LETTER

Using Bregmann Divergence Regularized Machine for Comparison of Molecular Local Structures

Raissa RELATOR[†a)], Nozomi NAGANO[††], *Nonmembers*, *and* Tsuyoshi KATO[†], *Member*

SUMMARY Although many 3D structures have been solved for proteins to date, functions of some proteins remain unknown. To predict protein functions, comparison of local structures of proteins with pre-defined model structures, whose functions have been elucidated, is widely performed. For the comparison, the root mean square deviation (RMSD) has been used as a conventional index. In this work, adaptive deviation was incorporated, along with Bregmann Divergence Regularized Machine, in order to detect analogous local structures with such model structures more effectively than the conventional index.

key words: *machine learning, Bregmann divergence, molecular structures, local structure comparison*

1. Introduction

Numerous macromolecules such as proteins exist in the cells of organisms. These proteins, performing various functions in the cells, comprise amino acids of 20 types that are covalently bonded sequentially through so-called peptide bonds and which are folded three-dimensionally. Such 3D forms of the proteins are called *tertiary structures* in the field of proteins. Each protein has a unique amino acid sequence, which can be folded into its own unique tertiary structure.

For the past few decades, more than a hundred thousand protein tertiary structures have been identified experimentally. Their 3D coordinate data for the atoms have been deposited in the database known as the Protein Data Bank (PDB) [1]. Some proteins, however, have unknown functionality despite their elucidated tertiary structures. It is crucially important to predict the functions of such proteins from their protein structure information in terms of biological sciences, protein engineering, and pharmaceutical sciences. In this work, a novel computational method is provided to predict the functions for such function-unknown proteins based on their protein structure information.

Although proteins usually consist of more than a hundred amino acid residues, only a few amino acid residues are adopted to perform their functions. Such amino acid residues are called *functional sites*. Even if their global structures mutually differ, the local structures of the functional sites might be similar. Such similar functional sites can carry out similar functions. For instance, functional sites in serine protease proteins (trypsin and subtilisin) can be mutually similar, even if their global structures are mutually distinct [2]. These two serine proteases perform the same function by catalyzing enzymatic reactions of the same type.

Protein sequences and tertiary structures can be clustered and classified based on their similarities. If the sequence patterns of two proteins are mutually similar with sequence identity higher than 30%, then these proteins can be clustered into the same protein category, a so-called "superfamily" or "homologous family." Even if their sequence identity is lower than 30%, the global structures of the two proteins can be mutually similar along with the same functional sites. In such cases, these proteins can also be clustered into the same "superfamily," which has been derived through divergent evolution. Relations between those proteins in the same superfamily are identifiable using sequence analyses. In contrast, two proteins that have similar functional sites showing similar function but having neither sequence similarity nor global structure similarity might result from so-called "convergent evolution." In such cases, it is impossible to identify the relations solely using sequence analysis or global structure comparison. To identify them, analogous structures of functional sites must be identified for those proteins that have neither sequence similarity nor global structure similarity.

Conventionally, local structure comparisons have been conducted to detect analogous functional sites for such proteins [3]–[5]. First, from a function-known protein, only atom coordinate data that constitute its functional site must be selected. Here, the dataset of these atoms is called a *model structure* for the functional site. Then local structures similar to the model structure must be searched to find function-unknown protein structures. Here, it is necessary to evaluate the degree of difference between a local structure candidate and the pre-defined model structure: (i) The distances between the corresponding atoms must be measured. (ii) The root mean square of the distances can be an index for the local structure similarity. This index is designated as the *root mean square deviation (RMSD)*.

The following two problems remain to be examined. The first problem is that the search performance is dependent on the number and type of the atoms in the model structure [3]. Even among the atoms within the functional site, some atoms are structurally conserved and fixed, although other atoms can differ considerably. If such spa-

Manuscript received May 1, 2015.
Manuscript revised September 18, 2015.
Manuscript publicized October 6, 2015.

[†]The authors are with the Department of Computer Science, Gunma University, Kiryu-shi, 376–8515 Japan.

[††]The author is with the AIST Computational Biology Research Center, Tokyo, 135–0064 Japan.

a) E-mail: relator-raissa@kato-lab.cs.gunma-u.ac.jp

DOI: 10.1587/transinf.2015EDL8104

tially differing atoms are included in the model structure, then the RMSD value between the local structure and the model structure can be too large to enable detection.

The second problem is that, once the atoms of the functional site are included in the model structure, they are processed similarly. Some atoms are crucially important to distinguish the functional site from other sites, although others are much less important. Nevertheless, local structure similarities are evaluated between the local structures and the pre-defined model structure simply by measuring the average squared deviation.

In this study, adaptive deviation measure is introduced to address the aforementioned two problems. A parametric deviation measure is employed to discriminate positive structures with negative structures, where positive structures have the same function as the model structure, and negative structure do not. A new machine learning algorithm is devised to determine the parameters of the deviation measure statistically from the set of positive structures and the set of negative structures collected in advance. In the machine learning algorithm, a loss function is devised. The parameters are determined by minimizing the average loss. Furthermore, for minimization of the average loss, we show that the optimization problem is in a class of Bregmann divergence regularized machines (BDRM) [6]. There is a similar popular algorithm called RFTL [7]. The update rule for both methods uses only one data point at each iterate. Compared to RFTL, the most prominent advantage of BDRM is that BDRM directly minimizes the sum of regularized losses, whereas RFTL tries to minimize the upper bound of this sum. Experimental results are shown to illustrate that the adaptive deviation measures are superior to the classical deviation measure.

2. Local Structure Comparison Method

In this paper, the following *local structure comparison problem* is posed. The input in this problem is composed of a model structure describing a functional site of interest and function-unknown macromolecule structures such as protein tertiary structures. The task is to identify the functional sites, if they exist, in the function-unknown macromolecule structure; otherwise, the comparison algorithm will output “Not Found”.

The typical procedure for the local structure comparison problem consists of two stages: Local Structure Search stage and Deviation Computation stage. In the Local Structure Search stage, a geometrical hashing algorithm such as TESS [4] or JESS [5] is used to enumerate local structures whose shapes are possibly similar to the model structure. The local structures contain every atom in the model structure.

Conventionally, the mean square deviation between the model structure and each candidate of local structures is computed in the Deviation Computation stage. Suppose a model structure has n atoms. Then each local structure candidate also contains exactly n atoms. These two sets of n atoms have a one-to-one correspondence. Basically, the type of the i-th atom in a local structure candidate is the same as the type of the i-th atom in the model structure. Let us denote by $\boldsymbol{m}_i \in \mathbb{R}^3$ the 3D coordinates of the i-th atom in the local site candidate, and by $\boldsymbol{u}_i \in \mathbb{R}^3$ the 3D coordinates for the model structure. Let $\mathcal{M} := \{\boldsymbol{m}_1, \dots, \boldsymbol{m}_n\}$ and $\mathcal{U} := \{\boldsymbol{u}_1, \dots, \boldsymbol{u}_n\}$. Structures are usually rotated and translated arbitrarily. The deviation between the corresponding atoms is computed using an appropriate rigid-body transformation $(\boldsymbol{A}, \boldsymbol{t})$ as $d_{\boldsymbol{A},\boldsymbol{t}}(\boldsymbol{u}, \boldsymbol{m}) := \|\boldsymbol{m} - \boldsymbol{A}\boldsymbol{u} - \boldsymbol{t}\|$. If we define a vector-valued function $\boldsymbol{d}_{\boldsymbol{A},\boldsymbol{t}}(\mathcal{U}, \mathcal{M})$ so that the i-th entry in the returned value is $d_{\boldsymbol{A},\boldsymbol{t}}(\boldsymbol{u}_i, \boldsymbol{m}_i)$, the classical deviation, RMSD, of an unknown local structure $\mathcal{U}$ from the model structure $\mathcal{M}$ is given by

$$D_{\boldsymbol{A},\boldsymbol{t}}(\mathcal{U}, \mathcal{M}) := \sqrt{\frac{1}{n} \|\boldsymbol{d}(\mathcal{U}, \mathcal{M}; \boldsymbol{A}, \boldsymbol{t})\|^2}$$

where $\boldsymbol{A} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{t} \in \mathbb{R}^3$ are parameters for rigid-body transformation; $\boldsymbol{A}$ and $\boldsymbol{t}$ represent a rotation matrix and a translation vector, respectively, with the rotation matrix satisfying $\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}$. The two parameters $(\boldsymbol{A}, \boldsymbol{t})$ that minimize the deviation are used, and the minimizer can be found in $O(n)$ computation.

We consider introducing a parametric adaptive deviation defined as the following quadratic form:

$$D_{\boldsymbol{A},\boldsymbol{t}}(\mathcal{U}, \mathcal{M}; \boldsymbol{w}) := \sqrt{\boldsymbol{d}_{\boldsymbol{A},\boldsymbol{t}}(\mathcal{U}, \mathcal{M})^\top \mathrm{diag}(\boldsymbol{w}) \boldsymbol{d}_{\boldsymbol{A},\boldsymbol{t}}(\mathcal{U}, \mathcal{M})} \tag{1}$$

where the n-dimensional non-negative vector $\boldsymbol{w}$ is the adaptive parameter adjusted by our learning algorithm.

3. Learning Algorithm

In this work, an adaptive deviation (1) is employed, and the model parameters $\boldsymbol{w}$ are determined by learning from a training data set.

A data set for training is constructed as follows.

1. Collect macromolecular structures whose functional sites are known.
2. Apply a geometric hashing algorithm to the collected macromolecular structures to get the local structures that are possibly similar to the model structure.
3. For each local structure obtained in the previous step, give a class label $y_i \in \{\pm 1\}$. If the local site is actually a functional site represented by the model structure, $y_i = +1$; otherwise, $y_i = -1$.

The number of atoms in each local structure is equal to the number of atoms in the model structure, n. Moreover, each atom in the local structure bijectively corresponds to one of the n atoms in the model structure. Suppose that ℓ local structures are obtained. Then, the 3D coordinates of the k-th local site are expressed as $\boldsymbol{u}_{1,k}, \dots, \boldsymbol{u}_{n,k} \in \mathbb{R}^3$, and for future reference, we define $\mathcal{U}^{(k)} := \{\boldsymbol{u}_{1,k}, \dots, \boldsymbol{u}_{n,k}\}$.

This work employs the regularized loss minimization approach using a regularization function and a loss function. The regularized loss minimization is similar to the MAP estimation [8] that maximizes the product of the likelihood function and the prior density function, where the likelihood function corresponds to the loss function and the prior density function to the regularization function. The loss function is designed so that, for positive structures, a non-zero loss is given if the deviation is smaller than a threshold θ, and for negative sites, a non-zero loss is given if the weighted square deviation is greater than the threshold θ. The definition of the loss function devised in this study is given by

$$\begin{aligned}&\mathrm{Loss}_{\boldsymbol{A}^{(k)},\boldsymbol{t}^{(k)}}(\boldsymbol{w},\theta;\boldsymbol{\mathcal{U}}^{(k)})\\ &:=\begin{cases}\left(D_{\boldsymbol{A}^{(k)},\boldsymbol{t}^{(k)}}(\boldsymbol{\mathcal{U}}^{(k)}\mathcal{M};\boldsymbol{w})^2-\theta\right)_+^2 & \text{if } y_k=+1\\ \left(\theta-D_{\boldsymbol{A}^{(k)},\boldsymbol{t}^{(k)}}(\boldsymbol{\mathcal{U}}^{(k)}\mathcal{M};\boldsymbol{w})^2\right)_+^2 & \text{if } y_k=-1\end{cases}\end{aligned}$$

The rigid-body parameters $(\boldsymbol{A}^{(k)},\boldsymbol{t}^{(k)})$ are set to the minimizers of $D_{\boldsymbol{A},\boldsymbol{t}}(\boldsymbol{\mathcal{U}}^{(k)},\mathcal{M})$, and the operator $(\cdot)_+$ is defined as $\forall x\in\mathbb{R},\ (x)_+=\max(x,0)$.

The regularization function is designed as follows. The prior distribution in the MAP estimation is often chosen so that the estimated model is non-informative if the training data is not given. Similarly, the regularization function employed in this study is minimized when no data is provided. In this work, the most non-informative parameters are equal weights:

$$w_1=\cdots=w_n=1/n.$$

Hence, we constructed the regularization function so that it is minimized at $\boldsymbol{w}_0=[w_{1,0},\dots,w_{n,0}]^\top:=\boldsymbol{1}_n/n$. The following two regularization functions are employed in this analysis:

- *ℓ_2 regularization*

$$r_2(\boldsymbol{w}):=\frac{1}{2}\|\boldsymbol{w}-\boldsymbol{w}_0\|^2.$$

- *KL(Kullback Leibler) regularization*

$$r_{\mathrm{KL}}(\boldsymbol{w}):=\sum_{i=1}^{n}w_i\log\frac{w_i}{w_{i,0}}-w_i+w_{i,0}.$$

Introducing such a regularizer, the regularized loss is defined as

$$\begin{aligned}&\mathrm{rLoss}_{\boldsymbol{A}^{(k)},\boldsymbol{t}^{(k)}}(\boldsymbol{w},\theta;\boldsymbol{\mathcal{U}}^{(k)})\\ &:=r(\boldsymbol{w})+\frac{\lambda}{2}(\theta-\theta_0)^2+\mathrm{Loss}_{\boldsymbol{A}^{(k)},\boldsymbol{t}^{(k)}}(\boldsymbol{w},\theta;\boldsymbol{\mathcal{U}}^{(k)}).\end{aligned}$$

where either ℓ_2 regularization or KL regularization is used as the regularization function $r(\cdot)$, λ is a positive constant, and θ_0 is set to 1 in the experiments described in the succeeding section. As a result, we obtain the following minimization problem:

$$\begin{aligned}&\min && \frac{1}{\ell}\sum_{k=1}^{\ell}\mathrm{rLoss}_{\boldsymbol{A}^{(k)},\boldsymbol{t}^{(k)}}(\boldsymbol{w},\theta;\boldsymbol{\mathcal{U}}^{(k)})\\ &\text{wrt} && \boldsymbol{w}\in\mathbb{R}_+^n,\quad \theta\in\mathbb{R}_+.\end{aligned}\tag{2}$$

3.1 Optimization Algorithm

Both ℓ_2 regularization function and KL regularization function are examples of Bregman divergence, which is defined as follows. Let φ be a seed function which is continuously-differentiable, real-valued, and strictly convex. A *Bregman divergence* D_φ : dom$\varphi\times$ri(domφ) → $[0,+\infty)$ is constructed with φ as

$$D_\varphi(\boldsymbol{x};\boldsymbol{y}):=\varphi(\boldsymbol{x})-\varphi(\boldsymbol{y})-\langle\boldsymbol{x}-\boldsymbol{y},\nabla\varphi(\boldsymbol{y})\rangle,$$

where domφ is the domain of φ, and ri(domφ) is the relative interior of domφ. For example, the seed function $\varphi(\boldsymbol{w})=\frac{1}{2}\|\boldsymbol{w}\|^2$ yields $D_\varphi(\boldsymbol{w};\boldsymbol{w}_0)=r_2(\boldsymbol{w})$, and $\varphi(\boldsymbol{w})=\sum_{i=1}^{n}w_i\log w_i$ generates $D_\varphi(\boldsymbol{w};\boldsymbol{w}_0)=r_{\mathrm{KL}}(\boldsymbol{w})$.

The learning algorithm (2) is an instance of BDRM [6] with either of the two regularization functions, $r_2(\cdot)$ and $r_{\mathrm{KL}}(\cdot)$, but with an additional constraint that the weight coefficients have to be non-negative when using $r_2(\cdot)$. The resulting algorithms are shown in Algorithm 1 for ℓ_2 regularization and in Algorithm 2 for KL regularization, where $\boldsymbol{a}_k=[a_{1,k},\dots,a_{n,k}]^\top$ is an n-dimensional vector whose entry is defined as $a_{i,k}:=y_kd_{\boldsymbol{A}^{(k)},\boldsymbol{t}^{(k)}}(\boldsymbol{u}_{i,k},\boldsymbol{m})$.

Algorithm 1 ℓ_2 Regularized Learning Algorithm

```
$(\boldsymbol{\alpha},\boldsymbol{\beta}):=(\boldsymbol{0}_\ell,\boldsymbol{0}_d)$; $(\boldsymbol{w},\theta):=(\boldsymbol{w}_0,\theta_0)$;
for $t=1,2,\dots$ do
    for $k=1,\dots,\ell$ do
        $\bar{\boldsymbol{w}}:=\boldsymbol{w}+\alpha_k\boldsymbol{a}_k$;        $\bar{\theta}:=\theta-\frac{1}{\lambda}\alpha_ky_k$;
        $\alpha_k:=\frac{(\langle\boldsymbol{a}_k,\bar{\boldsymbol{w}}\rangle-y_k\bar{\theta})_+}{1/\lambda+1/c_k+\|\boldsymbol{a}_k\|^2}$;
        $\boldsymbol{w}_{-1/2}:=\bar{\boldsymbol{w}}-\alpha_k\boldsymbol{a}_k$;        $\theta:=\bar{\theta}+\frac{1}{\lambda}\alpha_ky_k$;
        $\boldsymbol{w}:=(\boldsymbol{w}_{-1/2}-\boldsymbol{\beta})_+$;        $\boldsymbol{\beta}:=\boldsymbol{w}^t_{-1/2}-\boldsymbol{\beta}-\boldsymbol{w}$;
    end for
end for
```

Algorithm 2 KL Regularized Learning Algorithm.

```
$\boldsymbol{\alpha}:=\boldsymbol{0}_\ell$; $\boldsymbol{v}:=\log(\boldsymbol{w}_0)$; $\theta:=\theta_0$;
for $t=1,2,\dots$ do
    for $k=1,\dots,\ell$ do
        $\bar{\boldsymbol{v}}:=\boldsymbol{v}+\alpha_k\boldsymbol{a}_k$;
        $\bar{\theta}:=\theta-\frac{1}{\lambda}\alpha_ky_k$;
        if $\langle\boldsymbol{a}_k,\exp(\bar{\boldsymbol{v}})\rangle\le y_k\bar{\theta}$ then
            $\alpha_k:=0$;
        else
            Find positive $\alpha_k$ satisfying a nonlinear equation
                $\langle\boldsymbol{a}_k,\exp(\bar{\boldsymbol{v}}-\alpha_k\boldsymbol{a}_k)\rangle=y_k\bar{\theta}+(1/\lambda+1/c_k)\alpha_k$.
        end if
        $\boldsymbol{v}:=\bar{\boldsymbol{v}}-\alpha_k\boldsymbol{a}_k$;        $\theta:=\bar{\theta}+\frac{1}{\lambda}\alpha_ky_k$;
    end for
end for
```

4. Experimental Results

We used protein structures in PDB datasets [1] to evaluate the performance of our methods. We used PDB entries for which information on functional classification and the amino acid residues of functional sites have been annoted [9]. A total of 31 enzyme classes having a sufficient number of proteins were chosen, and these model structures were generated from the amino acid residues in their active sites. The resulting dataset contains 5,531 protein structures, some of them containing multiple active sites. At least 1,176 structures were used as positive sites and 5,518 structures were used as negative sites in the experiments.

Each amino acid in the active site is classified into one of four types: catalytic-site residues, cofactor binding site residues, modified residues, and mainchain catalytic residues. For cofactor binding site residues, all atoms are included in the model structure, whereas atoms from the sidechains of residues were included in the model structure for modified residues and catalytic-site residues. For mainchain catalytic residues, only mainchain atoms were included in the model structure.

Two proposed methods, ℓ_2 and KL Regularized BDRMs (abbreviated to L2 and KL), were examined. The two methods were compared with the classical deviation, RMSD.

Half of the 5,531 protein structures in the data set were chosen randomly and used for training, and the rest is used for testing. This is repeated three times to get three divisions. Sensitivity at 95% specificity and ROC score are obtained from each of testing data set. Sensitivity is defined as the ratio of true positives that are correctly identified whereas specificity is the ratio of true negatives that are correctly identified. ROC score is the area under the ROC curve, which plots the ratio of true positives against the ratio of false positives for different possible thresholds. Since 31 model structures are used in three divisions, 93 sensitivities and 93 ROC scores were obtained. The averages are reported hereinafter.

Figure 1 illustrates the average ROC score and the average sensitivity. For both evaluation measures, L2 achieved the highest performance. The performance of RMSD was the lowest among the three methods. To detect the statistical difference, one-sample t-test was performed. P-value for the difference of ROC scores between L2 and RMSD was 1.14×10^{-6} and P-value for KL and RMSD was 0.0022. For sensitivity, P-values for the difference of RMSD from L2 and KL were 0.0001 and 0.0020, respectively. These facts imply that the adaptive deviations are significantly better than RMSD. We also considered iteratively optimizing $(\boldsymbol{A}, \boldsymbol{t})$ and $(\boldsymbol{w}, \theta)$, but no significant improvement was obtained for the L2 method, and there was no change in the results for the KL method.

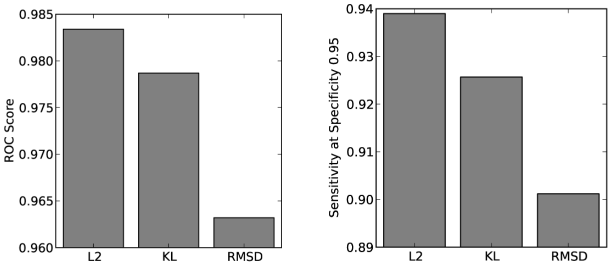

Fig. 1 Performance Comparison.

5. Conclusions

In this paper, we presented a new machine learning algorithm that determines the model parameters for the adaptive deviation used for local structure comparison. Experimental results on the real-world data revealed the effectiveness of the adaptive deviation learned with the new machine learning algorithm. The adaptive deviation combined with Bregmann Divergence Regularized Machine (L2 and KL) enabled us to detect analogous functional sites more effectively than classical deviation measure (RMSD) that are adopted by previous methods, which will lead to more efficient prediction of protein functions from their tertiary structures.

References

[1] P.W. Rose, A. Prlic, C. Bi, W.F. Bluhm, C.H. Christie, S. Dutta, R.K. Green, D.S. Goodsell, J.D. Westbrook, J. Woo, J. Young, C. Zardecki, H.M. Berman, P.E. Bourne, and S.K. Burley, "The rcsb protein data bank: views of structural biology for basic and applied research and education," Nucleic Acids Res, vol.43, no.D1, pp.D345–D356, Jan. 2015.

[2] C.S. Wright, "Comparison of the active site stereochemistry and substrate conformation in α-chymotrypsin and subtilisin BPN'," J. Mol. Biol., vol.67, no.1, pp.151–63, June 1972.

[3] G.J. Kleywegt, "Recognition of spatial motifs in protein structures," J. Mol. Biol., vol.285, no.4, pp.1887–1897, Jan. 1999.

[4] A.C. Wallace, N. Borkakoti, and J.M. Thornton, "TESS: a geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases, application to enzyme active sites," Protein Sci, vol.6, no.11, pp.2308–2323, 1997.

[5] J.A. Barker and J.M. Thornton, "An algorithm for constraint-based structural template matching: application to 3D templates with statistical analysis," Bioinformatics, vol.19, no.13, pp.1644–1649, Sept. 2003.

[6] T. Kato, W. Takei, and S. Omachi, "A discriminative metric learning algorithm for face recognition," IPSJ Transactions on Computer Vision and Applications, vol.5, pp.85–89, 2013.

[7] E. Hazen, Optimization for Machine Learning, ch. The Convex Optimization Approach to Regret Minimization, pp.281–297, MIT Press, 2011.

[8] C.M. Bishop, Pattern Recognition and Machine Learning, Springer Science+Business Media, LLC, New York, USA, 2006.

[9] N. Nagano, "EzCatDB: the enzyme catalytic-mechanism database," Nucleic Acids Res, vol.33, no.Database issue, pp.D407–D412, Jan. 2005.