Naturally occurring circular permutations in proteins

Shai Uliel, Amit Fliess and Ron Unger

Faculty of Life Sciences, Bar-Ilan University, Ramat-Gan, 52900, Israel

1To whom correspondence should be addressed.
E-mail: ron@biocom11s.biu.ac.il

A pair of proteins is defined to be related by a circular permutation if the N-terminal region of one protein has significant sequence similarity to the C-terminal of the other and vice versa. To detect pairs of proteins that might be related by circular permutation, we implemented a procedure based on a combination of a fast screening algorithm that we had designed and manual verification of candidate pairs. The screening algorithm is a variation of a dynamic programming string matching algorithm, in which one of the sequences is doubled. This algorithm, although not guaranteed to identify all cases of circular permutation, is a good first indicator of protein pairs related by permutation events. The candidate pairs were further validated first by application of an exhaustive string matching algorithm and then by manual inspection using the dotplot visual tool. Screening the whole Swissprot database, a total of 25 independent protein pairs were identified. These cases are presented here, divided into three categories depending on the level of functional similarity of the related proteins. To validate our approach and to confirm further the small number of circularly permuted protein pairs, a systematic search for cases of circular permutation was carried out in the Pfam database of protein domains. Even with this more inclusive definition of a circular permutation, only seven additional candidates were found. None of these fitted our original definition of circular permutations. The small number of cases of circular permutation suggests that there is no mechanism of local genetic manipulation that can induce circular permutations; most examples observed seem to result from fusion of functional units.

Keywords: circular permutations/database searches/domain fusion/edit distance

Introduction

Several recent laboratory experiments have analyzed the structural and functional properties of proteins that were artificially manipulated such that an N-terminal fragment of the protein was moved to the C-terminal end of the sequence. Usually, these manipulations were performed on the DNA level by cutting the gene at a certain position along the sequence and joining together the two original termini, thereby creating a circular permutation of the original gene. A graphical view of a circular permutation of a protein sequence is shown in Figure 1.

The effects of such permutations were studied by comparing the structural and functional properties of the mutated proteins with those of the original. Some examples include studies by various groups [Goldenberg and Creighton, 1983; Ay et al., 1998; Lugner et al., 1989; Heinemann and Hahn, 1995a (review)]. In a very detailed study (Hennecke et al., 1999), a large number (71 permutations derived from a protein of size 189 residues) of possible circular permutations of the protein DsbA were studied in a functional in vivo assay. In another recent detailed study, Iwakura et al. performed a systematic circular permutation of all residues in dihydrofolate reductase and showed that only circular permutation in a small number of specific positions which are related to folding elements disrupted the correct conformation (Iwakura et al., 2000). The conclusion emerging from these studies is that in most cases, circularly permuted proteins retain their three-dimensional structure and biological function.

Since proteins subjected to artificial circular permutation usually maintain their function, it is possible that circular permutation may serve as a constructive mechanism in evolution. Indeed, there are some cases in which such permutations were suggested to have occurred naturally. Several pairs of natural proteins have been described, which appear to be related by circular permutation. Some examples include lectins (Cunningham et al., 1979; Hemperly and Cunningham, 1983), saposin (Ponting and Russell, 1995) and β-glucanase (Heinemann and Hahn, 1995b). A comprehensive review was published by Lindqvist and Schneider (Lindqvist and Schneider, 1997).

There are two possible mechanisms that can give rise to pairs of proteins related by circular permutation. In the first, a 'parent' gene may undergo a direct local genetic manipulation of its DNA sequence to form a circularly permuted mutant. It has been suggested that a possible mechanism for this might be a duplication of a gene followed by deletion of both termini leaving a permuted gene in the middle. The other alternative is that proteins that are related by circular permutation were formed by fusion of two smaller components to form a larger unit. This process could occasionally occur independently at least twice in evolution, each time in a different order.

The major distinction between laboratory experiments of circular permutations and naturally occurring examples is that in laboratory experiments the two circularly permuted forms of the protein are identical, other than the permutation. In nature, it is unlikely that a pair of proteins would be exact circular permutations of each other. A more reasonable evolutionary scenario would be one in which a gene is mutated into another protein by a permutation or fusion event; thereafter both proteins would further diverge by the standard evolutionary events of insertions, deletions and substitutions.

Thus, in the context of natural sequences we define two sequences $S_1$ and $S_2$ to be related by a circular permutation if $S_1 = xy$ and $S_2 = y'x'$, where the fragment $x$ is similar to $x'$ and the fragment $y$ is similar to $y'$ under some sequence similarity measure. Note that an effort to search for circular permutations in a systematic manner requires a precise definition of what is considered to be a circular permutation and
S. Uliel, A. Fliess and R. Unger

Fig. 1. A schematic example of a circular permutation. The original termini (left) are fused to form a continuous part of the chain and new termini are formed by cutting the polypeptide chain elsewhere (right).

Fig. 2. The sequence-based definition is expected to identify cases of ‘proper’ circular permutation (A). Cases such as (B) where the permutation spans only a fragment of the protein or (C) where the proteins are very different in length are not considered.

what is not. We suggest a stringent definition for circular permutation (see below), that enforces the requirement that the N-terminus of one protein be similar to the C-terminus of the other and vice versa and that the permuted regions will cover a significant portion of both proteins (see the scheme in Figure 2A). This sequence-based definition enabled us to design a fast screening algorithm for such cases (Uliel et al., 1999), which is used here as a first step in a systematic search for examples of circular permutations in the entire Swissprot database. Note that our definition will exclude cases where the circular permutation is within a partial region of the protein (Figure 2B). These cases might be confusing and such examples have sometimes been described in the literature as ‘circular permutations’ (see Results). However, we strongly argue that they are not circular permutations of proteins and that they should be analyzed as ‘swaps’ which would require other computational tools.

The definition also excludes cases where the two proteins are of very different length (Figure 2C). To verify that this definition does not preclude a significant number of cases, we also conducted a study in which circular permutations were identified on the level of recognized protein domains. In this comparison, two proteins are considered to be related by circular permutation if they share the same domains as defined in the Pfam database (Bateman et al., 2000), but these domains appear in a circular permuted order in the two proteins. This comparison avoids some of the restrictions of the sequence-based definition.

In this study, we used computational algorithms that enabled us to screen the entire database of known proteins to identify novel examples of circular permutations, to assess how commonly this phenomenon occurs and to gather evidence that might help to distinguish between the two possible mechanisms for the evolutionary generation of circularly permuted protein pairs.

Methods

Available tools

It was noted (Lindqvist and Schneider, 1997; Russell and Ponting, 1998) that the standard tools of sequence comparison are not suitable for the detection of circular permutations: regular global alignment algorithms [e.g. the Needleman–Wunsch algorithm (Needleman and Wunsch, 1970)] are sequential in nature and fail to align sequences which contain sequence permutations. In some cases, the relatedness between the proteins would still be detected based on one common domain alone, but their specific relationship by circular permutation will remain undetected. Furthermore, in cases where a protein has undergone a circular permutation early in its evolution and both proteins accumulated a significant number of other genetic mutations later, it is possible that their alignment will not reveal any detectable similarity. However, if the original cyclic permutation can be identified and ‘corrected’, it is conceivable that sufficient residual similarity exists, which may be identified.

Another alternative is using database search methods such as Blast (Altschul et al., 1990) to search for circular permutations. These methods identify sequences that share short common fragments and report these best local matches. In principle, the list of significant local matches can be analyzed to check for evidence of possible circular permutations. In practice, we found that it is difficult to automatically screen Blast output for circular permutations. This is mainly due to the fact that Blast [and even the gapped-Blast variant (Altschul et al., 1997)] tends to break alignments into smaller fragments, thus making an unambiguous reconstruction problematic. An additional serious problem is that sequences that contain repeats and duplications are difficult to differentiate from true examples of circular permutations. Another problem is that in cases where the sequence similarity is low, two fragments with local similarity that, if taken together, could have been
indicative of a circular permutation might be below the Blast detection level.

While standard automatic methods are not suitable for the detection of circular permutations, visual inspection tools such as dot matrix plots allow the detection of circular permutations, but in a slow manual procedure. Dot matrix plots (Maizel and Lenk, 1981) (also known as dotplot) are a simple, yet effective, sequence comparison tool utilizing manual visualization to identify relationships (such as similarity, repeats and self-complementarity) between sequences (Unger et al., 1986). The existence of pairs of diagonal lines that originate from different rows and columns and are off the main diagonal is a characteristic feature of circular permutation (see Figure 3).

Screening strategy

We have previously described the development and testing of an algorithm that allows the automated screening of protein sequence data for the existence of circularly permuted sequence pairs (Uliel et al., 1999). The algorithm is briefly reviewed here, together with a description of how it was implemented for the screening of the entire Swissprot database: pairs of proteins that passed this screening test were subjected to more careful and computationally demanding algorithms. Pairs that passed the second test were subjected to manual analysis to validate their status.

The exact algorithm

We start with an overview of the straightforward exact algorithm to identify permuted proteins, since it provides the basis for understanding the much faster approximation variant that we have designed. The global edit distance between two sequences [using the Needleman–Wunsch algorithm (Needleman and Wunsch, 1970)] measures the number of genetic operations of insertion, deletion and substitution needed to change one sequence into the other. To detect a possible circular permutation between a pair of sequences, one can check if there exists a circular permutation of one of them that minimizes their edit distance. Thus, we defined the circular edit distance to be \( \text{CED}(S_1, S_2) = \text{MIN}(\text{ED}[S_1, \text{Perm}_i(S_2)]) \), where \( \text{ED}(S_1, S_2) \) is the standard edit distance between two strings and \( \text{Perm}_i(S) \) is an exact circular permutation of \( i \) characters from the suffix of \( S \) to the prefix of \( S \) [e.g., \( \text{Perm}_3(\text{AAAGCTG}) = \text{CTGAAAG} \)] and the minimum is over all possible circular permutations.

This representation of the question lends itself to the following exact algorithm: (a) for each pair of proteins create all possible circular permutations of one protein relative to another; (b) for each permutation, calculate the regular edit distance by dynamic programming; and then (c) choose the lowest edit distance value. Furthermore, the set of all circular edit distances \( \text{ED}[S_1, \text{Perm}_i(S_2)] \) between two sequences can be used as a statistical ensemble to evaluate the significance of the value of the minimum, \( \text{CED}(S_1, S_2) \). Thus it provides a way to estimate whether a certain value of \( \text{CED} \) is statistically significant.

This algorithm requires a time complexity of \( N^3 \), where \( N \) is the length of the proteins (\( N \) iterations of the standard \( N^2 \) edit distance algorithm). Recently, a theoretical algorithm was suggested (Landau et al., 1998) that solves the problem in asymptotic quadratic time. However, the pre-processing required for each pair of compared proteins and the large constant factors involved in running this algorithm make it impractical for comparison of a large number of pairs of proteins.

A time requirement of \( N^3 \) for a comparison between a pair of proteins is computationally expensive. It becomes prohibitive for a full pairwise comparison of the entire database. In our implementation, a single \( N^2 \) comparison of a pair of sequences each 300 amino acids in length takes about 0.005 s (using a Silicon Graphics R10000 processor). An exhaustive search of all possible cyclic permutations for a single protein pair should therefore require about 1.5 s (0.005×300). Thus, a complete survey of all pairs from the current protein database, which includes about 80 000 proteins, would take many CPU years.

A fast screening algorithm

As an alternative, we designed an efficient algorithm that, although not guaranteed to find the optimal permutation, does perform very well in practice. The details of the algorithm and a demonstration of its effectiveness in a designed test case were described previously (Uliel et al., 1999). Here, we review the main premise of the algorithm.

The fast algorithm is a variation of the edit distance algorithm in which one sequence is compared with a duplication of the other sequence. Namely, for sequences \( S_1 \) and \( S_2 \), \( S_1 \) is compared with \( S_2 S_1 \) (see Figure 4). The principle is that in this way, all consecutive permutations, each consisting of a C-terminal fragment of \( S_1 \) followed by an N-terminal fragment of \( S_2 \), are simultaneously compared with \( S_1 \).

To implement this algorithm, two modifications to the classical edit distance algorithm were introduced (see Figure 4 for a schematic example): The first is initialization of the first row of the edit distance with zeros, instead of the usual initialization as 0, 1, 2, . . . , \( N \). The standard sequential initialization for the first column is maintained. The second modification is to look for the value of the best alignment among all of the entries in the bottom row of the matrix. Actually, each entry on the right side of the last row, i.e. of columns \( N + 1, N + 2, \ldots, N + j, \ldots, 2N \), can be seen as an approximation of the exact edit distance for a circular permutation of \( j \) characters, respectively, and therefore the last row represents an approximation for the ensemble of all possible circular permutations between the two sequences. Thus, a significant minimum in the series of values in the last row suggests that an alignment with a specific computational circular permutation might be due to a true biological circular permutation event. Note that this algorithm is only an approximation, since alignments that do not reflect a valid circular permutation can be detected, for example aligning sequence \( S_1 \) with a duplication of fragments from \( S_2 \) (paying a penalty for the necessary deletions). However, the optimal edit distance under a circular permutation is always a possible solution of the algorithm. Thus, every other solution that the algorithm produces must have lower value than the ‘correct’ value. In other words, the exact value for the edit distance for each permutation is an upper limit for the approximation (see Figure 5 for comparison of BGLS_BUTFI with BGLB_CLOTM). This algorithm was shown to perform well in several artificial tests (Uliel et al., 1999).

The time and space complexity of this algorithm is clearly \( 2N^2 \), since the size of the dynamic programming matrix has been doubled in one dimension.

Passing this approximate algorithm is not sufficient to establish a case of circular permutation. Even the exact algorithm (i.e. calculating the edit distance for each possible permutation) does not prove that a pair of proteins is related.
Fig. 3. A dotplot view of several circularly permuted protein pairs. In the right matrix the original sequences are compared. The location of the two diagonal lines indicates that the N-terminus of the first sequence is highly similar to the C-terminus of the second and that the C-terminus of the first is highly similar to the N-terminus of the second. After one of the sequences is circularly permuted relative to the other (left matrix) the two lines merge, suggesting that the two proteins are related by circular permutation. Three examples are shown: (A) β-glucosidase, (B) lipopolysaccharide and (C) endoglucanase.

by circular permutation. Two of the obvious problems are that (1) it is not always clear how to define 'a significant minimum' in the series of edit distances produced by the exact algorithm and (2) the internal structure of a protein (e.g. repeats or low-complexity regions) may blur the significance of the results. Thus, as is common in biological applications of string matching algorithms, the algorithmic part can only identify potential candidates that need to be further examined.
The screening algorithm was used as the ‘main engine’ of a large-scale survey of circular permutations in the protein database. We started with all proteins from SwissPROT version 34.0. Since, by our definition, proteins of very different length are not candidates for circular permutation, only pairs of proteins of comparable length were tested. To this end, proteins were divided into 35 length groups: every 10 residues from length 50 to 100, then 10–119, etc., every 20 residues from length 300 to 400, every 25 from length 400 to 500 and every 50 from length 500 to 1000. Proteins shorter than 100 or longer than 1000 residues were not considered. On average, each length group contained about 1200 proteins.

Comparisons were made between all proteins that belong to length groups that differ by not more than two lengths, i.e. each protein was compared with all proteins in its own length group and in each of the two longer and shorter groups. For example, a protein of length 295 residues was compared with all proteins from size 270 to 340. The comparisons were performed using the approximate algorithm described above with a unit edit distance penalty of one for insertion, deletion and substitution.

For each comparison, we screened the last row for a possible minimum. A possible minimum was defined if the value in a certain cell of the last row of the edit distance matrix was smaller than 95% of the average value of all cells in the last row. A permissive threshold of only 5% below the average was used, since all of candidate pairs were further screened by the exact algorithm. About 300 000 pairs of proteins were identified for advance to the exact algorithm.

With the exact algorithm, a threshold of 90% of the lowest value achieved among all edit distance comparisons (i.e. one comparison for each possible circular permutation) relative to the average value was required. This stage left about 8000 pairs of proteins for further examination. Note that the exact algorithm also identified the position of the putative circular permutation (i.e. the location of the minimum).

As mentioned above, a minimum in the exact algorithm does not necessarily prove that an evolutionary event of circular permutation indeed took place. Thus, for each pair of candidate proteins additional tests were performed.

The first test involved visual inspection of the dotplot matrices, before and after the circular permutation was performed. Short diagonals that are off the main diagonal before the permutation and merge together along the main diagonal after the permutation provide a strong indication of a circular permutation (see Figure 3). In particular, a dotplot is a good tool to eliminate cases where the original proteins include internal repeats that can result in a false positive in the automated analysis. Next, the results of the standard sequence alignment procedure performed between the pairs of proteins before and after the permutation were compared. This was done using the GAP procedure of the GCG package with standard parameters. Naturally, the option of penalizing end gaps similarly to regular gaps was used to avoid the possibility of ignoring the terminal regions, which are important for detection of circular permutations.

The GAP procedure can calculate the Z-score of the significance of the alignment. This is done by comparing the quality (in terms of edit distance score) of the alignment with the quality of a large number of random alignments produced by a random shuffle of one of the sequences. Thus, we could compare the Z-score of the original alignment of the two proteins with the Z-score achieved after one of the proteins was circularly permuted. A large improvement in the Z-score of the alignment after circular permutation versus the Z-score of the alignment without circular permutation suggests that it is more likely that the two proteins are related by circular permutation than by regular alignment. We used a stringent cut-off of Z-score improvement after the circular permutation, by at least five standard deviation (SD) units. We validated that such an improvement in the Z-score is not observed when a large number of (non-permutated) sequences were subject to random circular permutations.

Pfam screening

As an independent test of the sensitivity of our screening procedure, we also employed a separate scheme based on a different definition of circularly permuted proteins. Thus, if
our algorithm did not detect a large number of circularly permutated protein pairs, we would expect these to be found by the second, independent screen. For this procedure, we used a totally different approach to detect circular permutations by finding pairs of proteins that contain the same recognized domains, where these domains appear in the two proteins in a circularly permuted order relative to each other. We used the Pfam database, version 6.0 (Bateman et al., 2000), which is based on multiple alignments of protein domains or conserved protein regions. Pfam supplies (in a file called SWISSPFAM) the domain structure of 71 415 proteins from the Swissprot database, according to Pfam definitions. We define two proteins to have a domain circular permutation if they contain the same domains, but these domains appear in one protein in an order that is a circular permutation of their order in the other protein. A direct comparison of each pair of Swissprot proteins to detect whether they contain circular permutations of domains will require \( N^2 \) operations.

To speed up the process, we designed the following procedure: for each protein in Swissprot, we stored in one line a list of its domains sorted to a canonical lexicographical order by their names. We also stored for each protein its distance from this order, i.e. the weighted number of swap operations needed to obtain to the canonical order from the original order. We then sorted these lines to find all proteins that share the same domains, which will appear as consecutive lines. From these lines we selected only pairs that have a different distance to the canonical order, indicating that they have a different original order. These protein pairs were then manually evaluated to determine whether their domains are ordered as a circular permutation of each other. Since this procedure is based on sorting, it has a run time of \( \log N \), where \( N \) is the number of proteins, rather than the naive \( N^2 \) comparisons.

Note that this domain-based definition of circular permutation avoids some of the restrictions imposed by our sequence-based definition. For example, the proteins do not have to be of similar sizes for a circular permutation to be considered. It suffices, for example, that each protein contains, in a different order, the same two recognized domains, but these domains do not have to cover the entire sequences. The domain-based definition excludes cases in which the permutation is within a partial region of the proteins (Figure 2B), which we do not consider as a circular permutation. Naturally, this definition depends on domains that are pre-defined in the Pfam database and thus it is less general than the direct sequence-based definition. Thus, in a sense, these two alternative definitions can be considered complementary.

Results

In this study, we used a fast screening algorithm followed by a full global alignment procedure to identify candidate pairs of proteins related by circular permutations. The 25 protein pairs that passed all the screening procedures are shown in Tables I–III. Table I shows the examples in which the proteins that are involved in the circular permutation have a similar function. Table II shows proteins that have related functions and Table III shows proteins that are not related by function or whose function is not known. For some of the examples, the improvement in Z-score is very large (\( >15 \) SD), which confirms them as clear cases of circular permutations. The pairs at the very bottom of our list, especially those that share no known functional connection, are less definite and at this stage should be considered to be unproven candidates.

We describe here, in some detail, six protein pairs identified by our algorithm, three previously known cases and three cases which were not recognized until now as examples of circular permutation events. Three well-known cases of circular permutations are \( \beta \)-glucosidase, lectins and endoglucanase. The results of the approximate algorithm and the exact algorithms that detected the circular permutation in \( \beta \)-glucosidase (comparing BGLS_BUTFI with BGLB_CLOTM) are shown in Figure 5. The existence of a circular permutation is very clear on both curves. The dotplot result for this case is shown in Figure 3A and a presentation of the alignment between the two proteins that clearly demonstrates the circular permutation is shown in Figure 6A.

The alignment in the case of lectins (LEC_BOWMI versus LECA_DIOGR) is shown in Figure 6B. The case of the circular permutation in lectins is very interesting, because the circular permutation is not on the genetic level, rather it is done as a post-translational modification (Carrington et al., 1985) which involves transposition and ligation within the initial polypeptide. The reason for this phenomenon and the extent of similar post-translational modifications in other proteins are still to be elucidated.

Another interesting case is the circular permutation between GUN2_THEFU and GUN2_THEF, two endoglucanase proteins. Each of these proteins contains two known motifs: a glycosyl hydrolase catalytic domain and a cellulose binding domain, which utilizes two tryptophan residues to facilitate cellulose binding. The relative order of these two domains is different in the two bacterial species, as can be seen in the dotplot presented in Figure 3C. This seems to suggest that the binding activity and the catalytic activity are spatially distinct and that their precise orientation is not crucial for the activity of the protein. These proteins are similar to the \( \beta \)-glucanase proteins which appears in the list of circular permutations compiled by Lindqvist and Schneider (Lindqvist and Schneider, 1997).

Our data also identify several examples that were not previously noted: one example is a circular permutation between PABS_STRGR (\( p \)-aminobenzoate synthase) from Streptomyces griseus and TRPE_RHIME (anthranilate synthase) from Rhizobium meliloti (Figure 6C). These two proteins are involved in related enzymatic activity. PABS_STRGR catalyzes the synthesis of \( p \)-aminobenzoate from chorismate and glutamine and TRPE_RHIME catalyzes the synthesis of anthranilate (\( o \)-aminobenzoate) from the same starting materials. The activity of the N-terminal domain of PABS_STRGR involves the removal of the ammonia group from a glutamine molecule and its subsequent transfer to a specific substrate. The chorismate binding is performed by the C-terminal domain. In TRPE_RHIME, these domains appear in the reverse order. It is interesting that in other organisms (e.g. Methanococcus jannaschii and Thermus aquaticus) these two domains are not part of the same gene. Rather, they are translated as two separate polypeptides which associate in vivo as subunits of a larger complex. Surprisingly, although the two domains are on different genes in Thermotoga maritima, the coding regions of these two genes overlap by more than 100 nucleotides.

An additional example is the circular permutation between two H-1 histone proteins, H1B_PLADU from the worm Platynereis dumerilii and the bovine H1L_BOVIN. The linker histone H-1 links nucleosomes into a higher order chromatin structure. A 3D NMR structure for the globular domain is
Naturally occurring circular permutations in proteins

### Table I. A list of circularly permuted protein pairs, identified in this study, with identical or very similar function with the names, lengths and function of the proteins and the predicted relative circular shift, and also the Z-scores for aligning the two sequences before (B-Z-score) and after (A-Z-score) the circular shift.

<table>
<thead>
<tr>
<th>Protein 1</th>
<th>Length 1</th>
<th>Protein 2</th>
<th>Length 2</th>
<th>Shift</th>
<th>B-Z-score</th>
<th>A-Z-score</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGLS_BUTFI</td>
<td>830</td>
<td>BGLB_CLOTM</td>
<td>754</td>
<td>476</td>
<td>−0.2</td>
<td>38.4</td>
<td>β-Glucosidases</td>
</tr>
<tr>
<td>LEC_BOWMI</td>
<td>240</td>
<td>LECA_DIOGR</td>
<td>237</td>
<td>115</td>
<td>2.4</td>
<td>39.8</td>
<td>Lectins</td>
</tr>
<tr>
<td>YHES_ECOLI</td>
<td>637</td>
<td>YEML_YEAST</td>
<td>610</td>
<td>524</td>
<td>12.7</td>
<td>36.5</td>
<td>ABC transporters</td>
</tr>
<tr>
<td>CIN_DROME</td>
<td>601</td>
<td>CNXI_ARATH</td>
<td>670</td>
<td>233</td>
<td>1.1</td>
<td>29.2</td>
<td>Molybdenum biosynthesis</td>
</tr>
<tr>
<td>RS2_HALMA</td>
<td>266</td>
<td>NABA_YEAST</td>
<td>251</td>
<td>55</td>
<td>−0.5</td>
<td>19.7</td>
<td>Ribosomal proteins</td>
</tr>
<tr>
<td>PT1A_ECOLI</td>
<td>711</td>
<td>PTF1_RHOCA</td>
<td>827</td>
<td>157</td>
<td>−0.2</td>
<td>33.7</td>
<td>PEP-phosphotransferases</td>
</tr>
<tr>
<td>K1CM_HUMAN</td>
<td>458</td>
<td>K2C1_XENLA</td>
<td>425</td>
<td>100</td>
<td>−0.7</td>
<td>19.0</td>
<td>Keratin cytoskeletons</td>
</tr>
<tr>
<td>ERR1_HUMAN</td>
<td>521</td>
<td>HNF4_HUMAN</td>
<td>465</td>
<td>98</td>
<td>−0.5</td>
<td>18.2</td>
<td>Hormone receptors</td>
</tr>
<tr>
<td>NOD1_BRAJA</td>
<td>306</td>
<td>CHVD_AGRTU</td>
<td>286</td>
<td>72</td>
<td>−1.7</td>
<td>13.5</td>
<td>ATP-binding proteins</td>
</tr>
<tr>
<td>GUN2_THEFU</td>
<td>441</td>
<td>GUNA_CELFI</td>
<td>449</td>
<td>309</td>
<td>2.1</td>
<td>13.1</td>
<td>β-Glucanase</td>
</tr>
<tr>
<td>H1B_PLADU</td>
<td>119</td>
<td>H11_BOVIN</td>
<td>104</td>
<td>75</td>
<td>0.3</td>
<td>9.6</td>
<td>Histones</td>
</tr>
<tr>
<td>HK25_XENLA</td>
<td>299</td>
<td>HXB5_BRARE</td>
<td>275</td>
<td>180</td>
<td>−0.7</td>
<td>7.3</td>
<td>Homeobox proteins</td>
</tr>
<tr>
<td>CU08_LOCMI</td>
<td>148</td>
<td>CU19_ARADI</td>
<td>120</td>
<td>33</td>
<td>−0.3</td>
<td>6.0</td>
<td>Cuticle proteins</td>
</tr>
</tbody>
</table>

### Table II. A list of circularly permuted protein pairs with related, but not identical function.

<table>
<thead>
<tr>
<th>Protein 1</th>
<th>Length 1</th>
<th>Protein 2</th>
<th>Length 2</th>
<th>Shift</th>
<th>B-Z-score</th>
<th>A-Z-score</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRPE_RHIME</td>
<td>729</td>
<td>PABS_STRGR</td>
<td>723</td>
<td>524</td>
<td>0.0</td>
<td>25.5</td>
<td>Aminobenzotate synthesis</td>
</tr>
<tr>
<td>RELB_MOUSE</td>
<td>558</td>
<td>REL_MELGA</td>
<td>530</td>
<td>88</td>
<td>15.5</td>
<td>33.2</td>
<td>Transcription factors</td>
</tr>
<tr>
<td>LPSZ_RHIME</td>
<td>440</td>
<td>LIPA_NEIME</td>
<td>400</td>
<td>275</td>
<td>1.6</td>
<td>17.0</td>
<td>Polysaccharide processing</td>
</tr>
<tr>
<td>ATF3_HUMAN</td>
<td>181</td>
<td>YAL3_SCHPO</td>
<td>171</td>
<td>63</td>
<td>−2.8</td>
<td>5.3</td>
<td>Transcription factors</td>
</tr>
<tr>
<td>MODC_ECOLI</td>
<td>352</td>
<td>PHRA_ECOLI</td>
<td>340</td>
<td>216</td>
<td>1.0</td>
<td>7.4</td>
<td>Molybdenum transporters</td>
</tr>
<tr>
<td>ATF4_MOUSE</td>
<td>349</td>
<td>FOS_AVINK</td>
<td>322</td>
<td>161</td>
<td>0.8</td>
<td>6.4</td>
<td>Basic leucine zipper</td>
</tr>
</tbody>
</table>

### Table III. A list of circularly permuted protein pairs with different function or cases in which the function of one of the sequences is unknown.

<table>
<thead>
<tr>
<th>Protein 1</th>
<th>Length 1</th>
<th>Protein 2</th>
<th>Length 2</th>
<th>Shift</th>
<th>B-Z-score</th>
<th>A-Z-score</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>VE4_BPV4</td>
<td>107</td>
<td>APA2_RAT</td>
<td>102</td>
<td>46</td>
<td>−0.2</td>
<td>6.5</td>
<td>1. Hypothetical protein</td>
</tr>
<tr>
<td>VM12_YEAST</td>
<td>218</td>
<td>YIM7_YEAST</td>
<td>206</td>
<td>122</td>
<td>−1.9</td>
<td>4.4</td>
<td>2. May affect HDL metabolism</td>
</tr>
<tr>
<td>HBB2_PANLE</td>
<td>146</td>
<td>RL7_THEMA</td>
<td>128</td>
<td>72</td>
<td>−1.0</td>
<td>4.6</td>
<td>1. Hemoglobin</td>
</tr>
<tr>
<td>RAN_PLAFA</td>
<td>214</td>
<td>GT28_SCHMA</td>
<td>211</td>
<td>176</td>
<td>−0.9</td>
<td>4.4</td>
<td>2. May affect HDL metabolism</td>
</tr>
</tbody>
</table>

available (1GHC in PDB) (Cerf et al., 1994) and was shown to be similar to other DNA binding domains. In H1B_PLADU, the N-terminal domain (residues 8–75) contains the globular domain (called linker-histone in the Pfam classification) and the C-terminal (residues 76–119) is lysine-rich. In H11_BOVIN, the N-terminus contains the lysine-rich domain, followed by the globular linker-histone domain. The organization found in the bovine protein is common in almost all organisms, but forms similar to H1B_PLADU are found in various sea urchins. It seems that the lysine-rich domain, which provides the general DNA binding function, is not required to be in a specific location in the protein. The results of this alignment are shown in Figure 6D.

Another case is the circular permutation between LFSZ_RHIME and LIPA_NEIME, two proteins that are involved in polysaccharide processing. LFSZ_RHIME is a cytoplasmic protein which is involved in the invasion of nitrogen fixation nodules and may be involved in the biosynthesis of lipopolysaccharides. LIPA_NEIME, which is probably an inner membrane protein, is involved in the phospholipid modification of the capsular polysaccharide, which is a requirement for its translocation to the cell surface. The dotplot comparing these two proteins, shown in Figure 3B, strongly suggests a circular permutation. Although the role of the two permuted segments is not clear, their ability to interchange may suggest again a two-domain structure, a binding domain and a catalytic domain.

A main goal of this study was to prepare a conclusive list of all pairs of known proteins that might be related by circular permutation. Hence we needed to determine whether our algorithm was sufficiently sensitive to detect all permuted proteins. To this end, we compared our systematic results with the compilation of such examples that have been described in the literature. The comprehensive review by Lindqvist and Schneider (Lindqvist and Schneider, 1997) was used as the source of these examples.

(a) The case of lectins was found by our method (Table I, LEC_BOWMI versus LECA_DIOGR). Note that this is the special case in which the circular permutation occurs as a post-translational modification on the protein level.

(b) The case of bacterial β-glucanase was identified (Table I, GUN2_THEFU versus GUNA_CELFI).

(c) The case of swaposin: this case does not fit our definition of circular permutation. First, the two proteins involved...
are of very different sizes (SAP_BOVIN is of length 80 amino acids and ASPR_CUCPE is of size 513 amino acids!); hence our procedure did not compare them. Furthermore, this is not a case of a full circular permutation where the N- and C-terminal regions are swapped. As discussed by Lindqvist and Schneider, in this protein pair the permutation occurs within a domain of the larger protein (Lindqvist and Schneider, 1997).

(d) The case of α-amylase and α-1,3-glucan-synthesizing glucosyltransferase: again because of the large size discrepancy between the two proteins (AMY2_ECOLI of size 495 residues and GTFD_Strmu is of size 1462), this case was not considered. Here, too, the circular permutation is within a domain of the longer protein.

(e) β-Glucosidase: this clear-cut case of circular permutation was detected by our procedure (Table I, BGLS_BUTFI versus BGLB_CLOTM)

(f) Transaldolase: a strong case for circular permutation between transaldolase B and aldolase has been made (Jia et al., 1996). However, their argument was based on the superposition of a small number of key residues based on structural alignment. As indicated by Lindqvist and Schneider (Lindqvist and Schneider, 1997) and shown by our dotplot (Figure 7A), the overall sequence similarity is rather weak and indeed our procedure did not detect a circular permutation in this case.

(g) Surface layer homology domain from bacterial cell wall-associated proteins: this case did not pass our procedure because these SLH domains repeat several times within each protein and in addition these domains are repetitive (Lupas et al., 1994). A dotplot (Figure 7B) demonstrates that there is no significant sequence similarities between the proteins and explains why our procedure did not identify this protein pair.

(h) C2 domain: this case also did not pass our criteria, (e.g. in comparing KPC2_DROME with KPCL_MOUSE) because the permutation here is within the N-terminal region and cannot be considered as a circular permutation of the entire protein.

Pfam screening

We screened Swissprot for all pairs of proteins that have a circular permutation of domains as defined by the Pfam database (Bateman et al., 2000). We compared all pairs of proteins that contain two to five domains. We found circular permutations mainly for proteins that include two domains.

The only exception is the case of the β-glucosidases proteins (BGLS_BUTFI and BGLB_CLOTM) that by the Pfam classification contain three domains. We did not find any other case of proteins that have three or more domains that are related by circular permutation. Ten cases of proteins that contain two domains in a different order were found. Of these 10 examples, three cases were found by our sequence-based procedure (GUN2_THEFU and GUNA_CELFI; CIN_DROME and CNX1_ARATH; HK25_XENLA and HXB5_BRARE); seven are novel cases which are shown in Table IV. Inspection of these cases revealed that they were not detected by our sequence-based algorithm owing to a large difference in protein sizes (the first four cases in Table IV) or because the permuted domains covered only relatively small segments of the protein lengths and thus the improvement in Z-score was not significant enough (the last three cases in Table IV).

Hence our sequence-based method was able to find significantly more cases than a method that is based on pre-defined domains. In addition, even if a more inclusive definition of circular permutation is used, mainly with regard to size differences, only a handful of additional protein pairs can be identified. This strengthens our conclusion that the number of cases of circular permutations in proteins is rather small.

Discussion

The algorithm that we employed was able to detect the previously described examples that represented protein pairs related by true circular permutation. All the suggested reported cases are actually other forms of mutations that cannot be considered as circular permutations of proteins. In addition, several novel examples are described. Hence it is likely that the list of pairs that we present here contains almost all the recognizable instances of circular permutation in the natural proteins that have been sequenced to date. However, we cannot rule out the possibility that additional cases of circular permutation have occurred during evolution but, owing to a large amount of evolutionary divergence, they are below the detection level of methods based on sequence similarity.

The list presented here adds several examples of possible circular permutation to those which were previously identified. About half of the instances are of longer proteins (12 of the 25 cases are longer than 300 residues). This preference for
 Naturally occurring circular permutations in proteins

Fig. 7. Dotplots comparing (A) transaldolase proteins and (B) surface layer homology domains. Although these pairs are both mentioned in the literature (Lindqvist and Schneider, 1997) as cases of possible circular permutations, the plots clearly show that no circular permutation exists on the sequence level.

Table IV. The seven cases of possible circular permutation identified by looking for pairs of proteins whose domains (according to the Pfam classification) are related by a circular permutation.

<table>
<thead>
<tr>
<th>Protein 1</th>
<th>Length 1</th>
<th>Protein 2</th>
<th>Length 2</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEE1_SCHPO</td>
<td>877</td>
<td>KG3H_DICDI</td>
<td>468</td>
<td>Protein kinases</td>
</tr>
<tr>
<td>EPA8_RAT</td>
<td>372</td>
<td>BYR2_SCHPO</td>
<td>659</td>
<td>Protein kinases</td>
</tr>
<tr>
<td>CLK2_MOUSE</td>
<td>499</td>
<td>CDK7_DICDI</td>
<td>362</td>
<td>Protein kinases</td>
</tr>
<tr>
<td>SUOX_CHICK</td>
<td>460</td>
<td>NIA_CHLVU</td>
<td>318</td>
<td>Nitrate reductase</td>
</tr>
<tr>
<td>PTPX_CANAL</td>
<td>597</td>
<td>VHP1_CAEEL</td>
<td>619</td>
<td>Protein-tyrosine phosphatase</td>
</tr>
<tr>
<td>ESC1_SCHPO</td>
<td>413</td>
<td>YMH7_CAEEL</td>
<td>380</td>
<td>Transcription factor</td>
</tr>
<tr>
<td>XYLZ_PSEPU</td>
<td>336</td>
<td>PHT2_PSEPU</td>
<td>324</td>
<td>Iron–sulfur oxidoreductase</td>
</tr>
</tbody>
</table>

*The first four examples were not considered by our sequence-based definition owing to a large difference in the size of the two proteins. In the last three cases, the permuted domains cover relatively small fragments of their corresponding sequences; thus the improvement in the Z-score was below our cut-off.

larger proteins suggests that independent fusion events of functional units in different orders are the driving mechanism, rather than direct circular permutation of a ‘parent’ gene to form a mutant. This and the small overall number of cases that we could find suggests that there is no direct genetic mechanism by which circular mutation can frequently occur.

The overall number of circularly permuted pairs is small, based on our analysis of protein sequences as appear in the Swissprot database. It must be noted that most of the sequence information in Swissprot is based on cDNA data and not on direct protein sequences. Thus, our procedure mainly compared putative sequences as translated from mRNA sequences and not actual protein sequences. The example of lectins shows that it is possible to generate circular permutations directly on the protein level by cleavage and ligation. Clearly, the frequency of circular permutations on the protein level, as well as other post-translational editing events, can be explored only when proteomics projects will provide sufficient direct data of protein sequences.

In the first part of our study we used a strict sequence-based definition of circular permutations. To confirm our results, we added a domain-based definition. Although we believe that these definitions taken together are inclusive, we must note that they do not cover other types of permutations that could occur, such as circular permutations within domains that do not span the whole protein (e.g. Figure 2B and C). These and other types of domain shuffling (see, for example, Hopfner et al., 1998) which might be more common in protein evolution require different tools for analysis and are part of a parallel study which is currently under way.

Acknowledgement
R.U. is partially supported by the Israel Science Foundation.

References


Received August 15, 2000; revised March 7, 2001; accepted May 11, 2001