Sequence analysis

Rapid motif-based prediction of circular permutations in multi-domain proteins

January Weiner 3rd\(^1\), Geraint Thomas\(^2\) and Erich Bornberg-Bauer\(^1,\,*\)

\(^1\)Division of Bioinformatics, School of Biology, Institute of Botany, The Westphalian Wilhelm’s
University of Münster Schlossplatz 4, D48149 Münster, Germany and \(^2\)Faculty of Biological Sciences,
The University of Leeds, Leeds, UK

Received on May 7, 2004; revised and accepted on October 8, 2004

ABSTRACT

Motivation: Rearrangements of protein domains and motifs such as swaps and circular permutations (CPs) can produce erroneous results in searching sequence databases when using traditional methods based on linear sequence alignments. Circular permutations are also of biological relevance because they can help to better understand both protein evolution and functionality.

Results: We have developed an algorithm, RASPODOM, which is based on the classical recursive alignment scheme. Sequences are represented as strings of domains taken from precompiled resources of domain (motif) databases such as ProDom. The algorithm works several orders of magnitude faster than a reimplementation of the existing CP detection algorithm working on strings of amino acids, produces virtually no false positives and allows the discrimination of true CPs from ‘intermediate’ CPs (iCPs). Several true CPs which have not been reported in literature so far could be identified from Swiss-Prot/Trembl within minutes.

Availability: Source codes, additional scripts, data and a web-based interface can be found on: http://www.uni-muenster.de/Biologie.

Contact: ebb@uni-muenster.de

INTRODUCTION

Domain rearrangements are considered to be an important mechanism of adaptation and evolution of new functions (Vogel et al., 2004; Soding and Lupas, 2003; Dooolittle and Bork, 1993; Heringa and Taylor, 1997). Known types of domain rearrangements encompass domain duplication (A → AA), swaps (AB → BA), domain deletion (ABC → AC) and circular permutations (CPs) (ABCD → CDAB).

Sequence analysis, widely used to assess protein function, is hampered by non-linear rearrangements of proteins. Heuristic algorithms for sequence comparison and search, such as BLAST (Altschul et al., 1990), rely on the sequence linearity. Therefore, they can only detect partial homologies if the linear order of domains (the string of domains) is preserved; a permutation drastically lowers the chance of the homology detection, even if the rearranged fragments still show a considerable degree of homology with the original protein. An example illustrating this problem is the case of two hepatitis B virus DNA polymerases, O91514 (gi 41152697) and Q69590 (gi 4377612). The list of hits in a BLAST search against the NR database with default parameters on the NCBI server for O91514 homologs does not contain Q69590. However, if O91514 is rearranged by placing the 300 N-terminal amino acids at the C-terminus, the first hit in the NR database is Q69590 sequence. Therefore, it is important to have algorithms allowing to find pairs of proteins which are related by such non-linear rearrangements.

In biological sequences, a circular or cyclic permutation is a rearrangement of a sequence (ABCD, i.e. comprising the four domains A, B, C and D in linear order) such that an N-terminal fragment of the protein (e.g. AB) is transferred to the C-terminus of the protein (Ponting and Russell, 1995; Jeltsch, 1999). This results in a non-linear rearrangement (CDAB). Such rearrangements can be caused by a gene duplication followed by subsequent insertion of new start and stop codons, shortening the protein at the C- and N-terminus (Jeltsch, 1999; Fig. 1), or by independent assembly of two proteins from existing sequence fragments (‘modules’, as defined in Bujnicki, 2002). We note here that the proposed mechanism of gene duplications, connected with partial domain loss, can lead to the creation of what we denote as incomplete or intermediate circular permutations (iCP; Fig. 1).

Several examples of CPs are known for biological sequences. Perhaps the best studied group of enzymes containing CP-variants are the DNA methyltransferases (Jeltsch, 1999; Bujnicki, 2002); other CP-variants are found for the bovine trypsin inhibitor (Goldenberg and Creighton, 1983), as well as some lectins, swaposins, glucosyltransferases and glucosidases (Lindqvist and Schneider, 1997).

Proteins have also been subjected to artificial CPs. It was shown (Cheltsov et al., 2001; Hennecke et al., 1999) that a large proportion of such arrangements shows normal activity in vivo. This

\*To whom correspondence should be addressed.
demonstrates that the function of the protein is preserved in spite of the reshuffling of the protein, as long as functional entities such as catalytic centres and conserved structural elements remain intact. It is therefore a reasonable assumption that CPs happen domain-wise, by rearrangements at the level of conserved functional domains rather than at the level of the amino acid sequence itself. This is in concordance with the general picture of protein evolution at the molecular level, in which protein domains are conserved units of selection.

A straightforward algorithm for the detection of CPs would be to generate all possible CPs of one sequence and subsequently align them with the second sequence. If the alignment score with a CP is better than in case of a simple alignment, then a CP is identified. Such an algorithm, however, has the time complexity of $O(n^3)$ (Uliel et al., 1999, 2001). Uliel et al. (1999) described a heuristic algorithm of $O(n^2)$ complexity (denoted UFAU in shorthand). Even though this algorithm is much faster than an exact algorithm, it still takes a considerable computational time and requires a further manual inspection of the results. This makes it unrealistic to identify all CPs in a large dataset, for example, the Swiss-Prot/TrEMBL sequence set.

Another heuristic approach has been chosen by Jung and Lee (2001) to investigate CPs of the domains from the SCOP database. Given a pair of proteins, they used a structural alignment algorithm to partially align the central part of one protein with either the N-terminal or C-terminal half of the second one. If such an alignment was successful, a possible CP point was selected. The result was labelled as a CP if the addition of the remaining part of the second protein improved the alignment. Their data suggested that CPs are very frequent since, from 3035 domains which were selected for analysis, 412 are involved in a CP with another domain. This would suggest that even the disruption of a functional domain can lead to another functional domain. However, whether such CPs within a domain sequence in general preserve the same functionality of that domain remains yet to be tested.

Both, the straightforward and the UFAU algorithm, work only at the level of amino acids. The premise of the presented algorithm is to exploit the existence of conserved domains. Since domains are widely disseminated in databases such as Pfam (Bateman et al., 2002), PROSITE (Hulo et al., 2004), BLOCKS (Henikoff et al., 2000), PRINTS (Attwood, 2002), SMART (Schultz et al., 1998) and ProDom (Sonnhammer and Kahn, 1994), we decided to search for CPs in terms of strings of domains. The second premise is that a duplication of a sequence contains all possible CPs of a sequence. Therefore, if two sequences are duplicated, the alignment between those two duplications will make it possible to reveal a CP. This way we are able to present a method which is 50,000 times faster then existing methods, has virtually no false positives and enables automatic updating from existing data resources.

**METHODS AND RESULTS**

**The dataset**

Generally, domains denote evolutionary conserved structural units. Motifs are commonly seen as the corresponding signatures derived from multiple sequence alignments. However, motifs can be much shorter than structural domains. Since the word *domain* is often used to denote motifs as well (see e.g. the name of the ProDom database), and the methods presented here works independently from the chosen definition, we will, in the following, use the word *domain* as a generic expression for any conserved unit, whether defined by structure or sequence.

**Table 1.** Numbers of sequences of domains in the datasets used for RASPODOM evaluation derived from the original ProDom dataset and numbers of positive results for the dataset evaluated

<table>
<thead>
<tr>
<th>ProDom version</th>
<th>2001</th>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sequences</td>
<td>371,996</td>
<td>552,890</td>
</tr>
<tr>
<td>Non-redundant set</td>
<td>115,024</td>
<td>153,646</td>
</tr>
<tr>
<td>Filtered set</td>
<td>66,319</td>
<td>90,300</td>
</tr>
<tr>
<td>Number of CPs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>36</td>
<td>93</td>
</tr>
<tr>
<td>Intermediate</td>
<td>32</td>
<td>60</td>
</tr>
</tbody>
</table>

We chose the ProDom database because for our purposes an extensive coverage of all proteins is more important than an accurate annotation. The ProDom database is generated by automatic annotation from the Swiss-Prot/TrEMBL protein database. Sequences were represented as strings of ProDom domains. For example, a more than 800 amino acids long sequence of hepatitis B virus DNA polymerase (Swiss-Prot entry, O91514) can be represented as the domain string ABCDE, where A, B, C, D, E correspond to the domains with ProDom annotations PD000626, PD347379, PD000646, PD000149 and PD000814. This reduction in string length considerably lowers the computational complexity of the problem. The mean length of a domain string in the input dataset from ProDom is around 6, whereas the mean sequence length in the Swiss-Prot/TrEMBL sequence database is 300, with many proteins having a length of well over 15,000 amino acids.

All sequences with domains contained in the ProDom database were included in the initial dataset. For brevity, we report here only results which have been obtained by applying a stringent filtering procedure as follows: first, sequences with identical strings of domains were removed, such that the remaining dataset became non-redundant. Second, all sequences with less than four domains were excluded from the dataset. Next, several sequences consisting of multiple repetitions of one or two identical domains were excluded as follows. Sequences, in which 30% or more of the total number of domains consisted of one domain type, or 40% or more of the total number of domains consisted of only two domain types were removed from the dataset. This operation permitted to exclude a large number of ICPS, which facilitates the evaluation of the algorithm. However, the algorithm is as well applicable to the non-redundant dataset. Filtering reduced the number of sequences that eventually required visual inspection by a factor of 3 (Table 1). The algorithm was developed using the (1) ProDom release 2001.3 and computations were repeated with (2) ProDom release 2003.1 (from April 2004).

**Algorithm**

The algorithm itself is a variation of the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) for sequence alignment. In the default setting we assign a score of 10 for matching domains, −1 for a mismatch and a gap penalty of 5. The usage of affine gap penalties did not show a significant impact on the results and was therefore abandoned. The first modification of the algorithm is that the first row and first column of the alignment lattice are initialized with zeroes, similarly as in the Smith–Waterman algorithm (Smith and Waterman, 1981). The second modification is the way the traceback is done to allow the distinction between CPs and intermediate CPs and to increase specificity.

First, the alignment lattice is arranged into four quadrants with identical contents and corresponding to the initial strings of domains: top right (TR), top left (TL), bottom right (BR) and bottom left (BL). Each cell in one of the quadrants has a corresponding cell in the other three, and all four cells correspond to the same pair of domains from the first and the second sequence. Precisely, if $n_1$ and $n_2$ are lengths of the not duplicated domain strings, and given a cell $(i, j)$ in the TL quadrant, then each of the cells $(i + n_1, j)$,
Fig. 2. The alignment lattice and traceback. Left: example of a true CP: the alignment between two oligopeptide binding proteins from Thermatoga maritima (sp Q9X0N4) and Archaeoglobus fulgidus (sp O28507). Right: example of an intermediate CP: sp Q9EWP5 and sp Q9FB18. TR, TL, BR, BL—top right, top left, bottom right, bottom left quadrants, corresponding each to one alignment between the two sequences. Cells with dark background correspond to identical domains in both sequences. Borders of cells indicate the start and end of the tracebacks for the optimal and suboptimal paths (thick lines; a colour version of the picture can be found in Supplementary material).

(i, j + n₂) and (i + n₁, j + n₂) corresponds to the same pair of domains in the initial domain strings. A match in the lattice is defined as a cell that corresponds to identical domains. Identity is defined by identical domain ID.

Next, after running a complete Needleman–Wunsch scheme over the four quadrants, the traceback procedure tries to decide if the two strings of domains are related by a true CP, an intermediate CP or no CP at all. Because of the definition, CPs in this layout will appear as follows. In case of a true CP (i.e. without repeats of domain string) there are exactly two alignment paths: one passing through the TR, TL and BR quadrants, and the other one passing the TR, BL and BR quadrants (see Fig. 2). Moreover, each of these paths will pass through the corresponding cells in the TL and BR quadrants. Furthermore, at least one match is present in the TL, BR and either TR or BL quadrants. In the case of an intermediate CP, i.e. one which comprises one or more repeats, other suboptimal alignment paths will exist as well.

Finally, these properties of CPs are exploited in the algorithm as follows. The algorithm searches not only for the optimal alignment, but also for other alignment paths which start with a match in the BR quadrant. The first traceback path is started from a match in the BR quadrant which has the maximum alignment score. Each traceback tags the cells, such that cells which belong to an already identified path can be recognized in further passes. To qualify as a valid traceback path indicating a CP (or an iCP) the back traced path must pass through three quadrants. That means, it has at least one match in the TL, BR and two matches in either BL or TR. If this condition is not met, no CP is identified and the algorithm finishes. If the condition is met, the algorithm searches for further suboptimal paths which are not connected to any paths identified during prior tracebacks. The algorithm identifies a pair of domain strings as related by a true CP if and only if the following conditions are met: (1) exactly two paths exist, (2) each of them passes through three quadrants, (3) the paths pass through a match in the TL quadrant which corresponds to a cell in the BR quadrant with which the path started. Additionally, if further suboptimal paths exist, then an iCP is identified, since at least one of the sequences contains multiple repeats (Fig. 2b).

Exactness and heuristics in RASPODOM
Consider two strings of domains which are related by a CP, allowing some domain insertions. Assume that in each string every domain is unique. Given the large alphabet of domain IDs, this assumption is reasonable and holds most of the time. Since no mismatches need to be considered, the algorithm is exact, because it will always find the CP.

Duplicating both strings improves specificity for CPs in less than ideal cases of CPs. Consider the following example of two domain strings: XXXABCXXX (where X stands for any non-matching domain) and ABCABC (see also Supplementary material). If only the first string was duplicated, there exists a path starting in the last row of the alignment lattice which passes through both halves of the lattice. Therefore, this pair would be—wrongly—identified as a CP. However, in the RASPODOM lattice there will be no path passing through three quadrants and the pair of domain strings will thus not be identified as a CP.

Since the alphabet of ProDom domains is large (|\Sigma| \approx 2.6 \times 10^5), and the mean length of the domain strings relatively small (n \approx 6), a large fraction of the examined domain string pairs will have less than three domains in common. It is obvious from the description of the algorithm that in such a case no CP can be defined. Therefore, before the algorithm starts, a hash table is used to lookup if the two strings have any domains in common. This step has time complexity O(n). The final traceback procedure is on the order of 2O(n). Since the algorithm itself works at O(4n^2) this is also the overall time limiting step.
Rapid prediction of circular permutations

Fig. 3. Multiple alignment of different nicotinamide nucleotide transhydrogenase sequences related to the bovine protein (NTNM). ENTHL, Entamoeba histolitica; EIMTE, Eimeria tenella; CAEEL, Caenorhabditis elegans; ACEAT, Acetabularia acetabulum; NEUCR, Neospora crassa. The ancestor of the orthologs in the protozoan branch has obviously been subjected to a circular permutation. Letters are used for brevity, the corresponding ProDom IDs are shown under the alignment. For further explanations, see text. For sequence alignment, see Supplementary material.

Performance

Scanning the whole non-redundant dataset (∼2.5 × 10^7 sequences) for pairs of sequences related by a CP takes just around 20 min for the filtered dataset and less than 1 h for the non-redundant dataset on a 2.4 GHz, 2 processor Linux machine. This includes all of Swiss-Prot/TrEMBL sequences which are found in the ProDom database. On the same platform, our reimplementation of the heuristic part of the UFAU algorithm (described by Uliel et al., 1999) was estimated to take several CPU months to complete the same task.

RESULTS

We report here results for the ProDom 3.1 and 4.2 version. Input consists of a dataset containing the domain strings for all protein sequences annotated in ProDom, stringently filtered as described above. The aim of this test run is to consider all possible pairs of protein sequences, represented as domain strings, and to determine whether one of them is a CP of the other. All positive results were confirmed by visual inspection of the amino acid sequence dot plots. Several new circular permutations, previously not described in literature, were found among proteins from the ProDom database. The 93 identified CPs (Table 1) can be divided into 25 groups, comprising 104 proteins involved in CPs, and corresponding to 30 unique circularly permuted proteins. A graphical representation of those groups can be found in Supplementary material. In one case, the DNA polymerase from the hepatitis B virus, we tracked down the CP to an erroneous annotation of the sp Q69590. Thus RASPODOM identified a CP between the P1 and the β subunit.

Examples of the identified CPs

RASPODOM found novel CPs in chitinases, bacterial enzymes which are able to break down the glycosidic bonds of chitin. The CPs in chitinases identified by RASPODOM contain two important domains: a glycosyl hydrolase domain, and a fibronectin domain. The fibronectin type III domain is a tandem repeat. Three of the four circularly permuted proteins were from different Bacillus species. The fourth came from Listeria monocytogenes, also belonging to the order Bacillales. This indicates that the CP has occurred early in the evolution of this taxon.

The adhesion protein P1 is a major surface antigen in Mycoplasma pneumoniae. The genome of this bacterium contains several repetitive regions overlapping with shorter, hypothetical paralogs of the P1 (Himmelreich et al., 1996). Both sequence duplication and recombination events are thus supposed to be common for this type of proteins. RASPODOM identified a CP between the P1 and the longest of the hypothetical paralogs, MPN370. The CP rearrangement concerns two repetitive regions. The underlying mechanism of sequence duplication, along with recombination, is thought to be a major cause of the antigenic variation of Mycoplasma strains.

CPs were identified between protozoan transhydrogenases and transhydrogenases from higher eukaryotes. The transhydrogenases are a large group of enzymes, found in both prokaryotic and eukaryotic organisms. In both groups three units of this enzyme can be distinguished: I, II and III. Units I and III are responsible for binding the substrates, while II is a transmembrane domain. In prokaryota, this enzyme is split into two different genes. The corresponding subunits, α and β, comprise the domain arrangements I–IIa and Iib–III, respectively. The transmembrane domain is split up between these two genes. In eukaryota, all three domains are encoded by one gene. In mammals and other higher eukaryotes the domain arrangement is I–II–III. In parasitic protozoans, however, the domain arrangement is Iib–III–I–IIa (Fig. 3). For protozoans, it has been suggested...
Choosing an appropriate BLOSUM matrix with adjusted gap parameters resolves this problem.

RASPODOM provides a re-implementation of the sequence-based approach. On our test set, RASPODOM was 50 000 times faster than our re-implementation of the sequence-based approach.

One advantage of RASPODOM is its speed, which allows a quick pairwise comparison of hundreds of thousands of sequences. Our domain-wise approach allows a quick identification of pairs of sequences which share a significant homology, such that pairs of unrelated sequences need not be compared. On our test set, RASPODOM was 50 000 times faster than our re-implementation of the sequence-based approach.

Secondly, the RASPODOM algorithm shows a high specificity of the results. A potential problem with an amino acid sequence-based approach is presumably its failure to draw a clear border between a normal homology and a CP. Let us assume that a homology exists between the N terminus of the first and C terminus of the second sequence. Even if there is a negligible and insignificant homology between the C terminus of the first sequence and the N terminus of the second sequence, the UFAU algorithm will nonetheless identify a CP, leading to a false positive result. A possible improvement would be to calculate the score contribution of both fragments of the CP alignment and set a threshold (e.g. score contribution of the smaller fragment must be at least 10% of the larger fragment). In doing so, we managed to improve the performance of our implementation of the UFAU algorithm. It has to be noted that the automatic, heuristic part of the algorithm, described by Uliel et al. (1999), requires, in a normal procedure, (1) a further check with the exact algorithm, and (2) a manual examination of the results (Uliel et al., 2001).

The selectivity can be hampered if a CP disrupts a domain in two halves, each of the halves will not be recognized and correctly annotated by ProDom. In such a case, RASPODOM might return a false negative. However, if the ProDom domains correspond to functional entities in proteins, as assumed, then such a disruption is not very likely.

Finally, an important feature of RASPODOM is its ability to distinguish between true and intermediate CPs. The way the traceback is performed assures that sequences containing recurring pattern of domains, characteristic for iCPs, will not be identified as a complete CP. Note that this does not mean that a CP cannot contain repeats. For example, the pair of domain sequences AABBBB and BBBBAA will be—correctly—identified as a true CP, whereas ABCABCABC and CABCABC will be identified as an iCP. As we have detailed above, this property depends on doubling both sequences.

The actual reason why RASPODOM, in spite of its simplicity, outperforms existing methods, is the exploitation of the biological paradigm of domain (motif) conservation.

The main drawback of RASPODOM is its reliance on ProDom data. First, sequences without a suitable domain annotation will not be included in the evaluated dataset. This may make it necessary to use other CP detection algorithms to complement RASPODOM.
Moreover, in many cases ProDom annotations tend to be restrictive and similar domains get different annotations. A straightforward approach would be to use high scoring segment pairs (HSPs) from the BLAST algorithm. However, since ProDom is updated automatically and profile methods appear to be more selective than pairwise hits, using precompiled resources appears to be both, more selective and efficient. An illustration of such a case is the dot plot of the sequences sp Q8RLC1 and sp Q8ZV34 (Fig. 4).

In conclusion, RASPODOM is certainly the first choice for rapid screening of large datasets. Due to the low false positive rate, even for in-depth analysis RASPODOM provides a good complement to the amino acid sequence-based algorithms.

REFERENCES


