BlastMultAl, a Blast Extension for Similarity Searching with Alignment Graphs

Pierre Nicodème

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

BlastMultAl, a Blast extension for similarity searching with alignment graphs

Pierre Nicodème

N ° 2911

Juin 1996

PROGRAMME 2

1996

ISSN 0249-6399

BlastMultAl, a Blast extension for similarity searching with alignment graphs[1]

Pierre Nicodeme[2]

Abstract. We describe a new method of processing similarity queries of a proteic multiple alignment with a set (database) of protein sequences, or similarity queries of a protein sequence with a set of protein alignments. We use a representation of multiple alignments as alignment-graphs. Comparisons with different classical methods is made. This new method allows the detection of subtle similarities which are not found by the other methods. It has direct applications for similarities querying with the database of protein domains ProDom.

BlastMultAl: une extension de Blast pour la recherche de similarités au moyen de graphes d'alignement.

Résumé. Nous décrivons une nouvelle méthode de recherche de similarités d'un alignement multiple de protéines avec un ensemble (une base de données) de séquences protéiques, ou de recherche de similarités d'une séquence protéique avec un ensemble d'alignements multiples de protéines. Nous comparons cette approche avec des approches classiques. Cette nouvelle méthode permet la détection de similarités subtiles qui ne sont pas mises en évidence par les autres méthodes; elle a des applications directes pour effectuer des recherches de similarités avec la base de domaines protéiques ProDom.

[1]This research is a joint work of INRIA (National Institute of Informatics and Automatics Research), INRA (National Institute of Agronomical Research), and the LIX laboratory (École Polytechnique); it was partially supported by the GREG grant No. 10794.

[2]Algorithms Project, INRIA-Rocquencourt, F-78153 Le Chesnay (France)

BlastMultAl, a Blast extension for similarity searching with alignment graphs[‡]

Pierre Nicodème

INRIA-Rocquencourt,
Domaine de Voluceau 78153 - Le Chesnay Cedex - France
and
LIX École Polytechnique, 91128 - Palaiseau Cedex - France

E-mail: Pierre.Nicodeme@inria.fr
Tel: 33-(1)-39.63.56.59 Fax: 33-(1)-39.63.53.30

June 24, 1996

Abstract

We describe a new method of processing similarity queries of a proteic multiple alignment with a set (database) of protein sequences, or similarity queries of a protein sequence with a set of protein alignments. We use a representation of multiple alignments as alignment-graphs. Comparisons with different classical methods is made. This new method allows the detection of subtle similarities which are not found by the other methods. It has direct applications for similarities querying with the database of protein domains ProDom.

1 Introduction

When a new gene or a new protein is discovered, homology methods allow the detection of similarities to sequences whose properties are understood. Finding such a similarity between a new protein and a sequence in a protein database often gives important insights into the functional, metabolic, or structural properties of the new protein. Similarities searching tools are therefore of great importance for biologists. We make in this introduction a brief review of the main alignment algorithms; we describe then the objectives of our new algorithm.

1.1 Historical Background

Homologies methods rely on the same principles when handling DNA sequences or protein sequences, alphabet size being 4 for DNA and 20 for proteins; however, efficient similarity matrices are available for proteins; they measure the probability of substitution of an amino-acid by another one during biological evolution; there is no direct equivalence for DNA. We consider from now on only proteins.

[‡]This research is a joint work of INRIA (National Institute of Informatics and Automatics Research), INRA (National Institute of Agronomical Research), and the laboratory LIX (École Polytechnique); it was partially supported by the GREG grant No. 10794.

1.1.1 Alignment of two sequences

We refer to Waterman book [20] for a detailed introduction to alignment issues; we restate here the main definitions.

Basically, one has to compare two proteic sequences,

$$x_1 x_2 \dots x_{n-1} x_n,$$
$$y_1 y_2 \dots y_{n-1} y_n,$$

where x_i and y_j are aminoacids.

We consider a similarity matrix $\mathcal{M}$ giving for each pair of aminoacids (a_i, a_j) a score value, with the notation $|\, a_i, a_j \,| = \mathcal{M}_{ij}$; this score is high for similar aminoacids and low for dissimilar ones. If p_i is the probability of the aminoacid a_i, we assume that

$$E(|\, a_i, a_j \,|) = \sum_{i,j} p_i p_j \; |\, a_i, a_j \,| < 0.$$

Methods allowing insertion-deletions (indels for short) assume also a penalty gap $\gamma < 0$; more advanced methods define a penalty γ_1 for gap opening and a penalty γ_2 for gap extensions. Choosing $\gamma = \infty$ is equivalent to forbid indels.

An alignment is a sequence of pairs (i_t, j_t), with $t = 0 \dots k$, associated with a score S, such that

$$(i_{t+1}, j_{t+1}) = \begin{cases} (i_t + 1, j_t + 1) & \text{no indels} \quad S \mathrel{+}= |\, a_{i_{t+1}}, a_{j_{t+1}} \,| \\ (i_t, j_t + 1) & \text{del. in x} \quad S \mathrel{-}= \gamma \\ (i_t + 1, j_t) & \text{del. in y} \quad S \mathrel{-}= \gamma \end{cases}$$

The best alignment of the sequences $\{x_i\}$ and $\{y_j\}$ is the sequence (i_t, j_t) with the best score S.

The original method for alignment of two sequences with indels goes back to Needleman and Wunsch [16] who make use of dynamic programming. The major improvement comes from Pearson and Lipman [17] with FASTA; FASTA speeds up the search by use of a hash table of small words of one of the two sequences. Dynamic programming is only done for getting thorough alignments of primary alignments reaching a high score.

When forbidding indels, BLAST [1] is probably the most popular method. Considering one of the sequences as the query, and the other as the target, for each small word with a given size (typically 3 or 4) of the query, any small word of same size with a score bigger than a threshold score T is inserted in a multi-string automaton, with a pointer back to the original small word of the query. Once built, the multi-string automaton is run with the target sequence as input string; for each match, or hit, corresponding to a pair of indices (i, j) for the two sequences $(\{x_i\}, \{y_j\})$, extension (without indel) is performed left and right from the hit position. Extension is done as long as the score is above zero; the record score and corresponding positions are memorized.

One advantage of BLAST is the solid probabilistic framework [13] [14] [15] giving the quality of an alignment.

1.1.2 Multiple sequence alignment

We turn now to the more general problem of searching similarities of a multiple alignment (multialignment for short) with a set of sequences, or of a sequence with a set of multialignments.

We focus on questions arising when the multialignments are already obtained, and do not consider therefore the different methods for building multialignments.

We review briefly here the two main methods currently used in this perspective, profiles and blocks.

A multialignment of n sequences of length smaller than l may be viewed as a two-dimensionnal array $A_{i,k}$, with $i = 1 \ldots l$ and $k = 1 \ldots n$. Some elements of the array may be gap characters.

A profile [7] [9] considers the aminoacids present in a column of the multialignment; in the more naive version of profiles, for a fixed index i of the sequence of columns, a score is computed for each aminoacid a_j by averaging the score of a_j with the aminoacids present in column i.

A dynamic programming method is used to align the profile and the sequence, with the same principle of additive scores as used to align two sequences.

Different weighting schemes are proposed by Gribskov and *al.* [8], such as weighted average or logarithmic weighting.

Two main applications are derived from profiles: PROFILESEARCH searches for similarities of a multialignment with a set of sequences; PROFILESCAN searches for similarities of a sequence with a set of profiles.

The protein blocks method [10] comes in filiation of profiles: a block is a short ungapped profile. To a family of homologous proteins corresponds an ordered set of blocks separated by unaligned regions. From all the families of proteins, Steven and Jorga Henikoff have built a database of blocks; the order relation induces a structure of directed graph on this database; the detection of a similarity between a protein and a protein family is characterized by a path in this graph.

1.2 Objectives

Our main motivation is to provide a method for similarity search of proteins sequences with the database of multialignments ProDom (Protein Domainer)[1]; the last versions of ProDom contains more than 7000 true multialignments containing more than two sequences.

Since the first construction of ProDom, similarity search with ProDom have been done by searching similarities with a database of consensus sequences of the multialignments.

We made an attempt with ungapped and unweighted profiles as representation of the multialignments; the results obtained, when comparing with similarity search with consensus sequences, did not give a clear advantage to the profiles.

We therefore worked out a complete new approach, characterized by use of alignment graphs. The scope of this paper is the description of this approach, and comparisons with results obtained with unweighted or weighted consensus, with unweighted or weighted profiles, and with the naive method of "flat sequences".

As far as this work has been advanced, gaps have not been taken in considerations, and comparisons are done with ungapped method; we intend to handle gaps in future work.

The biological intuition underlying our approach relies upon the hypothesis that the variable subsequences composing the poorly conserved parts of a multialignment have weaker structural constraints and may recombine around a skeleton built over the well-conserved parts of a multialignment.

This approach leads to the construction of simple graphs that we call Alignment-Graphs; as with profiles, this Alignment-Graph approach has two applications: similarity search of a sequence with a database of multialignments, which was our original goal, or similarity search of a multialignment with a database of sequences.

We name BlastMultAl the software associated with this method, which reuses some features of Blast.

[1] Web URL entry for ProDom is http://protein.toulouse.inra.fr/prodom.html

2 Alignment Graphs and Best Path method

2.1 Overview of the method

As with the block approach, we consider that the well-conserved and aligned regions are joined by the non-conserved regions; but, in a somewhat complementary way to that approach, we take the consensus of the well-conserved regions and replace the non-conserved regions by parallel subsequences; when extending a path of similarity with a query sequence, and going through a non-conserved region, this one of the parallel sequences of the region matching the best the query will be selected.

We describe in section 2.2.1 how we discriminate between conserved or non conserved positions to build conserved regions; we discuss also heuristics choices made to keep the method efficient and to avoid noisy hits.

We define the Alignment-Graph of a multialignment in the following intuitive way:

- In a first step, we define conserved regions and take the *consensus* of these regions; we will speak of *consensus regions* for these regions.
- We call a *branch* a subsequence spanning a whole non-conserved regions; we will speak of *branch region* for a non conserved region.
- Extremities of branches are connected to the contiguous consensus sequences with respect to the ordering of the multialignment.

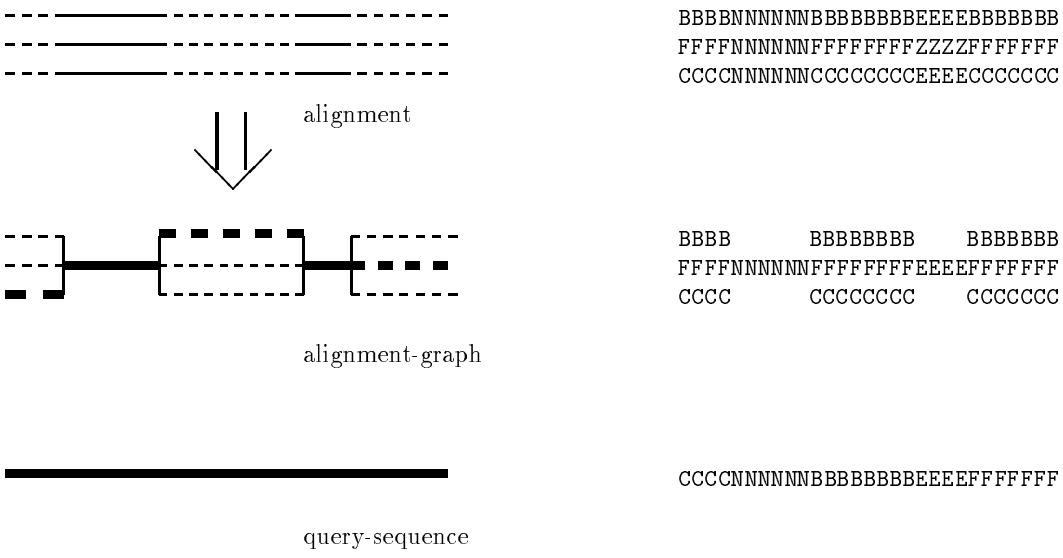

Figure 1: An *Alignment-Graph* and a *Best-Path*; on the conserved regions (in solid lines), only the consensus is kept; parallel edges, or branches are kept on the non-conserved regions (in dashed lines); combination of these branches is allowed by the algorithm, producing a best path (in bold in the left part of the figure). Note that the toy example of the right part of the figure corresponds to exact matching while our algorithm performs as Blast approximate matching, by use of a similarity matrix.

- Considering that there is no gap in the multialignment, an *admissible path* is the concatenation of aminoacids at contiguous positions, with the condition that inside a non-conserved region, once a branch is selected, all amino-acids for this region must belong to the same branch.

Figure 1 shows on the right size a multialignment, and on the left size the corresponding schematic Alignment-Graph.

Equipped with this Alignment-Graph, with a similarity matrix $\mathcal{M}$, and with an additive scoring scheme, we can define the score of an admissible path of length l with a subsequence of same length of a query sequence. Our resulting alignment will be the admissible path producing the highest score with a subsequence of the query sequence; we call this path *Best-Path*.

2.2 Getting the conserved and non-conserved regions from the multialignment

2.2.1 Definition of a coefficient of conservation for a multialignment position

We now describe how we discriminate between conserved and non-conserved positions:

Definitions: Given a multialignment of n sequences of length l, let a_{ij} be the aminoacid at position i in the sequence j. Let $|ab|$ be the *score of similarity* of two aminoacids a and b in the similarity matrix $\mathcal{M}$. Let c_i be the *consensus* amino-acid at position i, such that $\Sigma_{k=1}^{n} |a_{ik}c_i|$ is maximal. Let $\overline{c_i}$ be the *anticonsensus* amino-acid at position i, such that $\Sigma_{k=1}^{n} |a_{ik}\overline{c_i}|$ is minimal. We define a *coefficient of conservation* τ_i at position i as follows:

$$\tau_i = \frac{1}{n} \times \frac{\Sigma_{k=1}^{n}(|a_{ik}c_i| - |\overline{c_i}c_i|)}{|c_ic_i| - |\overline{c_i}c_i|} \tag{1}$$

Thus, the denominator of the fraction defining τ_i is n times the range of the consensus aminoacid in the similarity matrix. By construction, τ_i will be positive, and for a perfectly conserved position, $\tau_i = 1$.

We then choose a threshold value $\tilde{\tau}$, which is constant for the multialignment. Depending on whether or not $\tau > \tilde{\tau}$, we say that the position is conserved or not. Different strategies may be adopted when considering a set of multialignments: choosing an unique value for all the multialignments, or choosing a $\tilde{\tau}$ in relationship with the number of sequences in the multialignment.

The statistical results given in next section show that the empirical choice of a constant $\tilde{\tau} = 95\%$ for practical applications of the method with BLOSUM62 as similarity matrix $\mathcal{M}$ is reasonable.

2.2.2 Statistical results for τ with BLOSUM62

We choose the multialignments of ProDom28 as sample set for a practical statistical analysis of the parameter τ, and BLOSUM62 as similarity matrix $\mathcal{M}$.

We have computed the distribution of the parameter τ of equation 1 for multialignments of n sequences, with $n = 2, 3, 10$, and $n \geq 50$; we have also computed the distribution of τ for randomly generated multialignments of n random sequences, with $n = 2, 3, 10$, and $n \geq 50$, and comparisons of the resulting distributions has been done; (multialignment containing more that 50 sequences have been grouped into the same analysis). Figure 2 shows these distributions. Values of τ for $n = 2$ (resp. 3) are strongly correlated to the distribution of scores for the BLOSUM62 matrix; this explains the small peak at $\tau = 72\%$ (resp. 78%). The peak obtained for $\tau > 90\%$ for multialignments reflects the property that aminoacids of a column of multialignment

are similar. However, as expected, when the number of sequences increases, the average value of τ decreases. As stated in preceding section, with $\tilde{\tau} = 95\%$, very few positions of random alignments are considered as conserved; positions with $\tau > 95\%$ have therefore been considered as conserved.

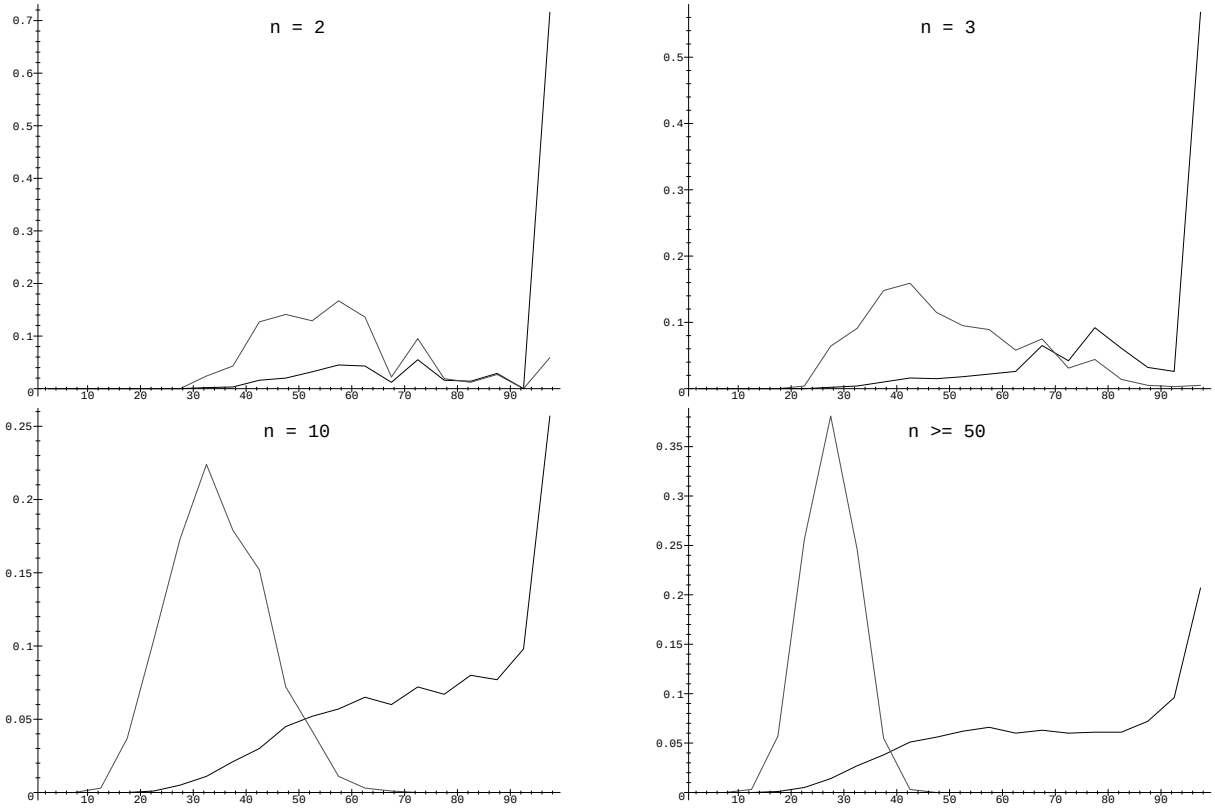

Figure 2: Density function of τ from ProDom28 multialignments (solid lines) and from random "multialignments" (dashed lines) computed with BLOSUM62.

2.3 Probabilistic framework and heuristics choices

One of the attractive features of Blast is its ability to provide a probability function of the score of a High Scoring Pair. For a sequence of l i.i.d random variables $\{X_i\}$, such that the expectation $E(X) < 0$, and $\{Prob(X > 0) > 0\}$, we have

$$Prob\{\mathcal{S}(l) > \frac{\log(l)}{\lambda} + x\} = 1 - e^{1 - K e^{-\lambda x}}, \tag{2}$$

where $\mathcal{S}(l) = Max_{1 \leq j < k \leq l}(S_k - S_j)$ is the maximum of the difference of the partial sums S_k, with $S_k = \sum_{i=1}^{k} X_i$; λ is the solution in t of the equation

$$E(e^{tX}) = 1, \tag{3}$$

and K is given by the equation,

$$K = \frac{\exp\left\{-2\sum_{k=1}^{\infty}\frac{1}{k}\left(E(e^{\lambda S_k}; S_k < 0) + Prob(S_k \geq 0)\right)\right\}}{\lambda E(Xe^{\lambda X})}. \quad (4)$$

Iglehart [12] derives equations 2 and 4 for a non-lattice i.i.d. variable X while computing extremes values in the GI/G/1 queues; his proof relies on random walk and on renewal theory. λ is a scaling factor transforming a specific renewal equation into the standard one. In the expression of K, we find an exp-log development applied to the Wiener-Hopf decomposition in random walk theory [3] [5]. This decomposition may be derived either by Fourier transforms on measures [3], or, more easily, by a Laplace transform of functions of a complex variable [5] [18]. Karlin obtains a version of equation 4 for lattice variables and gives bounds for K [15]; this last result is used by Blast to compute K.

Arnold and *al.* [2] shows that the limiting distribution $\exp(-\exp(-x))$ of equation 2 is one of three possible distributions for sample maximums. Searching for $S(l)$ is equivalent to getting the maximum of record values of independent random walk sections, and the conditions for the convergence to the double-exponential are met.

Turning back to the BestPath approach, we check what conditions may keep Karlin model applicable; the distribution of scores over branches corresponds to sums of i.i.d random variable X. Le $B_{k,i}$ $(i = 1 \ldots n)$ be the scores of the branches of a branch region of length k in an alignment with a query sequence, with n the number of parallel branches. The score W_k of the best branch has distribution of $\max_{i=1..n}(B_i)$ (the B_i being correlated despite the fact branches belong to non-conserved regions). It is therefore theoretically feasible to compute the distribution of a random variable Z such that W_k is the k^{th}-convolution of Z ($\sum_{i=1\ldots k} Z_i = W_k$). A mix of the initial variable X with these random variables $Z(n,k)$, depending of the number and of the shape of the branch regions will give a random variable $Y_{\mathcal{A}}$, function of the multialignment $\mathcal{A}$, and the Karlin model will be appliable, replacing X by $Y_{\mathcal{A}}$, in a rough approximation.

However, a major assumption for applicability of Iglehart-Karlin formula is the negative expectation of X. With a fixed k, as a function of n, $E(Z(n,k))$ is increasing, and the second assumption of Karlin model, $P\{X > 0\} > 0$ makes $E(Z)$ drift towards positive values as n increases. With big values of n, the mix variable $Y_{\mathcal{A}}$ is also likely to have a positive expectation.

These considerations lead us to make heuristic choices concerning branches:

- to keep $E(Y_{\mathcal{A}}) < 0$, we eliminate too short branches, taking the consensus for non-conserved regions of length smaller than 3;
- we restrict also the number of parallel branches for a non-conserved regions to 5, collapsing together similar branches during the preprocessing step of the algorithm, described in the next section.

Therefore, with BLOSUM62 as similarity matrix, a non-conserved region will be a sequence of more than 2 contiguous positions with $\tau > 95\%$

Practical experiments show that, with these heuristic choices, equation 2 may be used for probabilistic evaluations of the scores obtained by our method. The parameters K and λ fitting variables $Y_{\mathcal{A}}$ must be computed for each multialignment (see section 4.2.2), which corresponds to the step of calibration for blocks.

```
------------------------------------------------------------------------
>1284 (10) DEHYDROGENASE (IMP INOSINE-5'-MONOPHOSPHATE DEHYDROGENASE)

GUAC_ASCLU 105  331   ..NGYSEVFVDFIRRVREQFPTHTIFAGNVVTGEMVEELILSGADVVKVGIG..
GUAC_ECOLI 91   327   ..NGYSEHFVQFVAKAREAWPTKTICAGNVVTGEMCEELILSGADIVKVGIG..
IMP1_HUMAN 30   514   ..QGNSVYQIAMVHYIKQKYPHLQVIGGNVVTAAQAKNLIDAGVDGLRVGMG..
IMP2_HUMAN 30   514   ..QGNSIFQINMIKYIKDKYPNLQVIGGNVVTAAQAKNLIDAGVDALRVGMG..
IMP_ACICA  9    455   ..HGHSAGVIERVRWVKQNFPQVQVIGGNIATGDAALALLDAGADAVKVGIG..
IMP_BACSU  11   455   ..HGHSQGVLNTVTKIRETYPELNIIAGNVATAEATRALIEAGADVVKVGIG..
IMP_ECOLI  32   477   ..HGHSEGVLQRIRETRAKYPDLQIIGGNVATAAGARALAEAGCSAVKVGIG..
IMP_LEIDO  28   510   ..QGNTIYQIAFIKWVKSTYPHLEVVAGNVVTQDQAKNLIDAGADGIRIGMG..
IMP_MESAU  30   514   ..QGNSIFQINMIKYMKEKYPNLQVIGGNVVTAAQAKNLIDAGVDALRVGMG..
IMP_MOUSE  30   514   ..QGNSIFQINMIKYIKEKYPSLQVIGGNVVTAAQAKNLIDAGVDALRVGMG..
------------------------------------------------------------------------
conserved positions 1 .. C               C      CCC C      C   C     CC C..
------------------------------------------------------------------------
conserved positions 2 .. C               C      CCCCC      C   C     CCCC..
------------------------------------------------------------------------
consensus sequence    ..HGHSVYQINMIKYVKEKYPNLQVIGGNVVTAEQAKNLIDAGADAIRVGMG..
------------------------------------------------------------------------
                      ..N YSEVFVDFIRRVREQF THTIFA     GEMVEE ILS ADVVK    ..
                      ..Q NSVYQIAMVHYIKQKY TKTICA     AAQAKN IDA VDGLR    ..
Alignment-Graph       ..HGHSAGVIERVRWVKQNFPHLQVIGGNVVTGDAALALLDAGVDALRVGMG..
                      ..H HSQGVLNTVTKIRETY QVQVIG     AEATRA IEA CSAVK    ..
                      ..H HSEGVLQRIRETRAKY ELNIIA     AAGARA AEA ADGIR    ..
------------------------------------------------------------------------
```

Figure 3: A part (positions 270 to 319) of alignment 1284 of ProDom28, and the corresponding preprocessing output; branches from step 1 which are too short are replaced by the consensus in step 2. Positions considered as conserved have a conservation coefficient τ such that $\tau > \tilde{\tau} = 95\%$. After step 1, 21.4% of the positions are considered as conserved, and 30.0% after step 2.

3 Implementation

We present here the main algorithmic features of our method.

3.1 Preprocessing step: construction of Alignments-Graphs

This step has two functions:

1. to define conserved positions and non conserved ones; this is trivial, once a value for $\tilde{\tau}$ has been chosen; too short branches (size ≤ 2) are also easily replaced by the consensus;
2. to reduce the number of parallel branches; we describe this last operation.

The objective of this second function is to reduce the set of branches by aggregating branches with high similarity in a set, and by taking the consensus for each of these sets. We detail how we reduce the number of sets from k to $k-1$.

We consider a partition of the set of n branches b_i in k sets $S_j, j = 1 \ldots k$; we have:

reduction step:
1- take the consensus C_j of each set S_j;
2- for each pair l, m, with $1 \leq l < m \leq k$, compute the score $\mid C_l C_m \mid$, as sum of the score of the individual positions; select the pair l', m' with highest score;
3- merge sets $S_{l'}$ and $S_{m'}$

Initialization of the preceding algorithm is made by considering a partition with a single branch in each set S_j; we iterate the reduction step until $k = 5$, the heuristic value given in the preceding section.

Example of preprocessing Figure 3 provides a typical example of preprocessing: a section of the multialignment 1284 of ProDom28, which contains 10 sequences, and the corresponding Alignment-Graph are shown.

3.2 Main step: searching for an alignment between a query sequence and an Alignment-Graph

This step is composed of two phases[2].

1. A Aho-Corasick [4] multi-string (AC) automaton of high scoring small words is built, pointing to positions in the query string, as in Blast.
2. The main processing phase is composed of three substeps:
   (a) detecting hits in the Alignment-Graph; a hit is the alignment of a small word of the query with a small word of the Alignment-Graph with score above T; it has two parameters, (1) its coordinate in the query sequence, and (2) its coordinate (in terms of position) in the Alignment-Graph; hits detection is done by running the AC automaton successively on the consensus sequences of the conserved regions, on the branch sequences of the non-conserved regions, and on the small words which can be generated at the connections between branch and consensus regions.
   (b) Extension of hits for longer High Scoring alignment Pairs (HSPs); HSPs are defined by four parameters: (1) relative position (possibly negative) of the left extremity of the query relatively to the left extremity of the Alignment Graph; (2) position of the beginning of the HSP; (3) position of the HSP end[3]; (4) score of the HSP.
   The principle of the extension is simple: extension is first performed rightwards from the hit; for each crossed branch region, each branch is tried, and the best branch is kept; the position corresponding to the record score is memorized; extension is similarly made leftwards. Practical implementation needs to differentiate the initialisation of extension in the cases where the small word of Alignment-graph of a hit spans only a consensus region, only a branch region, or both consensus and branch region, with a special case "branch-consensus-branch". We detail in figures 4 and 5 the case where the hit spans a consensus region; the other cases are handled in the same spirit.
   (c) The last substep eliminates the alignments with little probabilistic significance.

[2] Steps 1 and 2(c) are identical to Blast.
[3] Positions of beginning and end of the HSP may be chosen indifferently relatively to the query sequence, or to the Alignment Graph.

```
Extend_Cons_Hit(Hit)
{
        Current_Region = Region_Containing_The_Hit
        Copy_Consensus(Current_Region)
        Extend_Right(Hit, Current_Region)
        Extend_Left (Hit, Current_region)
        Alignment_Parameters =
              ( (Rel_Orig = Hit_Query - Hit_AlignGraph),
                (Begin    = Query_Best_Left_position),
                (End      = Query_Best_Right_position),
                (Score    = Best_Left_Score + Best_Right_Score) )
}

/* AUXILIARY DATA STRUCTURE */
string W                          /* Working String */

/* SUBROUTINES */
Copy_Consensus( Current_Region )
        /* Copy Consensus of Current_Region to corresponding */
        /* positions of the Working String W */

Copy_Branch_Number_i( Current_Region )
        /* Copy Branch number i of Current_Region to corresponding */
        /* positions of the Working String W */

Score Blastlike_RightExtension( Hit, FromPos, FromScore, EOR )
{
        Score = FromScore;        Current_Pos = FromPos

        Compute Current_Query_Pos and Current_AlignGraph_Pos from Hit and Current_Pos

        Record_Score = -∞
        while ( Current_Pos ≤ EOR ) {
              Score += SimilarityMatrix( Current_Query_Pos++, Current_AlignGraph_Pos++ )
              Current_Pos++
              if (Score > Record_Score) {
                    Record_Score = Score;     Query_Record_Pos = Current_Query_Pos
              }
              if ( Score ≤ 0 )
                    return ( Record_Score, Query_Record_Pos)
        }
        return ( Record_Score, Query_Record_Pos )
}
```

Figure 4: BestPath algorithm: the top function for extension of a Hit inside a consensus region, and some subroutines.

```
Extend_Right(Hit, Current_Region) {
        FromPos = Hit;                  FromScore = 0
        Copy_Consensus( Current_Region )

        forever {
                /* CONSENSUS REGION */
                EOR = End_of( Current_Region )
                Best_Right_Score = Blastlike_RightExtension( Hit, FromPos, FromScore, EOR )
                Query_Best_Right_Pos = Query_Record_Pos

                if ( Extension_Terminated_Before_End_of_Region )        return
                if ( Current_Region == Right_Last_Region )              return

                Current_Region = Next_Right_Region_of( Current_Region )
                EOR = End_of( Current_Region )

                /* BRANCH REGION */
                FromPos = Beginning_of(Current_region);     FromScore = Best_Right_Score
                Temp_Record_Score = -∞;                      End_of_Region = FALSE

                for (i=0; i < number_of_branches; i++) {
                        Branch_Record_Score = 0
                        Copy_Branch_Number_i( Current_Region )
                        Branch_Record_Score =
                                Blastlike_RightExtension( Hit, FromPos, FromScore, EOR )
                        if ( Branch_Record_Score > Temp_Record_Score ) {
                                Temp_Record_Score = Branch_Record_Score
                                Query_Best_Right_Pos = Query_Record_Pos
                        }
                        if ( Extension_Terminated_at_End_of_Region )
                                End_of_Region = TRUE
                }
                Best_Right_Score = Temp_Record_Score

                if ( End_of_Region == FALSE )                   return
                if ( Current_Region == Right_Last_Region )      return

                Current_Region = Next_Right_Region_of( Current_Region )

                FromPos = Beginning_of(Current_region);     FromScore = Best_Right_Score
                Copy_Consensus( Current_Region )
        }
}
```

Figure 5: Best-Path algorithm: right extension of a Hit.

4 Applications

As mentionned earlier, two main applications are possible with the BestPath approach, like with the profile one.

- The first classical application concerns family analysis; knowing a family of homologous protein sharing sufficient similarity to build a multialignment, we want to detect in a database of sequences which ones are related to the family. We give in section 4.1 a comparison of six different methods applied to an analysis of similarity with a subset of the globins family.
- The second application looks in reverse direction for similarities between a sequence and a database of multialignments; after reviewing some aspects of the ProDom28 database, we develop this approach in 4.2.

4.1 Similarity searching between a multialignment and a set of sequences (family analysis); comparisons of different methods

We restricted comparisons to ungapped similarity methods, leaving all issues relative to similarity searching with gaps to future work.

Figures 6, 7, 8, 9, 10 and 11 show the results of similarity search with 6 differents methods.

- (A) Best Path method with Alignment Graph (BlastMultAl).
- (B) Unweighted Consensus.
- (C) Weighted Consensus.
- (D) Unweighted Profile (profile with average score at each position).
- (E) Weighted Profile (profile with weighted score at each position).
- (F) Flat sequences: with n sequences S_m in the multialignment, for each test sequence S_t, the sequence S_m scoring best with S_t has been selected, and the corresponding record score memorized.

4.1.1 Weighting schemes

Weighting schemes for consensus and profile of sequence alignments have been computed by using ClustAlv [11] and TreeWgt [6].

ClustAlv produces a binary phylogenetic-type tree for sequences belonging to a multialignment; each sequence is assigned to a node in the tree and a tree topology gives similarity level between sequences.

With such a topologic tree as input, TreeWgt gives weights for the sequences, and these weights may be used to compute weighted consensus or weighted profiles.

4.1.2 Description of the tests

Each sequence with odd rank in Multialignment #1 (globin family) of ProDom28 has been selected to build a multialignment, the Probe-Family.

Test sequences are:

- globins sequences of SwissProt32 which do not belong to domain #1 of Prodom28 (66 sequences);
- non globins sequences randomly chosen inside SwissProt32 (1875 sequences).

For each method, Z'-scores have been computed as follows.

1. For each test sequence, the alignment with the Probe-family producing the best score S in the additive scheme described previously has been kept.
2. Expectation E and variance V of these scores for the non-globins sequences has been computed.
3. A Z-score has been computed as
$$Z = \frac{S - E}{V}.$$
4. The probabilistic results of section 2.3 shows that a linear regression applied on the plot $Z = f(log(L))$, where L is the length of the sequences, will have a slope $\frac{1}{\lambda}$; computing λ by this linear regression, we define a Z'-score as
$$Z' = Z - \frac{1}{\lambda}\log(L),$$
to get rid of the deviation caused by the sequences length.

4.1.3 Discussion of the plots

The 66 globins sequences are represented by dark points in the plots, and the 1875 non globins sequences by grey points.

Most of the test globins sequences have a length close of 150, giving many points on the vicinity of the vertical line $\log(L) = 5$.

The six methods discriminate clearly six globins highly similar to the Probe-Family; however a seventh one has very high scores with all methods except BlastMultAl, which gives a lower score.

All the six methods do not recognize a part of the test globins sequences, which stay deep inside the cloud of random sequences.

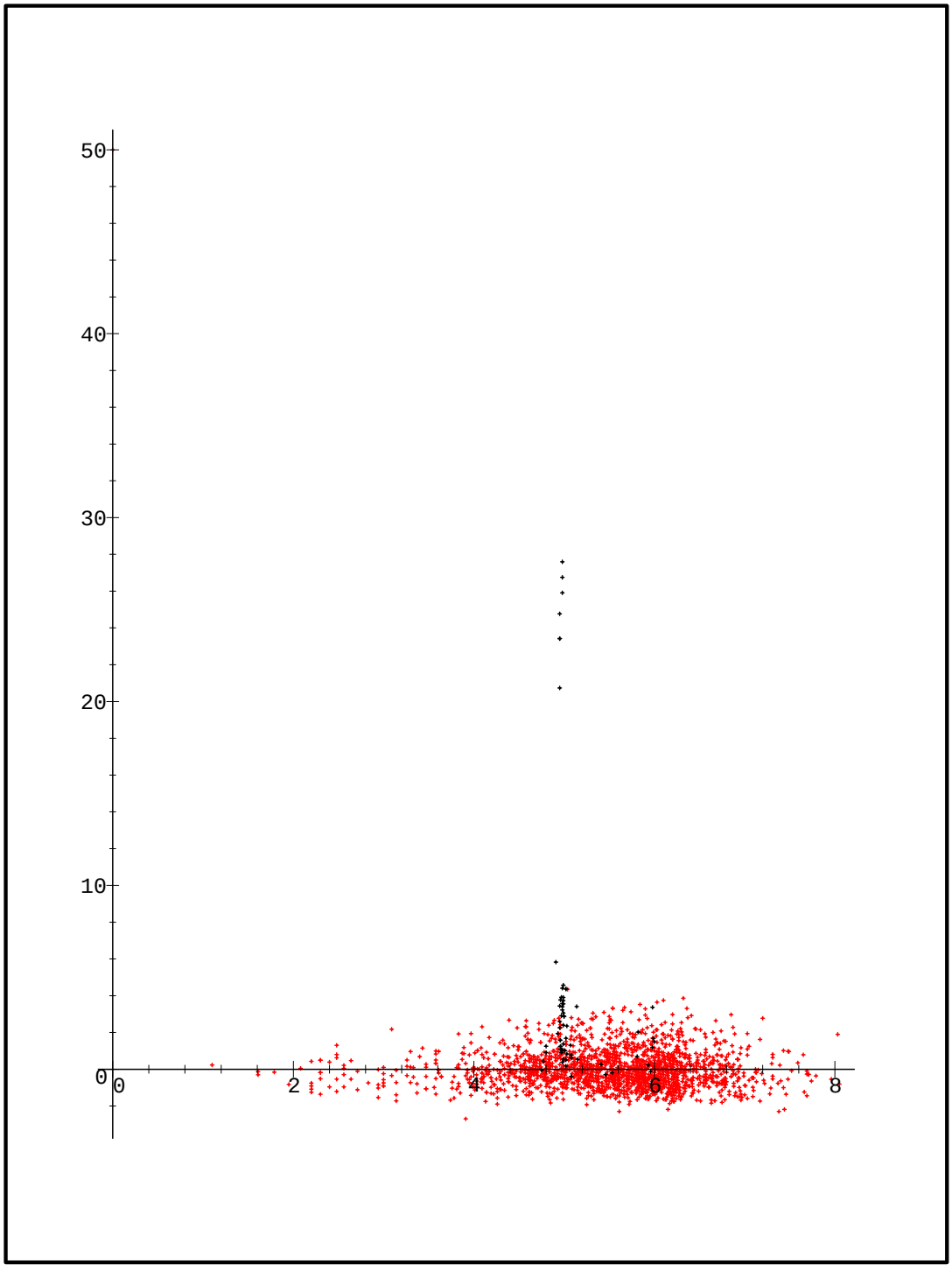

Figure 6: Z'-scores for BlastMultAl (A)

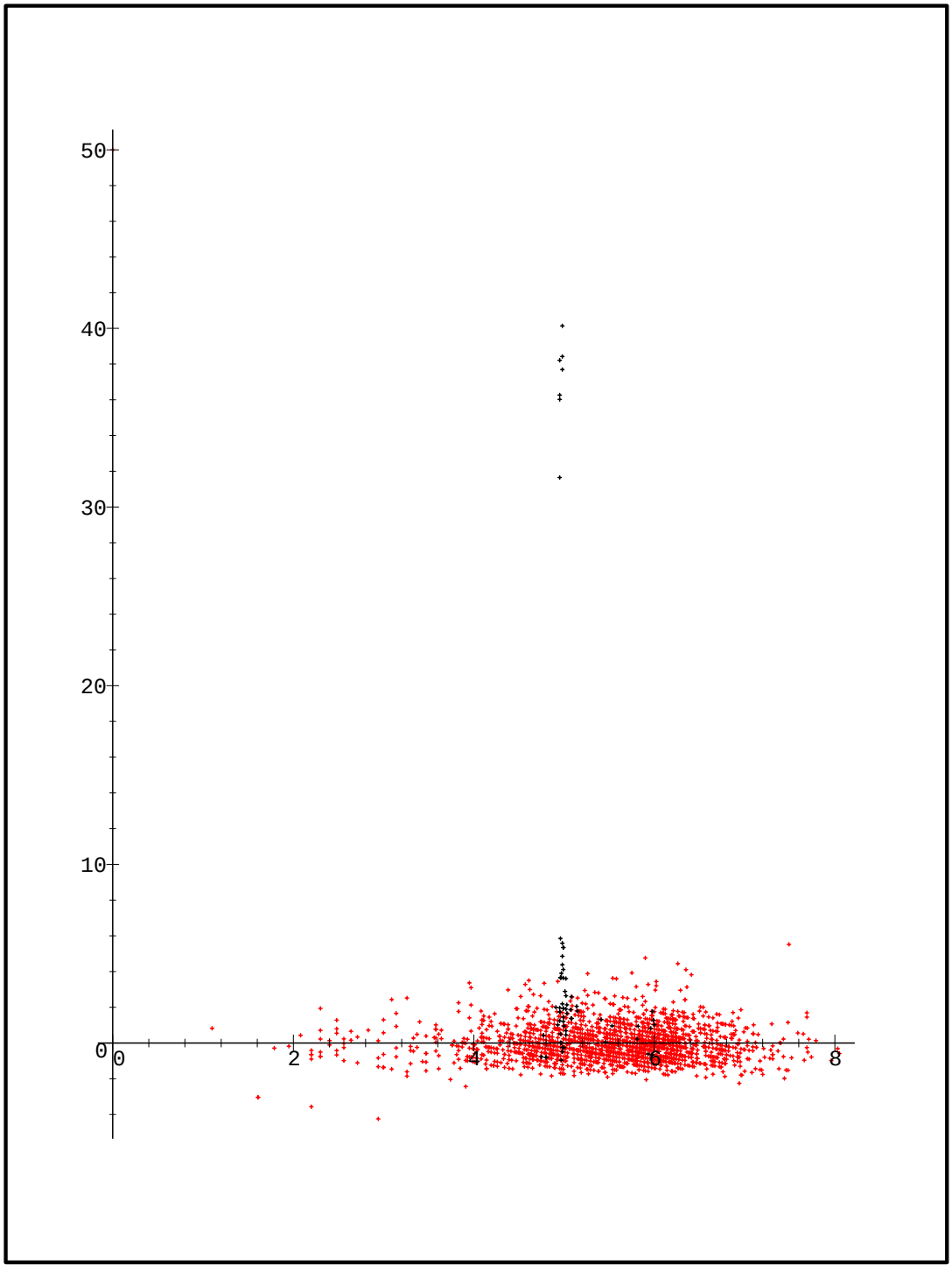

Figure 7: Z'-scores for the Unweighted Consensus method (B)

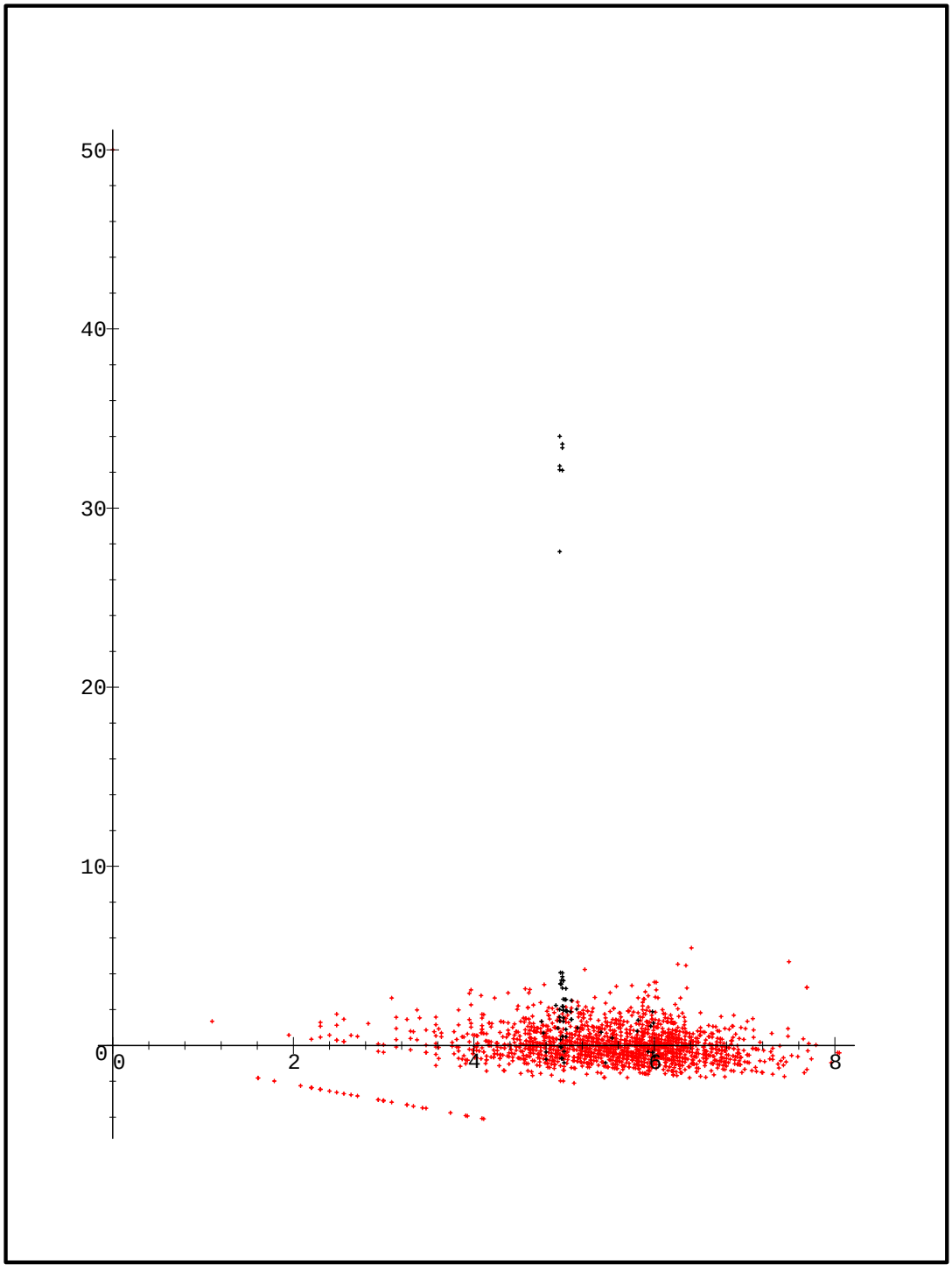

Figure 8: Z'-scores the Weighted Consensus method (C)

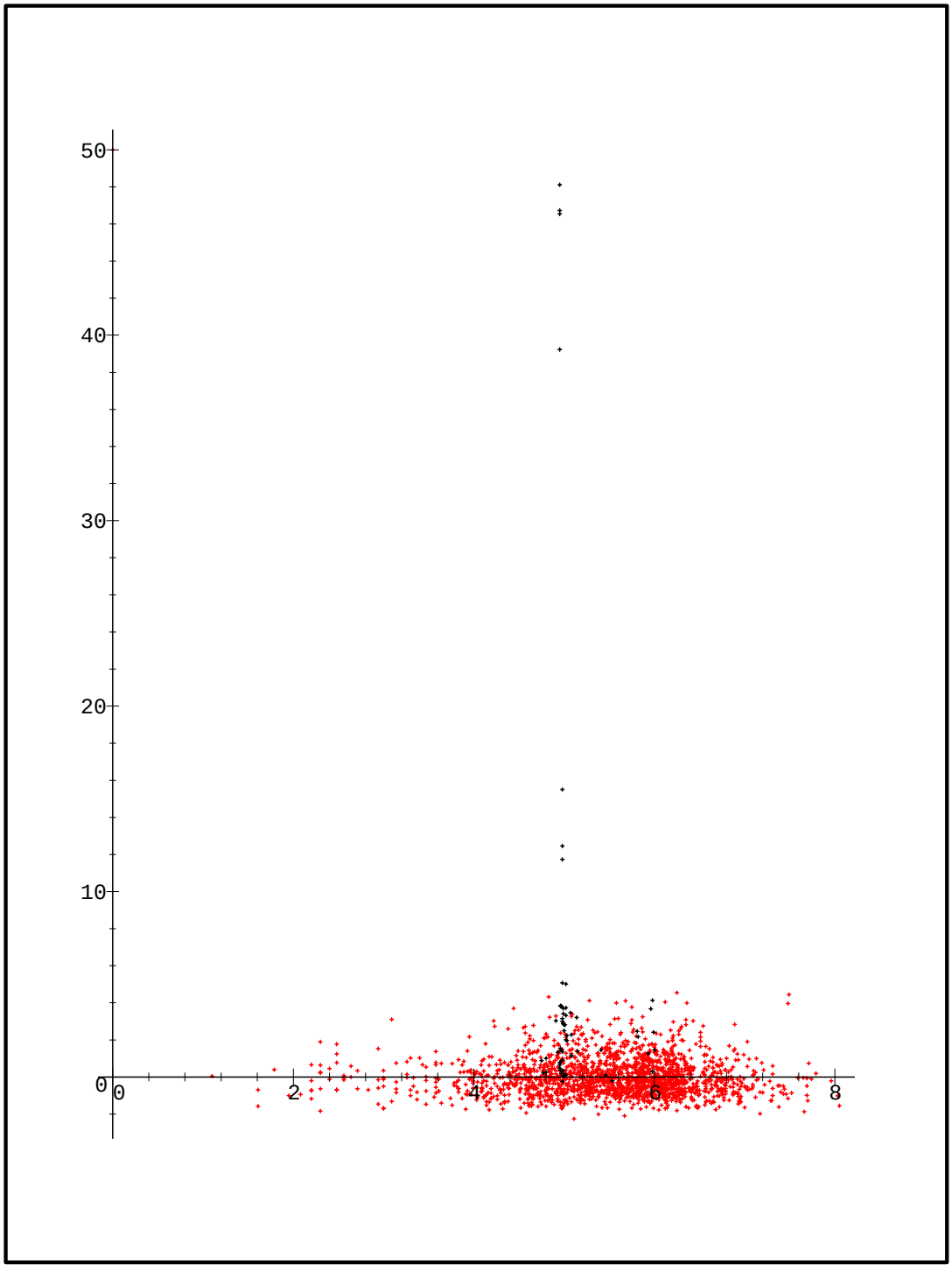

Figure 9: Z'-scores for the Unweighted Profile method (D)

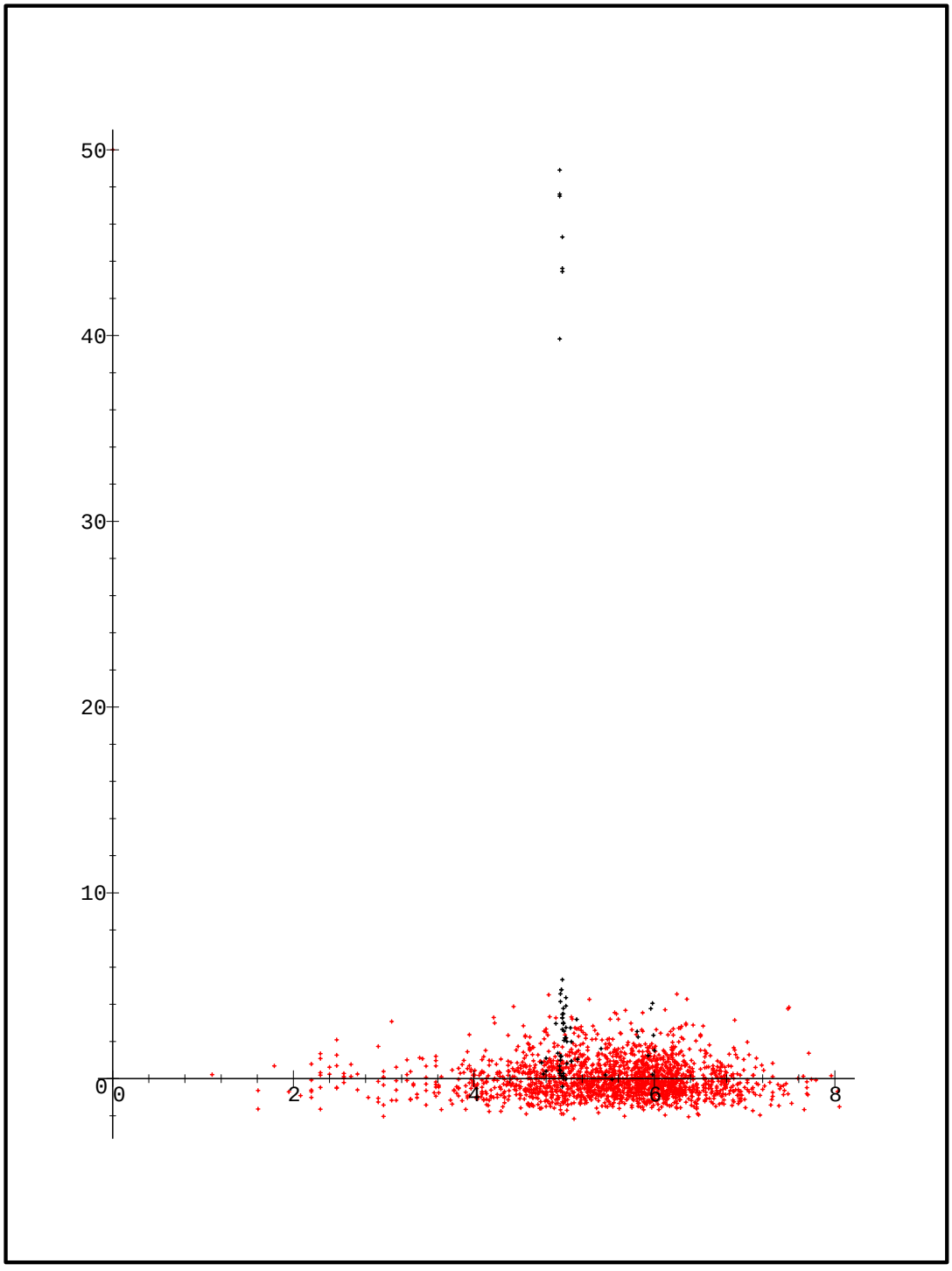

Figure 10: Z'-scores for the Weighted Profile method (E)

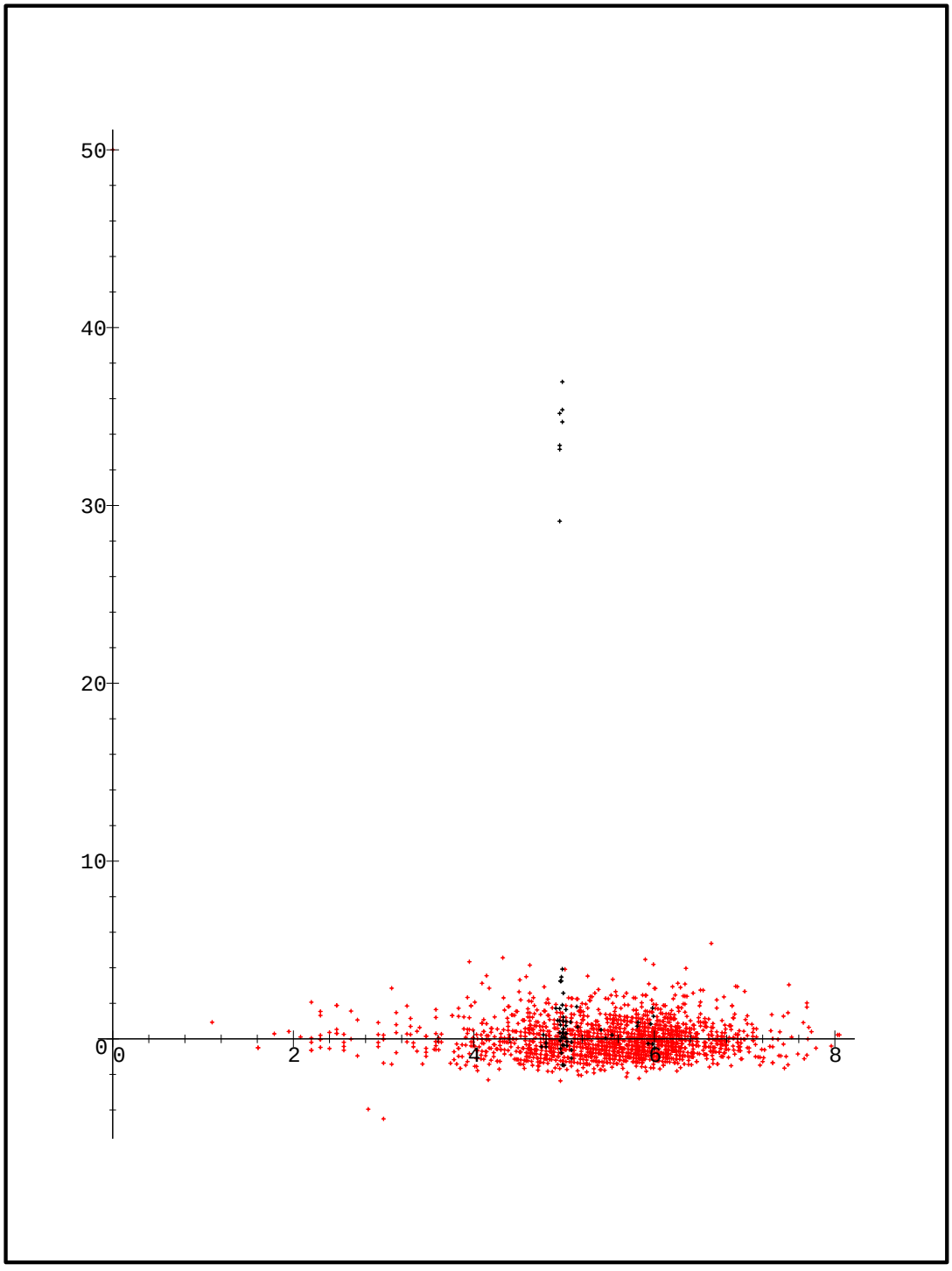

Figure 11: Z'-scores for the Flat Sequences method (F)

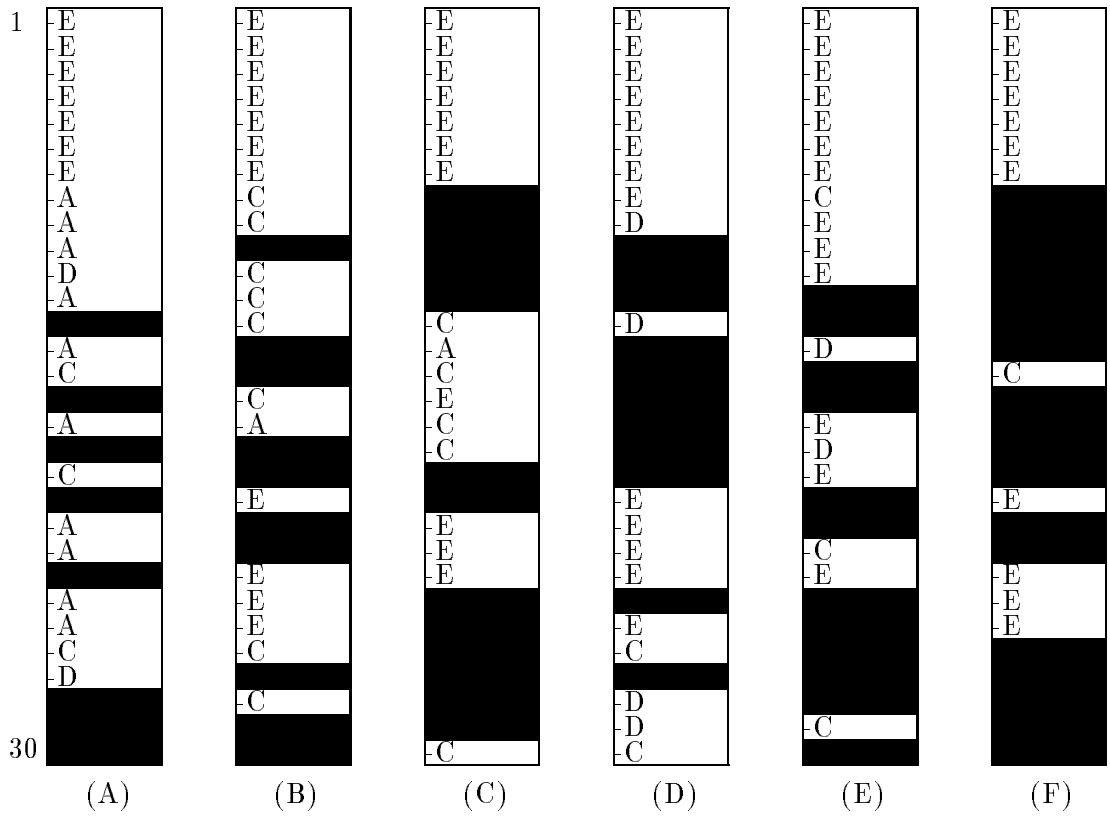

Figure 12: the 30 best Z'-scores results obtained with a globin Probe-Family from a target set of 66 globins and 1875 non globins; (A) BlastMultAl; (B) Unweighted Consensus; (C) Weighted consensus; (D) Unweighted Profile; (E) Weighted Profile; (F) Flat Sequences. Non Globins are represented by a black rectangle. Letters -A,...,-F indicate the method giving the highest Z'-score for the corresponding protein.

	(A)	(B)	(C)	(D)	(E)	(F)
GLOBINS						
$3 \leq Z'$	0.364	0.303	0.273	0.364	0.348	0.182
$2 \leq Z' < 3$	0.121	0.106	0.136	0.182	0.212	0.015
$Z' < 2$	0.515	0.591	0.591	0.455	0.439	0.803
NON GLOBINS						
$3 \leq Z'$	0.007	0.011	0.010	0.013	0.011	0.010
$2 \leq Z' < 3$	0.038	0.024	0.022	0.031	0.032	0.033
$Z' < 2$	0.954	0.965	0.968	0.956	0.957	0.958

Table 1: Comparisons of the Z'-scores for the six methods of figure 12.

4.1.4 Best ranking comparisons

Figure 12 shows how many non-globins are found in the 30 highest Z'-scores with each of the six methods; BlastMultAl has the smallest number of nonglobins in the 30 results, while flat sequences have the largest. For each protein in this table is also given the method with the highest Z'-score. For both consensus and profiles approaches, weighting the sequences improves clearly the results.

The Weighted Profile gives in most of the cases the best Z'-scores for the proteins of figure 12; however, BlastMultAl detects a set of distant proteins better than the other methods (letters -A in first column of figure 12). BlastMultAl is therefore a useful complement to Weighted Profiles which are the more sensitive approach.

4.1.5 Z'-score global comparisons

Table 1 compares the proportion of Z'-scores in the ranges $[0, 2]$, $[2, 3]$, and $[> 3]$ for the six methods.

The number of true positives over $Z' = 3$ is similar for BlastMultAl (0.364), and the profiles methods (0.364 and 0.348), but BlastMultAl has the smallest number of false positives with $Z' > 3$ (0.007).

In the middle range $[2, 3]$, BlastMultAl does not perform very well. When considering $Z' \geq 2$, Profiles find more true positives (0.546 and 0.560) than BlastMultAl (0.485).

Flat sequences performs very badly for true positive (0.182); this result gives evidence of the advantage obtained when sophisticating the methods.

4.1.6 Finding distant similarities with Alignment Graphs

Tables 2 and 3 show how the sequences with the 30 best BlastMultAl scores are found with the other methods; table 2 gives the rank obtained by the sequences and table 3 the corresponding Z'-score. These tables show that some distant similarities are detected by the BlastMultAl, and not by the other methods.

- GLB2_CALSO, a mollusc globin, and LGB3_SESRO, a leghemoglobin , are better found by BlastMultAl (leghemoglobins are hemoproteins present in the root nodules of leguminous plants; their function is to provide oxygen to the bacteroids).
- GLB4_LUMTE, a subunit of globin of worm is found by BlastMultAl and by Profiles, but not by the other methods; another worm globin subunit, GLB1_LUMTE, is not found by any method, except by BlastMultAl.
- Interestingly, the insect globin GLBT_CHITH is only found by BlastMultAl; the giant hemoglobins of worms GLB1_PHESE, although preceded by 5 non-globins, is also better found by BlastMultAl.

Rank obtained by similarity search of a globin Probe-Family of ProDom28 with 66 globins and 1875 non-globins						
Name	Blast-MultAl	Cons. UnWght	Cons. Weight	Prof. UnWght	Prof. Weight	Flat Sequences
HBB_MICGA	1	1	2	7	6	1
HBB_LYNLY	2	4	3	5	5	4
HBB_APTFO	3	2	6	6	4	2
HBA2_ARCGA	4	3	1	1	1	3
HBA1_ARCGA	5	5	4	3	3	5
HBA_LYNLY	6	6	5	2	2	6
HBA_MICGA	7	7	7	4	7	7
GLB2_CALSO	8	90	65	45	47	134
LGB3_SESRO	9	98	94	53	49	794
GLB2_TYLHE	10	1296	518	1065	1238	1174
GLB4_LUMTE	11	458	530	9	14	313
GLBT_CHITH	12	579	348	785	867	632
RL22_HALMA •	13	938	681	148	249	1718
GLB1_CALSO	14	878	1016	202	261	788
LGB1_MEDTR	15	11	17	30	29	627
MCP2_ECOLI •	16	551	978	124	109	1648
GLB1_LUMTE	17	879	630	604	573	970
YFH5_YEAST •	18	1530	1564	683	559	646
LGB4_MEDSA	19	12	18	26	22	485
GLYA_SALTY •	20	1275	934	214	285	1680
GLB2_LUMTE	21	1077	1567	39	67	1885
LGB2_SESRO	22	97	93	52	48	793
CYS1_DICDI •	23	466	1342	869	772	1432
GLB1_PHESE	24	133	149	403	554	606
GLB3_LUMTE	25	86	84	37	41	119
LGB2_MEDTR	26	13	30	350	36	620
HMPA_ECOLI	27	127	108	13	18	129
SSO1_YEAST •	28	298	1035	169	536	301
COAT_MISV •	29	711	572	766	755	1090
THYL_RAT •	30	27	365	929	937	21
Non globins proteins are marked by a •						

Table 2: Similarities found by BlastMultAl of the globin Probe-Family with globins GLB2-_TYLHE, GLBT_CHITH, GLB1_CALSO, GLB1_LUMTE and GLB1_PHESE are not detected by the other methods. Similarities with GLB2_CALSO and LGB3_SESRO are better detected.

Z'-scores obtained by similarity search of a globin Probe-Family of ProDom28 with 66 globins and 1875 non-globins						
Name	Blast-MultAl	Cons UnWght	Cons Weight	Prof UnWght	Prof Weight	Flat Sequences
HBB_MICGA	27.59	40.14	33.57	11.73	43.44	36.95
HBB_LYNLY	26.75	37.70	33.36	15.50	43.62	34.70
HBB_APTFO	25.91	38.43	32.11	12.46	45.30	35.38
HBA2_ARCGA	24.77	38.22	34.01	48.11	48.91	35.17
HBA1_ARCGA	23.42	36.27	32.35	46.54	47.50	33.38
HBA_LYNLY	23.42	36.03	32.14	46.72	47.61	33.15
HBA_MICGA	20.74	31.65	27.57	39.23	39.82	29.11
GLB2_CALSO	5.83	2.00	2.24	3.04	2.96	1.72
LGB3_SESRO	4.57	1.93	1.96	2.87	2.96	0.10
GLB2_TYLHE	4.41	-0.49	0.51	-0.24	-0.44	-0.35
GLB4_LUMTE	4.37	0.69	0.48	5.01	4.36	0.98
GLBT_CHITH	4.37	0.45	0.90	0.10	-0.02	0.31
RL22_HALMA •	4.35	-0.05	0.25	1.76	1.24	-1.05
GLB1_CALSO	3.92	0.01	-0.10	1.44	1.20	0.11
LGB1_MEDTR	3.90	5.34	3.62	3.41	3.49	0.32
MCP2_ECOLI •	3.86	0.50	-0.08	1.96	2.13	-0.95
GLB1_LUMTE	3.77	0.02	0.33	0.37	0.43	-0.10
YFH5_YEAST •	3.75	-0.77	-0.72	0.27	0.45	0.30
LGB4_MEDSA	3.73	5.34	3.62	3.70	3.78	0.55
GLYA_SALTY •	3.65	-0.47	-0.03	1.38	1.10	-1.00
GLB2_LUMTE	3.58	-0.24	-0.73	3.15	2.65	-1.47
LGB2_SESRO	3.56	1.93	1.96	2.87	2.96	0.10
CYS1_DICDI •	3.52	0.68	-0.48	-0.02	0.12	-0.65
GLB1_PHESE	3.44	1.73	1.58	0.76	0.46	0.35
GLB3_LUMTE	3.41	2.05	2.03	3.21	3.19	1.81
LGB2_MEDTR	3.40	4.86	3.21	0.92	3.27	0.33
HMPA_ECOLI	3.37	1.77	1.89	4.14	4.05	1.73
SSO1_YEAST •	3.36	1.08	-0.12	1.63	0.52	1.02
COAT_MISV •	3.32	0.22	0.42	0.14	0.15	-0.24
THYL_RAT •	3.32	3.63	0.83	-0.08	-0.09	3.35
Non globins proteins are marked by a •						

Table 3: Z'-scores corresponding to the preceding table.

4.2 Searching similarities of a sequence with a database of multialignments

We turn now to the second application of our method; we get a query sequence, and we want to know which ones of a set of multialignments are similar to the query. This approach has a direct application with the database ProDom. We applied our method to the ProDom28 version, which contains ungapped alignments.

4.2.1 Overview of ProDom28 database

ProDom28 [19] has been generated from SWISSPROT28[4] as follows:

- pairwise comparisons of SWISSPROT sequences produce a set of HSPs (High Scoring Segment Pairs);
- from overlapping HSPs are built bigger HSSs (High Scoring Segment Sets) and a graph of HSSs;
- multiple alignments are produced by concatenation of HSSs belonging to paths of this graph which contain a maximal number of sequences.

Table 4 gives an overview of ProDom28 database; it gives also an idea of the structure of the Alignment Graphs generated by the preprocessing phase, with value $\tilde{\tau} = 95\%$.

number of sequences in the multialignment	number of multialignments	number of branch regions	number of branch positions	total number of positions	percentage of branch positions
1	15074			2595435	0.0
2	3046	16031	68697	547066	12.56
3	1370	8682	45549	243514	18.70
4	836	5444	30566	168549	18.13
5	506	3906	24325	99597	24.42
6	336	3028	21679	72094	30.07
7	274	2705	21197	61398	34.52
8	183	1998	17098	43195	39.58
9	161	1706	14323	34913	41.02
≥ 10	1319	12991	147659	260516	56.68
≥ 2	8031			983776	

Table 4: Overview of ProDom28 and preprocessing results with $\tilde{\tau} = 95\%$ (branch segments smaller than 3 have been eliminated).

[4] Generation of ProDom32 is on the way; ProDom32 will be generated from SWISSPROT32 and will contain multialignments with gaps.

4.2.2 Customizing the probabilistic parameters K and λ from simulation results

Each multialignment containing $s \geq 2$ sequences has been processed with BlastMultAl against a database of 10000 random sequences. Using approximation of the exponential for small values of the exponent, a variable transform in equation 2 gives

$$\mathcal{F}(S(l_q l_r)) \approx Prob(S(l_q l_r) > y) \approx K l_q l_r e^{-\lambda y}, \tag{5}$$

where $\mathcal{F}(S)$ is the repartition function of the scores obtained by simulation, l_q is the length of the query sequence and l_r the length of the random sequences (chosen as 300). The maximum score has been kept for each random sequence, and the logarithm of the repartition function $\mathcal{F}(S)$ has been plotted against the score y. The parameters K and λ have then been evaluated directly by linear regression for each multialignment; practical computations show the good quality of this regression and the negligible influence of the approximation of the exponential on the results. This gives numerical evidence that the Iglehart-Karlin formula is valid for alignments $\mathcal{A}$-graphs.

Intuitively, similarity searching with a multialignment, when applying our BestPath approach is roughly equivalent to searching similarity with a longer sequence containing the same potential alignments. We call "*complexity length* of the alignment" the length of this potentially equivalent longer single sequence. Let L be the length of a multialignment, and b be the number of branches in branch regions and r the number of branch regions. We define the *complexity length* $\mathcal{L}^c$ as

$$\mathcal{L}^c = L \times r^b,$$

or,

$$\log(\mathcal{L}^c) = \log(L) + b \log(r).$$

We plot in Figure 13 the results of simulation for λ as a function of $\mathcal{L}^c$ for $b = 2$ and $b = 5$; note that the Alignment Graphs with $b = 5$ have been generated by multialignments containing $s \geq 5$ sequences. Results of simulation for K give an expectation of 0.19, with a variance of 0.1, independent of the value of $\mathcal{L}^c$. A result from direct computation of equation 4 with a consensus database gives $K \approx 0.13$.

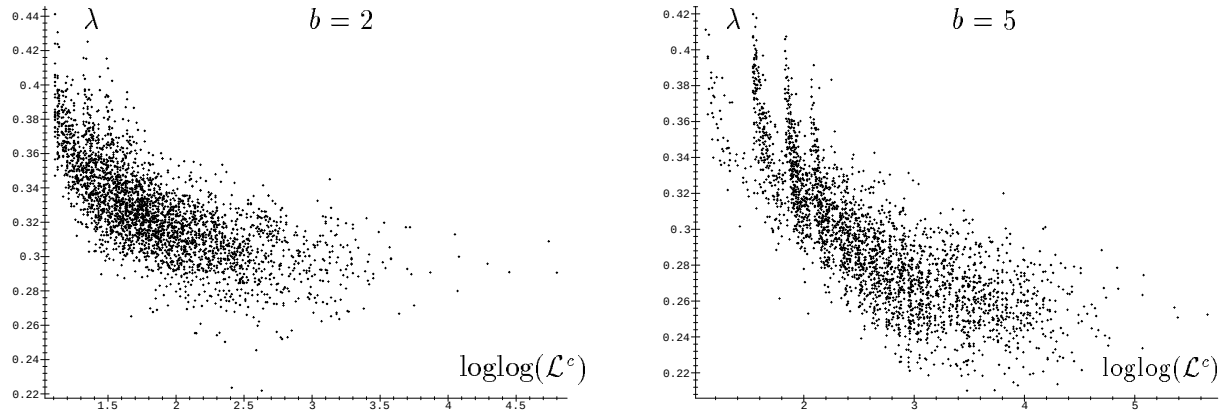

Figure 13: Values of λ from ProDom28 alignments containing $b = 2$ and $b = 5$ branches by branch region.

```
------------------------------------------------------------------------------
                                                                    Smallest
                                                                    Poisson
                                                          High      Probability
Sequences producing High-scoring Segment Pairs:           Score     P(N)      N

554   (21)   GLOBIN PRECURSOR. PRECURSOR (ERYTHROCRUORIN  ...     94  0.0057    1
2647   (5)   GLOBIN, COMPONENT POLYMERIC MAJOR MONOMERIC P...     50  0.23      1
1740   (7)   HEMOLYSIN PROTEIN). (CLYII-A II) (CYTOLYSIN A...     62  0.28      2
2121   (6)   . PRIMASE DNA Q05271 P07362 P30103 P02923 P33...     65  0.88      1
3140   (4)   (CLONE PRECURSOR ABUNDANT EMBRYONIC USP92). P...     52  0.93      1
1284  (10)   DEHYDROGENASE (IMP INOSINE-5'-MONOPHOSPHATE ....     76  0.93      1
1261  (10)   ENZYME SYSTEM, (PHOSPHOTRANSFERASE PHOSPHOTR....     70  0.98      1
1158  (11)   EXTRACELLULAR GLOBIN (ERYTHROCRUORIN). (GLOB....     63  0.98      1
982   (13)   HEAVY MYOSIN CHAIN, (MHC CHAIN SKELETAL MUSC....     73  0.9997    1
1051  (12)   CHEMOTAXIS METHYL-ACCEPTING CHEMORECEPTOR PR....     63  0.9999    1
1579   (8)   RIBOSOMAL 30S S15. MITOCHONDRIAL P21771 P0637...     60  0.99995   2
------------------------------------------------------------------------------
```

Figure 14: Result of a BlastMultAl query on ProDom28; the query sequence is the respiratory protein GLBT_CHITH.

```
------------------------------------------------------------------------------
                                                                    Smallest
                                                                    Poisson
                                                          High      Probability
Sequences producing High-scoring Segment Pairs:           Score     P(N)      N

554  (21)   GLB3(2) GLBW(1) GLBV(1)... GLOBIN PRECURSOR. ...     87  9.7e-07   1
470  (24)   LGB3(3) LGBA(2) HBPL(2)... LEGHEMOGLOBIN I. H...     39  0.88      2
177  (46)   TETO(3) TETM(2) RF3(1)... FACTOR ELONGATION (...     41  0.98      2
377  (28)   NUCM(4) PHNL(3) NUOD(1)... LARGE HYDROGENASE ...     53  0.98      1
4429  (3)   DIKINASE PYRUVATE,ORTHOPHOSPHATE PRECURSOR P2...     41  0.993     2
3140  (4)   EA87(1) EA30(1) (CLONE PRECURSOR ABUNDANT EMB...     52  0.995     1
------------------------------------------------------------------------------
```

Figure 15: Blastp query on the unweighted consensus of ProDom28 alignments; the query sequence is GLBT_CHITH.

4.2.3 Querying ProDom with BlastMultAl

The simulations described in the preceding section allow us to give probabilistic significance of alignments obtained with BlastMultAl[5]. An example of query with the respiratory insect protein GLBT_CHITH is given in figure 14. It is to be compared with the same query made on the consensus sequences of ProDom28 (figure 15). The results obtained with the two methods are each other complementary. Only HSPs with true multialignments, containing two or more sequences, are exhibited on figures 14 and 15, (the number of sequences in the multialignments is indicated between brackets). BlastMultAl detects twice as many similarities as Blastp, but does not detects similarities with ProDom alignments 470, 177, 377 and 4429; for these alignments, we have $\lambda \approx 0.25$, to compare to $\lambda \approx 0.32$ computed with equation 3, used for Blastp with the consensus sequences; these low λ values make BlastMultAl miss the similarities of GLBT_CHITH with the corresponding multialignments.

[5] Blast probabilistic and output routines have been used.

From the similarities not detected with the consensus, and found with BlastMultAl, the similarity of GLBT_CHITH with the alignment 1158 of ProDom28 is specially interesting; while GLBT_CHITH is an insect globin, this alignment is composed of globins of earthworms (LUMTE and PHESE), a marine worm (TYLHE), a giant deep-sea tube worm (LAMSP) and a sludge worm (TUBTU); the detection of this distant similarity is not possible only with the consensus.

Figure 13 shows a strong bias of λ for short alignments (say with fewer than sixty positions); this bias is also clear when computing λ by simulation for alignments composed of a single sequence, the case where theoretical computation is applicable. As an example, for the alignment 2647, we have length $L = 26$, and $\lambda \approx 0.37$. A more refine analysis should filter out the false similarity of this alignment with GLBT_CHITH.

Efficiency. Querying ProDom28 for similarities with GLBT_CHITH requires 120s for BlastMultAl and 6s for Blastp on a 133 MHZ Silicon Graphics computer.

5 Conclusions

We described a new method for homology search between a query sequence and multialignments; this method takes advantage of the combinatorial possibilities given by alignment graphs, which differentiate conserved subparts of the multialignment and non-conserved ones.

The BlastMultAl software implements this approach and allows queries on the protein multialignments database ProDom. Comparisons with methods such as unweighted or weighted consensus, unweighted or weighted profiles are good; comparison with a "flat sequences" approach is clearly in the advantage of BlastMultAl. We detect distantly related similarities, and particularly, similarity of an insect globin with a probe globin family containing no such globins. We can perform similarity queries with ProDom28 in reasonnable processing time, detecting twice as many similarities as with a database of consensus sequences. Our approach is complementary with simple approaches like consensus and with popular ones like profiles.

Future work includes handling multialignments with gaps, and getting better probabilistic insights to the parameters of the algorithm.

Acknowledgement. I am grateful to Daniel Kahn, Florence Corpet, Claude Chevalet, Mireille Régnier, Elizabeth Greene and Jean-Marc Steyaert for frequent discussions about the algorithm proposed in this paper, and to Eithne Murray for correcting it. I am indebted to Jean-Jacques Codani for simulating random queries on a powerful net of computers to evaluate the parameters K and λ for the multialignments of ProDom28.

References

[1] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. Basic Local Alignment Search Tool. *J. Mol. Biol. 215* (1990), 403–410.

[2] Arnold, B. C., Balakrishnan, N., and Nagaraja, H. N. *A First Course in Order Statistics.* John Wiley, 1992.

[3] Chung, K. L. *A Course in Probability theory.* Academic Press, 1974. second edition.

[4] Crochemore, M., and Rytter, W. *Text Algorithms.* Oxford University Press, 1994.

[5] Feller, W. *An Introduction to Probability theory and Its Applications.* John Wiley, 1966. second edition.

[6] Gerstein, M., Sonnhamer, E., and Chothia, C. Volume changes in protein evolution. *J. Mol. Biol. 236* (1994), 1067–1078. appendix.

[7] Gribskov, M. Profile analysis. *Methods in Molecular Biology 25* (1994), 247–266.

[8] Gribskov, M., Lüthy, R., and Eisenberg, D. Profile analysis. *Methods in Enzymology 183* (1990), 146–159.

[9] Gribskov, M., MacLachlan, A., and Eisenberg, D. Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad Sci. USA 84* (1987), 4355–4358.

[10] Henikoff, S., and Henikoff, J. G. Automated assembly of proteins blocks for database searching. *Nucleic Acids research 19*, 23 (1991), 6565–6572.

[11] Higgins, D., Fuchs, R., and Bleasby, A. J. CLUSTAL V: improved software for multiple sequence alignment. *CABIOS 8* (1992), 189–191.

[12] Iglehart, D. L. Extremes values in the GI/G/1 queues. *The annals of Mathematical Statistics 43*, 2 (1972), 627–635.

[13] Karlin, S., and Altschul, S. F. Methods for assessing the statistical significance of molecular sequences features by using general scoring schemes. *Proc. Natl. Acad Sci. USA 87* (march 1990), 2264–2268.

[14] Karlin, S., and Brendel, V. Chance and statistical significance in protein and DNA sequence analysis. *Science 257* (july 1992), 39–49.

[15] Karlin, S., and Dembo, A. Limit distribution of maximal segmental score along Markov-dependant partial sums. *Adv. Appl. Prob. 24* (1992), 113–140.

[16] Needelman, S. B., and Wunsch, C. D. A general method applicable to the search of similarities in the amino acid sequences of two proteins. *J. Mol. Biol. 48* (1970), 443–453.

[17] Pearson, W. R., and lipman, D. J. Improved tools for biological sequences comparisons. *Proc. Natl. Acad Sci. USA 85* (1988), 2444–2448.

[18] Robert, P. Cours de DEA de files d'attente. Tech. rep., Laboratoire de Probabilités, Université Paris VII, 1993.

[19] Sonnhamer, E. L., and Kahn, D. The modular arrangement of proteins as inferred from analysis of homology. *Protein Science 3* (1994), 482–492.

[20] Waterman, M. S. *Introduction to Computational Biology.* Chapman & Hall, 1995.