

# ProteMiner-SSM: a web server for efficient analysis of similar protein tertiary substructures

Darby Tien-Hau Chang, Chien-Yu Chen, Wen-Chin Chung, Yen-Jen Oyang\*,  
Hsueh-Fen Juan<sup>1</sup> and Hsuan-Cheng Huang<sup>2</sup>

Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, ROC,  
<sup>1</sup>Institute of Biotechnology and Department of Chemical Engineering, National Taipei University of Technology,  
Taipei, Taiwan, ROC and <sup>2</sup>Institute of Biological Chemistry, Academia Sinica, Taipei, Taiwan, ROC

Received February 15, 2004; Revised April 1, 2004; Accepted April 12, 2004

## ABSTRACT

**Analysis of protein–ligand interactions is a fundamental issue in drug design. As the detailed and accurate analysis of protein–ligand interactions involves calculation of binding free energy based on thermodynamics and even quantum mechanics, which is highly expensive in terms of computing time, conformational and structural analysis of proteins and ligands has been widely employed as a screening process in computer-aided drug design. In this paper, a web server called ProteMiner-SSM designed for efficient analysis of similar protein tertiary substructures is presented. In one experiment reported in this paper, the web server has been exploited to obtain some clues about a biochemical hypothesis. The main distinction in the software design of the web server is the filtering process incorporated to expedite the analysis. The filtering process extracts the residues located in the caves of the protein tertiary structure for analysis and operates with  $O(n \log n)$  time complexity, where  $n$  is the number of residues in the protein. In comparison, the  $\alpha$ -hull algorithm, which is a widely used algorithm in computer graphics for identifying those instances that are on the contour of a three-dimensional object, features  $O(n^2)$  time complexity. Experimental results show that the filtering process presented in this paper is able to speed up the analysis by a factor ranging from 3.15 to 9.37 times. The ProteMiner-SSM web server can be found at <http://proteminer.csie.ntu.edu.tw/>. There is a mirror site at <http://p4.sbl.bc.sinica.edu.tw/proteminer/>.**

## INTRODUCTION

One of the fundamental issues in drug design is analysis of protein–ligand interactions (1,2). The detailed and accurate analysis of protein–ligand interactions involves calculation of binding free energy based on thermodynamics and even quantum mechanics (3,4). However, this approach is highly expensive in terms of computing time. As a result, conformational and structural analysis of proteins and ligands has been widely employed as a screening process in computer-aided drug design (5–8).

In this paper, a web server designed for efficient analysis of similar protein tertiary substructures, named ProteMiner-SSM, is presented. Figure 1 illustrates one application that the design of ProteMiner-SSM addresses. In this application, the biochemist is given the crystal structure of a protein bound with a specific ligand and wants to conduct a search in the Protein Data Bank (PDB) database (9) for the other proteins that contain a similar binding site and therefore could interact with the specific ligand. In one experiment reported in this paper, ProteMiner-SSM has been exploited to investigate whether some proteins in the caspase family contain a similar binding site to the structure of integrin reported in (10). The experimental results provide biochemists with some valuable clues that conform to a biochemical hypothesis.

In terms of the application illustrated in Figure 1, it is apparent that only the substructures in the caves of the protein tertiary structure are of interest. Therefore, in order to expedite the analysis process, it is desirable to incorporate a mechanism that can effectively extract the residues in the caves of the protein tertiary structure. In this paper, an efficient filtering process with  $O(n \log n)$  time complexity is employed, where  $n$  is the number of residues in the protein. In comparison with the  $\alpha$ -hull algorithm (11), which is a widely used algorithm in computer graphics for identifying those instances on the contour of a three-dimensional (3D) object, the filtering

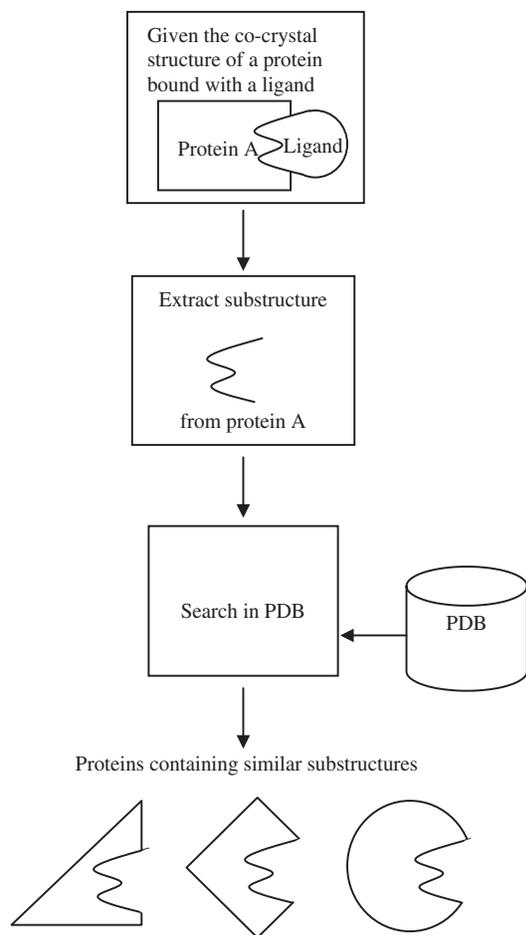
\*To whom correspondence should be addressed. Tel: +886 2 23625336 #431 Fax: +886 2 23688675; Email: yjoyang@csie.ntu.edu.tw

Correspondence may also be addressed to Chien-Yu Chen. Email: cychen@mars.csie.ntu.edu.tw

Present address:

Chien-Yu Chen, Graduate School of Biotechnology and Bioinformatics, Yuan-Ze University, Chung-Li, Taiwan

The online version of this article has been published under an open access model. Users are entitled to use, reproduce, disseminate, or display the open access version of this article provided that: the original authorship is properly and fully attributed; the Journal and Oxford University Press are attributed as the original place of publication with the correct citation details given; if an article is subsequently reproduced or disseminated not in its entirety but only in part or as a derivative work this must be clearly indicated.



**Figure 1.** One application addressed by the design of ProteMiner-SSM.

process employed in this paper features a lower time complexity,  $O(n \log n)$  versus  $O(n^2)$ . Experimental results show that the filtering process presented in this paper is able to speed up the analysis by a factor ranging from 3.15 to 9.37 times.

The next section of this paper elaborates the software design of ProteMiner-SSM. We then report the experiments conducted to evaluate the performance of ProteMiner-SSM. Finally, concluding remarks are presented, and two appendices give the mathematical basis of Equations 1 and 2 in the text.

## SOFTWARE DESIGN OF ProteMiner-SSM

ProteMiner-SSM carries out analysis in two steps. In the first step, a filtering process based on an efficient kernel density estimation algorithm is invoked to identify the crucial tertiary substructures on the contour of the protein that the analysis should focus on. In the second step, the geometric hashing algorithm in computer graphics (12,13) is invoked to compare the crucial substructures of the target protein and the binding/active site of the reference protein. In this paper, we refer to the protein that contains the binding/active site of interest as the reference protein and the proteins in PDB against which the alignment is to be performed as the target proteins.

ProteMiner-SSM conducts analysis at the residue level with each residue represented by its alpha carbon in the vector space. In other words, a protein substructure is defined by the coordinates of the alpha carbons included in the

substructure. The efficient kernel density estimation algorithm that forms the basis of the filtering process treats the set of residues  $\{s_1, s_2, \dots, s_n\}$  of a protein as  $n$  samples randomly taken from a probability distribution in the 3D vector space and employs the learning algorithm that we have recently proposed (14,15) to construct an approximate probability density function of the following form:

$$\hat{f}(\mathbf{v}) = \frac{1}{n} \sum_{i=1}^n \left( \frac{\beta}{\lambda \cdot \sigma_i} \right)^m \exp \left( -\frac{\|\mathbf{v} - \mathbf{s}_i\|^2}{2\sigma_i^2} \right), \quad 1$$

where

- (i)  $\mathbf{v}$  is a vector in an  $m$ -dimensional vector space and in this paper  $m=3$ ,
- (ii)  $\beta$  is the parameter that controls the smoothness of the approximation function,

$$(iii) \quad \sigma_i = \beta \delta_i = \beta \frac{R(s_i) \sqrt{\pi}}{\sqrt{(k+1) \Gamma((m/2)+1)}},$$

where  $R(s_i)$  is the distance between sample  $s_i$  and its  $k$ -th nearest neighbor,  $k$  is a parameter to be set by the user, and  $\Gamma(\cdot)$  is the Gamma function (18),

$$(iv) \quad \lambda = \sum_{h=-\infty}^{\infty} \exp(-h^2/2\beta^2).$$

One interesting observation is that, regardless of which  $\beta = (\sigma_i/\delta_i)$  ratio is employed, we have  $(\lambda/\beta) \cong \sqrt{2\pi}$ . If this observation can be proved to be generally correct, then we can further simplify Equation 1 and obtain

$$\hat{f}(\mathbf{v}) = \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{\sqrt{2\pi} \cdot \sigma_i} \right)^m \exp \left( -\frac{\|\mathbf{v} - \mathbf{s}_i\|^2}{2\sigma_i^2} \right). \quad 2$$

The mathematical basis of Equations 1 and 2 is elaborated in the appendices.

As the approximate probability density function shown in Equations 1 and 2 is a continuous and smooth function in the vector space, we can expect that the function values at the residues located on the contour of the protein tertiary structure are generally smaller than the function values at the inner residues. Accordingly, we can set a threshold of the function values to distinguish those residues that are located on the contour from those that are not.

With the residues on the contour of the protein tertiary structure been successfully identified, the next task of the filtering process is to further classify each of these residues depending on whether it is located in a cave or not. This task can be carried out by applying Equation 1 or 2 again but with a larger  $\beta$  value. Applying Equation 1 or 2 with a larger  $\beta$  value implies that the approximate probability density function obtained is smoother. As a result, the function values at those residues that are located in a cave will be generally larger than the function values at those residues that are on the contour of the protein tertiary structure but not in a cave. Accordingly, a threshold can be set to classify these residues.

With the filtering process applied to both the reference protein and the target protein, the next task that ProteMiner-SSM carries out is conducting structural alignment on the crucial substructures identified. In ProteMiner-SSM, we have adopted the common practice for carrying out protein

structural alignment with the geometric hashing algorithm (5–8,16). With this practice, the coordinate systems examined by the geometric hashing algorithm are limited to those defined by the two backbone bonds connected to the alpha carbon of each residue. With the filtering process incorporated, in our implementation, the geometric hashing algorithm further narrows down its search space to only the coordinate systems defined by the residues located in the caves. In the design of ProteMiner-SSM, the likelihood of residue substitution is also taken into account. If the entry in the PAM 250 matrix (1,17) corresponding to a pair of residues aligned by the geometric hashing algorithm is  $<2$ , then this pair of residues is excluded from the list of those successfully aligned.

The discussions presented in this section so far elaborate the basics of the software design of ProteMiner-SSM. Additional details, including parameter settings and time complexity analysis, can be found in the supplementary material.

## EXPERIMENTAL RESULTS

This section reports two experiments conducted to evaluate the performance of ProteMiner-SSM. The main objective of the first experiment is to test the accuracy of ProteMiner-SSM. The second experiment demonstrates how biochemists can exploit ProteMiner-SSM to facilitate their research works.

In the first experiment, three datasets, each of which contains a reference structure and a number of target proteins, are used to test whether ProteMiner-SSM is able to identify the region on the contour of the target protein that contains a similar substructure as the reference protein. Table 1 shows the characteristics of these three reference protein structures.

**Table 1.** Characteristics of the reference proteins in the first experiment

PDB ID	Number of residues	Number of residues in the binding/active site	Number of residues remaining with the filtering process applied
1HDZ	748	14	307
1BL5	414	8	130
1L5G	1470	18	833

**Table 2.** Experimental results for the first experiment

Reference protein	1HDZ					1BL5					1L5G	
	3HUD	1HTB	1HDY	1DEH	1HDX	1IDE	1HJ6	1IDC	1IDD	1IDF	1JV2	1M1X
Number of residues in the active/binding site	14	14	14	14	14	8	8	8	8	8	18	18
Geometric hashing without filtering												
Execution time of geometric hashing in seconds	66.69	67.07	67.23	66.88	67.00	10.76	10.80	10.76	10.73	10.77	447.14	444.12
Number of residues in the active/binding site that are successfully aligned	14	14	14	14	14	8	7	8	8	8	18	18
RMSD of aligned pairs	0.79	0.37	0.47	0.36	0.49	0.52	0.42	0.55	0.49	0.43	1.21	1.22
Geometric hashing with filtering applied												
Execution time of filtering in seconds	0.13	0.14	0.13	0.14	0.14	0.06	0.06	0.06	0.06	0.06	0.33	0.33
Execution time of geometric hashing in seconds	11.95	11.32	11.86	11.78	11.59	1.25	1.10	1.10	1.15	1.09	140.99	140.74
Number of residues in the active/binding site that are successfully aligned	14	14	14	14	14	8	7	8	8	8	18	18
RMSD of aligned pairs	0.96	0.37	0.54	0.36	0.66	0.56	0.54	0.62	0.5	0.57	1.23	1.22
Speedup due to the filtering process	5.52	5.85	5.61	5.61	5.71	8.21	9.31	9.28	8.87	9.37	3.16	3.15

RMSD = root-mean-square deviation.

The first two reference structures are two enzymes in PDB, PDB ID = 1HDZ (alcohol dehydrogenase) and 1BL5 (Isocitrate dehydrogenase), and the third reference structure, PDB ID = 1L5G, contains an integrin  $\alpha V\beta 3$  bound with a peptide ligand as reported in (10). For each of the two enzyme proteins, five proteins from the same family in PDB are employed as the target proteins. For integrin, the alternative structures of integrin  $\alpha V\beta 3$  with different bindings, PDB ID = 1JV2 and 1M1X, are employed as the target proteins. Table 2 reports the results of the first experiment. The experimental results show that, with a high degree of accuracy, ProteMiner-SSM is able to identify the residues in the binding/active sites of the target protein. The only miss occurs when protein 1HJ6 is aligned with reference protein 1BL5. However, as Table 2 shows, the miss is not due to the filtering process invoked to expedite the analysis. Without the filtering process, the geometric hashing algorithm still can only successfully align seven out of the eight residues in the active site of protein 1HJ6 with the residues in the active site of the reference protein.

In the second experiment, ProteMiner-SSM is invoked to figure out whether some proteins in the Caspase family may contain a similar binding site to the structure of integrin reported in (10). Table 3 shows the results output by ProteMiner-SSM. It is observed that caspase-7, PDB ID = 1F1J and 1K86, Procaspase-7, PDB ID = 1GQF, caspase-8, PDB ID = 1F9E and caspase-9, PDB ID = 1JXQ, have the largest numbers of residues successfully aligned with the residues in the binding site of integrin. This result is in conformity with a hypothesis theorized by biochemists. However, the outputs of ProteMiner-SSM can only be regarded as interesting clues and, as shown in Table 4, it is typical that multiple possible alignments are found. Therefore, more in-depth analyses, such as protein docking or protein affinity analysis, must be conducted to further confirm the hypothesis.

The results in Tables 1–4 also show that the filtering process incorporated in ProteMiner-SSM is able to speed up the analysis by a factor ranging from 3.15 to 9.37 times. However, for the case reported in Table 3, the experimental results reveal that a certain degree of accuracy has been traded for efficiency. On the other hand, no such tradeoff has been observed for

**Table 3.** Output of ProteMiner-SSM for the second experiment

PDB ID of the target protein	1C15	1CWW	1CY5	1FIJ	1F9E	1GQF	1JXQ	1K86	1K88	1NME	2YGS
Number of residues	97	102	93	469	1476	530	940	464	461	238	92
Number of residues remaining with filtering applied	26	31	40	184	542	103	329	110	112	66	33
Geometric hashing without filtering											
Execution time of geometric hashing	5.85	6.13	5.38	65.37	429.70	81.19	217.50	63.42	63.76	21.01	5.33
Number of residues in a cave that are successfully aligned	9	10	9	16	15	14	14	14	13	12	9
RMSD of aligned pairs	3.91	3.46	3.37	4.24	3.99	4.75	4.24	4.09	4.04	4.14	3.51
Geometric hashing with filtering applied											
Execution time of filtering	0.01	0.01	0.01	0.07	0.32	0.08	0.17	0.07	0.07	0.03	0.01
Execution time of geometric hashing	0.92	1.08	1.34	14.81	91.48	9.22	44.52	8.81	9.12	3.38	1.12
Number of residues in a cave that are successfully aligned	9	8	9	13	15	13	12	12	13	11	9
RMSD of aligned pairs	3.91	4.08	3.37	4.01	4.25	5.06	4.16	4.15	4.04	3.87	3.51
Speedup due to the filtering process	6.29	5.62	3.99	4.39	4.68	8.73	4.87	7.14	6.94	6.16	4.72

**Table 4.** Two possible mappings from the second experiment of the residues in the crucial substructures of caspase-8 to the residues in the binding site of integrin  $\alpha V\beta 3$ 

Protein integrin $\alpha V\beta 3$ (reference protein)			Protein caspase-8 (PDB ID = 1F9E)			PAM250 Score
Chain	Residue index	Residue type	Chain	Residue index	Residue type	
A	178	TYR	D	320	TYR	10
A	218	ASP	A	297	GLU	3
B	119	ASP	B	388	GLN	2
B	121	SER	B	339	SER	2
B	122	TYR	B	340	TYR	10
B	123	SER	B	378	SER	2
B	126	ASP	B	351	GLN	2
B	158	ASP	A	291	GLN	2
B	215	ASN	B	381	ASP	2
B	216	ARG	B	384	LYS	3
B	217	ASP	D	323	ASP	4
B	219	PRO	D	322	PRO	6
B	220	GLU	D	324	GLU	4
B	251	ASP	B	374	ASN	2
A	150	ASP	K	289	ASN	2
A	178	TYR	K	290	TYR	10
A	218	ASP	L	385	GLN	2
B	119	ASP	K	170	ASN	2
B	121	SER	K	236	SER	2
B	122	TYR	K	244	TYR	10
B	126	ASP	K	178	ASP	4
B	127	ASP	K	180	ASN	2
B	215	ASN	K	239	ASP	2
B	216	ARG	K	240	LYS	3
B	217	ASP	K	286	GLN	2
B	218	ALA	K	284	ALA	2
B	219	PRO	L	387	PRO	6
B	220	GLU	K	283	GLN	2
B	251	ASP	V	4604	ASP	4

the cases reported in Table 2. Nevertheless, our experience is that the loss of accuracy due to the filtering process is generally within an acceptable range. In the supplementary material, we present in-depth discussions on parameter setting.

## CONCLUSION AND FUTURE WORK

In this paper, a web server designed for efficient analysis of similar protein tertiary substructures is presented. In one experiment presented in this paper, ProteMiner-SSM has

been exploited to investigate whether some proteins in the caspase family contain a similar binding site to the structure of integrin  $\alpha V\beta 3$ , and the experimental results are in conformity with the biochemical hypothesis. However, the predictions made by ProteMiner-SSM can only be regarded as interesting clues that require more in-depth investigations to be conducted. The experimental results also show that the filtering process presented in this paper is able to speed up the analysis process by a factor ranging from 3.15 to 9.37 times.

As the experiences from this research work have been encouraging, it is of interest to investigate how to extend the ideas presented in this paper to other protein analysis problems. Possible topics include protein function prediction, protein structural clustering and protein structural classification.

## SUPPLEMENTARY MATERIAL

Supplementary Material is available at NAR Online.

## ACKNOWLEDGEMENTS

This research is sponsored by National Science Council of ROC under contract NSC 92-2323-B-002-013 and NSC 92-3112-B-027-001.

## REFERENCES

1. Krane, D.E. and Raymer, M.L. (2002) *Fundamental Concepts of Bioinformatics*, Benjamin Cummings.
2. Lesk, A.M. (2002) *Introduction to bioinformatics*, Oxford University Press, New York.
3. Atkins, P.W. and Depaula, J. (2001), *Physical Chemistry, 7th edn.* W H Freeman & Co.
4. Bourne, P.E. and Weissig, H. (eds) (2003) *Structural Bioinformatics*, Wiley-Liss Inc., New Jersey.
5. Boutonnet, N.S., Rooman, M.J., Ochagavia, M.E., Richelle, J. and Wodak, S.J. (1995) Optimal protein structure alignments by multiple linkage clustering: application to distantly related proteins. *Protein Eng.*, **8**, 647–662.
6. Orengo, C. and Taylor, W. (1996) SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol.*, **266**, 617–635.
7. Pennec, X. and Ayache, N. (1994) An  $O(n^2)$  algorithm for 3D substructure matching of proteins. In Califano, A., Rigoutsos, I. and Wolson, H.J. (eds), *Shape and Pattern Matching in Computational Biology. Proceedings of the First International Workshop, Seattle*, Plenum Publishing, pp. 25–40.
8. Pennec, X. and Ayache, N. (1998) A geometric algorithm to find small but highly similar 3D substructures in proteins. *Bioinformatics*, **14**, 516–522.
9. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
10. Xiong, J.P., Stehle, T., Zhang, R., Joachimiak, A., Frech, M., Goodman, S.L. and Arnaout, M.A. (2002) Crystal structure of the extracellular segment of integrin alpha Vbeta3 in complex with an Arg-Gly-Asp ligand. *Science*, **296**, 151–155.
11. Edelsbrunner, H. and Mücke, E.P. (1994) Three-dimensional alpha shapes. *ACM Trans. Graphics*, **13**, 43–72.
12. Haim, J.W. (1997) Geometric hashing: an overview. *IEEE Comput. Sci. Eng.*, **4**, 10–21.
13. Lamdan, Y. and Wolfson, H. (1988) Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. *Proceedings of International Conference on Computer Vision*, pp. 238–249.
14. Oyang, Y.-J., Chang, D.T.-H., Chen, C.-Y. and Hwang, S.-C. (2003) Expediting Protein Structural Analysis with an Efficient Kernel Density Estimation Algorithm. *Proceedings of IEEE 5th International Symposium on Multimedia Software Engineering*, Taichung, Taiwan.
15. Oyang, Y.-J., Hwang, S.-C., Ou, Y.-Y., Chen, C.-Y. and Chen, Z.-W. (2002) A Novel Learning Algorithm for Data Classification with Radial Basis Function Networks. *Proceedings of 9th International Conference on Neural Information Processing (ICONIP-2002)*, Singapore.
16. Tu, J.-T. (2003) Protein active site prediction by matching 3D structural data. Master thesis, Department of Computer Science and Information Engineering, National Taiwan University.
17. Altschul, S.F. (1991) Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.*, **219**, 555–565.
18. Artin, E. (1964) *The Gamma Function*, Holt, Rinehart and Winston, New York.

## APPENDIX A

The efficient kernel density estimation algorithm that forms the basis of the filtering process employed in this paper treats a given set of instances  $\{s_1, s_2, \dots, s_n\}$  in the vector space as  $n$  samples randomly taken from a probability distribution and constructs an approximate probability density function of the following form:

$$\hat{f}(\mathbf{v}) = \sum_{i=1}^n w_i \exp\left(-\frac{\|\mathbf{v} - s_i\|^2}{2\sigma_i^2}\right), \quad \text{A.1}$$

where  $\mathbf{v}$  is a vector in the vector space and  $\|\mathbf{v} - s_i\|$  is the distance between vectors  $\mathbf{v}$  and  $s_i$ . Accordingly, the task that the efficient kernel density estimation algorithm carries out is to determine the values of  $w_i$  and  $\sigma_i$  in Equation A.1, so that  $\hat{f}$  provides a good approximation of the original probability density function  $f$ . In fact, the kernel density estimation problem described here can be transformed to a kernel smoothing problem, if we employ the following equation to estimate the values of  $f$  at  $s_i$ ,  $i = 1, 2, \dots, n$ :

$$f(s_i) \cong \frac{(k+1)}{n} \cdot \left[ \frac{R(s_i)^m \pi^{m/2}}{\Gamma((m/2)+1)} \right]^{-1}, \quad \text{A.2}$$

where

- (i)  $m$  is the dimension of the vector space,
- (ii)  $R(s_i)$  is the distance between instance  $s_i$  and its  $k$ -th nearest neighbor,
- (iii)  $[R(s_i)^m \pi^{m/2} / \Gamma((m/2)+1)]$  is the volume of a hypersphere with radius  $R(s_i)$  in an  $m$ -dimensional vector space,
- (iv)  $\Gamma(\cdot)$  is the Gamma function (18) and
- (v)  $k$  is a parameter to be set by the user.

As shown in Equation A.1, the efficient kernel density estimation algorithm places one spherical Gaussian function at each instance. For an instance  $s_i$ , the efficient kernel density estimation algorithm conducts a mathematical analysis on a synthesized data set. The synthesized data set is derived from two ideal assumptions and serves as an analogy of the distribution of the instances in the proximity of  $s_i$ . The first ideal assumption is that the sampling density in the proximity of  $s_i$  is sufficiently high and, therefore, the variation of the probability density function  $f$  at  $s_i$  and its neighboring instances approaches 0. The second ideal assumption is that the instances in the proximity of  $s_i$  are evenly spaced by a distance determined by the value of  $f(s_i)$ . The details of the synthesized data set are elaborated in the following:

- (i) Instance  $s_i$  is located at the origin and the neighboring instances are located at  $(h_1\delta_i, h_2\delta_i, \dots, h_m\delta_i)$ , where  $h_1, h_2, \dots, h_m$  are integers and  $\delta_i$  is the average distance between two adjacent instances in the proximity of  $s_i$ . How  $\delta_i$  is determined will be addressed later on.
- (ii) The values of the probability density function at the instances in the synthesized data set, including  $s_i$ , are all equal to  $f(s_i)$ . The value of  $f(s_i)$  is estimated based on Equation A.2.

The efficient kernel density estimation algorithm begins with an analysis on the synthesized data set to figure out the values of  $w_i$  and  $\sigma_i$  that make function  $g_i(\cdot)$  defined in the following

virtually a constant function equal to  $f(s_i)$ ,

$$g_i(\mathbf{x}) = w_i \left[ \sum_{h_1=-\infty}^{\infty} \sum_{h_2=-\infty}^{\infty} \cdots \sum_{h_m=-\infty}^{\infty} \exp\left(-\frac{\|\mathbf{x} - (h_1\delta_i, h_2\delta_i, \dots, h_m\delta_i)\|^2}{2\sigma_i^2}\right) \right] \cong f(s_i).$$

**A.3**

In other words, the objective is to make  $g_i(\mathbf{x})$  a good approximator of  $f(\mathbf{x})$  in the proximity of  $s_i$ . Let  $\mathbf{x} = (x_1, x_2, \dots, x_m)$ , then we have

$$g_i(\mathbf{x}) = w_i \cdot \sum_{h_1=-\infty}^{\infty} \exp\left(-\frac{(x_1 - h_1\delta_i)^2}{2\sigma_i^2}\right) \cdot \sum_{h_2=-\infty}^{\infty} \exp\left(-\frac{(x_2 - h_2\delta_i)^2}{2\sigma_i^2}\right) \cdots \sum_{h_m=-\infty}^{\infty} \exp\left(-\frac{(x_m - h_m\delta_i)^2}{2\sigma_i^2}\right).$$

It is shown in Appendix B that, with  $\sigma_i = \delta_i$ ,

$$2.5066282745 - 1.34 \times 10^{-8} \leq \left[ \sum_{h=-\infty}^{\infty} \exp\left(-\frac{(y - h\delta_i)^2}{2\sigma_i^2}\right) \right] \leq 2.5066282745 + 1.34 \times 10^{-8}$$

Therefore, with  $\sigma_i = \delta_i$ ,  $g_i(\mathbf{x})$  defined in Equation A.3 is virtually a constant function. In fact, it can be shown that, as long as  $\sigma_i \geq 0.45 \cdot \delta_i$ ,  $g_i(\mathbf{x})$  is virtually a constant function. Accordingly, the next thing to do is to find the appropriate value of  $w_i$  that makes  $g_i(\mathbf{x})$  approximately equal to  $f(s_i)$ . We have

$$\begin{aligned} g_i(s_i) &= g_i(0, \dots, 0) \\ &= w_i \left[ \sum_{h_1=-\infty}^{\infty} \sum_{h_2=-\infty}^{\infty} \cdots \sum_{h_m=-\infty}^{\infty} \exp\left(-\frac{(h_1^2 + h_2^2 + \dots + h_m^2)\delta_i^2}{2\sigma_i^2}\right) \right] \\ &= w_i \left[ \sum_{h=-\infty}^{\infty} \exp\left(-\frac{h^2}{2\beta^2}\right) \right]^m, \end{aligned}$$

where  $\beta = \sigma_i/\delta_i$ . Therefore, we need to set  $w_i$  as follows:

$$w_i \left[ \sum_{h=-\infty}^{\infty} \exp\left(-\frac{h^2}{2\beta^2}\right) \right]^m = f(s_i).$$

If we employ Equation A.2 to estimate the value of  $f(s_i)$ , then we have

$$w_i = \frac{(k+1) \cdot \Gamma(m/2+1)}{\lambda^m \cdot n \cdot R(s_i)^m \cdot \pi^{m/2}}, \text{ where } \lambda = \sum_{h=-\infty}^{\infty} \exp\left(-\frac{h^2}{2\beta^2}\right). \quad \mathbf{A.4}$$

So far, we have found that if we set an appropriate ratio of  $\beta = \sigma_i/\delta_i$  and set  $w_i$  according to Equation A.4, we can make

$g_i(\mathbf{x})$  a good approximator of  $f(\mathbf{x})$  in the proximity of  $s_i$ . The only remaining issue is to derive a closed form of  $\sigma_i$ . In this paper,  $\delta_i$  is set to the average distance between two adjacent instances in the proximity of sample  $s_i$ . In an  $m$ -dimensional vector space, the number of uniformly distributed instances,  $N$ , in a hypercube with volume  $V$  can be computed by  $N \cong V/\alpha^m$ , where  $\alpha$  is the spacing between two adjacent instances. Accordingly, we set

$$\delta_i = \frac{R(s_i)\sqrt{\pi}}{\sqrt[2m]{(k+1)\Gamma((m/2)+1)}}. \quad \mathbf{A.5}$$

Finally, with Equations A.4 and A.5 incorporated into Equation A.1, we obtain an approximate probability density function of the form shown in Equation 1 in the main text.

**APPENDIX B**

Let  $q(y) = \sum_{h=-\infty}^{\infty} \exp(-((y - h\delta)/2\sigma)^2)$ , where  $\delta$  and  $\sigma$  are two coefficients and  $y$  is a real number. We have

$$q'(y) = \frac{dq(y)}{dy} = \left(-\frac{1}{\sigma^2}\right) \sum_{h=-\infty}^{\infty} (y - h\delta) \exp\left(-\frac{(y - h\delta)^2}{2\sigma^2}\right).$$

Since  $q(y)$  is a symmetric and periodical function, if we want to find the global maximum and minimum values of  $q(y)$ , we only need to analyze  $q(y)$  within the interval  $[0, \frac{\delta}{2}]$ . Let  $y_0 \in [0, \frac{\delta}{2}]$  and  $y_0 = (\delta/2) \cdot (j/n) + \epsilon$ , where  $n \geq 1$  and  $0 \leq j \leq n-1$  are integers, and  $0 \leq \epsilon < \frac{\delta}{2n}$ . We have

$$q(y_0) = q\left(\frac{j\delta}{2n}\right) + \int_{j\delta/2n}^{(j\delta/2n)+\epsilon} q'(t) dt.$$

Let us consider the special case with  $\sigma = \delta$ . Then, we have

$$\begin{aligned} q(y_0) &= \sum_{h=-\infty}^{\infty} \left[ \exp\left(-\frac{1}{2}\left(\frac{j}{2n} - h\right)^2\right) \right. \\ &\quad \left. - \frac{1}{\sigma^2} \int_{j\delta/2n}^{(j\delta/2n)+\epsilon} (t - h\delta) \exp\left(-\frac{(t - h\delta)^2}{2\sigma^2}\right) dt \right]. \end{aligned}$$

Let  $r(h) = -1/\sigma^2 \int_{j\delta/2n}^{(j\delta/2n)+\epsilon} (t - h\delta) \exp(-\frac{(t - h\delta)^2}{2\sigma^2}) dt$ . Since  $(-1/\sigma^2)(t - h\delta) \exp(-\frac{(t - h\delta)^2}{2\sigma^2})$  is a decreasing function for  $t \in [(h-1)\delta, (h+1)\delta]$  and is an increasing function for  $t \notin [(h-1)\delta, (h+1)\delta]$ , we have

$$\begin{aligned} \text{(i)} \quad r(0) &\leq \epsilon \left(\frac{-1}{\sigma^2}\right) \left(\frac{j\delta}{2n}\right) \exp\left[-\frac{1}{2\sigma^2} \left(\frac{j\delta}{2n}\right)^2\right] \\ &= \epsilon \left(\frac{-1}{\sigma}\right) \left(\frac{j}{2n}\right) \exp\left[-\frac{1}{2} \left(\frac{j}{2n}\right)^2\right]; \end{aligned}$$

$$\begin{aligned} \text{(ii)} \quad r(1) &\leq \epsilon \left(\frac{-1}{\sigma^2}\right) \left(\frac{j\delta}{2n} - \delta\right) \exp\left[-\frac{1}{2\sigma^2} \left(\frac{j\delta}{2n} - \delta\right)^2\right] \\ &= \epsilon \left(\frac{-1}{\sigma}\right) \left(\frac{j}{2n} - 1\right) \exp\left[-\frac{1}{2} \left(\frac{j}{2n} - 1\right)^2\right]; \end{aligned}$$

(iii) for  $h \neq 0$  and  $h \neq 1$ ,

$$r(h) \leq \varepsilon \left( \frac{-1}{\sigma^2} \right) \left( \frac{(j+1)\delta}{2n} - h\delta \right) \times \exp \left[ -\frac{1}{2\sigma^2} \left( \frac{(j+1)\delta}{2n} - h\delta \right)^2 \right] = \varepsilon \left( \frac{-1}{\sigma} \right) \left( \frac{(j+1)}{2n} - h \right) \exp \left[ -\frac{1}{2} \left( \frac{(j+1)}{2n} - h \right)^2 \right].$$

Therefore,

$$q(y_0) = \sum_{h=-\infty}^{\infty} \left[ \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right) + r(h) \right] \leq \left[ \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right) \right] + \varepsilon\theta,$$

where

$$\theta = \left( -\frac{1}{\sigma} \right) \left( \frac{j}{2n} \right) \exp \left[ -\frac{1}{2} \left( \frac{j}{2n} \right)^2 \right] + \left( \frac{-1}{\sigma} \right) \left( \frac{j}{2n} - 1 \right) \exp \left[ -\frac{1}{2} \left( \frac{j}{2n} - 1 \right)^2 \right] + \left( \frac{-1}{\sigma} \right) \sum_{\substack{h=-\infty \\ h \neq 0,1}}^{\infty} \left( \frac{(j+1)}{2n} - h \right) \exp \left[ -\frac{1}{2} \left( \frac{(j+1)}{2n} - h \right)^2 \right].$$

If  $\theta \geq 0$ , then we have for any  $0 \leq \varepsilon < \frac{\delta}{2n}$

$$\left[ \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right) \right] + \varepsilon\theta \leq \left[ \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right) \right] + \frac{\delta}{2n}\theta.$$

On the other hand, if  $\theta < 0$ , then we have for any  $0 \leq \varepsilon < \frac{\delta}{2n}$

$$\left[ \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right) \right] + \varepsilon\theta \leq \left[ \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right) \right]. \tag{B.2}$$

Combining Equations B.1 and B.2, we obtain, for all  $y \in [0, \frac{\delta}{2}]$ ,

$$q(y) \leq \lim_{n \rightarrow \infty} \text{maximum}_{0 \leq j \leq n-1} \left\{ \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right), \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right) + \frac{\delta}{2n}\theta \right\}.$$

Similarly, we can show that

$$q(y) \geq \lim_{n \rightarrow \infty} \text{minimum}_{0 \leq j \leq n-1} \left\{ \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right), \sum_{h=-\infty}^{\infty} \exp \left( -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right) + \frac{\delta}{2n}\rho \right\},$$

where

$$\rho = \left( \frac{-1}{\sigma} \right) \left( \frac{j+1}{2n} \right) \exp \left[ -\frac{1}{2} \left( \frac{j+1}{2n} \right)^2 \right] + \left( \frac{-1}{\sigma} \right) \left( \frac{j+1}{2n} - 1 \right) \times \exp \left[ -\frac{1}{2} \left( \frac{j+1}{2n} - 1 \right)^2 \right] + \left( \frac{-1}{\sigma} \right) \sum_{\substack{h=-\infty \\ h \neq 0,1}}^{\infty} \left( \frac{j}{2n} - h \right) \exp \left[ -\frac{1}{2} \left( \frac{j}{2n} - h \right)^2 \right].$$

**B.1** If we set  $n = 100\,000$ , then we have, with  $\sigma = \delta$ ,  $2.506628261 \leq q(y) \leq 2.506628288$ , for  $y \in [0, \frac{\delta}{2}]$ .