Domain Tree-Based Analysis of Protein Architecture Evolution

Kristoffer Forslund,[1] Anna Henricson,†[1] Volker Hollich,† and Erik L. L. Sonnhammer*†*

*Stockholm Bioinformatics Centre, Albanova, Stockholm University, Stockholm, Sweden; and †Department of Cell and Molecular Biology, Karolinska Institutet, Stockholm, Sweden

Understanding the dynamics behind domain architecture evolution is of great importance to unravel the functions of proteins. Complex architectures have been created throughout evolution by rearrangement and duplication events. An interesting question is how many times a particular architecture has been created, a form of convergent evolution or domain architecture reinvention. Previous studies have approached this issue by comparing architectures found in different species. We wanted to achieve a finer-grained analysis by reconstructing protein architectures on complete domain trees. The prevalence of domain architecture reinvention in 96 genomes was investigated with a novel domain tree–based method that uses maximum parsimony for inferring ancestral protein architectures. Domain architectures were taken from Pfam. To ensure robustness, we applied the method to bootstrap trees and only considered results with strong statistical support. We detected multiple origins for 12.4% of the scored architectures. In a much smaller data set, the subset of completely domain-assigned proteins, the figure was 5.6%. These results indicate that domain architecture reinvention is a much more common phenomenon than previously thought. We also determined which domains are most frequent in multiply created architectures and assessed whether specific functions could be attributed to them. However, no strong functional bias was found in architectures with multiple origins.

Introduction

Protein domains constitute the evolutionary units of proteins (Murzin et al. 1995). A domain can fold independently (Jaenicke 1987) and may combine with other domains on the same protein chain to form multidomain proteins (Rossmann et al. 1974). In eukaryotes, a majority of proteins have multiple domains, whereas prokaryotes have fewer multidomain proteins (Apic et al. 2001; Ekman et al. 2005; Wang and Caetano-Anollés 2006). The domain architecture of a protein is defined by the particular order of domains on the protein sequence. Domain architectures may arise by way of domain rearrangement or duplication, inserting or deleting domains. Most domain architectures appear to have originated a single time only, and if functional, they will be copied and spread across many species (Doolittle 1995). As a consequence, the presence of identical domain architectures in different species normally indicates a common origin. Nevertheless, some protein architectures have arisen multiple times independently due to functional necessity or random chance. Such cases can give insights into protein function and evolution.

Several previous studies have examined various aspects of domain architecture evolution, such as gene fusion and fission events and circular permutations (Ekman et al. 2005; Kummerfeld and Teichmann 2005; Weiner et al. 2006; Weiner and Bornberg-Bauer 2006; Fong et al. 2007). In contrast, the prevalence of multiple independent domain architecture invention events has hardly been studied at all. In one study, Gough (2005) searched for such convergent evolution events among domain architectures from 62 genomes in the SUPERFAMILY database. Only 1.9% (59/3041) of the analyzed architectures were found to be likely candidates for convergent domain architecture evolution. Possible reasons for this low number are that the data set was heavily biased toward prokaryotes and that a species tree was used leading to that only events between the chosen species were recorded. Compared with this work, we use a completely different approach based on phylogenetic trees and also include more species in the data set, substantially expanding the proportion of eukaryotic genomes.

In this paper, we present a novel algorithm based on domain trees to investigate evolution of protein architectures and address the question of multiple independent domain architecture creation in a more comprehensive way than previously. Our approach infers ancestral architectures using the maximum parsimony criterion separately for each domain by processing the full phylogenetic tree of the domain family. A maximum parsimony approach was recently described to analyze domain architecture evolution (Fong et al. 2007), but their method only operated on a species tree and was mainly used to study fusion and fission events.

The main technical novelty in our method is that it does not look for events between nodes in a species trees. A drawback with using a species tree is that the true species tree is often unknown. Another source of error from using a species tree are horizontal gene transfer (HGT) events, which easily appear as independent creation events. Because our domain tree-based approach uses sequence similarity and ignores the species, it removes any potential bias stemming from HGT. A major benefit of using the domain tree is that multiple origins of an architecture can be detected within one species, which is not possible when operating on a species tree. Also, the evolutionary pattern of an architecture can be further corroborated by combining the results from the phylogenetic trees for the individual domains. We are aware that a general drawback of using phylogenetic reconstruction is the inherent uncertainty in the reconstructed tree. Therefore, to ascertain robustness of the results, we use a bootstrapping approach to determine whether the conclusions for a given architecture are reliable.

The novel domain tree–based method was applied to domain architectures derived from Pfam, but only to proteins that passed a quality control to ensure accurate annotations and to avoid multiple spliceforms. With this method, independent creation events of the same domain architecture become detectable in 12.4% of all scored architectures.

[1] These authors contributed equally to this work.

Key words: protein, domain, architecture, evolution.

E-mail: erik.sonnhammer@sbc.su.se.

Mol. Biol. Evol. 25(2):254–264. 2008
doi:10.1093/molbev/msm254
Advance Access publication November 19, 2007

In a much smaller data set of only completely domain-assigned proteins, the figure was 5.6%. This indicates that domain evolution is a highly dynamic process. We further analyzed the content and character of the most frequently reinvented architectures. However, we could find no strong functional bias for multiple independent evolutionary events of an architecture. Knowledge of which architectures are readily reinvented, and which domains are most versatile to this end, provides us with a glimpse into nature's rules of functional domain rearrangement.

Materials and Methods

We selected 96 species (see supplementary table S1, Supplementary Material online) for the analysis. From the curated part of the Pfam database (Pfam-A) release 21 (Finn et al. 2006), we extracted the full multiple alignments of all protein domain families found in the selected species (see fig. 1). The alignments of domain families of type repeat or motif were excluded (ca. 2.5% of Pfam) because it is much harder to estimate their phylogeny. From the selected multiple alignments, we excluded sequences with discontinuous domains, that is, a domain that is interrupted by the insertion of another domain; such discontinuous domains are also rare (<1% of Pfam). Also, for nonfungi eukaryotes, only the longest spliceform in terms of number of domains was included. If several spliceforms had the same number of domains, the longest with regards to number of residues was chosen. Throughout this study, protein architectures are represented as strings of domains (N- to C-terminal) separated by asterisks.

In Pfam, hidden Markov models (HMMs) are constructed for each domain family alignment. An HMM match state models the distribution of amino acids in the corresponding column in the alignment. Some of the resulting alignments contain very divergent sequences, which means that if bootstrapping is performed, there is a risk that the resampled pseudoalignment will contain sequence pairs that have no positions in common, leading to that distance methods such as Neighbor-Joining are unable to handle them. In order to avoid this, we only retained sequences that share at least 50% of the match states with the HMM (thereby excluding sequences that are highly divergent). Because some families have only a few match states overall, the 50% cutoff sometimes still did not ensure that each pair of sequences in the pseudoreplicate alignments had defined distances to each other. Hence, we also only retained sequences that shared at least 10 of the HMM match states. After these filtering steps, alignments containing 3 sequences or more were subjected to tree reconstruction using Neighbor-Joining with the Scoredist distance estimator as implemented in Belvu (Sonnhammer and Hollich 2005). Each tree was taken as the basis for ancestral architecture inference.

The ancestral architecture inference algorithm is based on the parsimony criterion and runs in 2 passes (see fig. 2*A*). In the first pass, the tree is traversed from the leaves to the root. The extant protein architectures at the leaves are used to initialize the tree. At each inner node, all possible ancestral architectures and their costs are enumerated. In order to avoid prior bias, an insertion or a deletion of a domain is assigned an equal cost. The set of potential ancestral architectures is determined by finding the shared subarchitectures between child nodes and enumerating all their possible permutations, with the constraint that the domain from which the tree was built has to be present (see fig. 2*B*). Following the maximum parsimony principle, the least expensive architecture is selected at each node in the tree. However, at a particular node, several ancestral architectures may share the same cost, in which case they are all kept. In the second pass, the tree is traversed from the root to the leaves. At each node, the ancestral architecture yielding the lowest total cost over the whole tree is selected. If several architectures give the same total cost, one of them is chosen randomly. The outcome is a phylogenetic tree with inferred ancestral architectures at all inner nodes and the extant architectures indicated at the leaves. For each protein architecture at the leaves, we counted how many times it had been independently created in the domain tree and also identified its origin by determining which sequences that

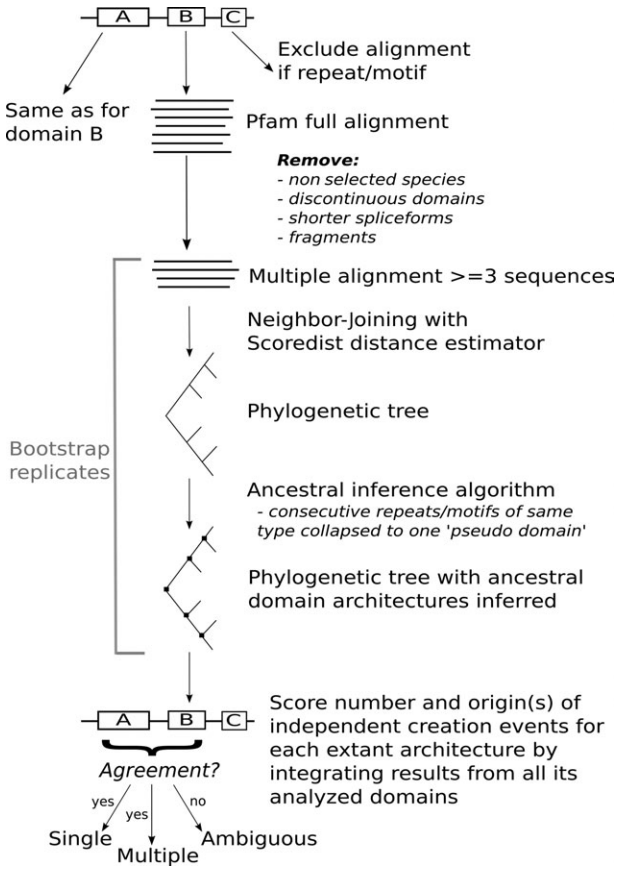

FIG. 1.—Overview of our analysis pipeline. We started with Pfam full alignments of domains that were not of the type repeat or motif. Subsequently, the alignments were exposed to several filtering steps in order to keep only wanted and useful sequences, as described in the text. Each domain architecture was classified into 3 different categories based on the agreement between domains and the number and origin of independent creation events. If the domains did not agree, the architecture was classified as ambiguous. If a majority agreed on one creation, it was classified as single. If a majority indicated multiple creations, the architecture was classified as multiple as long as the phylogenetic origins of the creations agreed.

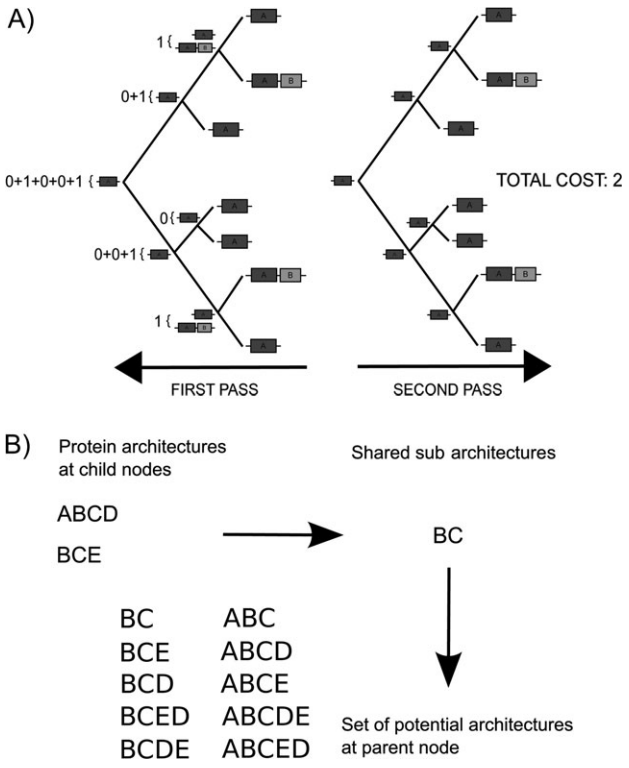

FIG. 2.—Illustration of the ancestral architecture inference algorithm. (*A*) Shown is a schematic of a phylogenetic tree for domain family A. In the first pass, the tree is traversed from the leaves and upward in the tree. At each inner node, all potential ancestral architectures and their costs are enumerated. The costs are propagated upward in the tree, and only the architectures with lowest costs are kept at each node. In the second pass, the tree is traversed in the opposite direction, starting from the root, and thereafter downward in the tree. All enumerated architectures at each inner node are evaluated to find the ones that give the lowest total tree cost. (*B*) Description of how the set of potential ancestral architectures at each parent node is determined. Two child architectures (ABCD and BCE) present in the hypothetical phylogenetic tree for domain family B are shown. From the 2 child architectures, we conclude that the shared subarchitecture is BC. All possible permutations containing the subarchitecture BC are enumerated, with the constraint that the domain from which the tree was built has to be present (domain B). From this set of potential ancestral architectures, the most parsimonious architecture is chosen as the ancestral one.

evolved from each creation. To assess the quality of the phylogenetic trees, we used a bootstrapping approach. For each domain family, a hundred bootstrap pseudoreplicate alignments were generated, and the entire above analysis was repeated for each pseudoreplicate. Finally, for each architecture in the data set, we collected the number of independent creation events and origins for that architecture from the individual domain trees.

To score the number of creation events for a given architecture, phylogenetic trees and all their bootstrap pseudoreplicate trees had to be available for at least 2 domains of the given architecture. If a majority of the phylogenetic trees for the individual domains of an architecture supported single or multiple evolutionary events and there was agreement regarding the origin of these events, that architecture was scored as "single" or "multiple," respectively. By "agreement regarding the origin," we mean that for a particular architecture, there was no conflict between the individual domain trees regarding which sequences each independent architecture creation event gave rise to. A conflict would mean that 2 sequences are found in the same cluster of sequences in one domain tree but in different clusters in another. As a result, we avoided drawing conclusions for architectures where the estimated phylogeny varied widely with sampling. Conversely, if there was no majority regarding the evolutionary pattern of an architecture, it was scored as "ambiguous."

Our algorithm handles repeats and motifs in such a way that a consecutive series of the same kind of repeat or motif is collapsed to one "pseudo domain," that is taken into account in the ancestral inference algorithm. However, as mentioned above, no phylogenetic trees are calculated for repeats and motifs. This means that a protein has to contain at least 2 domains that are not repeats or motifs to be amenable to our analysis. This approach was chosen because current algorithms have difficulties with assigning the precise number of repeats and motifs due to their short length. Furthermore, it is questionable whether the exact number of repetitions is important for the protein function. For a few domain families, the ancestral architecture inference could simply not be completed due to computational time and memory constraints. This occurred for families with a high number of different domains mapping to architectures in the tree and especially when the domain from which the tree was constructed was repeated many times in the architectures. Because the latter is true especially for repeats, this further motivates our handling of repeats and motifs.

Pure loss of a domain can be considered to be a more trivial event than other rearrangements of domains in an architecture. Consequently, we also scored architecture reinvention events with pure domain loss excluded. In the phylogenetic tree with ancestral architectures inferred, we scored the number of independent evolutionary events as previously described with the modification that if the only difference in the architecture between a child node and its parent was the loss of one or several domains, the architecture of the child and parent node was considered to be the same.

A problem when studying protein architectures is that sequence regions without assigned domains are not amenable to analysis. Either the domains have evolved beyond recognition of current methods or they are simply not represented in the domain databases. To address this issue, we generated 2 different data sets. The so-called *no-limit* data set contains all sequences from our selected species present in Pfam-A. The other data set (*max50*), only includes sequences with unassigned N-, C-terminal, or interdomain regions of 50 residues or shorter. We classified regions as unassigned if there was no assignment of any type of Pfam-A region (family, domain, repeat, or motif). Sequences present in the *max50* data set are also present in the *no-limit* data set. This implies that ideally, the architectures found to have evolved through convergent evolution in the *max50* data set should comprise a subset of the architectures found in the *no-limit* data set. The choice of the 50 residues cutoff is a trade-off between quality and quantity of the data set. Approximately 95% of the Pfam families of type family or domain are longer than this

cutoff, making it unlikely that an unknown domain is residing in an unassigned region in the *max50* data set. Both data sets were prepared as stated above to keep only wanted and useful sequences (see fig. 1).

To assess the difference in number of multiple independent domain creation events found in our study and Gough (2005), we analyzed the same set of genomes as in Gough using our method. From the Pfam full alignments, we extracted sequences that belonged to the species in the Gough study. For the 3 nonfungi eukaryotes, we consistently chose the longest spliceform according to the rules stated above. We analyzed possible convergent evolution with a maximum of 50 residues for unassigned regions, which is in essence similar to the limit used by Gough. The data set was prepared as stated above to keep only wanted and useful sequences (see fig. 1). Hereafter, this data set will be referred to as the Gough data set.

Algorithm

```
Data types:
node {
parent : node
children : array of node
found_architecture : architecture
potential_architectures : set of [architecture, child_-
architecture_left,
child_architecture_right, cost]
}
Main algorithm:
Pass 1:
traverse tree in DFS (depth first search) order starting
with leaf
{
if node is leaf
node.potential_architectures = { (architecture, null,
null, 0) }
else
calculate potential architectures and costs
}
Pass 2:
traverse tree in BFS (breadth first search) order starting
with root
{
node.found_architecture = least expensive architec-
ture from potential_architectures
}
Calculation of potential architectures and costs:
for all combinations from children's potential_archi-
tectures set
{
identify shared subarchitectures
calculate all combinations of non-shared subarchitec-
tures
if architecture already exists in potential_architectures
update cost if cost is lessened
else
add architecture and cost to potential_architectures
}
```

GO Term Analysis

In order to investigate whether any specific biological process was significantly over- or underrepresented in the set of reinvented architectures, we employed the Gene Ontology (The Gene Ontology Consortium 2000) Biological Process terms for all proteins in the *no-limit* data set. The unfiltered UniProtKB GO annotations file (submission date 5/1/2007) was downloaded from the Gene Ontology Consortium Web site (http://www.geneontology.org/) to retrieve GO term assignments. Each architecture in the *no-limit* data set was associated with all GO terms assigned to proteins exhibiting the architecture in question. To ensure sufficient sample size, we only included GO terms present in at least 5% of the multiply originated architectures and to leave out too general annotations only terms at depth $>= 3$ in the GO hierarchy were included. The frequency of each GO term in the singly and multiply originated architectures was calculated; correcting for the fact that the multiply originated architectures had on average a greater number of annotations compared with the singly originated. We also calculated the probability of these observations under the null hypothesis that they are not enriched in either set using a hypergeometric distribution.

More in detail, we consider the sampling of a subset (annotations in the multiply originated architectures) from another set (annotations in both singly and multiply originated architectures). Let Y be a stochastic variable denoting the number of times a given annotation is observed in the sampled subset. n is the size of the subset, whereas N is the size of the full set. For a given annotation term, k is the number of times it is observed in the subset and r is the number of times it is observed in the full set. Then the probability of our sample containing exactly k architectures associated with the annotation is

$$P(Y=k)=\frac{\binom{r}{k}\times\binom{N-r}{n-k}}{\binom{N}{n}}.$$

If $k/n > r/N$, the annotation is enriched in the sample. The probability of sampling more than k architectures annotated by chance becomes

$$P(Y \geq k)=\sum_{k_i=k}^{\min(r,n)} P(Y=k_i).$$

If $k/n < r/N$, the annotation may have been depleted in the sample. The probability of sampling less than k architectures annotated by chance becomes

$$P(Y \leq k)=\sum_{k_i=0}^{k} P(Y=k_i)$$

Results

For the *no-limit* data set, we extracted 8,367 unique multidomain architectures where consecutive repeats and motifs of the same kind had been collapsed to one pseudo domain (see table 1). The equivalent number for the *max50*

Table 1
Candidates for Convergent Protein Architecture Evolution Found in the Various Data Sets

Data Set	N_{total}	$N_{ambiguous}$	N_{single}	$N_{multiple}$
No-limit	8367	1605	4613	650
Max50	1798	301	1172	70
Max50, Gough	1388	205	900	48

NOTE.—N_{total} is the number of architectures included in the analysis, $N_{ambiguous}$ is the number of architectures were results from the individual domains were not conclusive, N_{single} is the number of single-origin architectures, and $N_{multiple}$ is the number of multiple-origin architectures.

data set was 1,798 architectures. Out of these, approximately 82% of the architectures in the *no-limit* data set and approximately 86% in the *max50* data set could be analyzed. We scored the number and origin of creation events for each domain architecture. The great majority of architectures appeared to have arisen only once (see table 1). There are also architectures where the number and/or origin of creation events is ambiguous because the domains show different results. Nevertheless, we found 650 cases, or 12.4% (650/5,263), of convergent protein architecture evolution in the *no-limit* data set (see tables 1 and supplementary table S2 [Supplementary Material online]). In the *max50* data set, there were 70 cases, or 5.6% (70/1,242) (see tables 1 and supplementary table S3 [Supplementary Material online]). When excluding origin events that purely involve domain loss, approximately 1/3 of the number of architectures with multiple independent origin remained in both data sets.

In the majority of cases, the architectures have evolved through 2 independent creation events (59% in the *no-limit* data set and 56% in the *max50* data set, respectively). However, there is also a substantial fraction of architectures that have 3 independent evolutionary origins (19% in the *no-limit* data set and 21% in the *max50* data set, respectively). Thereafter, the number of architectures is rapidly declining with increasing number of independent creation events in both data sets (data not shown). Of the 70 candidates for convergent evolution in the *max50* data set, 36 were also found in the *no-limit* data set and 22 of them were consistent across both data sets with regards to inferred number of creation events and origins (see supplementary table S4, Supplementary Material online). The remaining 14 architectures have different numbers of independent creation events in the 2 data sets. In all these cases, the inclusion of more sequences in the *no-limit* data set has resulted in more species being represented in the phylogenetic domain trees and subsequently an increased number of convergent evolutionary events.

For both data sets, a substantial fraction of the candidates for convergent evolution was found in eukaryotes (see fig. 3). For the *no-limit* data set, more than half of the candidates were found in eukaryotes only. According to the *no-limit* data set, bacteria exhibit a rate of convergent evolution that is lower compared with eukaryotes. In the *max50* data set, the opposite pattern is seen. However, we believe that this is due to the nature of the *max50* data set, where especially eukaryotic architectures will be excluded due to the unassigned region cutoff of 50 residues. Archaea display the lowest number of architectures with multiple independent creation events across both data sets. Moreover, most candidates are only found in one type of kingdom.

When analyzing the Gough data set using our method with a 50 residues interdomain region cutoff, which is roughly similar in scope to the approach used by Gough, we found 1,388 unique continuous multidomain architectures where consecutive repeats and motifs of the same kind had been collapsed to one pseudo domain. Out of these, evolutionary events could be scored for approximately 83% of the architectures. We found 48 cases, or 5.1% (48/948), of convergent evolution in the Gough data set. As for the other 2 data sets, approximately 1/3 of the number of architectures with multiple independent origin remained when architectural origins that purely involve domain loss had been excluded.

We detected convergent evolution among proteins belonging to the spermidine/spermine synthase family. The architecture (S-adenosylmethionine decarboxylase * Spermine/spermidine synthase) has arisen twice independently in bacteria; once in *Bdellovibrio bacteriovorus* and once in *Azoarcus* sp. (see fig. 4). The spermine/spermidine synthase synthesizes spermine and spermidine from putrescine and decarboxylated S-adenosylmethionine (Li et al. 2001). Among prokaryotes, spermine/spermidine synthase and

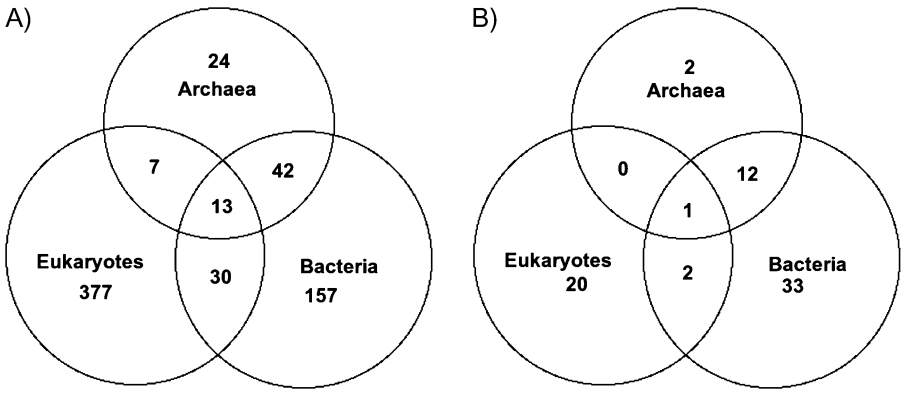

FIG. 3.—The distribution of domain architectures exhibiting multiple independent creation events across the 3 kingdoms. (*A*) the *no-limit* data set and (*B*) the *max50* data set.

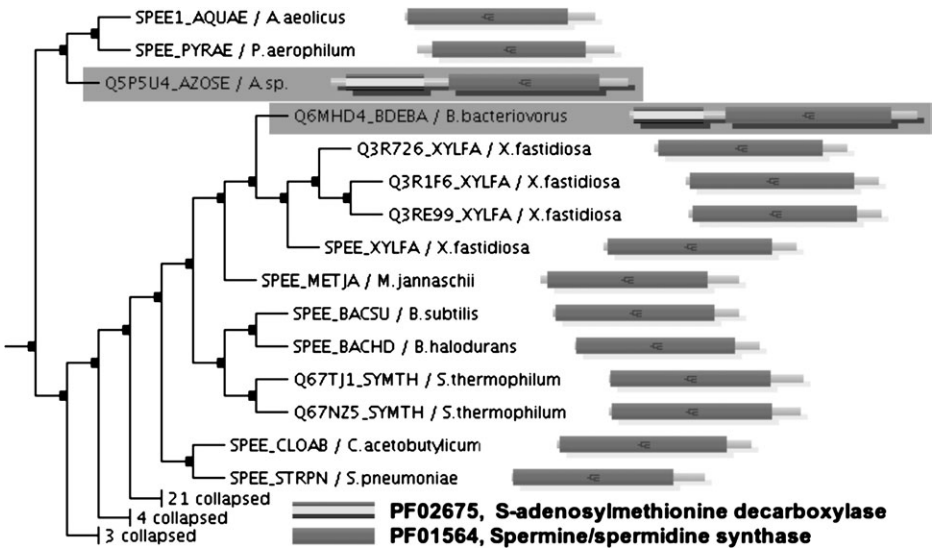

FIG. 4.—Example of independent protein architecture creation. Shown is the topology of the tree for Pfam domain Spermine/spermidine synthase (PF01564). The architecture (S-adenosylmethionine decarboxylase * Spermine/spermidine synthase) ([PF02675 * PF01564]) has arisen twice independently in bacteria (shaded boxes): once in *Bdellovibrio bacteriovorus* and once in *Azoarcus* sp. The tree shown is a subtree of the full phylogenetic tree, and it is not drawn to scale.

S-adenosylmethionine decarboxylase are usually present as single-domain proteins. However, in some prokaryotes, the process seems to have been made more effective by fusing the domains into a multidomain protein with dual functions. This can be advantageous for the spermine/spermidine synthesis process as the 2 catalytic steps can happen in direct succession.

Looking beyond our data set, in total 8 prokaryotes have the 2-domain protein in the Pfam database. Only 2 of these were included in our selection of genomes. None of the 8 prokaryotes seem to have maintained the S-adenosylmethionine decarboxylase as a single-domain protein, and only half of them have the single-domain protein spermine/spermidine synthase. Previous studies have suggested that the need for a compact genome can lead to evolution of multidomain proteins from interacting single-domain proteins (Brocchieri and Karlin 2005). *Bdellovibrio bacteriovorus* is a parasite of other gram-negative bacteria, and it has a smaller genome compared with the free-living soil bacteria *Azoarcus* sp. Interestingly, *Azoarcus* sp. still has a single-domain protein of spermine/spermidine synthase in its genome, whereas *B. bacteriovorus* has lost it. The other 3 prokaryotes with a single-domain spermine/spermidine synthase all have genome sizes larger than *Azoarcus* sp. The prokaryotes deficient in single-domain spermine/spermidine synthase on the other hand have small genomes, mostly smaller than *B. bacteriovorus*. In fact, one of them (*Candidatus Pelagibacter ubique*) has the smallest genome of any cell known to replicate independently in nature.

An example of independent protein architecture creation found in eukaryotes is shown in fig. 5. The architecture (Glycosyl hydrolases family 17 * X8 domain * X8 domain) has arisen twice independently in plants; once in *Arabdidopsis thaliana* and once in *Oryza sativa*. The N-terminal domain in the architecture belongs to the glycosyl hydrolases, a widespread group of enzymes that hydrolyze the glycosidic bond between 2 or more carbohydrates or between a carbohydrate and a noncarbohydrate moiety (Henrissat and Davies 2000). The X8 domain is thought to be involved in carbohydrate binding by the formation of disulphide bridges. *Arabdidopsis thaliana* and *O. sativa* have proteins with none, one, or 2 X8 domains C-terminally of the glycosyl hydrolase (see fig. 5). Also, single-domain X8 proteins are present in the genomes of these 2 plants. The fusion of a glycosyl hydrolase and an X8 domain may help to attune the hydrolase to a particular substrate and allows more specific regulation. Furthermore, one can speculate that the need for a different substrate specificity has led to an independent duplication of the X8 domain in some proteins.

Another example of a protein architecture found to have arisen multiple times in both data sets is the architecture (Glutamine amidotransferase class-I * Glycosyl transferase family, helical bundle domain * Glycosyl transferase family, a/b domain) (see fig. 6), which has been created independently in bacteria—once in *Escherichia coli*/*Salmonella typhimurium* and once in *Thermotoga maritima*. The protein is anthranilate synthase component II and is part of the tryptophan synthesis pathway (Romero et al. 1995). Together with anthranilate synthase component part I, an oligomer is formed. The anthranilate synthase catalyzes the second step in the pathway leading from chorismate to tryptophan (Roberts et al. 2002). Chorismate is used as a substrate not only for the tryptophan pathway but also for producing other amino acids such as phenylalanine and tyrosine. Also, folate and ubiquinone synthesis require chorismate as substrate. Apparently, the tryptophan synthesis pathway has been optimized, either for efficiency or for regulatory control, by independent creation events of the 3-domain architecture in different bacteria.

Are some domains more versatile than others with respect to the architectures in which they can function? We

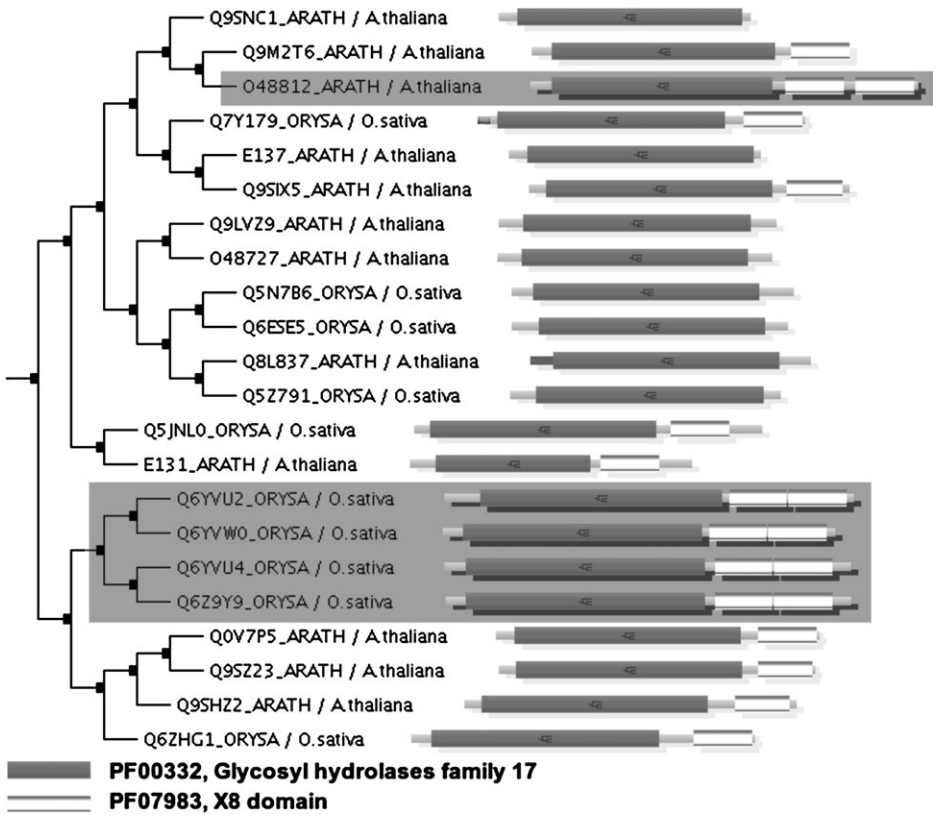

FIG. 5.—Example of independent protein architecture creation. Shown is the topology of the tree for Pfam domain Glycosyl hydrolases family 17 (PF00332). The architecture (Glycosyl hydrolases family 17 * X8 domain * X8 domain) ([PF00332*PF07983*PF07983]) has arisen twice independently in eukaryota (shaded boxes): once in *Arabidopsis thaliana* and once in *Oryza sativa*. The tree shown is a subtree of the full phylogenetic tree, and it is not drawn to scale.

would expect some domains to occur in convergent evolution more often than others, reflecting the ability to function well in different types of architectures. Table 2 lists the domains present in architectures with more than 10 independent creation events in the *no-limit* data set. Many of the top-ranked domains in the list are frequent in general, but some exceptions exist. For instance, C1 (PF03107) and C1-like (PF07649) domains both occurred in 13 reinvented domain architectures but only in 28 and 31 architectures, respectively, in total. The function of these domains is variable and largely unknown. However, C1 and C1-like domain–containing proteins are known to be involved in signaling pathways in a variety of organisms (Hurley et al. 1997).

Are proteins connected to some biological processes more likely to involve reinvented architectures? To investigate this, we evaluated whether any Gene Ontology Biological Process terms were significantly enriched in the set of reinvented architectures. At the 95% confidence level, 22 terms were significantly enriched or depleted in the set of architectures with multiple origin (see table 3). GO terms associated with signal transduction were only found among the terms significantly enriched in multiple origin architectures. In contrast, GO terms significantly enriched in single origin architectures were dominated by functions associated with metabolic processes, whereas only one such case was found among terms enriched in multiple origin architectures.

Discussion

We have presented a novel algorithm for analyzing protein architecture evolution based on domain trees. The algorithm uses maximum parsimony to infer ancestral architectures. Given the ancestral architectures on the tree, we were able to track the origin of each architecture. We used this to search for cases of domain architectures with multiple evolutionary origins, and our results suggest that such cases are more frequent than previously thought. In the *no-limit* data set, 12.4% of the explored architectures showed evidence of reinvention. In the more restricted *max50* data set, the figure was 5.6%. Compared with a previous study (Gough 2005) that suggested a proportion of 1.9%, our figures are significantly higher. To assess the possible reasons for this discrepancy, we analyzed the same set of genomes used in Gough (2005) using our method with a 50 residues cutoff for unassigned regions, which is in essence the same limit used by Gough. Using our method for preparing the data set, we only found approximately half the number of multidomain architectures as Gough did in the same genomes. A number of reasons can be responsible for this, including our rigorous filtering steps and differences between SUPERFAMILY and Pfam domain annotation. We also note that Gough did not explicitly remove alternate splice forms. The results from our analysis of the Gough genomes indicate approximately the same proportion of

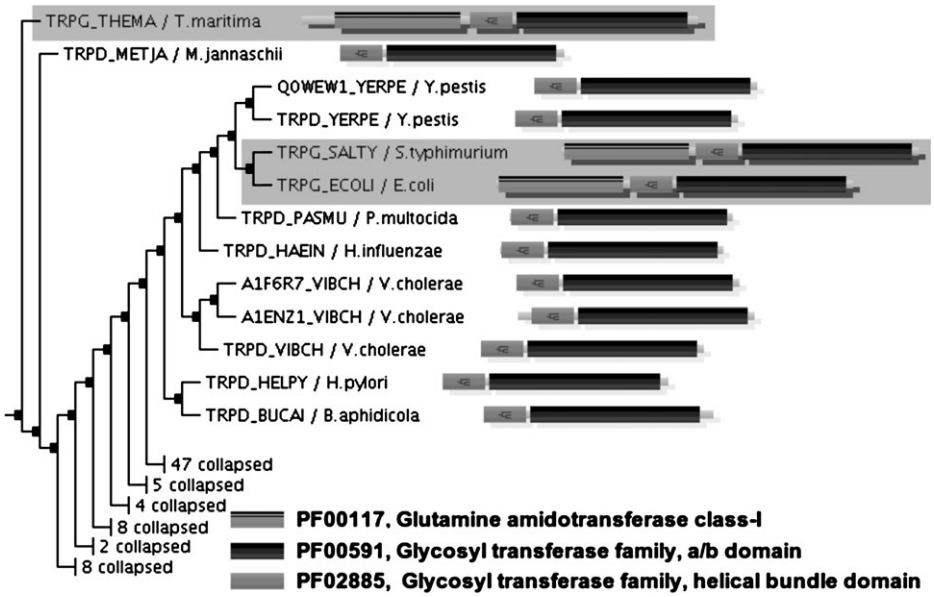

FIG. 6.—Example of independent protein architecture creation. Shown is the topology of the tree for Pfam domain Glycosyl transferase family, helical bundle domain (PF02885). The architecture (Glutamine amidotransferase class-I * Glycosyl transferase family, helical bundle domain * Glycosyl transferase family, a/b domain) ([PF00117 * PF02885 * PF00591]) has arisen twice independently in bacteria (shaded boxes): once in *Escherichia coli* and *Salmonella typhimurium* and once in *Thermotoga maritima*. The tree shown is a subtree of the full phylogenetic tree, and it is not drawn to scale.

convergent evolution (5.1%) as in our *max50* data set (5.6%). This suggests that the main reason for our much higher figure is differences in the methods, whereas differences in the data set have a small effect.

There are several major differences between our method and the method implemented by Gough (2005). Our method utilizes domain trees instead of a species tree that allows us to detect protein architecture reinvention at

Table 2
Most Frequent Domains in Multiple Origin Architectures

Domain	Pfam AC	Multiple Origin Architectures	All Architectures
Histidine kinase-, DNA gyrase B-, and heat shock protein 90-like ATPase	PF02518	44	268
His Kinase A (phosphoacceptor) domain	PF00512	33	195
Response regulator receiver domain	PF00072	27	177
Reverse transcriptase (RNA-dependent DNA polymerase)	PF00078	23	88
Integrase core domain	PF00665	20	87
PAS fold	PF00989	20	192
PAS fold	PF08447	20	164
Zinc knuckle	PF00098	19	102
PAS fold	PF08448	17	172
Retrotransposon gag protein	PF03732	16	67
Chromo' (CHRromatin Organization MOdifier) domain	PF00385	16	50
PH domain	PF00169	16	125
SH3 domain	PF00018	16	103
GGDEF domain	PF00990	15	94
4Fe-4S–binding domain	PF00037	13	58
Retroviral aspartyl protease	PF08284	13	42
C1-like domain	PF07649	13	31
C1 domain	PF03107	13	28
Helicase conserved C-terminal domain	PF00271	13	100
Zinc finger, C3HC4 type (RING finger)	PF00097	12	90
GAF domain	PF01590	12	128
PDZ domain (Also known as DHR or GLGF)	PF00595	11	73
C2 domain	PF00168	11	66

NOTE.—Listed here are domains occurring in more than 10 unique architectures with multiple origins in the *no-limit* data set. For each domain, the number of architectures it occurs in is shown, both for multiple origin architectures as well as for all architectures in the data set.

Table 3
GO Biological Process Terms over- or Underrepresented among Architectures with Multiple Origins in the *No-limit* Data Set at 95% Confidence Level

GO ID	Description	%Multiple	%Single	Ratio	*P* Value
GO:0006278	RNA-dependent DNA replication	0.7	0.2	3.99	2.9×10^{-2}
GO:0009064	Glutamine family amino acid metabolic process	1.2	0.4	3.13	1.4×10^{-2}
GO:0032446	Protein modification by small protein conjugation	2.7	1.4	1.85	2.0×10^{-2}
GO:0016567	Protein ubiquitination	2.5	1.4	1.77	3.1×10^{-2}
GO:0006512	Ubiquitin cycle	2.9	1.7	1.66	3.6×10^{-2}
GO:0007242	Intracellular signaling cascade	5.3	3.5	1.54	1.5×10^{-2}
GO:0007165	Signal transduction	16.4	12.9	1.27	9.7×10^{-3}
GO:0044249	Cellular biosynthetic process	4.7	6.3	0.75	4.3×10^{-2}
GO:0005975	Carbohydrate metabolic process	1.5	3.6	0.42	9.3×10^{-4}
GO:0044255	Cellular lipid metabolic process	0.6	1.4	0.41	2.1×10^{-2}
GO:0044262	Cellular carbohydrate metabolic process	0.8	2.0	0.40	1.2×10^{-2}
GO:0015980	Energy derivation by oxidation of organic compounds	0.8	2.3	0.35	3.7×10^{-3}
GO:0006366	Transcription from RNA polymerase II promoter	0.4	1.0	0.33	3.8×10^{-2}
GO:0009611	Response to wounding	0.2	0.8	0.28	3.3×10^{-2}
GO:0000278	Mitotic cell cycle	0.1	0.5	0.24	4.8×10^{-2}
GO:0009110	Vitamin biosynthetic process	0.1	0.5	0.24	4.8×10^{-2}
GO:0019318	Hexose metabolic process	0.1	0.5	0.22	3.7×10^{-2}
GO:0008610	Lipid biosynthetic process	0.2	1.1	0.21	8.4×10^{-3}
GO:0005996	Monosaccharide metabolic process	0.1	0.5	0.21	3.2×10^{-2}
GO:0006633	Fatty acid biosynthetic process	0.1	0.6	0.20	2.8×10^{-2}
GO:0016051	Carbohydrate biosynthetic process	0.1	0.7	0.17	1.3×10^{-2}
GO:0006066	Alcohol metabolic process	0.1	0.9	0.13	3.9×10^{-3}

NOTE.—%Multiple and %Single denote the percentage of multiply and singly originated architectures that were associated with the respective GO term. Ratio denotes the ratio between the frequencies in the multiple origin set (650 architectures) and the single origin set (4,613 architectures). A ratio >1 indicates that the GO term is overrepresented in the multiply originated architectures, whereas a ratio <1 means that it is underrepresented. The *P* value is the probability of the observed counts of GO terms in both sets given a hypergeometric distribution.

any node in the tree, whereas the Gough method was limited to detection between species. This makes our analysis much more fine grained. Another advantage is that we do not need to worry about the correctness of a species tree. Instead a tree is built for each domain family, making it possible to gain further support for architecture reinvention events by comparing the results for all the domains in the architecture. Also, studies based on a species tree run the risk of finding false cases of convergent protein architecture evolution due to HGT. HGT is a phenomenon that occurs quite frequently in prokaryotes, although very rare in eukaryotes (Choi and Kim 2007). By using a domain tree, we are studying the sequences directly, irrespective of which species they are found in, and therefore, our approach is not sensitive to HGT. A majority of our candidates for convergent evolution in the *no-limit* data set are found in eukaryotes, although there are more prokaryotic sequences present in the data set. In the *max50* data set, more candidates are found in prokaryotes but this is correlated to a decrease in eukaryotic sequences as the cutoff for unassigned regions is applied. We would have expected prokaryotic architecture reinvention to be seen much more often if our method could be tricked by HGT.

Our rationale for creating and analyzing 2 different data sets (*no-limit* and *max50*) is that they both have strengths and weaknesses. The *max50* data set is restricted to fewer but more completely annotated sequences, which is likely to result in a better resolved phylogeny for a particular architecture. However, the limited number of sequences could bias the ancestral architecture inference and thus, multiple independent creation events could be missed or be falsely detected. In particular, entire groups of sequences that have evolved independently might be excluded by the filtering steps. In contrast, the *no-limit* data set covering more sequences should allow a more accurate ancestral architecture inference, although with a higher risk of including unknown domains that could alter the inference had they been known. Another reason for the 2 data sets is that especially eukaryotic architectures are penalized by the 50 residues cutoff. Consequently, to gain a better understanding of eukaryotic domain architecture evolution, the *no-limit* data set is an important complement to the more restricted *max50* data set. Ideally, the architectures that were found to have multiple independent creation events in the *no-limit* data set should also be detected in the *max50* data set, that is, the *max50* data set should constitute a subset of the *no-limit* data set. However, the difference in sequence coverage between the 2 data sets can result in different phylogenetic trees and consequently, the inference of ancestral architectures can also differ. Therefore, not all candidates found in the *max50* data set are likely to be found in the *no-limit* data set. Indeed, of the 70 candidates for convergent evolution found in the *max50* data set, 34 are not found in the *no-limit* data set. Inspection of these cases showed that the main reason for this discrepancy was that, in the *no-limit* data set, more proteins exhibited the architecture in question, thereby shifting its invention closer to the root of the phylogenetic tree. Alternatively, the results for the architectures in question were ambiguous in the *no-limit* data set.

The main reasons for why convergent evolution could not be scored for some multidomain architectures was that either the phylogenetic trees and/or some of their bootstrap pseudoreplicate trees could not be generated for at least 2

domains of the given architecture (18% for *no-limit* and 14% for *max50*). Missing phylogenetic trees, in turn, are mainly due to our handling of repeats and motifs or that our filtering steps to obtain an adequate multiple alignment quality simply excluded the domain family. A possible drawback of our method is that it depends on Neighbor-Joining trees, which cannot be reliably generated in all cases, and therefore, we introduced a quality filter based on bootstrapping. We conclude that, due to this filtering step, the method has high reproducibility despite the risk of phylogenetic uncertainty. Another potential limitation is that the phylogenetic trees are binary. Even though binary trees are most commonly used in phylogeny, sometimes the resolution at a particular node is so low that it is impossible to determine the correct branching order; in such cases, allowing unresolved tree nodes should give more robust results.

In our ancestral inference algorithm, we use an equal cost for the loss or gain of a domain. Although, previous studies (Kummerfeld and Teichmann 2005; Fong et al. 2007) have attempted to answer whether fusion or fission events are more common, we felt that the matter is not thoroughly enough studied, particularly for Pfam domain architectures. Therefore, we chose an equal cost model. Another reason for our choice of not weighting the 2 types of evolutionary events differently is that higher penalties for domain gain in this context would lead to ancestral architecture assignments where gain events would tend to occur only close to the root of the tree. This we believe to be unreasonable as it would tend to lead to ancestral architectures with more domains than the leaves. Also, some domains are known to co-occur in so-called supradomains (Vogel et al. 2004) and that would have to be taken into account if differentiated costs for gain or loss were to be implemented. However, not all such supradomains are likely to be known, and there may also be other relationships between domains that affect how likely they are to be gained or lost together.

Our figures are dependent on the completeness of Pfam and to some extent on how good the domain recognition is. Because of the incompleteness of domain annotations, the multiply originated architectures deduced by this study cannot cover all cases of convergent architecture evolution, yet at the same time not all the inferences might be true. The method does, however, retrieve a reasonably complete set to which further analysis can be applied, and both specificity and sensitivity will increase in the future as more domain assignments become available. Our approach should be seen as a tool for finding candidates for convergent evolution that should subsequently be manually validated. However, already our results clearly show that domain architecture reinvention is relatively frequent even in a small data set. We find no strong functional bias among the architectures with multiple independent evolutionary events, suggesting that the process of convergent domain architecture evolution is driven by chance rather than functional necessity.

Supplementary Material

Supplementary tables S1–S4 are available at *Molecular Biology and Evolution* online (http://www.mbe.oxfordjournals.org/).

Acknowledgments

This work has been supported by grants from the Swedish Research Council, Pfizer Corporation, and the Swedish Knowledge Foundation through the Industrial PhD program in Medical Bioinformatics at the Strategy and Development Office at Karolinska Institutet.

Literature Cited

Apic G, Gough J, Teichmann SA. 2001. Domain combinations in archaeal, eubacterial and eukaryotic proteomes. J Mol Biol. 310:311–325.

Brocchieri L, Karlin S. 2005. Protein length in eukaryotic and prokaryotic proteomes. Nucleic Acids Res. 33:3390–3400.

Choi I-G, Kim S-H. 2007. Global extent of horizontal gene transfer. Proc Natl Acad Sci USA. 104:4489–4494.

Doolittle RF. 1995. The multiplicity of domains in proteins. Annu Rev Biochem. 64:287–314.

Ekman D, Bjorklund AK, Frey-Skott J, Elofsson A. 2005. Multi-domain proteins in the three kingdoms of life: orphan domains and other unassigned regions. J Mol Biol. 348:231–243.

Finn RD, Mistry J, Schuster-Bockler B, et al. (13 co-authors). 2006. Pfam: clans, web tools and services. Nucleic Acids Res. 34:D247–D251.

Fong JH, Geer LY, Panchenko AR, Bryant SH. 2007. Modeling the evolution of protein domain architectures using maximum parsimony. J Mol Biol. 366:307–315.

Gough J. 2005. Convergent evolution of domain architectures (is rare). Bioinformatics. 21:1464–1471.

Henrissat B, Davies G. 2000. Glycoside hydrolases and glycosyltransferases. Families, modules, and implications for genomics. Plant Physiol. 124:1515–1519.

Hurley JH, Newton AC, Parker PJ, Blumberg PM, Nishizuka Y. 1997. Taxonomy and function of C1 protein kinase C homology domains. Protein Sci. 6:477–480.

Jaenicke R. 1987. Folding and association of proteins. Prog Biophys Mol Biol. 49:117–237.

Kummerfeld SK, Teichmann SA. 2005. Relative rates of gene fusion and fission in multi-domain proteins. Trends Genet. 21:25–30.

Li YF, Hess S, Pannell LK, White Tabor C, Tabor H. 2001. In vivo mechanism-based inactivation of S-adenosylmethionine decarboxylases from Escherichia coli, Salmonella typhimurium, and Saccharomyces cerevisiae. Proc Natl Acad Sci USA. 98:10578–10583.

Murzin AG, Brenner SE, Hubbard T, Chothia C. 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol. 247:536–540.

Roberts CW, Roberts F, Lyons RE, et al. (18 co-authors). 2002. The shikimate pathway and its branches in apicomplexan parasites. J Infect Dis. 185(Suppl 1):S25–S36.

Romero RM, Roberts MF, Phillipson JD. 1995. Anthranilate synthase in microorganisms and plants. Phytochemistry (Oxf). 39:262–276.

Rossmann MG, Moras D, Olsen KW. 1974. Chemical and biological evolution of nucleotide-binding protein. Nature. 250:194–199.

Sonnhammer ELL, Hollich V. 2005. Scoredist: a simple and robust protein sequence distance estimator. BMC Bioinformatics. 6:108.

The Gene Ontology Consortium. 2000. Gene ontology: tool for the unification of biology. Nat Genet. 25:25–29.

Vogel C, Berzuini C, Bashton M, Gough J, Teichmann SA. 2004. Supra-domains: evolutionary units larger than single protein domains. J Mol Biol. 336:809–823.

Wang M, Caetano-Anollés G. 2006. Global phylogeny determined by the combination of protein domains in proteomes. Mol Biol Evol. 23:2444–2454.

Weiner J 3rd, Beaussart F, Bornberg-Bauer E. 2006. Domain deletions and substitutions in the modular protein evolution. FEBS J. 273:2037–2047.

Weiner J 3rd, Bornberg-Bauer E. 2006. Evolution of circular permutations in multi-domain proteins. Mol Biol Evol. 23:734–743.

Michele Vendruscolo, Associate Editor

Accepted October 25, 2007