monbijoustr. 2A, D-10117 Berlin, Germany.

Correspondence e-mail: cornelius.froemmel@charite.de

3D structure; algorithm; computer-aided ligand search; superposition

Abstract

A program NeedleHaystack is presented which computes molecular superpositions. A possible large molecule (target=haystack) is scanned for the occurrence of a given molecular motif (model=needle) within tolerances. The advance over other methods is that it can handle very large atom sets, e.g. targets of up to 100000 atoms and has not necessarily to include any chemical information and connectivity between atoms: chemical constraints to restrict the solution space are not used to pertain non-obvious superpositions based on geometric fit. The specialization to the needle-haystack-problem allows a very fast algorithm. Trade-offs between runtime and tolerance levels are possible. The program is fast enough to screen large databases in an acceptable time. An executable for LINUX on Pentium compatible machines is available free of charge at `http://www.charite.de/bioinf/haystack`. It is a command line executable well suited for scripts and distributed computing. A web interface for immediate testing is also available at this address.

1. Introduction

Superposition of atomic motifs is widely recognized as an efficient method to generate new hypothesises of molecular interactions (Lemmen & Lengauer, 2000). Two major applications are pharmacophore elucidation and 3D database searching (Miller *et al.*, 1999). Superposition methods should be able to handle large molecules, yet be sensitive to shapes at single-atom level, include flexibility, regard physicochemical propensities, and be rapid enough for database searches. Fulfilling all demands in one single program is not possible and therefore very different approaches have been proposed (reviewed in (Lemmen & Lengauer, 2000; Brown *et al.*, 1996; Jonassen *et al.*, 1999)). Flexibility increases the computational complexity dramatically. It can for instance be handled if the molecules are small and physicochemical interaction points (Lemmen *et al.*, 1998*b*; Mills *et al.*, 2000) or key atoms (Wallace *et al.*, 1997) instead of atoms which have to be superposed. For database searches a limited set of conformers per molecule together with a rigid superposition method yields better practical results (Thorner *et al.*, 1997; Miller *et al.*, 1999). Also, the size of the molecules increases the computational complexity. Previously, large proteins are represented by one (Toh, 1997; Nussinov & Wolfson, 1991) or two points (Kleywegt, 1999) per residue.

The algorithm presented here has similarities to Geometric Hashing (Nussinov & Wolfson, 1991; Leibowitz *et al.*, 1999; Wallace *et al.*, 1997). The main differences are:

(i) the restriction to the needle-haystack-problem,

(ii) omitting a cluster algorithm allowing the processing of millions of anchor matches[1],

(iii) the brute-force approach to find the best anchor matches which are subsequently

[1] Anchor matches are rudimentary superpositions of only a few atoms. They are starting points for complete superpositions.

scored guarantees finding a solution if it exists within certain tolerance levels,

(iv) simultaneous discrete and analytical optimization, and

(v) superposition of all (non-hydrogen) atoms.

2. Theoretical background

Definition 1. Given two mathematical sets S_1, S_2 of Cartesian coordinates (describing centers of atoms) a superposition is

(i) a selection of a subset $S_1' \subset S_1$ with cardinality n,

(ii) an injective map (assignment) $\sigma : S_1' \hookrightarrow S_2$,

(iii) a Cartesian transformation t.

Superpositions are measured by the RMSD (root mean square deviation) of the atomic coordinates which is $\sqrt{\frac{1}{n}\sum_{A\in S_1'}\left(|A - t(\sigma(A))|\right)^2}$. For given S_1, S_2 and n we are interested in the superpositions showing minimal RMSD or in all superpositions with RMSD below a given threshold. This involves two subproblems: a combinatorial task (fixing an assignment of related atom pairs) and an analytical task (fixing the Cartesian transformation for the model to the target reference frame). With a given coordinate transformation the assignment problem is relatively easy and vice versa(Kuhl *et al.*, 1984; Kabsch, 1978). The crux is the combination of the two problems.

Let $n_1 = |S_1|$, $n_2 = |S_2|$, $n = |S_1'|$. Then there are

$$N = \frac{n_1! n_2!}{n!(n_1 - n)!(n_2 - n)!} \tag{1}$$

possible ways to select n pairs of elements of S_1 and S_2, respectively (with the convention $0! := 1$) (Levi, 1972).

The analytical part of the problem is the choice of a Cartesian transformation. It forms a 6-dimensional non-discrete space, which reveals a large amount of local minima (Kuhl *et al.*, 1984) and contains singularities which cause severe numerical problems. Instead, we use a 7-dimensional space where the rotation is represented by a 4-dimensional quaternion space (Griewank *et al.*, 1979; Kearsley, 1989). Both the discrete and non-discrete space interrelate and must be optimized simultaneously which is difficult. Several approaches separate discrete and the non-discrete space to allow for a successive optimization. Lemmen et al. (Lemmen *et al.*, 1998*a*) use the Fourier space. Unfortunately, it aggravates the complexity of the non-discrete optimizations which makes it unsuitable for large atom sets. On the other hand, approaches which use the difference distance matrix (Escalier *et al.*, 1998; Lesk, 1997) circumvent the non-discrete optimization. The drawback is that the discrete optimization is much harder if the Cartesian transformation is not known. The complexity of the problem is reduced if the relevant subtask asserts additional assumption the algorithm may utilize.

2.1. Special superposition tasks

Depending on n compared to n_1 and n_2 the superposition task can be classified into three types:

Motif-Search The task is to locate common local structures of two or more molecules. Therefore the algorithm must be applicable in cases where $n \ll n_1$, $n \ll n_2$.

Needle-Haystack-Problem is a combination of motif-search and similarity search. The majority of atoms of S_1 is to be superposed to a subset of an (usually much larger) atom set S_2. $n \gtrsim \frac{n_1}{2}$ is assumed. The algorithm must be applicable in cases where $n \ll n_2$. It is the task we concentrate on.

Similarity-Search The majority of both atom sets has to be superposed. $n \gtrsim \frac{n_1}{2}$ and $n \gtrsim \frac{n_2}{2}$ is assumed. The relevant parts of molecules (interfaces to other proteins, interfaces of secondary structures) have to be known (Preissner *et al.*, 1998).

The similarity-search problem is a special case of the needle-haystack-problem which itself is a special case of the motif-search. The problem complexities behave accordingly in spite of being all NP hard. The motif-search as the most general problem has been investigated in many papers—some of them are discussed below. The computational complexity however is so large that additional constraints or heuristics must be introduced.

2.2. Needle-haystack problem

In the needle-haystack problem we are looking for a superposition of most of the atoms of S_1, called the model (the needle in the metaphor), and a subset of S_2, called the target (the haystack in the metaphor), such that the RMSD is small. The case $n = n_1$ will be called *complete superposition*. Otherwise we say that $n_1 - n$ *skips* are occurring. To compare superpositions with different numbers of skips we modify definition 2 (RMSD).

Definition 2. With the above notation of definition 1 the **skip-penalized RMSD** (pRMSD) of a superposition is

$$\sqrt{\frac{1}{n_1}\left((n_1 - n)p + \sum_{A \in S_1'} \left(|A - t(\sigma(A))|\right)^2\right)}, \tag{2}$$

where $p \in \mathbb{R}$ is called the **skip penalty**. This is the scoring function.

3. NeedleHaystack algorithm

The course of the algorithm is:

1. **Fix anchor:** Choose an anchor from the model, three non-colinear atoms. Several anchors can be chosen in which case all the following steps are repeated for each anchor.

2. **Anchor match:** Find a set of atoms in the target which approximately superpose with the anchor—the anchor match. The following steps are repeated for every anchor match.

3. **Transformation:** Compute the Cartesian transformation determined by the anchor match.

4. **Transform model:** Apply the inverse of the transformation to the model.

5. **Assign:** For each atom in the transformed model try to find one near atom in the target. The result is a superposition which is valuated by its pRMSD.

6. **Improve:** For the superpositions with the lowest pRMSDs use small modifications of the transformation and a repetition of step 5 to improve the superposition in terms of the pRMSD.

3.1. Anchor

The anchor is determined with the following method. The two atoms with the largest distance are chosen. The third atom is the atom with the largest distance to the line connecting the first and the second atom. This method is simple, efficient, and very stable.

Let the side lengths of the triangle be l_1, l_2, and l_3. Let tolerances ϵ_1, ϵ_2, and ϵ_3 be accepted. I use every point T_3 in the target to be assigned to the first anchor atom. The second anchor atom will be assigned to all atoms T_2 in the target within the distance range $(l_1 - \epsilon_1, l_1 + \epsilon_1)$. The third anchor atom is assigned to every target atom T_1 which has a distance $(l_2 - \epsilon_2, l_2 + \epsilon_2)$ to T_3 and $(l_3 - \epsilon_3, l_3 + \epsilon_3)$ to T_2.

3.2. Anchor matches

Three points in the target are chosen as anchor match if its corresponding side lengths are approximately equal to the side lengths of the anchor. The respective tolerance threshold is the first adjustable parameter of the algorithm (in case of proteins between 1 and 2Å).

The idea to match anchors is also a feature of the Geometric Hashing technique (Nussinov & Wolfson, 1991). However, the further course in Geometric Hashing is to cluster these matches. The enormous amount of possible anchor matches for full atom sets would make the cluster algorithm infeasible for large sets of atoms.

The algorithm proposed here looks at the anchor matches one-by-one which is much more straightforward. From the anchor match transformation the algorithm proceeds directly to the whole model where the small size of the model is an essential precondition.

3.3. Computation of the transformation using the anchor match

Assume the coordinates of three non-colinear points from both model and target are given. A Cartesian transformation which transforms the three model points to the three respective target points is needed.

With three non-colinear points P_1, P_2, and P_3 we define a coordinate system with P_1 as the coordinate base, the normalization $\overrightarrow{P_1P_2}$ is the $\overrightarrow{\mathbf{x}}$-axis, the orthonormalisation of $\overrightarrow{P_1P_3}$ in respect to $\overrightarrow{P_1P_2}$ forms the $\overrightarrow{\mathbf{y}}$-axis, and the $\overrightarrow{\mathbf{z}}$-axis is defined by the tensor product $\overrightarrow{\mathbf{z}} = \overrightarrow{\mathbf{x}} \times \overrightarrow{\mathbf{y}}$.

On the basis of the two coordinate systems the computation of the transformation is very fast since it involves only few vector operations. Likewise, the inversion of a transformation is a straightforward computation.

3.4. Transformation of the model coordinates

An application of the transformation computed in the previous step puts the model atoms in the coordinate frame of the target. This does not consume much computation time since the model is assumed to be small.

3.5. Assignment of related atom pairs

After a transformation is fixed the assignment of related atom pairs is trivial except in case of conflicts. Every atom M_i in the transformed model set is assigned to the target point T_j such that the distance $|M_i - T_j|$ is minimal. In the implementation a hash-list for the atoms of the target reduces the search time for atoms to a small fraction of the total running time.

A model atom is counted as a skip if there is no target atom in the vicinity of the transformed coordinate of the model atom. Two adjustable parameters have to be considered: the *adjacency threshold* (usually between 1 and 3Å) and the *allowed number of skips* (usually between 0 and 3 but for exhaustive searches up to half the number of atoms in the model is allowed).

A conflict occurs if a transformed model atom M_i is assigned to a target atom T_j and another model atom M_k is also assigned to T_j. Since the superposition task requires a one-to-one correspondence the algorithm has to decide between M_i and M_k to be assigned to T_j and to find another target atom T_l which the respective other model atom is assigned to. In this case we choose the assignment which pairing contributes less to the pRMSD.

3.6. Improvements

Because the anchor match transformation has been chosen on the basis of three points only it cannot expected to be a good transformation for the whole model.

Instead, it is a starting point for an optimization of the pRMSD regarding all points of the model.

Assuming that the assignment of the atom pairs and that the rotation is fixed there is a straightforward method to find the optimal translation: compute the center of mass (every atom is a unit mass point) of both the transformed model set (except the skipped atoms) and the atoms of the target a model atom is assigned to. The vector between the two centers of mass forms the optimal transformation. Similarly, one rotation angle can be optimized in one step if the other two angles and the translation are fixed (Sippl & Stegbuchner, 1991). Since translation and rotation interrelate an alternating optimization of translation and rotation is used until no further improvements can be done (Sippl & Stegbuchner, 1991). Though the separate optimizations are very fast the convergence of the combined method is slow. The one-step optimization of the three rotation angles using Eigenvalues computed algebraically (Kearsley, 1989) is much faster. Due to our experience, however, the simultaneous optimization of translation and rotation with a steepest descent algorithm is even better. We use the four-dimensional quaternion representation for a rotation to avoid singularities (Griewank *et al.*, 1979; Kearsley, 1989). The parameters of the gradient method were tuned according to a stable and fast convergence. Then on the basis of the improved transformation it is checked if there exists a better assignment of atoms for which the optimization of the transformation is repeated.

3.7. Pruning

Most anchor matches lead to a superposition of model and target which reveals an unacceptably large pRMSD. Improving each insufficient anchor match would result in a very slow algorithm. In computer science pruning is a basic technique to increase the efficiency of search algorithms. If a branch of the calculation is unlikely to yield

an improvement it need not to be further pursued. Pruning requires methods to guess if a branch may yield good solutions before it is exhaustively considered. As a first pruning method, an assignment which exceeds the allowed number of skips is discarded immediately. The second pruning technique involves two adjustable parameters: the *acceptable pRMSD* a (between 0.3 and 1.5Å) and an additional tolerance e (usually twice the value of a). After the assignment of the i-th model atom the anchor match is discarded if

$$kp + \sum_{A \in S^i} |A - t\,(\sigma(A))|^2 > ia^2 + e^2, \tag{3}$$

where k is the number of skips occurred in the assignment of the first i atoms, p is the skip penalty, S^i is the set of the target atoms assigned up to the i-th step of the assignment step (excluding skipped atoms). As a third pruning technique only the best m superpositions (m adjustable, usually 5,...,50) in terms of the pRMSD are considered in the improvement step. This is crucial for the algorithm since for one superposition the improvement is the most computer time consuming part of the calculation.

3.8. Complexity

The number of anchors is linearly dependent on the number of atoms in the target because the number of atoms in a bounded volume is bounded. The time to find them is bounded by $O(\log^2 n)$ (Lueker, 1978; Willard, 1979), where n is the cardinality of the target. For each anchor the time critical step is the search of the nearest atom to the transformed model point. It can also be computed in $O(\log^2 n)$. This totals $O(kn \log^2 n)$ for the search part of the algorithm, where k is the cardinality of the model. The optimization is not time-critical and may not be considered.

3.9. Proof of completeness

Proposition. *If the model is a Cartesian transformed subset of the target the NeedleHaystack-algorithm finds it.*

The proof of this proposition is trivial since for any chosen anchor in the model there is a perfect anchor match which leads to a superposition with pRMSD 0. This will remain on top of the solution list throughout any improvements of other superpositions.

The proposition is also true if n skips are allowed, where $3(n+1)$ must be smaller than the number m of atoms in the model. In this we perform the algorithm for $n+1$ disjoint anchors. At least in one anchor contains no skips and will lead to a superposition with pRMSD smaller than $\sqrt{\frac{pn}{m}}$ where p is the skip penalty.

The proposition is also true if the atom coordinates of the model are modified up to a certain degree, depending on the parameter setting. Assume that there is a Cartesian transformation such that the transformed model is completely contained in the target where the observed distance of the transformed model point and the respective target point is at most d. We assume that the correct anchor match is found if an anchor match error of $2d$ is allowed. If the anchor represents the maximal distant points in the model the maximal distance of the model point transformed with the anchor match transformation and the respective target point is at most $2d$. Therefore, the pRMSD of the superposition is not larger than $2d$ and will not be discarded, suitable parameter setting presumed. If there are no other superpositions with a lower pRMSD it will be recognized.

4. NeedleHaystack program

4.1. Data

NeedleHaystack is designed to find a model of 10 to 200 non-hydrogen atoms (usually a ligand, a binding region, or an interaction pattern—atoms from a pair of binding partners) in a set of either all or the water accessible non-hydrogen atoms of a molecule of arbitrary size. It can also be applied for targets of only the main chain atoms and the Cα atoms, respectively. However NeedleHaystack is not intended to compete with fold recognition programs on accuracy or runtime. NeedleHaystack is capable to process data where also the hydrogens are included.

4.2. Parameters and Runtime

The size of model and particularly the target is most important for the runtime. Searching in 100 atoms is faster than reading the input file from the hard disk, searching in 100000 atoms of a protein can be a matter of hours. If the target is the subset of all surface accessible atoms of a molecule the runtime is considerably reduced—more than the reduction in the number of atoms would suggest. Usually runtimes longer than some minutes indicate poor parameter settings. High cutoff values may allow millions of solutions all of which are computed. This is desirable if the computation time is irrelevant and an extremely low likelihood of missing possible superpositions is required.

In the recent version 2.1.0 there are many command-line parameters a few of which are important for the yield in superpositions and the runtime:

(i) The cutoff for the distance of assigned atoms. Standard is 2Å. Lowering this number speeds computation but also changes the scoring function—more skips are required.

(ii) The allowed number of skips. Computing superpositions with many skips re-

quires much computation time. A good value for 30 model atoms are 3 skips (used in subsection

The ability of the program to find similarities go far beyond what can be guaranteed (subsection

5. Databases and other methods used

The Protein Data Bank (Bernstein *et al.*, 1977) was used as the primary protein 3D-structure data source. PISCES (Wang & Jr, 2002) is used to create representative subsets of it. The program "gap" in the GCG/Wisconsin package (Womble, 2000) (with the parameters gap=50, len=3) was used to score sequence similarity.

6. Testing the program

6.1. Finding surface patches on subtilisin-type proteases

The first task is to find 2 surfaces patches of subtilisin Carlsberg (PDB code 1CSE) as needles in members of the subtilase family (Siezen & Leunissen, 1997) and few other proteases as haystacks. The first model is the active site. It is the set of 37 atoms of Subtilisin Carlsberg which are in van-der-Waals contact to the atoms of the ligand (inhibitor Eglin C) both taken from PDB structure 1CSE. The second model is a surface patch set on the opposite side of the subtilisin molecule. It is a set of 37 solvent accessible atoms forming a weak depression. It shares some properties with a typical binding site (Peters *et al.*, 1996) despite there is no binding function described. The detailed atom list can be found in table

Due to the observation that the specificity of subtilases and the binding site

is quite similar (Gron *et al.*, 1992; Rheinnecker *et al.*, 1994), one would expect from a automatic method that it can find the correct position of the active site of the proteases type. Because the active site is more conserved than any other surface area the recognition rate of the control patch can be expected to be lower (Irving *et al.*, 2001).

67 structures are used as target molecules including all subtilases and a few other molecules as controls. For the best superposition of each model and target it is checked visually if the match is indeed located at the particular region. See results in tables

The pro-subtilisin (PDB code 1SPB, (Gallagher *et al.*, 1995)) contains no active site at the surface but is otherwise similar—the result of the program clearly reflects this. The non-subtilases have quite bad scores and the best superposition was not located at the active site. The back-site patch was not found for Subtilisin E, Savinase and Subtilisin (bac. lentus) and the non-Subtilisin subtilases. It could also not be found for the other structures (considered beside the pro-subtilisin , see above).

It is a well-known phenomenon that active sites and other binding regions are structurally more conserved than other regions of protein molecules (Irving *et al.*, 2001). Therefore we are not surprised that the molecular patch defined on the back site shows a larger variation than the active site and shows in some cases such a high score that it could not be separated from noise. Due to the low sequence variation in the subtilisin family the high success rate of NeedleHaystack may be caused by a low structural variation of the binding site. In the next section we apply the program to a larger, more variable protein family.

6.2. Identifying trypsins by their active side

The program is also tested by the task to find the active side of bovine γ-chymotrypsin (PDB code 1GG6) in members of the trypsin-fold family and a representative PDB selection. The surface patch is the 30 atoms of the cleavage site of chymotrypsin identified by the atom which are in van-der-Waals contact to the atoms of the ligand (inhibitor n-acetyl-phenylalanine trifluoromethyl ketone) both taken from PDB structure 1GG6. The atoms can be found in table **??**.

The trypsin dataset was derived from all 619 structures of the "Trypsin-like serine proteases" superfamily of the database SCOP (Release 1.61, Nov 2002) (Murzin *et al.*, 1995) without viral proteins. To ensure an automatic analysis of the results we used only 559 structures (list in tables

The average sequence similarity score (see section

The NeedleHaystack program searched for an occurrence of a atomic motif similar to the active side patch of 1GG6 in every surface of both the trypsin and the control dataset. 517 of the 559 trypsin structures (92.5%) superposed correctly with a score of less than 1.1Å. An analysis of the 42 "false negatives" (given in table

14 of the 643 structures of the control dataset were superposed (with the same program parameters) with a score of less than 1.1Å 7 of which also belong to the trypsin dataset. This corresponds to a rate of false positives of 1.1%. The runtime for this computation was 5h56min13s, 33.14 s for each target, on a dual CPU (AMD MP 1800+) based system.

6.3. Complete PDB search with the trypsin active site

The surface patch given in subsection

7. Availability

Our implementation of the algorithm is available free of charge at `http://www.charite.de/bioinf/haystack`. We provide an executable for LINUX on Pentium compatible machines. A manual guides through the first steps with some application examples. Also, this web address leads to a simple web-interface of the program.

The authors wish to thank the DFG for supporting this work in the course of the project Schn317/6-5.

References

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanoushi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535–542.

Brown, N. P., Orengo, C. A. & Taylor, W. R. (1996). *Comput. Chem.* **20**, 359–380.

Escalier, V., Pothier, J. & Soldano, H. (1998). *J. Comput. Biol.* **5**(1), 41–56.

Gallagher, T., Gilliland, G., Wang, L. & Bryan, P. (1995). *Structure*, **3**(9), 907–914.

Griewank, A. O., Markey, B. R. & Evans, D. J. (1979). *J. Chem. Phys.* **71**(8), 3449–3454.

Gron, H., Meldal, M. & Breddam, K. (1992). *Biochemistry*, **31**, 6011–6018.

Irving, J. A., Whisstock, J. C. & Lesk, A. M. (2001). *Proteins*, **42**(3), 378–382.

Jonassen, I., Eidhammer, I. & Taylor, W. R. (1999). *Proteins*, **34**, 206–219.

Kabsch, W. (1978). *Acta Cryst.* A**34**, 827–828.

Kearsley, S. K. (1989). *Acta Cryst.* A**45**, 208–210.

Kleywegt, G. J. (1999). *J. Mol. Biol.* **285**, 1887–1897.

Kuhl, F. S., Crippen, G. M. & Friesen, D. K. (1984). *Comput. Chem.* **5**, 24–34.

Leibowitz, N., Fligelman, Z. Y., Nussinov, R. & Wolfson, H. J. (1999). *ISBN*, pp. 169–177.

Lemmen, C., Hiller, C. & Lengauer, T. (1998*a*). *J. Comput.-Aided Mol. Design*, **12**, 491–502.

Lemmen, C. & Lengauer, T. (2000). *J. Comput.-Aided Mol. Design*, **14**(2), 215–232.

Lemmen, C., Lengauer, T. & Klebe, G. (1998*b*). *J. Medical Chem.* **41**, 4502–4520.

Lesk, A. M. (1997). *Folding & Design*, **2**(3), 12–14.

Levi, G. (1972). *Calcolo*, **9**, 341–352.

Lueker, G. S. (1978). In *Proc. 19th Annu. IEEE Sympos. Found. Comput. Sci.*, pp. 28–34.

Miller, M. D., Sheridan, R. P. & Kearsley, S. K. (1999). *JMC*, **42**(9), 1505–1514.

Mills, J. E. J., Esch, I. J. P., Perkins, T. D. J. & Dean, P. M. (2000). *CAMD*, **15**(1), 81–96.

Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995). *J. Mol. Biol.* **247**, 536–540.

Nussinov, R. & Wolfson, H. J. (1991). *Proc. Natl. Acad. Sci. USA*, **88**(23), 10495–10499.

Peters, K. P., Fauck, J. & Frömmel, C. (1996). *J. Mol. Biol.* **256**, 201–213.

Preissner, R., Goede, A. & Frömmel, C. (1998). *J. Mol. Biol.* **280**(3), 535–550.

Rheinnecker, M., Eder, J., Pandey, P. S. & Fersht, A. R. (1994). *Biochemistry*, **33**, 221–225.

Siezen, R. J. & Leunissen, J. A. (1997). *Protein Sci*, **6**(3), 501–523.

Sippl, M. J. & Stegbuchner, H. (1991). *Comput. Chem.* **15**(1), 73–78.

Thorner, D. A., Willett, P., Wright, P. M. & Taylor, R. (1997). *J. Comput.-Aided Mol. Design*, **11**(2), 163–174.

Toh, H. (1997). *Comput. Appl. Biosci.* **13**(4), 387–396.

Wallace, A. C., Borkakoti, N. & Thornton, J. M. (1997). *Protein Sci.* **6**, 2308–2323.

Wang, G. & Jr, R. L. D. (2002). *Bioinformatics*. Submitted.

Willard, D. E. (1979). *Predicate-oriented database search algorithms*. Ph.D. thesis, Aitken Comput. Lab., Harvard Univ., Cambridge, MA.

Womble, D. D. (2000). *Methods Mol. Biol.* **132**, 3–22.

Table 1. *Atom list of the two surface patches on 1CSE*

Active side patch
Cγ, Cδ_2, Cϵ_1, Nϵ_2 of H64; Cδ_1 of L96; Cα, C, O of G100; O of S125; Cα, C, Cβ of L126; N, Cα, C, O of G127; Cα of G128; N of A129; O, Cβ of A152; Cα of G154; Cγ, Nδ_2 of N155; Cα, Cβ, Cγ, Cδ_1, Cδ_2 of L217; N, O of N218; Cα of G219; Oγ_1 of T220; N, Cβ, Oγ of S221; Sδ, Cϵ of M222
back-side patch
O of A169; O of K170; O of Y171; O of D172; O of V174; Cβ of A176; O of A194; Cα, C, O, Cβ, Cγ of E195; O of L196; Cα, Cβ, Cγ, Cδ, Oϵ_1, Oϵ_2 of E197; Cγ, Cδ, Nϵ, Cζ, Nη_1, Nη_2 of R247; Oδ_1 of N248; O, Cβ, Oγ of S251; Cβ, Oγ of S260; Cα of G264; Cα, Cβ, Cδ, Cϵ, Nζ of K265

Table 2. *Best superpositions found by the NeeedleHaystack program. The model are selected atom sets of Subtilisin Carlsberg (PDB code 1CSE) described in the text. The targets are the surface atoms of the proteins identified by its PDB code. An uppercase letter follows to indicate the chain. The pRMSD and the number of skips for the best match is given.*

target	PDB code	sequence similarity	binding site (37 atoms) pRMSD	skips	inactive site (37 atoms) pRMSD	skips
Subtilisin carlsberg	1cseE	100	0.00	0	0.00	0
	1sbc	100	0.68	1	0.91	2
	2secE	98.5	0.14	0	0.22	0
	1sel	98.5	0.76	3	0.72	3
Subtilisin Bac. licheniformis	1af4	100	0.51	1	0.66	2
	1scb	100	0.61	2	0.70	2
	3vsb	100	0.72	2	0.80	4
	1av7	100	0.71	2	0.80	3
	1be6	100	0.55	1	0.46	0
	1be8	100	0.60	1	0.52	1
	1bfu	100	0.44	1	0.80	4
	1scd	99.5	0.55	1	0.73	2
	1scnE	99.5	0.58	1	0.47	1
	1sca	99.5	0.53	1	0.59	1
	1vsb	99.5	0.68	2	0.70	2
	1avt	99.5	0.62	2	0.61	1
	1bfk	99.5	0.77	3	0.47	1
	1bh6	83.4	0.66	1	0.54	2
Subtilisin E	1scjA	60.3	0.60	1	1.04	2
Subtilisin Savinase	1svn	57.5	0.61	2	1.30	4
Subtilisin Bac. lentus	1jea	57.5	0.62	2	—	-
	1gci	55.6	0.62	2	—	-

Table 2. *continued.*

target	PDB code	sequence similarity	binding site (37 atoms) pRMSD	skips	inactive site (37 atoms) pRMSD	skips
Subtilisin BPN'	1yjb	60.5	0.56	1	0.98	4
	1sbnE	59.9	0.47	1	0.99	2
	1suaA	59.6	0.74	2	1.05	3
	1yjc	59.5	0.55	1	1.06	4
	2sniE	59.4	0.47	1	1.10	2
	2st1	59.4	0.67	2	1.06	4
	1st2	59.4	0.68	2	—	-
	1s01	59.2	0.63	2	1.13	4
	2sicE	59.2	0.50	0	1.04	2
	1sud	59.1	0.69	2	1.13	4
	3sicE	58.9	0.52	1	1.10	3
	5sicE	58.6	0.60	1	1.16	4
	1aqn	58.3	0.53	1	1.12	4
	1sue	58.2	0.71	2	—	-
	1au9	58.1	0.52	1	1.12	4
	1sub	58.0	0.67	2	1.12	4
	1yja	57.8	0.54	1	1.11	4
	1s02	57.7	0.71	3	1.06	3
	1sbi	57.7	0.58	1	—	-
	1sbh	57.6	0.52	1	—	-
	1sibE	57.6	0.52	1	0.95	2
	1a2q	57.3	1.25	3	1.07	4
	1suc	57.1	0.83	3	1.15	3
	1sup	57.1	0.87	4	1.03	3
	1ak9	54.7	0.54	1	1.14	3

Table 2. *continued.*

target	PDB code	sequence similarity	binding site (37 atoms) pRMSD	skips	inactive site (37 atoms) pRMSD	skips
Mesentericopeptidase	1meeA	62.5	0.35	0	1.07	4
M-proteinase	1mpt	55.0	0.82	2	—	-
Serine protease PB92	1ah2†	52.1	—	-	—	-
Thermitase	3tecE	40.3	0.35	0	—	-
	2tecE	39.1	0.33	0	—	-
	1tecE	38.9	0.55	1	—	-
	1thm	38.5	0.61	2	—	-
Proteinase K	2prk	29.6	0.57	1	—	-
	3prkE	29.6	0.71	1	—	-
	1ptk	29.6	0.72	0	—	-
	1cnm	29.6	0.58	1	—	-
	1bjrE	28.6	0.71	1	—	-
	2pkc	27.8	0.58	1	—	-
	1pekE	25.4	0.69	2	—	-

† first model chosen

Table 2. *continued.*

target	PDB code	sequence similarity	binding site (37 atoms) pRMSD	skips	inactive site (37 atoms) pRMSD	skips
Pro-subtilisin†	1spbS	57.9	—	-	1.05	3
Myeloblastin	1fujA	20.0	—	-	—	-
Trypsin	5ptp	22.6	1.31	4	—	-
	1try	19.5	1.24	3	—	-
Thrombin	1vr1H	19.3	1.30	3	—	-
Chymotrypsin	4gch	19.8	1.22	2	—	-

† The catalytic side is not at the surface.

Table 3. *Atom list of the chymotrypsin active side on 1GG6*
O of F41; Sγ of C42; Cδ_2, Nϵ_2 of H57; O, Cβ of S190; Cα, C, O of C191; Cα, C, Cβ, Cγ, Cϵ of M192; N, Cα of G193; N, Cβ, Oγ of S195; Cγ_1 of V213; O of S214; Cα, C, O, Cβ of W215; N, Cα, C of G216; N, O of S217

Table 4. *Data set for the Trypsin example*
Prokaryotic proteases

alpha-Lytic protease	1tal, 2ull, 2alp, 1p12e, 1p11e, 1gbja, 1p02a, 1p01a, 6lpra, 7lpra, 1gbaa, 1gbfa, 1gbba, 1p05a, 1gbda, 1gbca, 5lpra, 1gbka, 1p03a, 3lpra, 1p09a, 1gbla, 9lpra, 2lpra, 8lpra, 1gbma, 1gbha, 1p10a, 1gbia, 1gbea, 1p06a, 1p04a,
Protease A	2sga, 3sgae, 1sgc, 4sgae, 5sgae
Glutamic acid-specific protease	1hpga
Trypsin	1sgt
Protease B	1sgpe, 1ct4e, 1ct2e, 3sgbe, 1sgre, 1ct0e, 1sgqe, 1ds2e, 4sgbe, 1csoe, 2sgpe
Epidermolytic toxin A	1exfa

Table 4. *continued*
Eukaryotic (chymo-)trypsin, thrombin, elastase

Trypsin(ogen) cow	1tpp, 1bty, 1tld, 2ptn, 3ptb, 1btxa, 1tnh, 1btwa, 1tnk, 1tpo, 2tgt, 1ppce, 1jrsa, 1jrta, 1tgt, 1btza, 1tnj, 1tps, 1tgsz, 1tng, 3ptn, 1pphe, 1may, 1tgc, 1mts, 1tnl, 1ntp, 1tni, 1tawa, 2tga, 1tyn, 1tioa, 1tpae, 1ppee, 1tgn, 2ptce, 1max, 2tioa, 3tpiz, 1gbt, 2tgpz, 1smfe, 1aq7, 1tgb, 4tpiz, 1btp, 2tpiz, 2tgd, 1tabe, 1xui, 1xug, 1sfia, 1bju, 1sbwa, 1xuj, 1ce5a, 1xuk, 1qcpa, 2bzaa, 1zzza, 1az8, 1bjv, 1xuf, 1yyy1, 1xuh, 1hj9a, 1j8aa, 1gi2a, 1gi1a, 1c1ta, 1c5ta, 1c1qa, 1c1pa, 1gi4a, 1c5ua, 1ghza, 1c1ra, 1c5sa, 1c2ha, 1c2ja, 1gi0a, 1c1na, 1c2ma, 1c1oa, 1c2ia, 1c5qa, 1c5pa, 1gi3a, 1gi6a, 1gj6a, 2btce, 1c5ra, 1c5va, 1c2la, 1c2ga, 1gi5a, 1c2ea, 1c1sa, 1g3ba, 1c2da, 1g3ca, 1g3da, 1g3ea, 1c2fa, 1c2ka, 3bthe, 1k1pa, 3btfe, 3btme, 1f2se, 1f0ta, 1f0ua, 1mtw, 1g36a, 1mtu, 1g34a, 1mtv, 3btke, 1ejma, 3btee, 1jira, 3btde, 1k1na, 3btte, 3btqe, 1k1oa, 1eb2a, 3btge, 1k1ja, 1k1ma, 1auj, 1k1ia, 3btwe, 1ql7a, 1k1la, 1d6ra, 1g9ie, 1ezxc, 1ql8a, 1c9ta
Trypsin(ogen) other organisms	1mcta, 1fnia, 1qqua, 1fn6a, 1avwa, 1epta, 1fmga, 1ldtt, 1aksa, 1an1e, 1avxa, 1tfxa, 1ejaa, 1c9pa, 1dpo, 3tgie, 1slub, 1fy8e, 1slwb, 1bra, 1slxb, 1anc, 1brbe, 1ane, 1and, 1ql9a, 1trma, 1slvb, 3tgje, 1brce, 1amha, 1anb, 2trm, 1f7za, 1f5ra, 3tgke, 1k9oe, 1trna, 1h4wa, 1hj8a, 1a0ja, 2stbe, 2stae, 1bit, 2tbs, 1bzxe, 1gdna, 1fn8a, 1fy4a, 1fy5a, 1gdqa, 1gdua, 1try
(α,γ)-Chymotrypsin(ogen)	1gg6a, 1ggda, 1ab9a, 3gcta, 1gcta, 8gch, 3vgca, 1k2i1, 1afqa, 5chaa, 2cgaa, 1choe, 2gcta, 4chaa, 1ghbe, 1gcd, 2gmt, 2vgca, 1vgca, 7gch, 1acbe, 6chaa, 1gmh, 1gl1a, 1ghae, 4vgca, 1gmda, 1gmca, 3gch, 4gch, 6gch, 1dlka, 2gch, 1ca0a, 1hjaa, 2cha, 1cgie, 1cgje, 1cbwa, 1gl0e, 5gch, 1mtna, 1chg, 1ex3a, 1eq9a
Thrombin	1h8dl, 1c5ll, 1ahtl, 1c5nl, 1doja, 1toml, 1a4wl, 1ghxl, 1ba8a, 1a3bl, 1h8il, 1gj5l, 1ay6l, 1ai8l, 1a3el, 1eb1l, 1c1ul, 1qbvl, 1ppbl, 1ihsl, 1joua, 1hbtl, 1k22l, 1k21l, 1gj4l, 1ghvl, 1ghyl, 1ihtl, 1de7l, 1g37a, 1c5ol, 1g32a, 1d6wa, 1vr1l, 1c1wl, 1lhcl, 1bcul, 1eoja, 1ae8l, 1bb0a, 1umal, 1a46l, 1aixl, 1ad8l, 1b5gl, 1afel, 1hxel, 1hxfl, 1qj1a, 1ca8a, 1a5gl, 1qurl, 1c1vl, 1g30a, 1a61l, 1a2cl, 7kmel, 1eola, 1hahl, 1tmtl, 8kme1, 1ktta, 1thsl, 1tbzl, 5gdsl, 1ditl, 1c4u1, 1qj7a, 1fpcl, 1hage, 1nrrl, 1hail, 1thpa, 2thfa, 1bhxa, 1qj6a, 1c4v1, 1awfl, 2hgtl, 1tmbl, 1abil, 1lhel, 1lhgl, 1b7xa, 1thrl, 1hgtl, 1qhra, 1d9ia, 1nrsl, 1dx5a, 4htcl, 1lhdl, 1fphl, 1abjl, 3hatl, 1ktsa, 1tmul, 1lhfl, 4thnl, 1bmml, 1dm4a, 1dwcl, 1bmnl, 1c4y1, 3htcl, 1uvsl, 1dwel, 1dwdl, 2hntl, 1dwbl, 1hdtl, 1hltl, 1haol, 1hapl, 1hutl, 1nrnl, 1awha, 2hpql, 1e0fa, 2hppl, 1nrpl, 1nrol, 1nrql, 1etrl, 1ucyl, 1etsl, 1mkxl, 1bthl, 1bbrl, 1mkwl, 1ettl, 1uvtl, 1ycpl, 1avgl, 1id5l, 1tbrl, 1tbql, 1hrtl, 1uvul, 1vitl, 1toca
Elastase	1ppfe, 1hnee, 1ppge, 1b0fa, 1gvkb, 1qnja, 1hazb, 1haxb, 1h9lb, 3est, 1btu, 1qgfa, 4este, 1nese, 1e36b, 1e38b, 1b0ea, 8este, 1hayb, 1eas, 1gwaa, 1qixb, 1qr3e, 1e34b, 1e37b, 1lvy, 6est, 9est, 7este, 1fzza, 1e35b, 1eat, 1eau, 1inc, 1flee, 5este, 1hb0b, 1brup, 1jim, 1c1ma, 1eaia, 2este, 1est, 1elt, 1m9ua

Table 4. *continued*
Other eukaryotic proteases

Neuropsin	1npma
Crab collagenase	1azza
HL collagenase	2hlca, 1hyla
Enteropeptidase (enterokinase light chain)	1ekbb
beta-Tryptase	1a0la
Cathepsin G	1cgha, 1au8a, 1kyna
Coagulation factor VIIa	1danh, 1jbuh, 1cvwh, 1fakh, 1dvah
Chymase (Proteinase II)	3rp2a, 1klt, 1pjpa
Kallikrein A	2pkaa, 1hiaa, 2kaia
Kallikrein-13	1ao5a
Kallikrein 6	1lo6a, 1l2ea, 1gvla
Tonin	1ton
7S NGF protease subunits	1sgfa
Factor B	1dlea
Factor D	1bio, 1dica, 1dsua, 1dst, 1dfpa, 1hfd, 1fdpa
Two-chain tissue plasminogen activator (TC)-T-PA	1rtfa, 1a5hc
Single chain tissue plasminogen activator	1bdaa, 1a5ia
Plasminogen activator from snake venom, TSV-PA	1bqya
Coagulation factor IXa, protease domain	1pfxc, 1rfna
Coagulation factor Xa (Chrismas factor), protease domain	1fjsa, 1f0ra, 1c5md, 1f0sa, 1hcga, 1ezqa, 1ksna, 1xkbc, 1xkac, 1g2ma, 1faxa, 1kigh
Coagulation factor Xa-trypsin chimera	1fxya
Activated protein c (autoprothrombin IIa)	1autc
Myeloblastin, PR3	1fuja
Urokinase-type plasminogen activator (LMW U-PA)	1gj7a, 1gjaa, 1gj8a, 1c5ya, 1gjda, 1gj9a, 1gjca, 1ejna, 1c5xa, 1gi7a, 1gi8a, 1gi9a, 1c5za, 1f5ku, 1c5wa, 1gjba, 1f5la, 1f92a, 1lmwa
Plasmin(ogen), catalytic domain	1buia,
Granzyme B	1fi8a, 1iaua, 1fq3a
Duodenase	1eufa
Beta-acrosin	1fiwl, 1fizl
Matriptase MTSP1	1eaxa, 1eawa

Table 5. *Locating trypsin binding sides: false positives*

Trypsinogen where the catalytic side is not active	2tgt, 1tgt, 1tgc, 2tga, 1tgb, 2tgd, 1ezxc
Chymotrypsinogen where the catalytic side is not active	1chg, 1ex3a
Prethrombin	1hage
α-thrombin	1nqf
Elastases	1qnja, 1e36b, 1hayb, 1lvy, 1e35b, 1inc, 1hb0b, 1est, 1m9ua
Coagulation factor VIIa	1jbuh
Kallikrein 6	1gvla
Tonin	1ton
7S NGF protease subunits	1sgfa
Factor D	1bio, 1dica, 1dsua, 1dst, 1dfpa, 1hfd, 1fdpa
Granzyme B	1fq3a
Alpha-Lytic protease	1tal, 2ull, 2alp, 1p02a, 1p09a, 1p10a, 1gbia, 1gbea, 1p06a
Epidermolytic toxin A	1exfa

Synopsis

A program NeedleHaystack is presented which computes molecular superpositions. A possible large molecule (target) is scanned for the occurrence of a given molecular motif (model) within tolerances both being represented by all non-hydrogen atoms.