BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 3 2014, pages 317–325
doi:10.1093/bioinformatics/btt694

*Sequence analysis*

Advance Access publication November 26, 2013

# Revisiting amino acid substitution matrices for identifying distantly related proteins

Kazunori Yamada and Kentaro Tomii*

Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST), 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan

Associate Editor: Alfonso Valencia

## ABSTRACT

**Motivation:** Although many amino acid substitution matrices have been developed, it has not been well understood which is the best for similarity searches, especially for remote homology detection. Therefore, we collected information related to existing matrices, condensed it and derived a novel matrix that can detect more remote homology than ever.

**Results:** Using principal component analysis with existing matrices and benchmarks, we developed a novel matrix, which we designate as MIQS. The detection performance of MIQS is validated and compared with that of existing general purpose matrices using SSEARCH with optimized gap penalties for each matrix. Results show that MIQS is able to detect more remote homology than the existing matrices on an independent dataset. In addition, the performance of our developed matrix was superior to that of CS-BLAST, which was a novel similarity search method with no amino acid matrix. We also evaluated the alignment quality of matrices and methods, which revealed that MIQS shows higher alignment sensitivity than that with the existing matrix series and CS-BLAST. Fundamentally, these results are expected to constitute good proof of the availability and/or importance of amino acid matrices in sequence analysis. Moreover, with our developed matrix, sophisticated similarity search methods such as sequence–profile and profile–profile comparison methods can be improved further.

**Availability and implementation:** Newly developed matrices and datasets used for this study are available at http://csas.cbrc.jp/Ssearch/.

**Contact:** k-tomii@aist.go.jp

**Supplementary information:** Supplementary data are available at *Bioinformatics* online

Received on August 26, 2013; revised on October 30, 2013; accepted on November 21, 2013

## 1 INTRODUCTION

Protein sequence comparison methods are fundamental tools in contemporary biology. Currently they are used widely in various fields of bioinformatics such as computational genomics and proteomics, and computational evolutionary biology. Pairwise alignment is the basis of protein sequence comparison methods. Consequently, improving pairwise amino acid sequence comparison methods can engender improved quality of studies in the field of computational biology.

A few factors are necessary to improve pairwise amino acid sequence comparison methods. Minimum units that govern the methods consist of comparison algorithms and amino acid substitution matrices, also called similarity/mutation/scoring matrices. Development and improvement of a matrix are crucial for improvement of the methods. Many studies have been undertaken since the first compilation of such matrices (Tomii and Kanehisa, 1996) and have been developed and improved along the following three lines. (i) Specific matrices are proteins with distinctive amino acid compositions. Consequently, it is reasonable to construct matrices for distinctive protein classes or proteins encoded in the genome under directional mutation pressures. For instance, matrices specialized for transmembrane regions (Muller *et al.*, 2001; Ng *et al.*, 2000) and for $\beta$-barrel membrane proteins (Jimenez-Morales and Liang, 2011; Jimenez-Morales *et al.*, 2008) have been developed. Matrices for particular proteins/organisms have also been constructed (Ali *et al.*, 2012; Brick and Pizzi, 2008; Dimmic *et al.*, 2002; Kuznetsov, 2011; Lemaitre *et al.*, 2011). Aside from those matrices, a general scheme for the compositional adjustment of matrices has been proposed (Yu *et al.*, 2003). (ii) Optimized matrices: starting from the existing matrices, some superior matrices have been derived by maximizing the ability to discriminate between homologs and non-homologs (Hourai *et al.*, 2004; Kann *et al.*, 2000; Saigo *et al.*, 2006), and to obtain accurate alignments (Qian and Goldstein, 2002) with optimization methods. (iii) Context-dependent matrices: from the pioneering work of constructing $400 \times 400$ doublet-type (= dipeptide) substitution matrices (Gonnet, *et al.*, 1994), several approaches along this line have been proposed (Crooks *et al.*, 2005; Gambin *et al.*, 2002; Huang and Bystroff, 2006; Jung and Lee, 2000; Liu and Zhao, 2010). More recently, a novel similarity search method called CS-BLAST, which uses information of neighboring residues extensively to improve similarity search, has been developed (Biegert and Soding, 2009). In this method, the similarity search is performed with no amino acid substitution matrix. CS-BLAST, which is reportedly of high detection performance, presents the possibility of displacement of a traditional amino acid substitution matrix (Angermuller *et al.*, 2012; Biegert and Soding, 2009).

Although many matrices have been proposed, we found previously that widely used matrices, so-called general purpose matrices such as PAM (Dayhoff *et al.*, 1978) and BLOSUM (Henikoff and Henikoff, 1992), have common characteristics in terms of the results of both hierarchical cluster analysis and reproduction from amino acid indices, despite the difference in

*To whom correspondence should be addressed.

datasets, methods, and models used for obtaining them (Tomii and Kanehisa, 1996). This fact might imply the existence of common ground among general purpose matrices. It is worth investigating the potential for development. To this end, we explore effective matrices for identifying distantly related proteins based on the prevailing matrices. We explored and identified effective matrice(s) in the PCA subspace through the intensive benchmark analyses, and inferred the most sensitive matrix. That matrix, designated as MIQS, with SSEARCH outperforms the existing matrices and CS-BLAST on an independent dataset, in terms of sensitivity. We argue that substitution matrices are useful for amino acid similarity search.

## 2 METHODS

### 2.1 Datasets

We used a non-redundant subset of SCOP (1.75 release) (Andreeva *et al.*, 2008) domain sequences from ASTRAL (Chandonia *et al.*, 2004) for both training and validation of our method. The subset, SCOP20, consists of all SCOP domains with 20% maximum pairwise sequence identity and includes 7074 sequences in total. We divided the sequences randomly into two sets for training and validation. The resulting sets respectively contain 3537 sequences.

To investigate the effects of dataset for training of our method, we prepared the other dataset with a higher threshold of sequence identity. The subset, which we call SCOP40-v, consists of SCOP domain sequences, except those included in the validation set above, with 40% maximum pairwise sequence identity. The resulting set contains 8598 sequences.

Furthermore, we prepared a test set that does not share homologous sequences with either the training or validation set to ensure the independence of the sets. We created the test set based on ever-growing CATH domain sequences (Sillitoe *et al.*, 2013). We first built a subset, the CATH20 dataset, which consisted of CATH domains (ver. 3.5.0) with 20% maximum pairwise sequence identity and which included 8203 sequences, using the PSI-CD-HIT program of CD-HIT (ver. 4.6.1) package (Fu *et al.*, 2012). Then, we obtained the test set, called CATH20-SCOP (1754 sequences), by excluding sequences related to entries in SCOP, according to SCOP/CATH mapping (Lewis *et al.*, 2013). Coordinate files of CATH domains for alignment quality evaluation were generated from corresponding PDB files using makedomains.perl (ver. 1.1) developed by Dr Martin (http://www.bioinf.org.uk/faqs/cath/).

To perform principal component analysis (PCA; see below), we used nine existing substitution matrices, in 1/3-bit units, of the three series, i.e. original BLOSUM, VTML (Muller *et al.*, 2002) and matrices developed by Benner *et al.* (1994). For brevity, we designate the last ones as BCG below. BCG1, BCG2 and BCG3 correspond to a matrix collected in 6.4–8.7, 22–29 and 74–100 PAM, respectively. The BLOSUM and BCG series are major members of the cluster containing matrices that are widely used in sequence alignments (Tomii and Kanehisa, 1996).

To ascertain whether a BLOSUM-type matrix performs well, we derived the BLOSUM20 matrix, which we called BLOP20, using the training set from SCOP20, and tested it. To construct BLOP20, we used the scripts provided by (Lemaitre *et al.*, 2011).

### 2.2 Derivation of matrices from the PCA subspace

The substitution matrix processed in this study is symmetric, and it consists of 210 elements. Therefore, the PCA was performed with the variance–covariance matrix of the nine 210-dimensional vectors as an input. Our aim is to explore and to identify sensitive region(s) to identify distantly related proteins in the PCA subspace, and to deduce and obtain the most sensitive matrix. To this end, in the PCA subspace, matrices derived from points around the existing matrices, which perform better among the nine matrices with the training set, were sampled and examined. We can produce systematically arbitrary matrices based on principal component scores (= coordinate as ($s_1$, $s_2$, $s_3$) in the PCA subspace) as follows:

$$\boldsymbol{M} = \boldsymbol{\mu} + \sum_{i=1}^{3} s_i \boldsymbol{U}_i^{\boldsymbol{T}} \tag{1}$$

Therein, $\boldsymbol{U}_i^{\boldsymbol{T}}$ represents the transpose of eigenvector PC$_i$; $s_i$ represents the coordinate on the PC$_i$ axis ($i = 1, 2, 3$). Furthermore, $\boldsymbol{M}$ represents a novel matrix; $\boldsymbol{\mu}$ represents the mean of nine matrices used for PCA. Then, elements in $\boldsymbol{M}$ are rounded off to the nearest integer values.

We used Kernel Density Estimation (KDE) to infer and confine the most sensitive region in the 3D PCA subspace based on the results of benchmarks for sampled matrices. At the KDE execution, we treated a value of ROC$_{50}$ (see below) as a density at each grid point sampled in the PCA subspace. Both PCA and KDE were conducted using R ver. 2.15.0 (R Development Core Team, 2012).

### 2.3 Gap penalty optimization and sensitivity benchmark of existing and derived matrices

To assess the sensitivity of both existing and derived matrices, we used SSEARCH (ver. 36.3.5) (Pearson, 1991) to conduct all-against-all sequence comparison of datasets. The SSEARCH results were sorted according to their statistical significance (E-value), with the most significant hits on top. Each hit is labeled as true or false positive, otherwise unknown. Above the threshold(s), we defined hits from the same *superfamily* with a query in SCOP (or the same *homologous superfamily* in CATH) as true positives, and defined hits from the different *fold* with a query in SCOP (or different *topology* in CATH) as false positives. Hits from the different *superfamily* (*homologous superfamily*), but from the same *fold* (*topology*), were classified as neither a true nor a false positive, but were classified as unknown because it is difficult to determine whether such hits are homologous or not.

With the training set from SCOP20, we tested all possible combinations of open gap penalty from −13 to −9 at 1 interval, and −2 and −1 as an extension gap penalty for each matrix. For each matrix and each combination of open and extension gap penalties, ROC$_{50}$ (see below) was calculated. Then the best (optimized) combination of open and extension gap penalties and the best ROC$_{50}$ value for each matrix was used for the subsequent analyses, i.e. for evaluation and for obtaining the most sensitive matrix. The best combination of the open and extension gap penalty and the corresponding ROC$_{50}$ value for the existing nine matrices is shown in Table 1.

### 2.4 Evaluation

Detection sensitivity and selectivity were measured using the receiver-operating characteristic (ROC) curve (Gribskov and Robinson, 1996). In this curve, the number of true positives is shown against the number of false positives with an arbitrary threshold of E-value. To compare and evaluate the performance of matrices (and also methods), ROC$_{50}$ was used as in previous reports (Lee *et al.*, 2008; Schaffer *et al.*, 2001). The ROC score is defined as the normalized area under the ROC curve; therefore, ROC$_{50}$ is the normalized area under the ROC curve up to the first 50 false positives, as

$$ROC_{50} = \frac{1}{50T}\sum_{i=1}^{50} t_i \tag{2}$$

In this equation, $T$ represents the total number of true positives in each dataset; $t_i$ represents the number of true positives up to $i$-th false positive. To observe the performance per query, we use ROC$_5$. The percentage of

**Table 1.** Optimized gap penalties and benchmark results with the training dataset of the existing nine matrices

| Matrix | Gap penalty | $ROC_{50}$ |
|---|---|---|
| BCG1 | (−11, −2) | 0.0249 |
| BCG2 | (−12, −1) | 0.0293 |
| BCG3 | (−12, −1) | 0.0358 |
| BLOSUM80 | (−13, −1) | 0.0254 |
| BLOSUM62 | (−9, −2) | 0.0299 |
| BLOSUM45 | (−13, −1) | 0.0288 |
| VTML160 | (−10, −2) | 0.0338 |
| VTML200 | (−9, −2) | 0.0361 |
| VTML250 | (−12, −1) | 0.0359 |

*Note*: Optimized open and extension penalties are shown in parentheses.

all test queries that yield larger $ROC_5$ than a given value is shown against the given value.

Furthermore, to evaluate the detection performance differences between methods statistically, we conducted the bootstrap analysis using the script provided in a report of an earlier study (Green and Brenner, 2002).

We also evaluated alignment quality: the alignment sensitivity and precision of matrices and methods. To assess the alignment quality, we compared sequence alignments with the structural alignments generated by Fr-TM-align (Pandit and Skolnick, 2008), which allows flexible alignments, and DaliLite (ver. 3.3) (Holm *et al.*, 2008) as reference alignments. The alignment sensitivity, the ratio of correctly aligned residue pairs to structurally equivalent residue pairs, is defined as $(N \cap S)/S$, where N is the number of residue pairs in the sequence alignment and S is the number of ones in the reference alignment. The alignment precision is the ratio of correctly aligned pairs to aligned pairs and is defined as $(N \cap S)/N$. First, we randomly selected a maximum of 10 domain pairs from each family in the CATH20-SCOP test set and structurally aligned each pair with Fr-TM-align and DaliLite. Among the obtained alignments, those with TM-scores >0.6 for Fr-TM-align and those with Z-scores >2 for DaliLite were used as reference alignments, respectively. Results show that reference alignments of 345 pairs from 433 different protein domains were obtained using Fr-TM-align and those of 588 pairs from 670 different domains were obtained using DaliLite. Then, the average alignment sensitivity and precision were computed respectively, and were binned by pairwise sequence identity in the reference alignment.

In evaluation of both detection sensitivity and alignment quality, we performed SSEARCH with the nine and obtained matrices, and also performed, with both default and optimized parameter set, SSEARCH, blastpgp (Altschul *et al.*, 1997) and the latest version of CS-BLAST with the K4000.crf library (Angermuller *et al.*, 2012). We used the –s option to enhance alignment quality by calculating the locally optimal Smith–Waterman alignments with blastpgp and CS-BLAST.

# 3 RESULTS

## 3.1 The PCA subspace

We performed principal component analysis (PCA) from three series of prevailing substitution matrices, i.e. BLOSUM, VTML and BCG, to uncover a region with higher sensitivity in the PCA subspace. In this study, we used three matrices for each type, i.e. nine matrices in total. Results showed that the first three components are dominant ones to represent the total variance of the nine matrices. The first three components of PCA described ~92.7% of the total variation. The first (PC1), second (PC2) and third (PC3) components described 61.1, 18.9 and 12.7%, respectively. Other components were responsible for 3.4% or less. Therefore, most necessary factors of the matrices are retained in this subspace consisting of the first three components. These three components can sufficiently explain the relation among the matrices.

In this subspace, we found substantial linearity between PC1 and the divergence of matrices for both the BLOSUM and VTML series. For BLOSUM, as the clustering threshold is decreased, their scores (= coordinate values) on PC1 are decreased. Higher-numbered VTML matrices have lower coordinate values of PC1. In fact, eigenvectors of diagonal elements are along PC1 (Supplementary Fig. S1). The divergence of BCG matrices is related mainly to PC2 instead of PC1. As the set of sequences used for constructing matrices is diverged, their coordinate values of PC2 are decreased.

## 3.2 Derivation of the most sensitive matrix

*3.2.1 Grid search* To identify a region with high sensitivity in the PCA subspace, the points around the existing sensitive matrices for the training set were sampled based on the results presented in Table 1. Among the existing matrices, BCG3, VTML200 and VTML250 are more sensitive than others in terms of $ROC_{50}$. In this study, points from −14 to 4, from −14 to 2 and from −18 to 2 at two intervals were sampled for PC1, PC2 and PC3 axes, respectively. They amounted to 990 $(= 10 \times 9 \times 11)$ samples (Fig. 1A). Matrices were calculated from PCA coordinates of those sampling points (see Section 2). For every matrix sampled in the PCA subspace, we assessed the detection sensitivity in the same manner, using SSEARCH and optimized gap penalties, as existing matrices with the training set. The best $ROC_{50}$ value for each matrix was used for the subsequent analysis. According to the result of the grid search, a matrix derived from (PC1, PC2, PC3) = (−6, −6, −6) demonstrated the best performance ($ROC_{50} = 0.0386$). Correspondingly, we also identified the most sensitive point with the SCOP40-v subset as the training set.

*3.2.2 KDE and refinement* To elucidate the most sensitive point (= matrix) in the PCA subspace, we performed Kernel Density Estimation (KDE) based on the results of the grid search above. As a density, we used the best $ROC_{50}$ value at each point, as described above. According to the result by KDE (Fig. 1B), the most densely populated, i.e. most sensitive point, was identified as (PC1, PC2, PC3) = (−4.57, −7.14, −6.57). Subsequently, we conducted grid search again to scrutinize matrices derived from around this point. We sampled and tested points (= matrices derived) from −5.5 to −4, from −8 to −6.5 and from −7.5 to −6 at 0.5 intervals for PC1, PC2 and PC3 axes, respectively. As a result of this grid search, a matrix derived from (PC1, PC2, PC3) = (−5.5, −8, −6.5) showed the best performance ($ROC_{50} = 0.0395$), with gap penalties of −10 for open and −2 for extension, in terms of $ROC_{50}$. We refer to this matrix as a Matrix to Improve Quality in Similarity search (MIQS). Correspondingly, we also identified the most sensitive matrix and gap penalties for the SCOP40-v subset. We refer to this as MIQS.SCOP40-v.

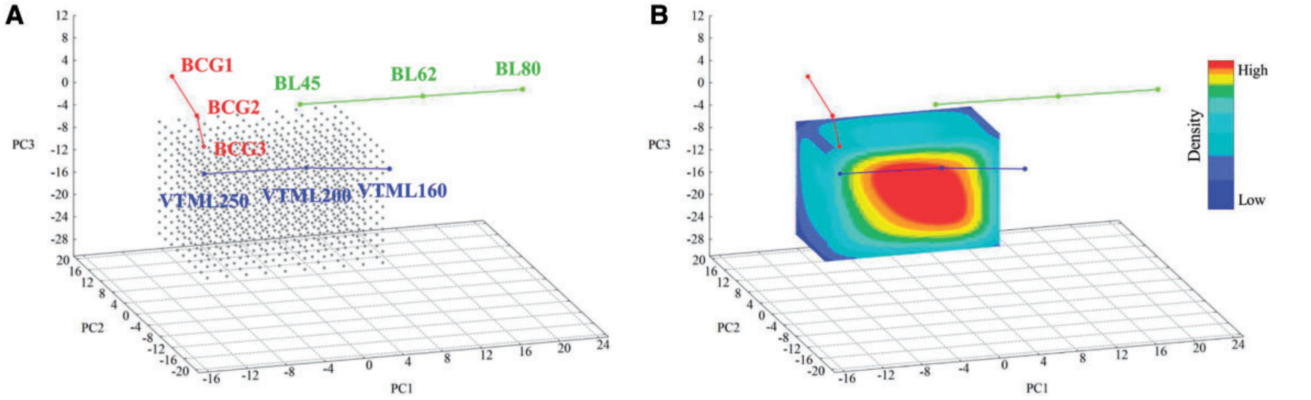

**Fig. 1.** PCA subspace constructed with the nine existing matrices and the result of grid search. The red, green and blue lines represent BCG, BLOSUM and VTML series, respectively. On the red line, each point represents BCG1, BCG2 and BCG3 in descending order of PC2 coordinate. On the green line, each point represents BLOSUM45, BLOSUM62 and BLOSUM80 in ascending order of PC1 coordinate. On the blue line, each point represents VTML250, VTML200 and VTML160 in ascending order of PC1 coordinate. (**A**) The 990 generated points for the detection performance benchmark. (**B**) Contour plot of the estimated density (= sensitivity). The color box on right side represents a kernel density corresponding to the detection performance. The cross-sectional view, which is parallel to the PC1-PC3 plane, passing through around the highest point is shown

## 3.3 Obtained matrix: MIQS

The obtained matrix was located between the VTML series and the BLOSUM series on the PC1–PC2 plane and at the lower than most of the existing matrices on the PC3 axis (Supplementary Fig. S1). According to relative entropy, our best matrix, MIQS (0.3004), is more diverse than BLOSUM62 (0.6979), even when compared with BLOSUM45 (0.3795). This might be expected because MIQS was derived using a diverse set of proteins with 20% maximum pairwise sequence identity, although the relative entropy of MIQS is slightly higher than that of BLOSUM40 (0.2851). As presented in Figure 2 (and Supplementary Fig. S2), diagonal elements, except for Gly, of the best matrix are smaller, from −1 to −4, than those of the popular matrix, BLOSUM62. By contrast, most off-diagonal elements of MIQS are larger, from 1 to 4, than those of BLOSUM62, although some off-diagonal elements of MIQS are smaller, from −1 to −3, than those of BLOSUM62. Notably, values for amino acid pair associated with Trp, such as W–C, W–Q and W–E pairs, are reduced in MIQS. Those differences remind us of the observation that chemical characteristics of amino acids are influential at high divergence (Benner *et al.*, 1994). On biplots, the W–C pair is far from 0 associated with PC1, PC2 and PC3, and W–Q and W–E pairs show extremal positions associated with PC3, reflecting the high variance in these mismatch scores between MIQS and BLOSUM (Supplementary Fig. S1).

## 3.4 Performance of obtained matrix

We evaluated both the detection performance and alignment quality of nine existing matrices, MIQS, MIQS.SCOP40-v and BLOP20 with their optimized gap penalties for the training set from SCOP20 using SSEARCH. For comparison with standard methods, we measured the performance of SSEARCH, blastpgp and CS-BLAST with both the default parameter set and the optimized gap penalties for the training set from SCOP20 in the same manner as that described above.

| | | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | -3 | 1 | 2 | 3 | 3 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | -1 | 1 | 0 | 1 | 0 | A |
| A | 3 | | -2 | 1 | 1 | 2 | 1 | -1 | 1 | 1 | 2 | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 | 1 | 2 | R |
| R | -1 | 6 | | -3 | 1 | 2 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 0 | 0 | 1 | 2 | 1 | N |
| N | 0 | 0 | 5 | | -3 | 1 | 1 | 1 | 1 | 2 | 0 | 0 | 1 | 2 | -1 | 2 | 0 | 2 | 1 | 1 | 2 | D |
| D | 0 | -1 | 3 | 6 | | -1 | 1 | 2 | 2 | 3 | 2 | 0 | 2 | 2 | 1 | 1 | 2 | 1 | -3 | 3 | 3 | C |
| C | 2 | -3 | -2 | -4 | 12 | | -4 | -1 | 1 | 0 | 2 | 1 | 0 | 1 | 3 | 2 | 0 | 1 | -2 | -1 | 1 | Q |
| Q | 0 | 2 | 1 | 1 | -3 | 4 | | -3 | 2 | 0 | 2 | 1 | 0 | 1 | 1 | 2 | 0 | 1 | -2 | 1 | 2 | E |
| E | 0 | -1 | 1 | 3 | -3 | 2 | 4 | | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | -1 | 1 | 1 | G |
| G | 0 | -2 | 0 | -1 | -2 | -2 | -1 | 8 | | -4 | 3 | 2 | 1 | 0 | 2 | 1 | 1 | 3 | 4 | -1 | 3 | H |
| H | -1 | 1 | 1 | 0 | -1 | 1 | 0 | -2 | 7 | | -1 | 1 | 2 | 0 | 1 | 0 | 1 | 0 | 3 | 1 | -1 | I |
| I | -1 | -2 | -4 | -5 | 0 | -2 | -3 | -5 | -2 | 5 | | -1 | 2 | 0 | 1 | 1 | 1 | 0 | 2 | 2 | 1 | L |
| L | -1 | -3 | -4 | -5 | -2 | -2 | -3 | -5 | -2 | 3 | 5 | | -3 | 1 | 1 | 2 | 0 | 1 | 0 | 1 | 1 | K |
| K | -1 | 3 | 1 | 0 | -3 | 2 | 1 | -2 | 0 | -2 | -2 | 4 | | -3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | M |
| M | -1 | -1 | -2 | -3 | 0 | 0 | -2 | -4 | -2 | 2 | 3 | -1 | 5 | | -2 | 1 | 1 | 1 | 3 | 1 | 1 | F |
| F | -2 | -3 | -3 | -6 | -3 | -2 | -4 | -5 | 0 | 1 | 2 | -4 | 1 | 7 | | -3 | 1 | 2 | 1 | -1 | 1 | P |
| P | 0 | -1 | -1 | 0 | -3 | 0 | 0 | -2 | -2 | -4 | -3 | 0 | -3 | -4 | 8 | | -3 | 0 | 0 | 1 | 1 | S |
| S | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | -3 | -3 | 0 | -2 | -3 | 0 | 3 | | -3 | -1 | 0 | 0 | T |
| T | 1 | -1 | 0 | 0 | 0 | 0 | 0 | -2 | 0 | -1 | -2 | 0 | -1 | -2 | 0 | 2 | 4 | | -1 | 2 | 1 | W |
| W | -4 | -4 | -5 | -5 | -6 | -5 | -6 | -5 | 0 | -1 | 0 | -4 | -2 | 4 | -4 | -4 | -5 | 15 | | -2 | 1 | Y |
| Y | -2 | -2 | -1 | -4 | -1 | -3 | -2 | -4 | 2 | -1 | 0 | -2 | -1 | 5 | -5 | -2 | -2 | 5 | 8 | | -2 | V |
| V | 0 | -2 | -3 | -3 | 2 | -2 | -2 | -4 | -2 | 3 | 2 | -2 | 1 | 0 | -3 | -1 | 0 | -3 | -1 | 4 | | |
| | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V | | |

**Fig. 2.** Comparison between obtained matrix and BLOSUM62. The obtained matrix (lower) and difference matrix (upper) obtained by subtracting BLOSUM62 from the obtained matrix are shown

*3.4.1 Detection sensitivity* When we measure the detection performance using $\mathrm{ROC}_{50}$ with the validation set, as depicted in Figure 3A, a matrix that showed the best detection performance among the existing nine matrices was VTML200, which was identical to a result with the training set. In contrast, the detection performance of our novel matrix indicated $\mathrm{ROC}_{50}$ of 0.0347, which was higher than that of VTML200 by ~10.4%, and which was almost identical to that of CS-BLAST. MIQS and MIQS.SCOP40-v showed almost identical detection sensitivity. However, $\mathrm{ROC}_{50}$ of BLOP20 was 0.0281 and it was inferior to MIQS by ~23.7%. When compared by the *superfamily*-weighted $\mathrm{ROC}_{50}$, the performance of MIQS was also higher than that of

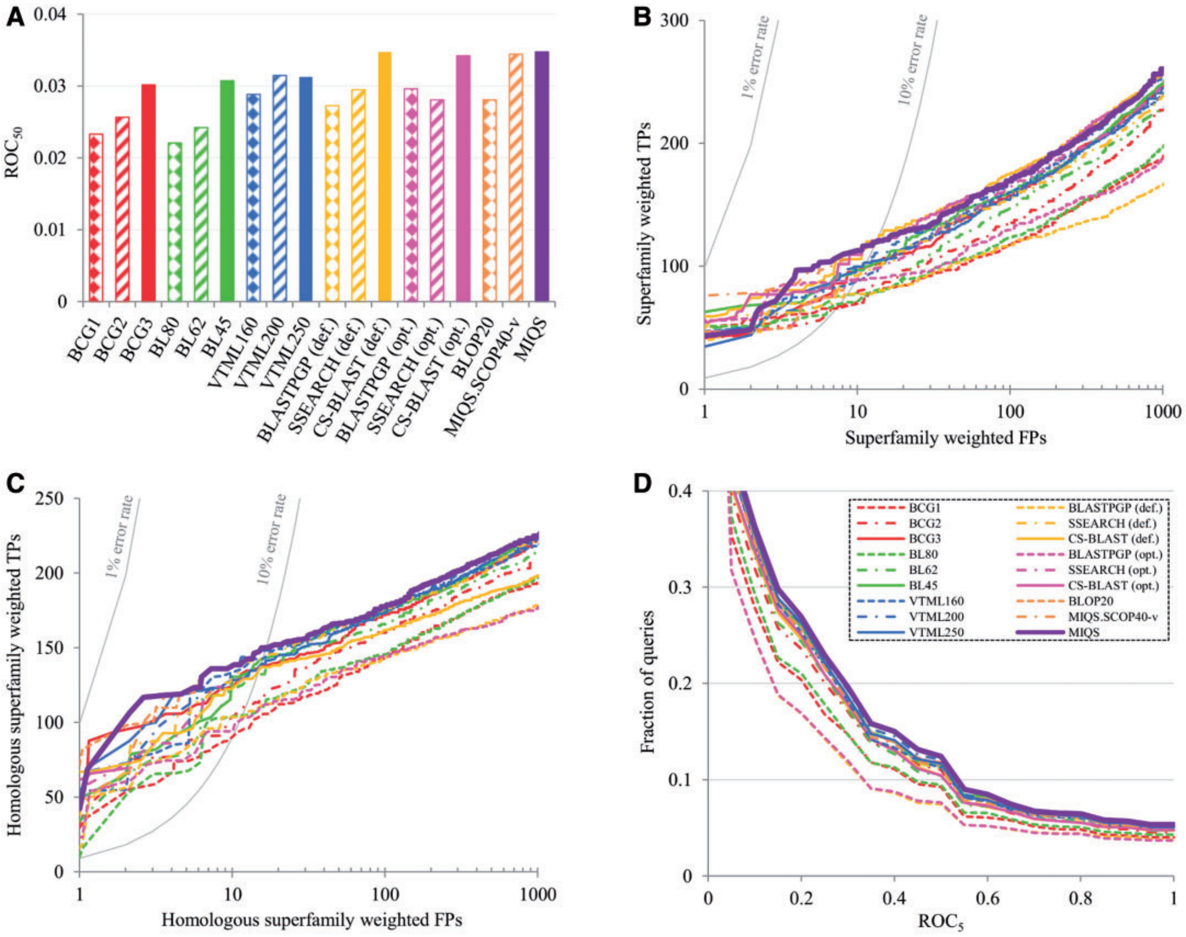

**Fig. 3.** Detection performance of developed matrix. (**A**) A comparison of ROC$_{50}$ value of the developed matrix and other existing matrices and other methods. (**B**) ROC curve of the developed matrix and other matrices and methods on the SCOP20 validation set. The purple and bold line represents the developed matrix. The number of true positive relation detected was weighted by the number of superfamily. (**C**) ROC curve on the CATH20-SCOP dataset. (**D**) ROC$_5$ curve on SCOP20 validation set. In SSEARCH, blastpgp, and CS-BLAST, (def.), present the result with the default gap penalties, and (opt.) shows the one with gap penalties optimized on the training set from SCOP20. BLOSUM50 is used for SSEARCH (def.) and (opt.), and BLOSUM62 is used for blastpgp (def.) and (opt)

VTML200, by ∼5.1%, and was almost identical to that of CS-BLAST (Fig. 3B).

Inconsistencies in SCOP classification have been noted. For example, the authors of CS-BLAST argue that SCOP has no well-classified *superfamilies* or *folds*. Then they avoid judging pairs as true or false within the four- to eight-bladed $\beta$-propellers (SCOP fold IDs: b.66−b.70), Rossman-like folds (c.2−c.5, c.30, c.66, c.78, c.79, c.111) and $\alpha$-helical and 4Fe-4S ferredoxins (a.1.2, d.58.1). The detection performance of MIQS exceeded that of the nine existing matrices and was lower than that of CS-BLAST by ∼5.8% if followed by the standard with *superfamily* weighting. Similarly, the performance of MIQS exceeded that of the existing matrices and was lower than that of CS-BLAST by ∼11.4%, when we used the ruleset for SCOP 1.61 benchmarks (Gough *et al.*, 2001).

Furthermore, we evaluated the detection performance of the matrices and methods against another test dataset, CATH20-SCOP, to ensure the independence of the dataset from the training set (Fig. 3C). Results show that, if compared by ROC$_{50}$, a matrix of the best performance detection among the existing matrices was also VTML200, whereas MIQS was slightly better than VTML200. When compared by the *homologous superfamily*-weighted ROC$_{50}$, the performance of MIQS was higher than that of VTML200 by ∼2.9% and was higher than that of CS-BLAST by ∼12.0%. In this case, the CS-BLAST performance was degraded drastically, partly because CS-BLAST can detect many homologies from a few, rather than from various, *superfamilies* (see below).

We also compared ROC$_5$ for search results of individual query (Fig. 3D). In this figure, the percentage of queries that exceeds an ROC$_5$ value is shown against the horizontal axis. This figure demonstrates how effective the testing method is in actual use because the ROC$_5$ analysis evaluates the detection performance of the testing method when very few false positives are detected: the number of false positives is five in this case. As depicted in Figure 3D, the fraction of queries of MIQS was the highest over the entire range of $ROC_5$ value among all tested matrices and methods: MIQS showed the highest performance for both easy problems (larger $ROC_5$ on horizontal axis) and difficult problems (smaller $ROC_5$ on horizontal axis).

Furthermore, enhanced evaluation supports our observation. When we perform the bootstrap analysis (Green and Brenner, 2002), at Error Per Query (EPQ) = 0.0285, which corresponds to 50 false positives in ROC curve, we observed that SSEARCH with MIQS was significantly better than CS-BLAST for the CATH20-SCOP dataset, although we were unable to find a significant difference between SSEARCH with MIQS and CS-BLAST for the SCOP20 validation dataset at EPQ = 0.0141, which corresponds to 50 false positives in ROC curve (Supplementary Fig. S3). The same results were observed at EPQ = 0.01 in both cases.

We compared the number of true positive relations detected at the number of false positives as 50 with MIQS and the other matrices and methods. In addition to this, we compared the number of true positive *superfamilies* detected with MIQS and the other matrices and methods. Comparison results are portrayed in Figure 4 as Venn diagrams. The number of true positives detected with MIQS was compared with that of VTML200 (Fig. 4A) and CS-BLAST (Fig. 4B). The number of true positive *superfamilies* detected by MIQS was also compared with that of VTML200 and that of CS-BLAST. Regarding true positive relations, MIQS was able to detect more unique true relations than VTML200 did, but less than CS-BLAST did. However, MIQS detected the same number of *superfamilies* as VTML200 (Fig. 4C) did, but more *superfamilies* than CS-BLAST did (Fig. 4D). These results indicate that, compared with VTML200, MIQS can detect more sequences from various different homologous relations. Compared with CS-BLAST, MIQS can detect various sequences from various different homologous relations.

*3.4.2 Alignment quality* The alignment quality is assessed using two standard measures: alignment sensitivity and precision, as described in Section 2.4. We compared sequence alignments with the structural alignments generated by Fr-TM-align (Fig. 5A and B) and DaliLite (Fig. 5C and D). Although, compared with the case with DaliLite, greater values of both sensitivity and precision, except for the quite low (5–10%) range of sequence identity, were observed using Fr-TM-align, which allows flexible alignments, for all matrices and methods, overall trends were preserved in both cases. Figure 5A and C shows the alignment sensitivity, and Figure 5B and D shows precision for various sequence identity bins in reference alignments. Regarding alignment sensitivity, in both cases, MIQS and BLOP20 showed comparable performance with the best one, VTML250, over almost the entire range of sequence identities in this test. In terms of alignment precision, MIQS.SCOP40-v and CS-BLAST are superior to other matrices and methods in the entire range.

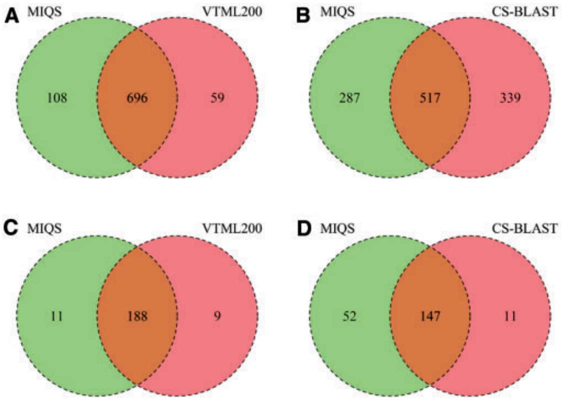

**Fig. 4.** Venn diagrams for the number of true positive relations and superfamilies detected. Comparison of true positive relations (**A** and **B**) and superfamilies (**C** and **D**) detected between the developed matrix and VTML200 and CS-BLAST

Generally, a tradeoff exists between sensitivity and precision. In terms of alignment sensitivity, SSEARCH with sensitive matrices is better than BLAST-based methods, partly because the BLAST algorithm including CS-BLAST, and also SSEARCH with MIQS.SCOP40-v, tends to generate shorter alignment (Supplementary Fig. S4). Instead, the performance of shorter alignment groups such as BLAST-based methods and SSEARCH with MIQS.SCOP40-v is exceeded in alignment precision comparison. In terms of both alignment sensitivity and precision, MIQS is balanced compared with existing matrix series, which tended to produce longer alignments, i.e. better sensitivity, with more diverged ones, and which tended to generate shorter alignments, i.e. better precision, with less diverged ones. Similar results were obtained when reference alignment was generated from SCOP20 validation dataset instead of CATH20-SCOP test dataset (Supplementary Fig. S5).

## 4 DISCUSSION

In this study, based on a previous finding, we empirically identified a region in the PCA subspace that represents a set of matrices that are suitable for detecting distantly related proteins, by combining benchmarks and PCA of the nine existing matrices from BCG, BLOSUM and VTML series. This approach differs from conventional approaches used to obtain optimized matrices. Consequently, we were able to provide a novel and highly sensitive substitution matrix, which we call MIQS, for distantly related protein sequence comparison. We were able to find that the MIQS performance with SSEARCH is superior to the sophisticated approach, CS-BLAST, in terms of detection sensitivity on an independent dataset, although CS-BLAST is clearly superior to other methods when we consider inconsistencies in the SCOP classification. This finding is expected to have a major influence on any field of protein sequence analysis that uses a substitution matrix, such as multiple alignment, profile–profile alignment and phylogeny inference.

We constructed the PCA subspace with the first three PC axes. It was thought to be sufficient for these three axes to present a relation among the existing matrices because the accumulative contribution of these axes reached ~93% of total variance. It was able to reproduce the existing matrices from the coordinate of the PCA subspace. We found relations between PC1 and the divergence of matrices, and between PC2 and the divergence of the set of sequences used for constructing matrices. We speculate that a relation exists between PC3 and the datasets, models and methods used for constructing matrices. These observations

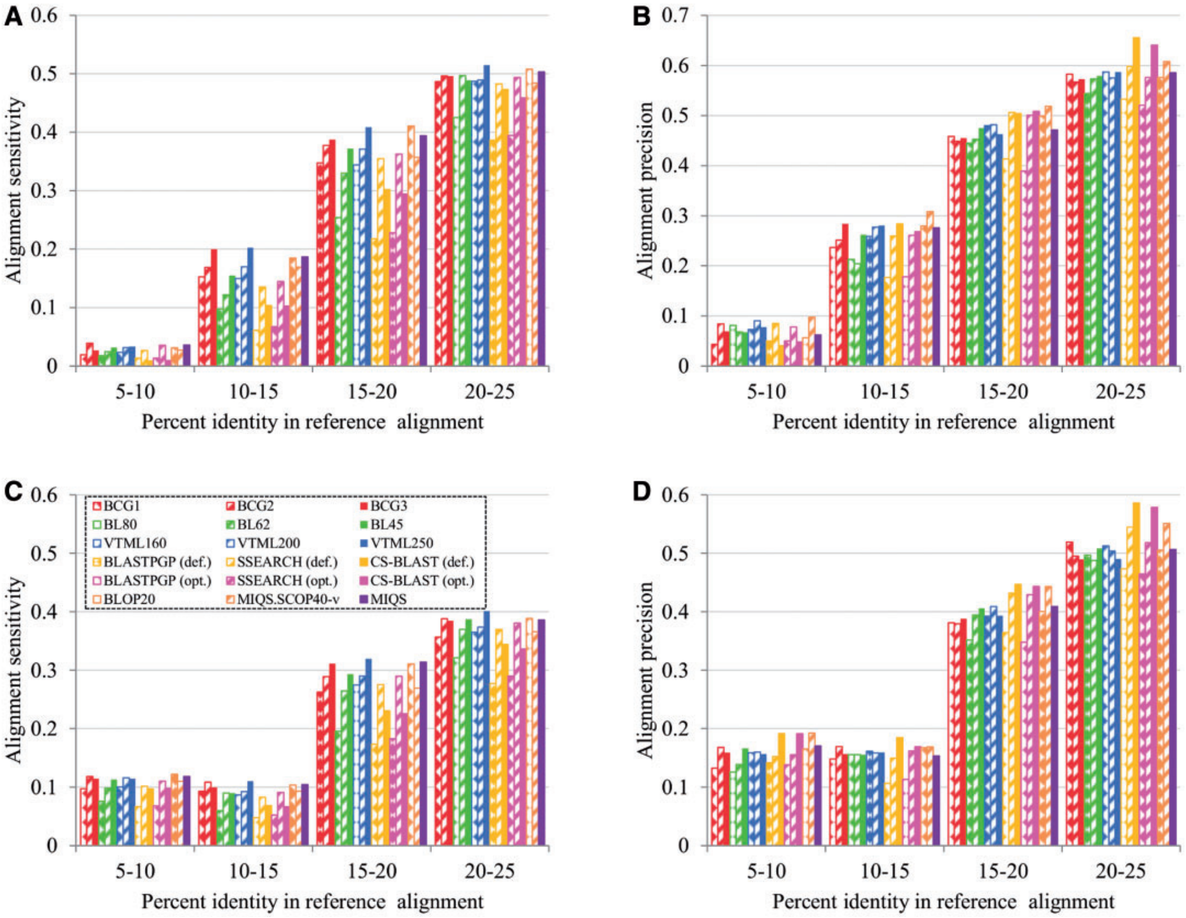

**Fig. 5.** Alignment quality of matrices and methods. Alignment sensitivity, defined as (N∩S)/S, of the developed matrix, MIQS, and other matrices and methods. Here, N denotes the number of residue pairs in an alignment and S denotes the number of residue pairs in a reference alignment. Therefore, sensitivity measures the fraction of correctly aligned residue pairs in the sequence alignment. Alignment precision, defined as (N∩S)/N, of the developed matrix and other matrices and methods. Precision measures the fraction of correctly reproduced alignment compared with the reference alignment. Reference alignments were generated, respectively, using Fr-TM-align (**A** and **B**) and DaliLite (**C** and **D**)

might imply that our projection to the PCA subspace was used to ‘de-noise’ data of amino acid substitutions, and that one can obtain an arbitrary general purpose matrix using the PCA subspace without computing actual amino acid substitutions. Associated with PC3, some mismatch pairs related to rare amino acids, such as Trp and Cys, were far from 0 (Supplementary Fig. S1). This apparently implies that PC3 has a role in adjusting the variation result from low background frequency of rare amino acids. Using our approach, one might also develop an amino acid matrix that is suitable for specific purpose such as a transmembrane matrix, i.e. an AT-biased matrix.

Our developed matrix, MIQS, exhibited extremely high-detection performance. In the ROC$_5$ curve, which is an analytical method suitable for actual sequence similarity search, MIQS showed the best detection performance over the entire range of ROC$_5$ values. We learned different characteristics of CS-BLAST from matrix-based methods. CS-BLAST can detect many true positive relations from the confined *superfamily* group, rather than from various *superfamilies*, as shown in the Venn diagrams (Fig. 4). For instance, CS-BLAST can detect a huge number of true positive relations within the c.37.1 (P-loop containing nucleoside triphosphate hydrolases) *superfamily* in SCOP. As described in Section 3, for the CATH20-SCOP test set, the performance of CS-BLAST was degraded drastically. These results suggest that CS-BLAST might be overfitted to the SCOP dataset or to some confined *superfamilies*/*homologous superfamilies* when its parameters were trained. However, according to the result portrayed in Figure 4B, combining CS-BLAST and conventional search with matrices including our developed one for similarity search is expected to be beneficial. In general, detecting the relations within larger *superfamilies* is more difficult, as pointed out in an earlier study (Green and Brenner, 2002). It is noteworthy that CS-BLAST is ~2-fold faster than SSEARCH when we search query sequences against a large database, here NCBI NR, although CS-BLAST takes time when we perform all-against-all sequence comparison (Supplementary Fig. S6).

The detection performance of a method is not necessarily proportional to the quality of sequence alignment (Vingron and Waterman, 1994). In the evaluation of alignment quality with

DaliLite, our approach is not the best, but it is well balanced and comparable to the best method(s). Another example shows that MIQS is of good alignment quality. Yu *et al*. performed compositional adjustment of amino acid substitution matrices for better alignment quality. They compared an alignment calculated using their composition-adjusted matrix and the original BLOSUM62 matrix in their report. We also aligned the same sequences they used and obtained a similar, though longer, alignment as they did without compositional adjustment (Supplementary Fig. S7). In addition, results obtained using POP (Edgar, 2009) show that MIQS (and BLOP20) is suitable for pairwise global protein alignments, although we used the Smith–Waterman local alignment method to derive MIQS (Supplementary Fig. S8). This result suggests that multiple alignment methods can be improved using MIQS.

The recently developed innovative method, CS-BLAST, does not require the use of any substitution matrix for similarity search. However, as shown in this study, the availability and importance of amino acid substitution matrices have remained. Our novel matrix, MIQS, can be useful for improving the performance of existing methods easily. In addition, strictly speaking, CS-BLAST actually requires an amino acid matrix to construct its context library. Moreover, MIQS might be useful with other advanced methods, such as profile–profile methods, to improve their performance. In the future, we will examine whether our developed matrix, MIQS, can enhance the performance of these methods.

## 5 CONCLUSION

We demonstrated in this study that, using the PCA subspace based on typical existing matrices, we were able to obtain a sensitive novel matrix, MIQS, empirically. Therefore, it is possible to use it to improve the homology detection of proteins, especially in the SCOP and CATH database, compared with existing matrices and CS-BLAST. We argue that MIQS can be useful for other database searches, and that this matrix can be influential for the improvement of sophisticated methods, such as PSI-BLAST and profile–profile comparison methods, in addition to any method using a substitution matrix in the field of bioinformatics.

## ACKNOWLEDGEMENT

The authors thank Kana Shimizu, Kenichiro Imai and Szu-Chin Fu for helpful discussions.

*Funding*: Platform for Drug Discovery, Informatics, and Structural Life Science from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

*Conflict of Interest*: none declared.

## REFERENCES

Ali,J. *et al*. (2012) The parasite specific substitution matrices improve the annotation of apicomplexan proteins. *BMC Genomics*, **13** (**Suppl. 7**), S19.

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Andreeva,A. *et al*. (2008) Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res*., **36**, D419–D425.

Angermuller,C. *et al*. (2012) Discriminative modelling of context-specific amino acid substitution probabilities. *Bioinformatics*, **28**, 3240–3247.

Benner,S.A. *et al*. (1994) Amino acid substitution during functionally constrained divergent evolution of protein sequences. *Protein Eng*., **7**, 1323–1332.

Biegert,A. and Soding,J. (2009) Sequence context-specific profiles for homology searching. *Proc. Natl Acad. Sci. USA*, **106**, 3770–3775.

Brick,K. and Pizzi,E. (2008) A novel series of compositionally biased substitution matrices for comparing Plasmodium proteins. *BMC Bioinformatics*, **9**, 236.

Chandonia,J.-M. *et al*. (2004) The ASTRAL Compendium in 2004. *Nucleic Acids Res*., **32**, D189–D192.

Crooks,G.E. *et al*. (2005) Pairwise alignment incorporating dipeptide covariation. *Bioinformatics*, **21**, 3704–3710.

Dayhoff,M.O. *et al*. (1978) A model of evolutionary change in proteins. *Atlas Protein Seq. Strut*., **5**, 345–352.

Dimmic,M.W. *et al*. (2002) rtREV: an amino acid substitution matrix for inference of retrovirus and reverse transcriptase phylogeny. *J. Mol. Evol*., **55**, 65–73.

Edgar,R.C. (2009) Optimizing substitution matrix choice and gap parameters for sequence alignment. *BMC Bioinformatics*, **10**, 396.

Fu,L. *et al*. (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**, 3150–3152.

Gambin,A. *et al*. (2002) Contextual alignment of biological sequences (Extended abstract). *Bioinformatics*, **18** (**Suppl. 2**), S116–S127.

Gonnet,G.H. *et al*. (1994) Analysis of amino acid substitution during divergent evolution: the 400 by 400 dipeptide substitution matrix. *Biochem. Biophys. Res. Commun*., **199**, 489–496.

Gough,J. *et al*. (2001) Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J. Mol. Biol*., **313**, 903–919.

Green,R.E. and Brenner,S.E. (2002) Bootstrapping and normalization for enhanced evaluations of pairwise sequence comparison. *Proc. IEEE*, **90**, 1834–1847.

Gribskov,M. and Robinson,N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem*., **20**, 25–33.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Holm,L. *et al*. (2008) Searching protein structure databases with DaliLite v.3. *Bioinformatics*, **24**, 2780–2781.

Hourai,Y. *et al*. (2004) Optimizing substitution matrices by separating score distributions. *Bioinformatics*, **20**, 863–873.

Huang,Y.M. and Bystroff,C. (2006) Improved pairwise alignments of proteins in the twilight zone using local structure predictions. *Bioinformatics*, **22**, 413–422.

Jimenez-Morales,D. and Liang,J. (2011) Pattern of amino acid substitutions in transmembrane domains of beta-barrel membrane proteins for detecting remote homologs in bacteria and mitochondria. *PLoS One*, **6**, e26400.

Jimenez-Morales,D. *et al*. (2008) Detecting remote homologues using scoring matrices calculated from the estimation of amino acid substitution rates of beta-barrel membrane proteins. *Conf. Proc. IEEE Eng. Med. Biol. Soc*., **2008**, 1347–1350.

Jung,J. and Lee,B. (2000) Use of residue pairs in protein sequence-sequence and sequence-structure alignments. *Protein Sci*., **9**, 1576–1588.

Kann,M. *et al*. (2000) Optimization of a new score function for the detection of remote homologs. *Proteins*, **41**, 498–503.

Kuznetsov,I.B. (2011) Protein sequence alignment with family-specific amino acid similarity matrices. *BMC Res. Notes*, **4**, 296.

Lee,M.M. *et al*. (2008) Simple is beautiful: a straightforward approach to improve the delineation of true and false positives in PSI-BLAST searches. *Bioinformatics*, **24**, 1339–1343.

Lemaitre,C. *et al*. (2011) A novel substitution matrix fitted to the compositional bias in Mollicutes improves the prediction of homologous relationships. *BMC Bioinformatics*, **12**, 457.

Lewis,T.E. *et al*. (2013) Genome3D: a UK collaborative project to annotate genomic sequences with predicted 3D structures based on SCOP and CATH domains. *Nucleic Acids Res*., **41**, D499–D507.

Liu,X. and Zhao,Y.P. (2010) Substitution matrices of residue triplets derived from protein blocks. *J. Comput. Biol*., **17**, 1679–1687.

Muller,T. *et al*. (2001) Non-symmetric score matrices and the detection of homologous transmembrane proteins. *Bioinformatics*, **17** (**Suppl. 1**), S182–S189.

Muller,T. *et al*. (2002) Estimating amino acid substitution models: a comparison of Dayhoff's estimator, the resolvent approach and a maximum likelihood method. *Mol. Biol. Evol*., **19**, 8–13.

Ng,P.C. *et al.* (2000) PHAT: a transmembrane-specific substitution matrix. Predicted hydrophobic and transmembrane. *Bioinformatics*, **16**, 760–766.

Pandit,S.B. and Skolnick,J. (2008) Fr-TM-align: a new protein structural alignment method based on fragment alignments and the TM-score. *BMC Bioinformatics*, **9**, 531.

Pearson,W.R. (1991) Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics*, **11**, 635–650.

Qian,B. and Goldstein,R.A. (2002) Optimization of a new score function for the generation of accurate alignments. *Proteins*, **48**, 605–610.

R Development Core Team. (2012) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Saigo,H. *et al.* (2006) Optimizing amino acid substitution matrices with a local alignment kernel. *BMC Bioinformatics*, **7**, 246.

Schaffer,A.A. *et al.* (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, **29**, 2994–3005.

Sillitoe,I. *et al.* (2013) New functional families (FunFams) in CATH to improve the mapping of conserved functional sites to 3D structures. *Nucleic Acids Res.*, **41**, D490–D498.

Tomii,K. and Kanehisa,M. (1996) Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Eng.*, **9**, 27–36.

Vingron,M. and Waterman,M.S. (1994) Sequence alignment and penalty choice. Review of concepts, case studies and implications. *J. Mol. Biol.*, **235**, 1–12.

Yu,Y.K. *et al.* (2003) The compositional adjustment of amino acid substitution matrices. *Proc. Natl Acad. Sci. USA*, **100**, 15688–15693.