Dynamic Programming for Protein Sequence Alignment

Zhi-min Zhou and Zhong-wen Chen

Department of Computer Science
Zhejiang Water Conservancy And Hydropower College, Hangzhou, China
E-mail: zhouzhm@zjwchc.com, chenzw@zjwchc.com

Abstract

Dynamic programming is a method for solving complex problems by breaking them down into simpler subproblems. This idea is very insightful for solving bioinformatics problems. Aligning distantly related protein sequences is a long-standing problem in bioinformatics and a key for successful protein structure prediction. A fast and valid algorithm can benefit the whole process of biology research. In this paper, we introduce an algorithm that given a certain evaluation function, will calculate the optimal alignment by dynamic programming.

Keywords: *algorithm, bioinformatics, protein sequence alignment*

1. Introduction

1.1. Background

In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences [1].

How to produce high-quality alignment of lengthy and extremely numerous sequences is the major difficulty facing bioinformatics researchers. Before fast algorithms such as BLAST and FASTA were developed, doing database searches for the protein or nucleic sequences was very time consuming by using a full alignment procedure like Smith-Waterman [2]. Various algorithms and programs are developed to solve these problems. This technology is also very useful in aligning other sequences in for example, natural language and financial data.

1.2. Our Result

Data set used in the illustration isCj1293 and HP0840 enzymes from the genome database of IMG (img,jgi.de.gov).

By applying multiple algorithms, we demonstrate a regular procedure of pairwise protein sequence alignment, and then compare their performances.

Before the introduction of any algorithms, we first want to make two concepts clear. The first one is Second one is pairwise and multiple alignment. The idea is easy, a pairwise alignment substitute a sequence to another while a multiple alignment to a number of sequences.

We compare 2 pairwise methods (dynamic programming and local search) and implement a multiple alignment task based on them.

Then we give a demo of showing how to query the whole genome of shewanella, a marine bacterium whose O-linked glycosylation pathway is not known, for protein similar to PseB.

2. Dynamic Programming

2.1. The Algorithm

Both global alignments (Needleman-Wunsch algorithm) and local alignment (Smith-Waterman algorithm) can be produced by applying dynamic programming.

Dynamic programming can be time consuming. In typical usage, protein alignments use a substitution matrix to assign scores to amino-acid matches or mismatches, and a gap penalty for matching an amino acid in one sequence to a gap in the other.

Figure1 depicts a Needleman-Wunsch alignment of the words "PELICAN" and "COELACANTH" [3].

2.1.1. Initialization: Construct the matrix. The number of rows and columns are the length of two sequences plus one.

2.1.2. Initialization: matrix initial configuration: Initialize the matrix, with zero on the upper left corner, and count from upper left to lower right, apply the gap penalty one by one. And arrows are added to indicate the alignment direction.

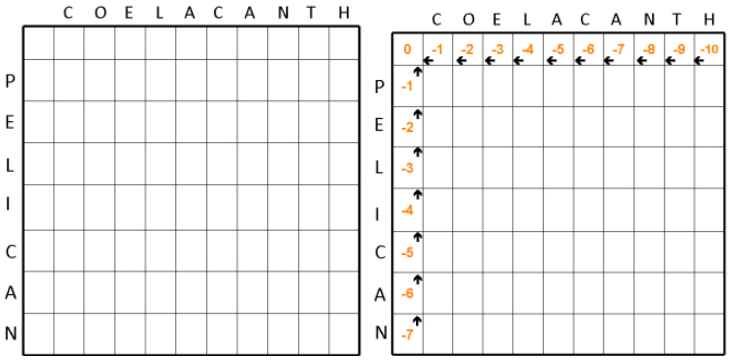


Figure 1. Initialization of the substitution matrix and the filling of axis

2.1.3. Induction: Algorithm aligns the letters sequentially, if two letters match add one point, otherwise minus one point. The new score is calculated three times, with the square to the left, above and upper left. The minimal score is chosen as the final alignment.

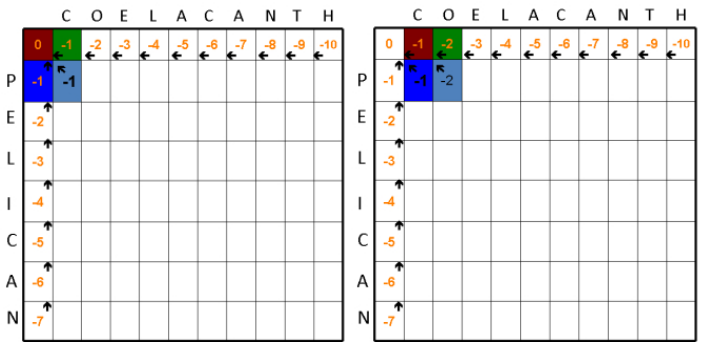


Figure 2. Induction or filling of the alignment matrix

2.1.4. Traceback: Once the matrix is completed, the optimal alignment is found from the lower right back to the beginning.

		C	O	E	L	A	C	A	N	T	H
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	10
P	-1	-1	-2	-3	-4	-5	-6	-7	-8	-9	10
E	-2	-2	-2	-1	-2	-3	-4	-5	-6	-7	-8
L	-3	-3	-3	-2	0	-1	-2	-3	-4	-5	-6
I	-4	-4	-4	-3	-3	-1	-2	-3	-4	-5	-6
C	-5	-3	-5	-4	-4	-4	0	-1	-2	-3	-4
A	-6	-4	-6	-5	-5	-3	-1	1	0	-1	-2
N	-7	-5	-7	-6	-6	-4	-2	0	2	1	0

Figure 3. Traceback of the optimal complete alignment

This algorithm meets the problem of large input scale, which requires memory of O(I2), where I is the input scale.

2.2. The Tool: BioConductor

When the sequences are reasonably short, this algorithm can be investigated.

While most researchers use sequence alignment software like ELAND, MAQ, and Bowtie to perform the bulkof short read mappings to a target genome, BioConductor contains a number of string matching/pairwisealignment tools in the Biostrings package that can be invaluable in answering complex scientificquestions.These tools are naturally divided into four groups (matchPDict, vmatchPattern, pairwiseAlignment, and OTHER) that contain the following functions[4]:

matchPDict :matchPDict, countPDict,whichPDict,vmatchPDict,vcountPDict,vwhichPDict

vmatchPattern :matchPattern, countPattern, vmatchPattern, vcountPattern, neditStartingAt, neditEndingAt, isMatchingStartingAt, isMatchingEndingAt

pairwiseAlignment :pairwiseAlignment, stringDist

OTHER :matchLRPatterns (nds singleton paired-end matches), trimLRPatterns (trims left and/orrightanking patterns), matchProbePair (nds theoretical amplicons), matchPWM (matches using aposition weight matrix)

2.3. The Result

Here is the result of pairwise sequence alignment being fit by the pairwise Alignment function. In this case evolutionary model is used with the BLOSUM50 matrix:

```
>data(BLOSUM50)
>BLOSUM50[1:4, 1:4]

A R N D
A 5 -2 -1 -2
R -2 7 -1 -2
N -1 -1 7 2
D -2 -2 2 8

> test<-pairwiseAlignment(AAString(s1),AAString(s2),substitutionMatrix=
+ BLOSUM50, gapOpening=0, gapExtension=-8)
➢  test

Global PairwiseAlignedFixedSubject (1 of 1)
MFNGKN------ILITGGTGSFGKTYTKVLLENYK...GQKVKDGFSYSSDNNPLWASEKELLEIINHTEVF
MPNHQNMLDNQTILITGGTGSFGKCFVRKVLDTTN...GQKVAPDFEYSSHNNNQWL-EPD--DLL---KLL
score: 1339
>compareStrings(test)
[1] "M?N??N------ILITGGTGSFGK??????L??????KII?YSRDELKQ?EMA??FN?P?MR?FIGDVRD?ERL??A???V
D??IHAAA?KHVPIAEYNP?ECIKTNI?GA?NVI?AC??N?????IALSTDKA?NP?NLYGATKL?SDKLFV?ANN??G??QT?F?
V?RYGNVVGSRGSVVPFFKKL????A?E?PITD?RMTRFWI?L??GV?FVL????RMHGGEIF?PKIPSMK?TDLA?ALAP????K
IIGIR?GEKLHE?MI??D?SHL??EFE????I?P?I?F+????D?????L?EKGQKV???F?YSS?NN??W?+E??++???++
+???"

>pid(test)
[1] 62.35294
```

2.4. Remarks

Dynamic programming can be applied to generate gapped alignments ith insertions and deletions. This algorithm is powerful and is sure to produce the optimal alignment corresponding to a certain scoring function. However, this method also requires lots of computational resources. Since the computational complexity is O(nm), with m, n be the lengths of each sequences, this algorithm could be too slow to be extended to align multiple sequences.

3. Local Search Alignment

3.1 The Algorithm

As we have shown in the last section, dynamic programming cannot perform well when the protein sequence is long. Researcher Stephen Altschul and colleagues wanted to bypass these challenges and develop a way for databases to be searched quickly on routinely used computers. In order to increase the speed of alignment, the BLAST algorithm was designed to approximate the results of an alignment algorithm created by Smith and Waterman in 1981 [5], but to do so without comparing every residue against every other [6-9]. BLAST is

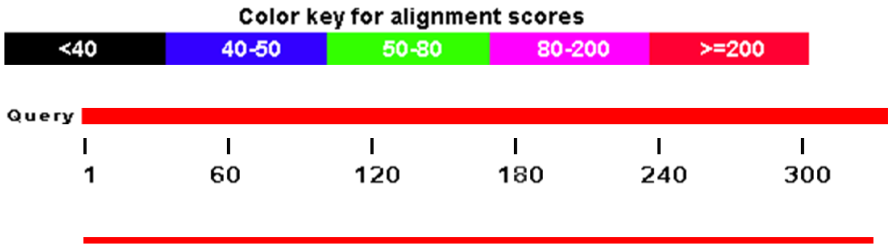
therefore heuristic in nature, meaning it has "smart shortcuts" that allow it to run more quickly [10]. However, in this trade-off for increased speed, the accuracy of the algorithm is slightly decreased.

3.2 The Tool: BLAST

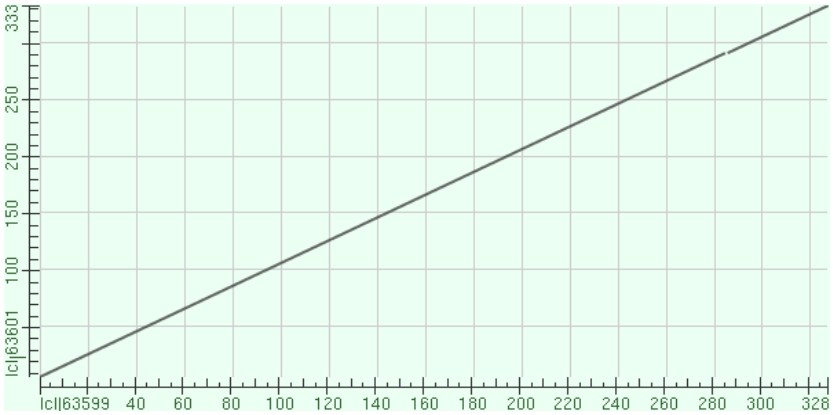
BLAST is one of the most commonly used tools for local search alignment. The main idea of BLAST is that there are often high-scoring segment pairs (HSP) contained in a statistically significant alignment. BLAST searches for high scoring sequence alignments between the query sequence and sequences in the database using a heuristic approach that approximates the Smith-Waterman algorithm. The exhaustive Smith-Waterman approach is too slow for searching large genomic databases such as GenBank. Therefore, the BLAST algorithm uses a heuristic approach that is less accurate than the Smith-Waterman algorithm but over 50 times faster. The speed and relatively good accuracy of BLAST are among the key technical innovations of the BLAST programs [11].

3.3 The Result

Distribution of Blast Hits on the Query Sequence:



Plot of cj1293 vs HP_0840:



```
>lcl|63601hp
Length=333

 Score =  446 bits (1147),  Expect = 4e-130, Method: Compositional matrix adjust.
 Identities = 211/328 (64%), Positives = 254/328 (77%), Gaps = 1/328 (0%)

Query  1    MFNGKNILITGGTGSFGKTYTKVLLENYKPNKIIIYSRDELKQFEMASIFNAPYMRYFIG  60
            M + + ILITGGTGSFGK + + +L+     KII+YSRDELKQ EMA  FN P MR+FIG
Sbjct  7    MLDNQTILITGGTGSFGKCFVRKVLDTTNAKKIIVYSRDELKQSEMAMEFNDPRMRFFIG  66

Query  61   DVRDKERLSAAMRDVDFVIHAAAMKHVPIAEYNPMECIKTNIRGAQNVIDACFENGVKKC  120
            DVRD ERL+ A+  VD  IHAAA+KHVPIAEYNP+ECIKTNI GA NVI+AC +N + +
Sbjct  67   DVRDLERLNYALEGVDICIHAAALKHVPIAEYNPLECIKTNIMGASNVINACLKNAISQV  126

Query  121  IALSTDKACNPVNLYGATKLASDKLFVAANNIAGNKQTRFGVTRYGNVVGSRGSVVPFFK  180
            IALSTDKA NP+NLYGATKL SDKLFV+ANN  G+ QT+F V RYGNVVGSRGSVVPFFK
Sbjct  127  IALSTDKAANPINLYGATKLCSDKLFVSANNFKGSSQTQFSVVRYGNVVGSRGSVVPFFK  186

Query  181  KLINEGAKELPITDTRMTRFWISLEDGVKFVLSNFERMHGGEIFIPKIPSMKITDLAHAL  240
            KL+   A E+PITD RMTRFWI+L++GV FVL + +RMHGGEIF+PKIPSMK+TDLA AL
Sbjct  187  KLVQNKASEIPITDIRMTRFWITLDEGVSFVLKSLKRMHGGEIFVPKIPSMKMTDLAKAL  246

Query  241  APHLSHKIIGIRAGEKLHEIMISSDDSHLTYEFENYYAISPSIKFVDKDNDFSINALGEK  300
            AP+   KIIGIR GEKLHE+MI  D+SHL  EFE+++ I P+I F     D+++  L EK
Sbjct  247  APNTPTKIIGIRPGEKLHEVMIPKDESHLALEFEDFFIIQPTISF-QTPKDYTLTKLHEK  305

Query  301  GQKVKDGFSYSSDNNPLWASEKELLEII  328
            GQKV   F YSS NN  W    +LL+++

  Sbjct  306  GQKVAPDFEYSSHNNNQWLEPDDLLKLL  333
```

4. Progressive Alignment

4.1 The Algorithm

Another method build upon the dynamic programming is used in multiplealignment. The idea is firstly align the 2 most closest sequences and then adds the next closest one iteratively. This algorithm progresses by dynamicly updating the positions of the indels [12]. If some part of the sequences is overpresented, this algoritm suffers from biased inference caused by the order of mofits.

This heutistic algorithm always gives good results [13]. The most reliable alignments are produced by aligning the most similar pairs of sequences. By this way, a hierarchical tree is constructed and tree analysis in unsupervised learning can be invested to find groups of alignment.

4.2 The Tool: ClustalW

One widely used implementation of profile-based progressive alignment is the CLUSTALW program. CLUSTALW works in much the same way as Feng-Doolittle method except for its carefully tuned use of profile alignment methods [14].

Algorithm:

- Construct a distance matrix of all N(N-1) pairs by pairwise dynamic programming alignment followed by approximate conversion of similarity score to evolutionary distances using the model of Kimura.
- Construct a guide tree by a neighbor-joining clustering algorithm by Saitou & Nei.

- Progressively align at nodes in order of decreasing similarity, using sequence-sequence, sequence-profile, and profile-profile alignment.

ClustalW is unabashedly ad hoc (designed for this, not generalizable) in its alignment construction and scoring stage [15]. In addition to the usual methods of profile construction and alignment, various heuristics of ClustalW contribute to its accuracy:

- Sequences are weighted to compensate for biased representation in large sub-families
- Substitution matrix used to score an alignment is chosen on the basis of the similarity expected of the alignment; closely related sequences are aligned with hard matrices (BLOSUM 80) and distant sequences are aligned with soft matrices (BLOSUM 50).
- Position-specific gap-open profile penalties are multiplied by a modifier that is a function of the residues observed at the position. Penalties are obtained from gap frequencies observed in large number of structurally based alignments.
- Gap-open penalties are also decreased if the position is spanned by a consecutive stretch of 5 or more hydrophilic residues.
- Both gap-open and gap-extend penalties are increased if there are no gaps in a column but gaps occur nearby in alignment. This rule tries to force all the gaps to occur in the same places in an alignment.
- In the progressive alignment stage, if the score of an alignment is low, the guide tree may be adjusted on the fly to defer the low-scoring alignment until later in the progressive alignment phase when more profile information has been accumulated.

4.3 The Result

The result given by ClustalW is shown in the following diagram.

```
CLUSTAL 2.1 multiple sequence alignment
cj            ------MFNGKNILITGGTGSFGKTYTKVLLENYKPNKIIIYSRDELKQFEMASIFNAPY      54
hp            MPNHQNMLDNQTILITGGTGSFGKCFVRKVLDTTNAKKIIVYSRDELKQSEMAMEFNDPR      60
          *::.:.************ :.: :*:. :.:***:******** ***  ** *
cj            MRYFIGDVRDKERLSAAMRDVDFVIHAAAMKHVPIAEYNPMECIKTNIRGAQNVIDACFE     114
hp            MRFFIGDVRDLERLNYALEGVDICIHAAALKHVPIAEYNPLECIKTNIMGASNVINACLK     120
    **:******* ***. *:..**: *****:**********:******* **.***:**::
cj            NGVKKCIALSTDKACNPVNLYGATKLASDKLFVAANNIAGNKQTRFGVTRYGNVVGSRGS     174
hp            NAISQVIALSTDKAANPINLYGATKLCSDKLFVSANNFKGSSQTQFSVVRYGNVVGSRGS     180
    *.:.: ********.**:********.******:***: *..**:*.*.***********
cj            VVPFFKKLINEGAKELPITDTRMTRFWISLEDGVKFVLSNFERMHGGEIFIPKIPSMKIT     234
hp            VVPFFKKLVQNKASEIPITDIRMTRFWITLDEGVSFVLKSLKRMHGGEIFVPKIPSMKMT     240
    ********::: *.*:**** *******:*::**.***..::********:*******:*
cj            DLAHALAPHLSHKIIGIRAGEKLHEIMISSDDSHLTYEFENYYAISPSIKFVDKDNDFSI     294
hp            DLAKALAPNTPTKIIGIRPGEKLHEVMIPKDESHLALEFEDFFIIQPTISFQTP-KDYTL     299
    ***:****: . ******.******:**..*:***: ***::: *.*:*.*    :*:::

  cj                    NALGEKGQKVKDGFSYSSDNNPLWASEKELLEIINHTEVF          334
hp                     TKLHEKGQKVAPDFEYSSHNNNQWLEPDDLLKLL------          333
    . * ******  .*.***.**  * . .:**:::
```

5. Online Tools Implementations

This family of algorithms are widely adapted by online databases and other online softwares. Following are some well-known examples.

5.1 MultiIdent

MultiIdent [16] is designed for the identification of proteins from 2-D gels. There are many properties of a protein that can be used to aid in its identification. These include:

- protein isoelectric point (estimated from a 2-D gel)
- protein molecular mass (estimated from a 2-D gel or by mass spectrometry)
- species of origin of the protein
- protein amino acid composition
- protein sequence data (sequence tag)
- peptide masses generated by enzymatic or chemical digestion followed by mass spectrometry of generated peptides.

the current version of the software works by first generating a list of best-matching proteins with amino acid composition, and then using other protein parameters (*e.g.*, protein pI, Mw, sequence tag, peptide mass data) to query this list and identify the protein of interest.

5.2 EGM

The Encapsulated Gene-by-gene Matching (EGM)[17] approach is a method that employs a graph matching strategy to identify gene orthologs and conserved gene segments. Given a pair of genomes, EGM constructs a global gene match for all genes taking into account gene context and family information. The Hungarian method for identifying the maximum weight matching in bipartite graphs is employed, where the resulting matching reveals one-to-one correspondences between nodes (genes) in a manner that maximizes the gene similarity and context. The EGM software, Supplementary information and other tools are available online from [18].

5.3 FASTA

The FASTA programs [19] find regions of local or global (new) similarity between Protein or DNA sequences, either by searching Protein or DNA databases, or by identifying local duplications within a sequence. Other programs provide information on the statistical significance of an alignment. Like BLAST, FASTA can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

5.4 FFAS

The FFAS03 [20] server provides an interface to the profile-profile alignment and fold recognition algorithm FFAS. A profile-profile alignment utilizes information present in sequences of homologous proteins to amplify the sequence conservation pattern defining the protein family. This method allows detection of remote homologies beyond the reach of other sequence comparison methods. Input into the FFAS03 server is a protein sequence provided by the user. From the sequence, a profile is generated that is then compared to several databases of sequence profiles of proteins and domains from public databases Databases. The databases are updated with the latest structural and sequence information.

References

[1] D. M. Mount, "Bioinformatics: Sequence and Genome Analysis (2nd ed.)", Cold Spring Harbor Laboratory Press: Cold Spring Harbor, NY, ISBN 0-87969-608-7, **(2004)**.
[2] I. Korf, M. Yandell and J. Bedell, "BLAST", O'Reilly Media Inc., ISBN: 9780596002992, **(2003)**.
[3] S. F. Altschul, *et al.*, "Issues in searching molecular sequence databases", Nature Genetics, vol. 6, **(1994)**, doi:10.1038/ng0294-119, pp. 119–129.
[4] G. Taubs, "Sense from sequences: Stephen F. Altschul on bettering BLAST", Science Watch, vol. 11, **(2000)**, pp. 3–4.
[5] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences", Journal of Molecular Biology, vol. 147, **(1981)**, doi:10.1016/0022-2836(81)90087-5, pp. 195–197.
[6] S. F. Altschul, et al., "Basic Local Alignment Search Tool", Journal of Molecular Biology, vol. 215, **(1990)**, doi:10.1016/S0022-2836(05)80360-2, pp. 403–410.
[7] G. Mirceva and D. Davcev, "HMM based approach for classifying protein structures", IJBSBT, vol. 1, no. 1, **(2009)**, pp. 37-46.
[8] S. Ismail, R. Othman, S. Kasim and R. Hassan, "Pairwise Protein Substring Alignment With Latent Semantic Analysis and Support Vector Machines To Detect Remote Protein Homology", IJBSBT, vol. 3, no. 3, **(2011)**, pp. 17-34.
[9] F. Abdullah, R. Othman, S. Kasim, R. Hassan, H. Asmuni and J. Taliba, "An optimal Mesh Algorithm for Remote Protein Homology Detection", IJBSBT, vol. 3, no. 2, **(2011)**, pp. 13-38.
[10] T. Madden, "The BLAST sequence analysis tool. In NCBI Handbook", J. McEntyre and J. Ostell, (Eds.), National Library of Medicine, Bethesda, MD, **(2005)**.
[11] I. Korf, M. Yandell and J. Bedell, "BLAST: An Essential Guide to the Basic Local Alignment Search Tool", O'Reilly, Sebastopol, CA, **(2003)**.
[12] S. F. Altschul, et al., "Gapped Blast and PSI-Blast: A new generation of protein database search programs", Nucleic Acids Research, vol. 25, **(1997)**, pp. 3389–3402.
[13] J. F. Collins and A. F. Coulson, "Applications of parallel processing algorithms for DNA sequence analysis", Nucleic Acids Research, vol. 12, (1984), pp. 181–192.
[14] W. Gish and D. J. States, "Identification of protein coding regions by database similarity search", Nature Genetics, vol. 3, **(1993)**, doi:10.1038/ng0393-266, pp. 266–272.
[15] P. Aboyoun, "Sequence Alignment of Short Read Data using Biostrings", **(2009)**, www.bioconductor.org/help/course-materials/.../MatchAlign.pdf.
[16] SIB Swiss Institute of Bioinformatics, "Instructions for MultiIdent Protein Identification Software", http://web.expasy.org/multiident/multiident-doc.html.
[17] K. Mahmond, *et al.*, "EGM: encapsulated gene-by-gene matching to identify gene orthologs and homologous segments in genomes", The International Society for Computational Biology, **(2010)**.
[18] K. Mahmond, *et al.*, "The EGM software", http://vbc.med.monash.edu.au/kmahmood/EGM, **(2010)**.
[19] FASTA Sequence Comparison at the U. of Virginia, http://fasta.bioch.virginia.edu/fasta_www2/fasta_list2.shtml
[20] Fold&Function Assignment, Godzik Lab, Samford Burnham Medical Research Institute, http://ffas.sanfordburnham.org/ffas-cgi/cgi/document.pl?ses=&rv=&lv=.

Authors

Zhou Zhimin

Zhou Zhimin, female, associate professor, was born in Baoding of Hebei Province in 1966. Her research areas include analysis & design of MIS, machine learning algorithms and application of Linux Operating System

Chen Zhongwen

Chen Zhongwen, male, was born in CangZhou of Hebei Province in 1967. His research areas include computer network technology and computer application design and development.