Clustering 3D-structures of Small Amino Acid Chains for Detecting Dependences from their Sequential Context in Proteins

Alexander Hinneburg, Daniel A. Keim
Institute of Computer Science
University of Halle, Germany
{hinneburg, keim}@informatik.uni-halle.de

Wolfgang Brandt
Institute of Bio-Chemistry
University of Halle, Germany
brandt@biochemtech.uni-halle.de

Abstract

In the past a good number of rotamer libraries have been published, which allow a deeper understanding of the conformational behavior of amino acid residues in proteins. Since the number of available high resolution X-ray protein structures has grown significantly over the last years, a more comprehensive analysis of the conformational behavior is possible today. In this paper, we present a method to compile a new class of rotamer libraries for detecting interesting relationships between residue conformations and their sequential context in proteins. The method is based on a new algorithm for clustering residue conformations. To demonstrate the effectiveness of our method we apply our algorithm to a library consisting of all 8000 tripeptide fragments formed by the 20 native amino acids. The analysis shows some very interesting new results, namely that some specific tripeptide fragments show some unexpected conformation of residues instead of the highly preferred conformation. In the neighborhood of two asparagine residues, for example, threonine avoids the conformation which is most likely to occur otherwise. The new insights obtained by the analysis are important in understanding the formation and prediction of secondary structure elements and will consequently be crucial for improving the state-of-the-art of protein folding.

1. Introduction

In the recent years the number of proteins stored in the PDB (Protein Data Bank) has grown significantly. Because of the technical progress a good number of high-resolution X-ray structures of proteins became available. Statistical methods have been applied to the PDB to extract knowledge about the conformational behavior of amino acid residues. Amino acid side chain conformations have been studied, for example, by [4, 5, 13]. These studies resulted in side chain rotamer libraries, which consist of a list of discrete conformations having a weight which corresponds to their frequency in the PDB. Since the PDB contains a multitude of high-resolution structures, it was also possible to determine rotamer preferences depending on the backbone conformation. Based on this idea, a number of weak correlations of rotamer distributions and secondary structures have been found [14, 16]. Recently, a backbone dependent side chain rotamer library has been presented by [9, 7]. The effectiveness of the backbone dependent rotamer libraries has been shown by [8, 2] for homology modeling and by [15] for NMR and X-ray structure refinement.

Although the idea of using rotamer libraries has already been applied successfully in the past, until now only a small fraction of its potential has been revealed. The backbone dependent side chain rotamer library mentioned above, for example, only uses 132 out of the about 2000 proteins with a resolution $\leq 2\mathring{A}$, which are available in the PDB. For the better understanding of tertiary structures, it is highly desirable to compile comprehensive rotamer libraries which are based on all protein structures available in the PDB. Using such a rotamer library, for instance, one would be able to determine how the conformation of an amino acid residue (in particular that of a side chain) in a protein depends on its neighbors in the sequence. To find and understand such relationships, a new method is required which is able to deal with large amounts of residue conformations and to classify them effectively. Our new method presented in this paper is based on a cluster analysis in the conformation space (cf. section 2). The basic idea is to model the conformation of amino acid residues or small peptide fragments as points in the multidimensional dihedral angle space. The cluster algorithms then determines clusters by assigning an influence function to each data point, by summing up all influence functions to determine the overall density function, and by finding the maxima of the overall density function using a gradient-based hill-climbing procedure (cf. section 3). The method is used to compile a new class of rotamer libraries

which allows new insights into interesting dependences between the 3D-structure of small peptide chain fragments and their sequential context. In section 5, we evaluate the effectivity and efficiency of our new approach and provide some interesting results showing, for example, that in the neighborhood of two asparagine residues, threonine avoids the conformation which is highly preferred otherwise. Note that our new method is generally applicable to arbitrary protein fragments. In this paper, however, we restrict ourselves to the evaluation and analysis of tripeptide conformations.

2. General Idea

In the backbone dependent rotamer library developed by [7] for each residue type a probability distribution of the side chain angle χ_1 is calculated for each node on an equidistant grid in the 2D (ϕ, ψ)-space. The distributions of χ_2, χ_3 and χ_4 only depend on the previous side chain dihedral angle. For detecting more global relationships this method becomes inefficient since the size of the grid grows exponentially in the number of considered angles. Another problem arises if one is interested in the probability distribution of more than one angle. Using Bayesian statistics, it is difficult to derive combined distributions of two or more angles.

The conformation of amino acid residues or small peptide fragments can be described by data points in a multidimensional dihedral angle space. The approach we are using partitions the multidimensional angle space corresponding to the observed data distribution by clustering the data in the dihedral angle space. More formally, this can be described as follows. Given is a set of protein sequences P. A sequence $p \in P$ is denoted as a string of linked amino acid residues a from the set of natural amino acids A:

$$p \in P,\ p = a_1 a_2 \ldots, a_l,\ a_i \in A, i = 1, \ldots, l.$$

In our approach, we do not directly use the real tertiary structure since the mapping of the 3D coordinates of the structure to dihedral angles basically contains all relevant information about the protein structure and is much easier to handle. For clear notations, we introduce the mapping D of the 3D atom coordinates of a protein $p \in P$ to a sequence of vectors of dihedral angles as:

$$p \in P,\ D(p) = s_1, s_2, \ldots s_l,\ s_i \in [-180, 180)^{d_i},\ i = 1, \ldots, l$$

with d_i being the number of dihedral angles for the residue a_i. For example, d_i is 3 for $a_i = G$ (glycine) and $d_i = 7$ for $a_i = A$ (arginine). The components of a vector s_i are

$$s_i = \begin{cases} (\phi, \psi, \omega) & ; d_i = 3 \\ (\phi, \psi, \omega, \chi_1, \ldots, \chi_{d_i - 3}) & ; 3 \leq d_i \leq 7 \end{cases}.$$

We use the dihedral angles of one residue as the smallest unit for detecting relationships. Note that one can easily modify the structure of the data by, for example, grouping the dihedral angles.

To produce the rotamer library for detecting relationships between the 3D-structure of a residue and in its sequential context two steps have to be performed.

Step 1 For all different residues do:
Determine all conformations of the residue in the protein structures of P and partition the conformational angle space according to the observed data distribution by using a cluster algorithm which identifies highly populated areas in the multidimensional dihedral angle space.

Step 2 For all residues in the protein sequences, replace the dihedral angle vector with the cluster-id of that vector and based on the resulting data, build frequency tables for all different residue fragments of fixed length by counting the occurrences of all fragments which correspond to the same sequence of cluster-ids.

In the resulting frequency tables, we obtain significant information about dependences between the residues in a given sequential neighborhood and the preference for a certain conformational structure. Note that most of these correlations can only be detected if the sequential context is considered. In the following, we discuss the most important step of our approach, namely building the frequency tables by clustering the dihedral angle space, in more detail.

3. Clustering the Dihedral Angle Space

In the clustering step, the densely populated areas in the conformational space of each of the 20 natural amino acid residues have to be identified. This can be done separately for each of the residues. As the first step of the clustering algorithm, the source data for the cluster analysis of a residue $a \in A$ is determined by collecting all conformations of a, which occur in the protein conformations in P. In addition to the residue conformation, the protein name and the position of the residue in the protein chain are stored. With this additional information it is possible to retrieve the sequential context of a residue conformation after the cluster analysis. The considered set of proteins P contains all proteins from the PDB, which have resolutions of the X-Ray structure of $\leq 2\mathring{A}$. With this condition, P contains about 2000 proteins. From this set of proteins, we get a conformational data set for each residue with a size ranging from 48.000 (for alanine) to 9000 (for tyrosine). In order to get a good number of classified tripeptide conformations for each of the 8000 possible tripeptide fragments, we used the complete data sets in the cluster analysis.

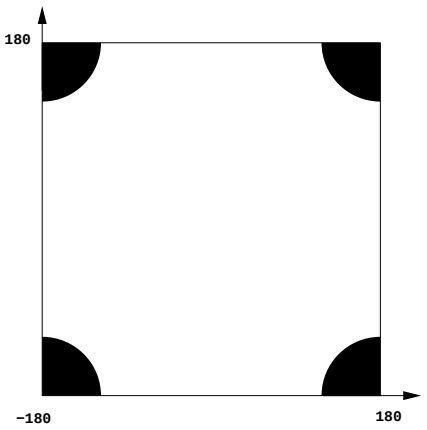


Figure 1. Effect of the Warp-Around

The task of the cluster algorithm is to group the objects from the given data sets into smaller, homogeneous, and meaningful subsets which are the clusters. In our case, the objects are conformations of one residue type described by a vector of dihedral angles. To formally define the term $homogeneous$, we need an appropriate distance measure on the objects. In case of dihedral angle vectors, it is rather straight-forward to extend the Euclidian distance to measure the shortest path of transformation between two residue conformations. This can be defined as

$$x, y \in [-180, 180)^d,\ dist(x,y) = \sqrt{\sum_{i=1}^{d} \begin{cases} |x_i - y_i|^2, & |x_i - y_i| \leq 180 \\ (360 - |x_i - y_i|)^2, & \text{else} \end{cases}}$$

The impact of this distance measure is the wrap-around at the borders. The effect is shown in figure 1 where the shaded area displays a two-dimensional sphere around the the point $(-180, -180)$. After defining the distance measure, we have to define an adequate notion of clusters. Since the definition of clusters largely depends on the data and the application, we first tried to get a visual impression of the structure of clusters in our application. For this purpose, we used the Ramachandran-Plot of the actual conformation set, which is a projection to the (ϕ, ψ)-plane. Figure 2 shows the (ϕ, ψ)-plot of glycine conformations as an example. Note that the plot is only a projection of the high-dimensional dihedral angle space to the two-dimensional display space, which involves loosing some information. Nevertheless, the plot reveals some important properties of the clustering in our data set. The figure shows densely populated areas which are separated by nearly empty space. It is well known from biochemistry that preferred areas exist in the conformation space, and the clusters in the plot correspond to preferred secondary structure elements in which the residues are involved. Two further observations can be

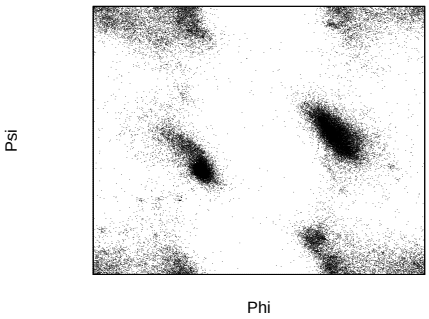


Figure 2. Ramachandran-Plot for Glycine

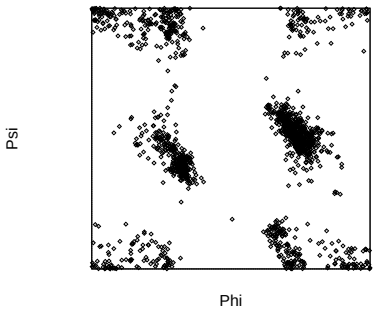
derived from the plots: First, the shape of the clusters is not fixed to certain shapes (e.g., spherical shapes) and second, the space between the clusters is filled with a significant number of outliers. Outliers are points which do not belong to any cluster. The observations lead to restrictive requirements for the cluster algorithm: The algorithm should be able to find clusters of arbitrary shape, handle a variable amount of outliers, and efficiently deal with a large multi-dimensional data set (up to 50.000 points). From the wide range of cluster algorithms which have been proposed in the literature [6, 10, 20], only few algorithms fulfill these requirements and none of them works efficiently on large amounts of multidimensional data. A new approach which has been recently proposed by the authors in the context of knowledge discovery in multimedia databases [11, 12] can be adapted to meet the requirements.

In the following, we briefly introduce the algorithm and describe the adaptation to the problem of clustering the conformation space of amino acid residues. For our definitions, we need a point density function which is determined based on kernel density estimation (KDE) [18, 17].

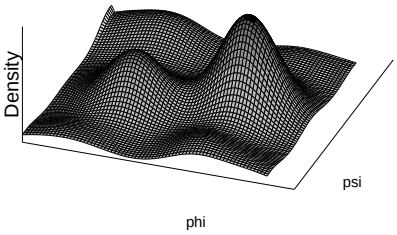
Definition 1 (Density Function - KDE)
Given N conformations described by a set of dihedral angle vectors $D = \{x_1, \ldots, x_N\} \subset [-180, 180)^d$ and h be the smoothness level. Then, the **density function** *$\hat{f}^D$ based on the kernel density estimator K is defined as:*

$$\hat{f}^D(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$$

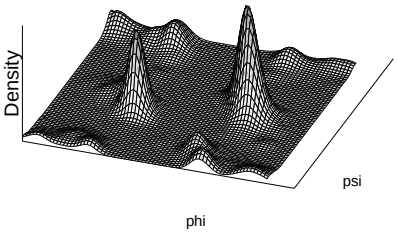
Kernel density estimation provides a powerful framework for finding clusters in large data sets. In the statistics literature, various kernels K have been proposed. Examples are square wave functions or Gaussian functions. An example for the density function of a two-dimensional data set ($\phi - \psi$ plot of glycine) using a Gaussian kernel and different smoothing levels $h = \{40, 10\}$ is provided in figure 3. A detailed introduction to kernel density estimation is beyond the scope of this article and can be found in [18, 17].



(a) Data Set



(b) $\sigma = 40$



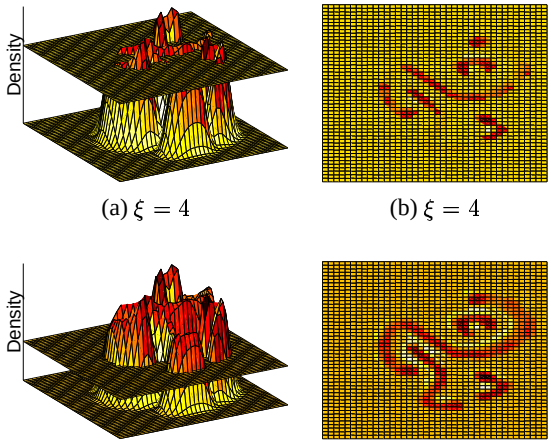
(c) $\sigma = 10$

Figure 3. Example for Density Functions

According to [11, 12], clusters can now be defined as the maxima of the density function, which are above a certain noise level ξ. Using the noise level, the algorithm can handle large amounts of outliers, which are responsible for local maximums with low point density.

Definition 2 (Center-Defined Cluster)
A center-defined cluster for a maximum x^* of the density function $\hat{f}^D$ is the subset $C \subseteq D$, with $x \in C$ being density-attracted by x^* and $\hat{f}^D(x^*) \geq \xi$. Points $x \in D$ are called outliers if they are density-attracted by a local maximum x_o^* with $\hat{f}^D(x_o^*) < \xi$.

According to this definition, each local maximum of the density function which is above the noise level ξ becomes a cluster of its own and consists of all points which are density-attracted by the maximum. The notion of *density-attraction* is defined by the gradient of the density function. The definition can be extended to clusters which are defined by multiple maxima and can approximate arbitrarily-shaped clusters.



(a) $\xi = 4$ (b) $\xi = 4$

(c) $\xi = 2$ (d) $\xi = 2$

Figure 4. Example of Multicenter-Defined Clusters for different ξ

Definition 3 (Multicenter-Defined Cluster)
A multicenter-defined cluster for a set of maxima X is the subset $C \subseteq D$, where

1. $\forall x \in C \ \exists x^* \in X : \ f_B^D(x^*) \geq \xi$, *$x$ is density-attracted to x^* and*
2. $\forall x_1^*, x_2^* \in X : \exists$ *a path $P \subset F^d$ from x_1^* to x_2^* above noise level ξ.*

Figure 4 shows the multicenter-defined clusters for different ξ. When ξ increases, more and more clusters get separated. Note that the resulting clusters may have an arbitrary shape and that all the resulting clusterings represent valid clusterings describing some characteristics of the data set. Our definitions of clusters have two important parameters, namely the smoothness h and the noise level ξ. The parameter h describes the influence of one data point on its neighborhood. There are two extreme values h_{max} and h_{min}. If $h \geq h_{max}$ the influence is propagated so far that the density function f^D has only one local maxima. The other extreme value is $h \leq h_{min}$ in which case the kernel functions become little peaks and each data point becomes a density attractor of its own. Choosing a good h can be done by considering different h and determining the largest interval between h_{max} and h_{min}, where the number of density attractors is almost constant. The clustering resulting from that approach can be seen as naturally adapted to the

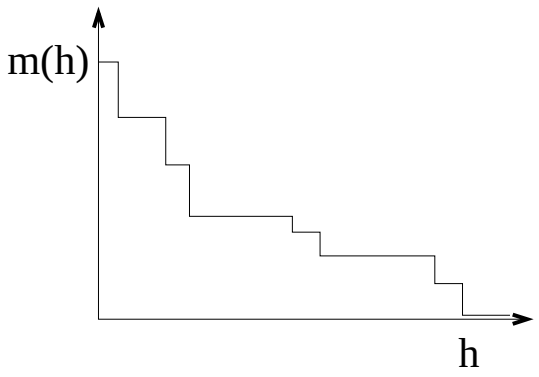


Figure 5. Number of Clusters Depending on h

data set. In figure 5 we provide an example for the number of density attractors $m(h)$ depending on h. The second parameter ξ describes the minimum density level above which a density attractor is considered significant. A good choice for ξ helps the algorithm to focus on the densely populated areas and to save computational time. Note that the border of a cluster may be in regions with a density lower than ξ. Important is that the density attractor x^* has $f^D(x^*) \geq \xi$. The details of the theoretical foundations and implementation of the DENCLUE and the OptiGrid algorithm are beyond the scope of this paper and are described in [11, 12].

4. Building Fragment Rotamer Libraries

With the results derived by the cluster analysis, we are now able to build the desired frequency table. The result from the cluster analysis is that each residue conformation in the protein structures $p \in P$ has become part in exactly one cluster (the outliers are grouped into a special cluster). Formally, we introduce a mapping function C from the residue conformations into the set of cluster identification CI. This function provides the identification of the cluster for a given residue conformation.

$$p \in P,\ D(p) = s_1, \ldots, s_{l_p}, \\ \forall j \in \{1, \ldots, l_p\} : C(s_j) = c_j,\ c_j \in CI$$

For a more convenient notation, we write for $p \in P$:

$$C(D(p)) := C(s_1), \ldots, C(s_{l_p}) = c_1, \ldots, c_{l_p}.$$

The fragment rotamer libraries can be compiled based on the mapping C for different fragment sizes without recomputing the cluster analysis in step 1. Let be $t > 1$ the fragment size for the desired library. The fragment rotamer library L can be considered as a relation, where each of the 20^t possible residue fragments of length t correspond to a finite set of t-tuples of cluster-ids. The set A^t is the set of the fragments with length t based on the set A of the 20 natural amino acid residues.

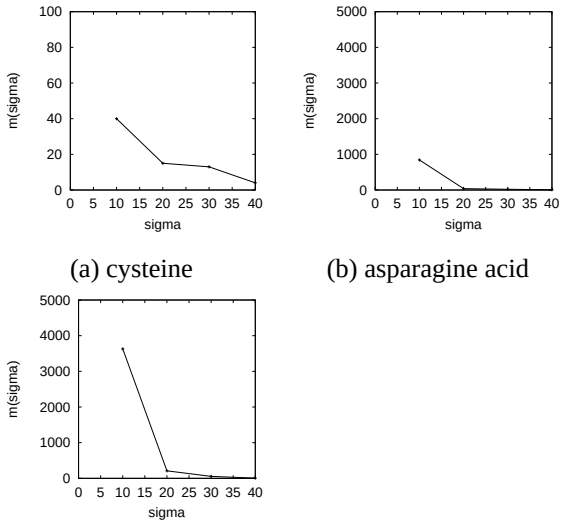
$$x \in A^t, L(x) = \{[(c_1^1, \ldots, c_t^1), h^1], \ldots, \\ [(c_1^{l_x}, \ldots, c_t^{l_x}), h^{l_x}]\}$$

Each tuple i matches to $h^i \geq 1$ occurrences of the corresponding fragment and its mapping given by the t-tuple in the protein structures $p \in P$. The frequency h of each tuple is provided in L as a statistical information about the preference of the corresponding fragment for the conformation space corresponding to the t-tuple of cluster-ids. The significance of such preferences depends strongly on the total number $\sum_{i=1}^{l_x} h^i$ of occurrences of the examined fragment $x \in A^t$ in the protein structures $p \in P$. Frequency tables can be seen as a basis for mining association rules [1, 19] or correlation rules [3], which express the found knowledge in a similar way to logic rules.

5. Results

In this section, we focus on the application of the cluster algorithm to conformational data of amino acid residues and provide as an example a rotamer library for tripeptids. The algorithm using a Gaussian kernel was applied to each of the 20 conformation data sets for different $h = \sigma$ ($\sigma_1 = 10$, $\sigma_2 = 20$, $\sigma_3 = 30$ and $\sigma_4 = 40$). To get a first impression of the results, we plotted the number of local maxima (not clusters) depending on σ. Figure 6 shows three typical cases which occurred in our analysis: (a) there exists an interval where $m(\sigma)$ is almost constant (b) $m(\sigma)$ slowly decreases (c) $m(\sigma)$ rapidly decreases (note the scaling). Residues with a behavior such as the one presented in case (c) of Figure 6 are mostly hydrophile and have long side chains described by $\chi_1, \ldots, \chi_4$. Hydrophile residues often occur at the surface of a protein and the side chain reaches into the water where no stable conformation is adapted. As a consequence, the data points are uniformly distributed in these dimensions and no preference can be detected. It seems to be more realistic to neglect χ_3 and χ_4 and postulate them as freely rotatable. Figure 7 shows the effect of neglecting χ_3 and χ_4 on the number of density attractors. From Figure 7 it is clear that the assumption that χ_3 andχ_4 can rotate freely leads to more realistic clusterings - a result which is also supported by [7].

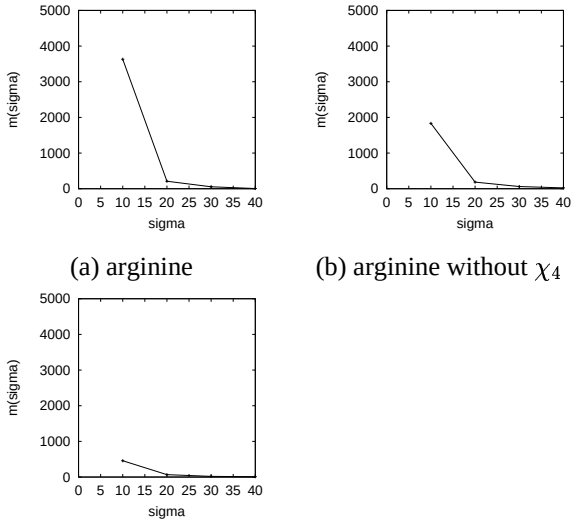
With these results we can build the mapping function C from the residue conformations to the cluster-ids we formally introduced in section 4. A cluster identification consists of the residue name and the cluster number. The clusters are numbered in the order of decreasing size. In Table 1, we provide an example of a clustering and in Table 2 an example of the mapping for the clusterings with $\sigma = 40$.



(a) cysteine (b) asparagine acid

(c) arginine

Figure 6. Number of Density Attractors Depending on σ



(a) arginine (b) arginine without χ_4

(c) arginine without χ_4, χ_3

Figure 7. Number of Density Attractors Depending on σ

- cluster (T,1); size: 8904/35212, 25.29%, center: $\phi = -107.1 \quad \psi = 129.2 \quad \omega = 178.7 \quad \chi_1 = -58.1$
- cluster (T,2); size: 8303/35212, 23.58%, center: $\phi = -113.4 \quad \psi = 160.6 \quad \omega = 178.4 \quad \chi_1 = 61.3$
- cluster (T,3) size: 8293/35212, 23.55%, center: $\phi = -88.3 \quad \psi = -14.5 \quad \omega = -179.6 \quad \chi_1 = 59.6$

Table 1. Cluster for Threonine (not complete), $\sigma = 40$

AC	ϕ	ψ	...	χ_2		AC	Cluster Id
S	-82.54	-23.18	...			S	(S,3)
C	-101.64	22.62	...			C	(C,1)
T	-83.39	159.20	...			T	(T,2)
H	63.44	17.12	...	-72.51	$\stackrel{C}{\Longrightarrow}$	H	(H, out)
F	-112.55	140.25	...	-82.41		F	(F, out)
P	-90.11	8.31	...	-23.44		P	(P,2)
G	-54.72	-24.68	...			G	(G,3)
N	-89.86	2.58	...	131.35		N	(N, out)
L	-61.54	-43.83	...	-172.62		L	(L,1)

Table 2. Example of the Mapping Function C

We stored these mappings and scanned them to determine the frequency of different mappings for each tripeptide fragment occurring in the protein sequences. Due to the large number of possible tripeptide fragments (8000), the set of proteins P (2000 proteins in our example) also provides a very large number of occurring tripeptide fragments (240000). In the derived frequency table, for each tripeptide fragment a list of the different triples of cluster-ids and their frequencies is stored. Table 3 shows a portion of the frequency table for $\sigma = 40$.

Table 3 contains a full list of cluster-id triples for the tripeptide fragment *N T N*. Two observations can be derived from the table. First, there is a significant preference for the first triple, and second, cluster 1 for threonine (T,1) occurs only once in the list. Cluster 1 is the cluster with the largest size in the clustering for the middle amino acid residue threonine and can be considered a good preference for threonine in the conformation space. The list shows that in the neighborhood of two asparagine residues threonine avoids the usually preferred conformation space of cluster 1.

This kind of frequency tables which can be considered as a new class of rotamer libraries, provides an easy way of detecting of a-priori unknown relationships. The next steps in our further research will be the development of an adequate visualization of such libraries combined with automated algorithms, allowing scientists a fast exploration of the libraries and enabling them to find rules which are hidden in the data set. Further investigations are intended to explore applications to the protein folding problem.

...			
(N,8)	(T,3)	(P,1)	2
ASN	THR	ASN	
(N,2)	(T,3)	(N,9)	109
(N,4)	(T,2)	(N,5)	13
(N,5)	(T,5)	(N,3)	12
(N,1)	(T,4)	(N,1)	11
(N,3)	(T,4)	(N,1)	7
(N,7)	(T,3)	(N,4)	6
(N,3)	(T,4)	(N,3)	5
(N,5)	(T,5)	(N,5)	4
(N,4)	(T,5)	(N,5)	3
(N,1)	(T,3)	(N,1)	3
(N,2)	(T,3)	(N,9)	3
(N,1)	(T,3)	(N,1)	2
(N,4)	(T,3)	(N,7)	2
(N,1)	(T,4)	(N,3)	2
(N,1)	(T,4)	(N,3)	1
(N,3)	(T,3)	(N,1)	1
(N,5)	**(T,1)**	**(N,2)**	**1**
(N,out)	(T,2)	(N,1)	1
(N,6)	(T,3)	(N,1)	1
(N,2)	(T,3)	(N,1)	1
(N,5)	(T,3)	(N,1)	1
(N,1)	(T,4)	(N,2)	1
(N,2)	(T,3)	(N,2)	1
(N,3)	(T,4)	(N,3)	1
(N,2)	(T,3)	(N,7)	1
(N,4)	(T,5)	(N,8)	1
(N,2)	(T,3)	(N,2)	1
(N,1)	(T,3)	(N,3)	1
(N,3)	(T,3)	(N,3)	1
(N,5)	(T,2)	(N,3)	1
ASN	THR	GLU	
(N,4)	(T,2)	(E,4)	6
...			

Table 3. Part of the Frequency Table for $\sigma = 40$

References

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, pages 207–216. ACM Press, 1993.

[2] M. Bower, F. Cohen, and R. Dunbrack. Homology modeling with a backbone-dependent rotamer library. *J. Mol. Biol.*, 267:1268–1282, 1997.

[3] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, pages 265–276. ACM Press, 1997.

[4] R. Chandrasekaran and G. Ramachandran. Studies on the conformation of amino acids. xi. analysis of the observed side group conformations in proteins. *Int. J. Pept. Prot. Res.*, 2:223–233, 1970.

[5] V. Cody, W. Duax, and H. Hauptman. Conformational analysis of aromatic amino acids by x-ray crystallography. *Int. J. Pept. Prot. Res.*, 5:297–308, 1973.

[6] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley and Sons, 1973.

[7] R. Dunbrack and F. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci.*, 6:1661–1681, 1997.

[8] R. Dunbrack and M. Karplus. Backbone-dependent rotmer library for proteins: Application to side-chain prediction. *J. Mol. Biol.*, 230:543–571, 1993.

[9] R. Dunbrack and M. Karplus. Conformational analysis of the backbone- dependent rotamer preferences of protein side chains. *Nature Struct. Biol.*, 1:334–340, 1994.

[10] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD 1996, Proceedings 3rd Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[11] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *KDD 1998, Proceedings 4th Int. Conf. on Knowledge Discovery and Data Mining*, pages 58–65. AAAI Press, 1998.

[12] A. Hinneburg and D. A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 506–517. Morgan Kaufmann, 1999.

[13] M. James and A. Sielecki. Structure and refinement of penicillo-pepsin at 1.8 a resolution. *J. Mol. Biol.*, 125:299–361, 1983.

[14] J. Janin, S. Wodak, M. Levitt, and B. Maigret. Conformations of amino acid side chains in proteins. *J. Mol. Biol.*, 125:357–386, 1978.

[15] J. Kuszewski, A. Gronenborn, and G. Clore. Improving the quality of nmr and crystallographic protein structures by means of conformational database potential derived from structure databases. *Protein Sci.*, 5:1067–1080, 1996.

[16] M. MCGregor, S. Islam, and M. Sternberg. Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. *J. Mol. Biol.*, 198:295–310, 1987.

[17] D. Scott. *Multivariate Density Estimation*. Wiley and Sons, 1992.

[18] B. Silverman. *Density Estimation*. Chapman & Hall, 1986.

[19] R. Srikant and R. Agrawal. Mining generalized association rules. In *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, pages 407–419. Morgan Kaufmann, 1995.

[20] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996*, pages 103–114. ACM Press, 1996.