# Evaluation of the suitability of a Zipfian gap model for pairwise sequence alignment

**Ramu Chenna**[1] **and Toby Gibson**[2]
[1]Biotechnology Center, Dresden University of Technology, Tatzberg 47-51, 01307 Dresden, Germany
[2]European Molecular Biological Laboratory, 1 Meyerhofstrasse, Postfach 10.2209, Heidelberg, Germany

**Abstract**— *Insertions and deletions occur during evolution of biological sequences resulting in gaps in sequence alignments. The quality of an alignment depends on the placement of the gaps. Reliable pairwise as well as multiple sequence alignments are useful in inferring protein protein interacton sites through residue conservation[23], [24]. It has been reported that the Zipfian distribution best approximates the observed gap-lengths in the sequence alignments. The probability of a gap of length N decreases, inversely related to length, as a function of $N^{-c}$ for some suitable c. We have analysed four different gap scoring models: affine, log, power and the new Zipf that is based on Zipfian distribution. When tested on pairwise alignments from the BAliBASE benchmark suite, the widely used affine gaps were outperformed by the three other models. Log, Power and Zipf gap models performed comparably well.*

**Keywords:** protein sequence, sequence alignment, gap penalty, parameter reduction, zipfian distribution, riemann zeta function

## 1. Introduction

Aligning a new protein sequence to a known sequence is an essential and first step to study the structural and functional information of the new protein molecule. Pairwise alignments are done through a method called dynamic programming, first applied to biological sequences by Needleman and Wunsch [13]. Historically local sequence alignments are calculated using the algorithm Smith-Waterman [19] and global alignments are calculated by Needleman-Wunsch [13], each having their own advantages. For example, local alignments are more suitable for identifying protein domains irrespective of their domain shuffling and global alignment is necessary when you want full length alignments.

Many variants of sequence alignment algorithms are used for searching sequence databases e.g. SSEARCH [16] as well as methods that use word search for example FASTA [15], BLAST [2], PSI-BLAST [3] etc. Alignment scores are used to rank sequences and provide statistics for the likelihood of homology with the query. Therefore alignment quality directly influences the signal-to-noise, hence the sensitivity of database searches.

Gaps are common in alignments of biological sequences. They occur more frequently between distantly related sequences. Gaps in pairwise or multiple sequence alignments represent insertion or deletion (indels) events in the evolution of biological sequences.

The quality of an alignment is obtained in part through scoring aligned pairs of residues. The indels are scored by pairing a residue in one sequence with a gap in another sequence. The placement of gaps influences the quality score and hence the quality of an alignment.

Thus placement of gaps is critical in sequence alignment and they have been studied extensively [19], [18], [11], [1], [12].

A number of different gap scoring models have been discussed. However three parameters, gap open, gap extension and length of the gap are common to most of the gap models.

It has recently been proposed that observed gap lengths obey a Zipfian distribution and that this could be used to derive an appropriate gap penalty model, although this was not tested [6]. Since the Zipfian equation is so simple, we were interested in evaluating its performance for pairwise alignment. Here we report the performance of Zipfian gap penalties using the BAliBASE testbed and compare it to other concave gap models.

## 2. Methods

### 2.1 Gap Models

Gaps in sequence alignment represent the insertions and deletions that occurred in the history of the protein family of sequences [4]. Placing gaps in the right place is essential to the quality of an alignment. We have studied four different gap models by modifying the Monotone pairwise alignment package [12]. The quality of alignment for different gap models are assessed with the BAliBASE benchmark database [20].

### 2.2 Affine gap model

The affine gap model is the most widely used gap scoring scheme in alignment algorithms.

$$gapcost = m \cdot x + c \quad , \qquad m < c \tag{1}$$

where $c$ is the cost for opening a gap and $m$ is the cost of extending a gap and $x$ is length of the gap. The default values in Monotone are for gap open $c = 9$ and gap extension

$m = 3$. However, $c = 10$ and $m = 1$ are commonly used for protein database searches.

The affine gap model is an extension of a linear gap model of the form $gapcost = mx$. In the affine model the condition $m < c$ is set to allow long insertions and deletions to be penalised less to overcome the deficiency of the linear gap model where short and long gaps are treated as equally likely. It has been shown by [6] that for gaps observed in aligned protein sequences, the affine gap is a poor approximation. The affine gap model was used to study the distribution of indel lengths [17] and they suggested a quadruple affine gap model as an alternative to plain affine gap model. This would be expensive to calculate. The linear gap model equation is in fact a straight line equation.

## 2.3 Log gap model

The log gap model [12] is of the form

$$gapcost = c + m \cdot log(x) \tag{2}$$

where $c$ is the gap open penalty, $m$ is the gap extension penalty and $x$ is length of gap. The default values in Monotone are for gap open $c = 9$ and gap extension $m = 3$.

## 2.4 Power gap model

The Power gap model [12] is of the form

$$gapcost = c + m \cdot x^d \qquad where \quad d > 0 \tag{3}$$

where $c$ is the gap open penalty, $m$ is the gap extension penalty, $d$ is gap power and x is the length of the gap. The power law is convex only for $0 < d \leq 1$.

The default values in Monotone are for gap open $c = 9$ and gap extension $m = 3$ and power $d = 0.5$.

## 2.5 The new Zipf gap model based on Zipfian distribution

Chang and Benner have studied the gap length distribution in a set of pairwise alignments and suggested that the Zipfian distribution can be used as a best approximation for scoring the gaps in an alignment [6]. Their detailed study shows that the number of gaps say n of length N decreases according to the expression

$$n = c_1 N^{-c_2} \tag{4}$$

where $c_1$ and $c_2$ are parameters empirically selected to fit the data.

Benner also suggested that this function is independent of the length of the gap and the extent of divergence. One caveat is that they used a dataset with just one gap per pairwise alignment. Even if the Zipfian holds for multiple gaps, the derived parameters may not. We have further tested their suggestion by incorporating the Zipfian gap scoring model into the Monotone [12] pairwise alignment package.

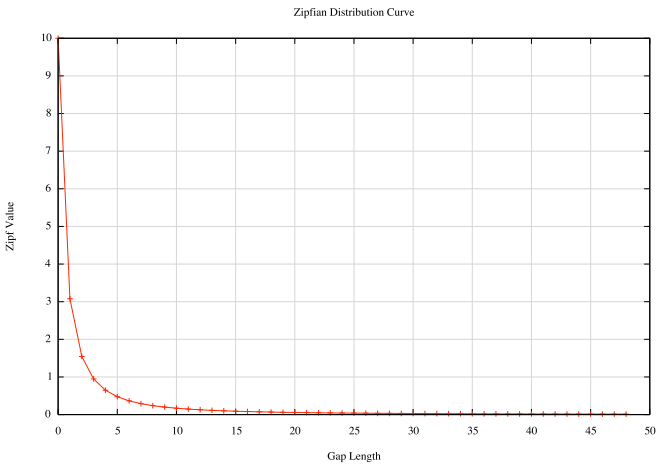


Figure 1: Gap penalty scores generated by the Zipfian curve for $10 * N^{-1.7}$ where N is the gap length. Starting value of $c_1 = 10$ is typical for pairwise alignment with the Blosum62 matrix.

Figure 1 shows the gap penalty scores generated by the Zipfian curve for $10 * N^{-1.7}$ where N is the gap length. A starting value of 10 is typical for pairwise alignment with Blosum62 [10]. The value of the curve at position N is added to the gap extension cost at position N-1. Since the curve converges, we cannot use the equation 1 as it is. Therefore we take the cumulative sum over the entire given gap length.

Define the cumulative sum

$$gapcost(n) = P = \sum_{N=1}^{n} \frac{c_1}{N^{c_2}} \tag{5}$$

Here the cumulative sum P can be used as the gap cost for inserting a gap of N (or n) symbols. This gap cost function is monotonically increasing where $gapcost(n) > gapcost(n-1)$ for all n. In other words it is a non-decreasing, concave gap function.

The equation 5 is in fact the partial sum of the infinite series of the famous Riemann Zeta function of the form

$$\zeta(p) = \sum_{n=1}^{\infty} n^{-p} \tag{6}$$

As a special case when $p = 1$, the $\zeta(p)$ becomes the logorithm function which is an advantage that one could mimic different gap models inside the alignment algorithm by changing the exponent $p$ of Riemann Zeta function [21].

Riemann Zeta function is extensively studied in number theory and has number of interesting properties. When one considers indels as infinite series of evolutionary events then it would be interesting to study these events in the light of Riemann Zeta function.

The Figure 1 shows the plot of equation 4. The curve is asymptotic to X-axis.

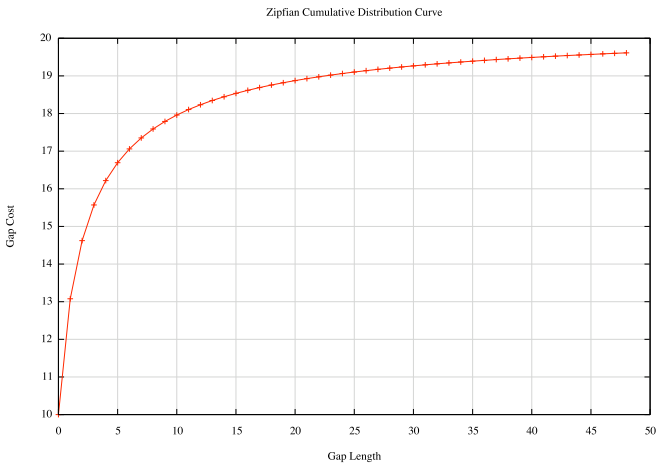


Figure 2: Shows the cumulative distribution plot of equation-5 showing the gap score for different gap lengths. The value of the curve at position n is added to the gap extension cost at position n-1. For gaps of more than 20 residues, the extension cost becomes very small. This curve diverges as the length of the gap increases. The costs of opening and then extending small gaps are high. However the curve becomes asymptotic and the costs of further extending long gaps become very small.

Figure 2 shows the Cumulative sum for the Zipfian values in Figure 1 (See equation 5). The costs of opening and then extending small gaps are high. However the curve becomes asymptotic and the costs of extending long gaps become very small.

# 3. Results

## 3.1 The Scoring Method

We used the BAliBASE Version 2.0 benchmark alignment database [20]. BAliBASE is designed for evaluating multiple sequence alignment algorithms. Alignments in BAliBASE were derived from visually inspected structure alignments. Therefore they are not biased toward any sequence alignment method.

BAliBASE consists of mainly five different reference alignment sets. Reference 1 consists of equidistant sequences, Reference 2 consists of related families with divergent, orphan sequences, Reference 3 consists of families of related sequences, Reference 4 consists of N and C terminal extension sequences, Reference 5 consists of internal insertions. Refer to the BAliBASE [20] paper for more details.

We have modified Richard Mott's software package Monotone [12] to allow Zipfian values to be computed and used for gap scoring. The Monotone package is elegantly designed and it was easy to incorporate the new Zipfian gap model. Monotone also comes with the affine, log and power gap models.

Monotone reads two sequences from two different files. So it was necessary to split the sequences from the BAliBASE sequence files. First the sequences from reference files were separated into single files and the multiple sequence alignment files were also separated with all the possible pairwise combinations intact. See table 1 for details.

Table 1: Pairwise alignments available in BaliBASE-2

| Set | Number of Files | Sequences | Pairwise alignments |
|---|---|---|---|
| Ref1 | 82 | 367 | 652 |
| Ref2 | 23 | 412 | 3544 |
| Ref3 | 12 | 266 | 2865 |
| Ref4 | 12 | 107 | 504 |
| Ref5 | 12 | 112 | 570 |

Each file in each Balibase reference set consists of different numbers of sequences. Note that the total pairwise $\frac{n(n-1)}{2}$ comparisons is based on the number of sequences in each file.

A script was written to generate all the possible pairwise alignment commands to run Monotone for different gap models. The program BaliScore was used to assess the quality of the alignment with reference to the test alignment. BaliScore gives SP, Sum of Pairs score and CS, Column Score. The SP score determines the extent to which the test program, in this case Monotone, succeeded in aligning the sequences. The CS score is designed to see whether the test program can align all of the sequences correctly in a multiple alignment (that is not relevant here).

We plotted the overall SP scores for all the different gap models using Blosum62 [10] and Gonnet PAM250 [5] matrices by varying the exponent of the Zipfian gap model by 0.1 increments over a range from 1.0 to 2.0.

## 3.2 Comparison of four penalty schemes

Figures 3 and 4 show the comparison of the four gap penalty models: affine, log, power and Zipfian. The X-axis shows the gap open penalty varied from 1 to 25 and the Y-axis shows the overall percentage BaliScore [20] for all five different reference sets (See Table.1). The range of baliscore is from 0, the lowest, to 1, the highest for each alignments. We used two different popular comparison matrices namely Blosum62 [10] and Gonnet PAM250 [5].

Examining the affine scores in Figures 3 and 4, the best scores are achieved with gap opening in the range 7.0 - 9.0 for both matrices, however the overall score is higher for PAM250, indicative of better alignments. Better quality alignments for the PAM250 matrix are in accordance with previous matrix comparisons (Vogt et al., 1995). Optimal gap opening penalties observed here are slightly lower than the typical values used as defaults in sequence alignment.

In both figures, the peak BALiBASE scores for affine gap are below the peak scores of the other gap functions. It is also clear that the affine gap penalty does not tolerate the higher penalty values as well as the other models. (This is not a major consideration provided that gap penalties are

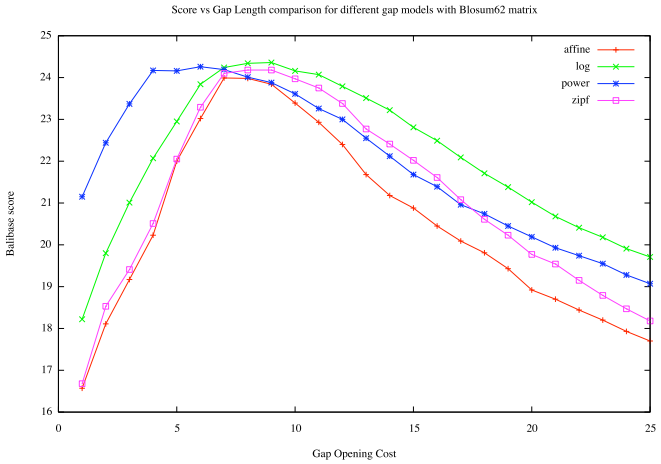


Figure 3: Comparison of 4 gap penalty schemes tested by pairwise BAliBASE scores for the Blosum62 exchange matrix. The obtained BAliBASE score (Y axis) is reported for the cost of opening a gap varied over a range of 2 to 25 (X axis). For all gap penalties, affine gaps always perform worse than the other functions. Peak performances of the log, power and Zipf functions are very close with log slightly ahead. See text for the equations describing the gap functions.

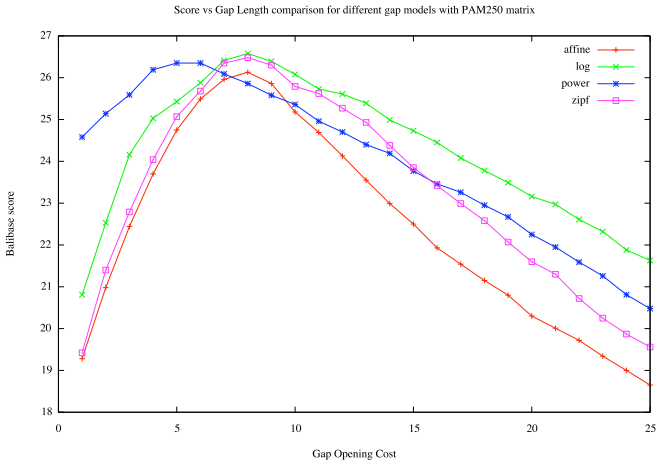


Figure 4: Comparison of 4 gap penalty schemes tested by pairwise BAliBASE scores for the Gonnet PAM250 exchange matrix, suitable for aligning highly divergent proteins. The obtained BAliBASE score (Y axis) is reported for the cost of opening a gap varied over a range of 2 to 25 (X axis). Affine gaps again perform worse than the other functions. Note that the better performance of the smooth models is more apparent with the more sensitive PAM250 matrix than with the Blosum62. Peak performances of the log, power and Zipf functions are very close again with log slightly ahead. See text for the equations describing the gap functions.

being set close to optimal perfomance). The peak difference is smaller than we expected, especially for Blosum62. The poorer performance of affine becomes clearer using the more sensitive PAM250 matrix. This may imply that alignment with better residue exchange parameterisation benefits more from the improved gap penalty models.

The log, power and Zipfian models outperform affine for both Blosum62 and PAM250. However, the peak performances of the three smooth models are all very close for both tested matrices. Based solely on performance in our tests, we would not be able to choose between the three models. Note that for log and power we ran the tests using the default Monotone gap extension value of 3.0. This value has already been well optimised for the smooth gap models supplied in the Monotone package. However, we observed very poor performance for affine with the Monotone defaults (data not shown) - a gap penalty of 3.0 is much higher than usually recommended for protein alignment. Therefore, in accordance with standard practice, we have kept the affine gap opening and gap extension penalties in the ratio of 10 to 1 for the tests.

From these figures, it is clear that the default gap opening penalty value 9.0 for Monotone could be set to higher.

### 3.3 Effect of varying the Zipfian exponent

The exponent of the equation 5 $c_2$ has been varied in the range of 1.0 to 2.0 keeping $c_1$ at a constant 1.0. The Figure 5 and Figure 6 are graphs showing the overall percentage BaliScore score distribution for the variations of $c_2$. The gap opening penalties are computed using the equation 5. With the exponent 1.7 the highest score 24.18 is achieved when gap opening penalty is 8 or 9 for Blosum62 matrix whereas for PAM250 matrix with the same exponent the highest score is 26.48 when the gap open penalty is 8. For exponent 1.8 the highest score 26.49 with gap open 8.0 for PAM250, and with Blosum62 it is 24.15 with gap open 9.0.

The results are in good agreement with the observed value of exponent $c_2 = 1.8$ [6]. Though the higher exponent tolerates large gap penalties they do not get higher score comparatively. In a progressive multiple sequence alignment scenario the exponent in the Zipfian could be used to adjust the gap openings dynamically for different divergence.

## 4. Discussion

The Zipf law has been used to study phenomena in many areas e.g. linguistic, audio signals [8] and also recently to study the human transcriptome [14]. The Zipf law suggests that the frequency of occurrence of a word is inversely proportional to its rank.

Chang and Benner showed that the gap-lengths can be approximated by the Zipfian distribution with the probability of a gap of length N decreasing as a function of the gap length [6].

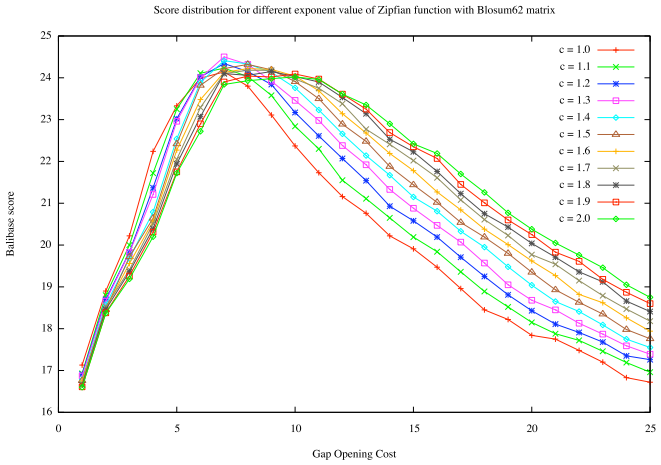


Figure 5: BAliBASE score distributions for different exponent values of the Zipfian function and the Blosum62 matrix. Varying the exponent yields a small difference for peak performance with $N^{-1.4}$ marginally best. However, this value performs relatively less well when gap penalties are set too high. Values closer to the Benner exponent of $N^{-1.8}$ have a broader peak and are more tolerant of higher gap penalties: These values may be appropriate when sequence similarity varies widely and gap penalty values are necessarily imprecise.

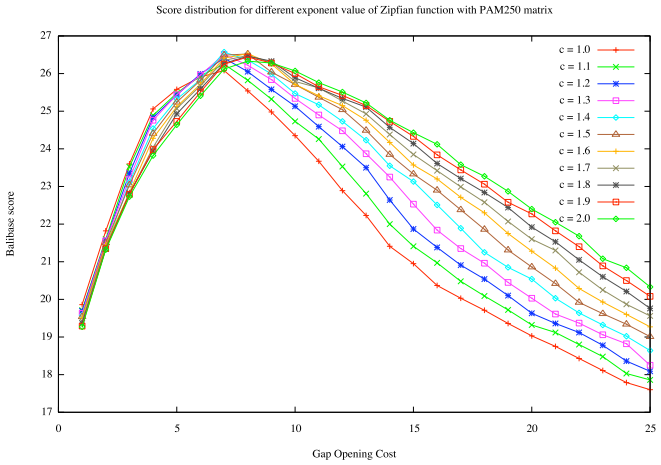


Figure 6: BAliBASE score distributions for different exponent values of the Zipfian function and the Gonnet PAM250 matrix. Varying the exponent yields even smaller differences than for Blosum62 for peak performance with $N^{-1.4}$ marginally best. Again, this value performs relatively less well when gap penalties are set too high. Values closer to the Benner exponent of $N^{-1.8}$ have a broader peak and are more tolerant of higher gap penalties: These values may be appropriate when sequence similarity varies widely and gap penalty values are necessarily imprecise.

Other authors have proposed concave gap functions. For long gaps, Gotoh used a piecewise linear gap-weighting function that approximated a smooth concave function [9]. Miller and Myers [11] and Waterman [22] also used concave weighting functions to compare sequences. Mott has shown that by using monotonic gap penalties, the chances of detecting a similarity containing a long gap is greater over affine gap penalties. We have seen that in Monotone the default value of gap opening 9 and gap extension 3 for affine gaps performed very poorly. The situation improved when we modified the affine gap extension as one tenth of gap opening penalty, which is typical for protein sequence alignment, but still the affine gap penalty was worse.

Despite the extensive literature on concave penalty functions, all widely used alignment and database search software continue to use affine gaps. One reason may be increased computational costs. Myers and Miller found affine gaps to be three times faster than other concave functions. Performance should be less important with recent computer hardware. Furthermore precalculation and array lookup can reduce the time penalty for any gap scheme that is more complicated than affine.

BAliBASE is the most widely used alignment benchmarking suite. Using BAliBASE we have now shown that the non-affine gap penalties are better suited for pairwise sequence alignments. Although it does not outperform the other smooth gap models, the Zipfian model shows promise as the simplest of the models tried, with the lowest parameter space. From the Figure 5 and Figure 6 it is clear that the higher exponent tolerates longer gaps.

We would like to suggest that in a progressive multiple alignment environment where the highly homologous sequences are aligned first, the gap opening in equation 4 could be adjusted automatically to fit to the extent of divergence of the sequence or profile that are already aligned with the new sequences or profile. This is quite logical because a fixed gap opening cannot perform well for merging group of sequences with varied degree of divergence.

BAliBASE covers a range of alignment test cases including long gaps. Though BAliBASE benchmark alignments are designed for testing multiple sequence alignment programs, BAliBASE can also be adopted for use with pairwise alignments. The pairwise test with local alignment approximates database search properties. Thus Zipfian gap model should be suitable for use in sequence database searches.

The Zipfian gap model might also be useful for nucleic acid alignment, since genomic sequence alignment needs to handle very large indels. For example, insertion of a Line-1 element creates a gap of more than 8000 bases and affine gaps are completely unsuitable for dealing with such long gaps.

In future work, we hope to examine the performance of Zipfian penalties for the progressive alignment algorithm of Clustalw [7]

## Acknowledgement

We would like to thank Richard Mott for providing the Monotone software package, and Julie Thompson for the BAliBASE benchmark suit and helpful discussions. I also thank Des Higgins, Mark Larkin, Ian Wallace, Lars Juhl Jensen and unknown referees for their careful reading and critical comments.


## References


[1] Altschul, S. F. (1989) Gap costs for multiple sequence alignment. *J Theor Biol*, **138** (3), 297–309.

[2] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990) Basic local alignment search tool. *J Mol Biol*, **215** (3), 403–410.

[3] Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, **25** (17), 3389–3402.

[4] Benner, S., Cohen, M. & Gonnet, G. (1993) Empirical and Structural Models for insertion and deletions in the divergent evolution of proteins. *J. Mol. Biol*, **229**, 1065–1082.

[5] Benner, S. A., Cohen, M. A. & Gonnet, G. H. (1994) Amino acid substitution during functionally constrained divergent evolution of protein sequences. *Protein Eng*, **7** (11), 1323–1332.

[6] Chang, M. S. S. & Benner, S. A. (2004) Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *J Mol Biol*, **341** (2), 617–631.

[7] Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T. J., Higgins, D. G. & Thompson, J. D. (2003) Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Res*, **31** (13), 3497–3500.

[8] Dellandr, E., Makris, P. & N, V. (2004) Zipf analysis of audio signals. *Fractals*, **1** (12), 73–85.

[9] Gotoh, O. (1990) Optimal sequence alignment allowing for long gaps. *Bull Math Biol*, **52** (3), 359–373.

[10] Henikoff, S. & Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, **89** (22), 10915–10919.

[11] Miller, W. & Myers, E. W. (1988) Sequence comparison with concave weighting functions. *Bull Math Biol*, **50** (2), 97–120.

[12] Mott, R. (1999) Local sequence alignments with monotonic gap penalties. *Bioinformatics*, **15** (6), 455–462.

[13] Needleman, S. B. & Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, **48** (3), 443–453.

[14] Ogasawara, O., Kawamoto, S. & Okubo, K. (2003) Zipf's law and human transcriptomes: an explanation with an evolutionary model. *C R Biol*, **326** (10-11), 1097–1101.

[15] Pearson, W. R. & Lipman, D. J. (1988*a*) Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A*, **85** (8), 2444–2448.

[16] Pearson, W. R. & Lipman, D. J. (1988*b*) Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A*, **85** (8), 2444–2448.

[17] Qian, B. & Goldstein, R. A. (2001) Distribution of Indel lengths. *Proteins*, **45**, 102–104.

[18] Sankoff, D. & Kruskal, J. (1983) *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison.* . Addison-Wesley, Reading, MA.

[19] Smith, T. F. & Waterman, M. S. (1981) Identification of common molecular subsequences. *J Mol Biol*, **147** (1), 195–197.

[20] Thompson, J. D., Plewniak, F. & Poch, O. (1999) BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, **15** (1), 87–88.

[21] Titchmarsh, T. (1951) *The Theory of Riemann Zeta Function.* Oxford Univ Press Oxford.

[22] Waterman, M. S. (1984) Efficient sequence alignment algorithms. *J Theor Biol*, **108** (3), 333–337.

[23] Panjkovich, A. and Aloy, P. (2010 ) Predicting protein-protein interaction specificity through integration of three dimentional structural information and the evolutionary records of protein domains. *Molecular Biosystems* **6**(4), 741-749.

[24] Want, B. and Wong, HS. (2006) Inferring protein-protein interacting residue conservation and evolutionary information. *Protein and Peptide Letters* **13**(10), 999-1005.