Protein domain architectures provide a fast, efficient and scalable alternative to sequence-based methods for comparative functional genomics [version 1; peer review: 1 approved, 2 approved with reservations]

Jasper J. Koehorst, Edoardo Saccenti, Peter J. Schaap, Vitor A. P. Martins dos Santos, Maria Suarez-Diez

1Laboratory of Systems and Synthetic Biology, Wageningen University and Research, Stippeneng, The Netherlands
2LifeGlimmer, LifeGlimmer GmBH, Berlin, Germany

Abstract
A functional comparative genome analysis is essential to understand the mechanisms underlying bacterial evolution and adaptation. Detection of functional orthologs using standard global sequence similarity methods faces several problems; the need for defining arbitrary acceptance thresholds for similarity and alignment length, lateral gene acquisition and the high computational cost for finding bi-directional best matches at a large scale.

We investigated the use of protein domain architectures for large scale functional comparative analysis as an alternative method. The performance of both approaches was assessed through functional comparison of 446 bacterial genomes sampled at different taxonomic levels.

We show that protein domain architectures provide a fast and efficient alternative to methods based on sequence similarity to identify groups of functionally equivalent proteins within and across taxonomic boundaries. As the computational cost scales linearly, and not quadratically with the number of genomes, it is suitable for large scale comparative analysis. Running both methods in parallel pinpoints potential functional adaptations that may add to bacterial fitness.

Keywords
Bacterial genomics, Bacterial functionome, Orthology, Horizontal gene transfer, clustering, semantic annotation
Introduction
Comparative analysis of genome sequences has been pivotal to unravel mechanisms shaping bacterial evolution like gene duplication, loss and acquisition\(^1\),\(^2\), and helped in shedding light on pathogenesis and genotype-phenotype associations\(^3\),\(^4\).

Comparative analysis relies on the identification of sets of orthologous and paralogous genes and subsequent transfer of function to the encoding proteins. Technically orthologs are defined as best bi-directional hits (BBH) obtained via pairwise sequence comparison among multiple species and thus exploits sequence similarity for functional grouping. Sequence similarity-based (SB) methods present a number of shortcomings. First, a generalized minimal alignment length and similarity cut-off need to be arbitrarily selected for all, which may hamper proper functional grouping. Second, sequence and function might differ across evolutionary scales. Protein sequences change faster than protein structure and proteins with same function but with low sequence similarity have been identified\(^5\),\(^6\). SB methods may fail to group them hampering a functional comparison. This limitation becomes even more critical when comparing either phylogenetically distant genomes or gene sequences that were acquired with horizontal gene transfer events.

Third, time and memory requirements scale quadratically with the number of genomes to be compared. Recent technological advancements are resulting in thousands of organisms and billions of proteins being sequenced\(^7\), rendering SB approaches of limited applicability for comparisons at the larger scales.

To overcome these bottlenecks, protein domains have been suggested as an alternative for defining groups of functionally equivalent proteins\(^8\),\(^9\) and have been used to perform comparative analyses of *Escherichia coli*, *Pseudomonas*\(^10\), *Streptococcus*\(^11\) and for protein functional annotation\(^12\),\(^13\). A protein domain architecture describes the arrangement of domains contained in a protein and is exemplified in Figure 1. As protein domains capture key structural and functional features, protein domain architectures may be considered to be better proxies to describe functional equivalence than a global sequence similarity\(^14\). The concept of using the domain architecture to precisely describe the extent of functional equivalence is exemplified in Figure 2. Moreover, once the probabilistic domain models have been defined, mining large sets of individual genome sequences for their occurrences is a considerably less demanding computational task than an exploration of all possible bi-directional hits between them\(^15\),\(^16\).

Building on these observations we aim at exploring the potential of domain architecture-based (DAB) methods for large scale functional comparative analysis by comparing functionally equivalent sets of proteins, defined using domain architectures, with standard clusters of orthogonal proteins obtained with SB methods. We compared the SB and DAB approach by analysing *i)* the retrieved number of singletons (*i.e.* clusters containing only one protein) and *ii)* the characteristics of the inferred pan- and core-genome size considering a selection of bacterial genomes (both gram positive and negative) sampled at different taxonomic levels (species, genus, family, order and phylum). We show that the DAB approach provides a fast and efficient alternative to SB

![Figure 1](Figure 1. Domain architecture as a formal description of functional equivalence. Although the proteins obviously share a common core, four distinct domain architectures involving six protein domains were observed in (1) Enterobacteriaceae, (2) *H. pylori*, (3) *Pseudomonas* and (4) Cyanobacteria.)
methods to identify groups of functionally equivalent/related proteins for comparative genome analysis and that the functional pan-genome is more closed in comparison to the sequence based pan-genome. DAB approaches can complement standardly applied sequence similarity methods and can pinpoint potential functional adaptations.

**Methods**

**Genome sequence retrieval**

Bacterial species were chosen on the basis of the availability of fully sequenced genomes in the public domain: two species (Listeria monocytogenes and Helicobacter pylori), three genera (Streptococcus, Pseudomonas, Bacillus), one family (Enterobacteriaceae), one order (Corynebacteriales), and one phylum (Cyanobacteria) were selected. For each a set of 60 genome sequences were considered, except for L. monocytogenes for which only 26 complete genome sequences were available. Maximal diversity among genome sequences was ensured by sampling divergent species (when possible) at each taxonomic level. Genome sequences were retrieved from the European Nucleotide Archive database (www.ebi.ac.uk/ena). A full list of genomes analyzed is available in the Data availability section.

---

**Figure 2. Relationship between domain architecture based (DAB) and sequence similarity based (SB) clustering.** Domains are probabilistic models of amino acids coordinates obtained by hidden Markov modeling (HMM) built from (structure based) multiple sequence alignments. Domain architectures are linear combinations of these domains representing the functional potential of a given protein sequence and constitute the input for DAB clustering. SB-orthology clusters inherit functional annotations via best bi-directional hits above a predefined sequence similarity cut-off score. The information content decreases when moving from the overall function to the sequence level.
De novo genome annotation

To avoid bias due to different algorithms used for the annotation of the original deposited genome sequences, all genomes were de novo re-annotated using the SAPP framework (1.0.0). In particular, the FASTA2RDF, GeneCaller (a semantic wrapper for Prodigal (2.6.2)) and InterPro (interproscan-5.17-56.0) (a semantic wrapper for InterProScan) modules were used to handle and re-annotate the genome sequences. This resulted in 446 annotated genomes (7 × 60 genomes and 1 × 26 genomes) with provenance. For each annotation step the provenance information (E-value cut off, score, originating tool or database) was stored together with annotation information in a graph database (RDF-model) and can be reproduced through the SAPP framework (http://semantics.systemsbiology.nl).

Retrieval of domain architecture

The positions (start and end on the protein sequence) of domains having Pfam, TIGRFAMs and InterPro identifiers were extracted through SPARQL querying of the graph database and domain architectures were retrieved for each protein individually. The domain starting position was used to assess relative position in the case of overlapping domains; alphabetic ordering was used to order domains with the same starting position or when the distance between the starting position of overlapping domains was < 3 amino acids. Labels indicating N-C terminal order of identified domains were assigned to each protein in such a way that the same labels were assigned to proteins sharing the same domain architecture.

Sequence similarity based clustering

To make a direct comparison possible only protein sequences containing at least one protein domain signature were considered for analysis. BBH were obtained using Blastp (2.2.28+) with an E-value cutoff of 10⁻³ and -max_target_seqs of 10⁵. OrthaGogue (1.0.3) combined with MCL (14-137) was used to identify protein clusters on the base of sequence similarity.

Domain architecture based clustering

Domain architecture based clusters were built by clustering proteins with the same labels using bash terminal commands (sort, awk). The number of proteins sharing a given domain architecture in each genome was stored in a 446 × 21054 (genomes × domain architectures) matrix; from this a binarized presence-absence matrix was obtained and used for multivariate analysis.

Heaps’ law fitting and pan-genome openness assessment

A Heaps’ law model was fit to the abundance matrices using 5 × 10⁸ random genome ordering permutations and the micropan R package.

Software

SAPP, a Semantic Annotation Pipeline with Provenance which stores results in a graph database, used for genome handling and annotation, is available at semantics.systemsbiology.nl. Matrix manipulations and multivariate analysis were performed using the R software (3.2.2).

Results

SB and DAB approaches were compared by considering eight sets of genome sequences sampled at different taxonomic levels, from species to order, preserving phylogenetic diversity (see Table 1). Each set contained 60 genome sequences, except for Listeria monocytogenes for which only 26 complete genomes were publicly available. To facilitate the comparison between DAB and SB clusters only protein sequences that contained at least one domain were considered. On average, 85% of the protein sequences contain at least one domain from the InterPro database (see Table 1). Values range from 77±4% for Cyanobacteria to 91 ± 4% for Enterobacteriaceae (which include E. coli). Since the overall results were the same for gram negative and gram positive bacteria, we will show and comment only on results for the latter. Results obtained for gram negative bacteria are shown in the Supplementary material section.

Cluster formation based on sequence similarity

A standard BBH workflow was used to obtain SB protein clusters for the eight sets. We started by calculating the total number of clusters, corresponding to the pan-genome size, as shown in Table 1. Then we considered protein cluster persistence, that is the number of genomes where at least one member of the cluster is present, divided by the total number of genomes considered. Results are shown in Figure 3.

### Table 1. Comparison between DAB and SB clustering

DAB has been performed using HMM from Pfam (29.0) and InterPro (interproscan-5.17-56.0). Fraction refers to the fraction of proteins with at least one protein domain. Core- and pan- indicate the sizes of the core- and pan- genomes (based on the sample) and singletons refers to the number of clusters with only one protein.

<table>
<thead>
<tr>
<th>Taxon</th>
<th>Name</th>
<th>Fraction</th>
<th>DAB Core</th>
<th>DAB Pan</th>
<th>DAB Singletons</th>
<th>InterPro Core</th>
<th>InterPro Pan</th>
<th>InterPro Singletons</th>
<th>SB Core</th>
<th>SB Pan</th>
<th>SB Singletons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Species</td>
<td>H. pylori</td>
<td>0.82 ± 0.01</td>
<td>724</td>
<td>1334</td>
<td>142</td>
<td>534</td>
<td>2888</td>
<td>853</td>
<td>1036</td>
<td>1503</td>
<td>295</td>
</tr>
<tr>
<td>Species</td>
<td>L. monocytogenes</td>
<td>0.89 ± 0.01</td>
<td>1333</td>
<td>2142</td>
<td>309</td>
<td>1414</td>
<td>3415</td>
<td>847</td>
<td>2294</td>
<td>2937</td>
<td>746</td>
</tr>
<tr>
<td>Genus</td>
<td>Bacillus</td>
<td>0.87 ± 0.03</td>
<td>792</td>
<td>5984</td>
<td>1474</td>
<td>342</td>
<td>16349</td>
<td>6745</td>
<td>885</td>
<td>9903</td>
<td>5505</td>
</tr>
<tr>
<td>Genus</td>
<td>Pseudomonas</td>
<td>0.88 ± 0.02</td>
<td>1113</td>
<td>6572</td>
<td>1554</td>
<td>646</td>
<td>19387</td>
<td>7444</td>
<td>1453</td>
<td>12204</td>
<td>4838</td>
</tr>
<tr>
<td>Genus</td>
<td>Streptococcus</td>
<td>0.87 ± 0.02</td>
<td>535</td>
<td>3435</td>
<td>845</td>
<td>244</td>
<td>8265</td>
<td>3276</td>
<td>716</td>
<td>4468</td>
<td>2116</td>
</tr>
<tr>
<td>Family</td>
<td>Enterobacteriaceae</td>
<td>0.91 ± 0.04</td>
<td>146</td>
<td>6690</td>
<td>1664</td>
<td>20</td>
<td>19590</td>
<td>8173</td>
<td>197</td>
<td>10899</td>
<td>6715</td>
</tr>
<tr>
<td>Order</td>
<td>Corynebacteriales</td>
<td>0.83 ± 0.05</td>
<td>475</td>
<td>6022</td>
<td>1719</td>
<td>130</td>
<td>22558</td>
<td>10554</td>
<td>605</td>
<td>12632</td>
<td>9087</td>
</tr>
<tr>
<td>Phylum</td>
<td>Cyanobacteria</td>
<td>0.77 ± 0.04</td>
<td>400</td>
<td>9752</td>
<td>4428</td>
<td>120</td>
<td>27421</td>
<td>16140</td>
<td>511</td>
<td>10575</td>
<td>11154</td>
</tr>
</tbody>
</table>
The ratio between the size of the core-genome (clusters with persistence of 1, i.e. present in all genomes) and the number of singletons decreased with evolutionary distance (see Table 1). It ranged from 3.51 and 3.07 at species level (H. pylori and L. monocytogenes respectively) to 0.05 and 0.06 when considering members of the same order (Corynebacteriales) and phylum (Cyanobacteria) respectively. A similar pattern is observed when directly comparing the sizes of the pan- and core- genomes of the sampled genomes. Within the gram negative bacteria this ratio ranges from 0.69 for members of the same species (H. pylori) to 0.05 for members of the same phylum (Cyanobacteria) with intermediate values (0.12) for sequences from the same genus (Pseudomonas).

Cluster formation based on domain architectures
Domain architectures directly rely on the definition of protein domain models; those were retrieved from Pfam, InterPro and TIGRFAMs databases. However, TIGRFAMs results were not further considered because of a lower coverage. As shown in Table 1, as expected partly overlapping results were obtained when different domain databases were used. The number of singletons was larger when using InterPro rather than Pfam and for the latter we also observed larger core-genome size. These discrepancies can be due to the fact InterPro aggregates different resources (including Pfam and TIGRFAMs) and domain signatures arising from different databases are integrated with different identifiers in InterPro. In light of this we focused on results obtained using Pfam whose current release (30.0) contains hidden Markov models for over 16300 domain families. Size and persistence of groups of functionally equivalent proteins obtained using Pfam domains are presented in Figure 4.

Similar to what has been observed in the SB case we observed a decrease of the ratio between the size of the core genome and the number of singletons when higher taxonomic levels are considered. For organisms of the same species (H. pylori and L. monocytogenes) the ratio was 5.09 and 4.30, respectively, while for member of the same order (Corynebacteriales) and phylum (Cyanobacteria) it was 0.55 and 0.009 respectively. Similarly, also the ratio between the size of the core- and pan-genome decreases as higher taxonomic levels are considered, ranging from 0.54 for H. pylori to 0.04 for Cyanobacteria.

**Figure 3.** Persistence of sequence similarity based (SB) clusters. Cluster persistence is defined as the relative number of genomes with at least one protein assigned to the cluster. The frequency of SB clusters according to their persistence is shown.
Comparison of DAB and SB clusters

We compared the clusters obtained using both approaches and the proteins assigned to them. The number of one-to-one relationships (indicating a complete agreement) between SB and DAB clusters is indicated in Table 2 and ranges from 648 (for *H. pylori*) to 1680 (in *Pseudomonas*) corresponding to 50% and 25% of the pan-genome. This indicates that results of SB and DAB clustering tend to be more similar when working at closer phylogenetic distances. However, more complicated cases occur when proteins in a single SB cluster are assigned to various DAB clusters including singletons and vice versa. An overview of the possible mismatches between SB and DAB clusters is in Figure 5. The observed frequency of the different types of cluster mismatches are given in Figures 6 (for SB clusters per DAB cluster) and Figure 7 (for DAB clusters per SB cluster). For *L. monocytogenes* we found 378 1d → 1s DAB cluster mismatches, (Figure 5, panel A, top

![Image](image.png)

**Figure 4. Persistence of domain architecture based (DAB) clusters.** The frequency of DAB clusters according to their persistence is shown.

**Table 2. Number of identical clusters found with SB and DAB.**

<table>
<thead>
<tr>
<th>Group</th>
<th>Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>H. pylori</em></td>
<td>648</td>
</tr>
<tr>
<td><em>L. monocytogenes</em></td>
<td>1085</td>
</tr>
<tr>
<td><em>Bacillus</em></td>
<td>1439</td>
</tr>
<tr>
<td><em>Pseudomonas</em></td>
<td>1680</td>
</tr>
<tr>
<td><em>Streptococcus</em></td>
<td>961</td>
</tr>
<tr>
<td>Enterobacteriaceae</td>
<td>1649</td>
</tr>
<tr>
<td>Corynebacteriales</td>
<td>1034</td>
</tr>
<tr>
<td>Cyanobacteria</td>
<td>1127</td>
</tr>
</tbody>
</table>
Figure 5. Summary of possible mismatches between DAB and SB clusters. Mismatches of SB and DAB derived clusters (marked by $S$ and $D$ respectively) can occur in two directions. Panel A: possible cases of mismatch when counting the number of SB clusters the sequences in a DAB cluster are assigned to. $1d \rightarrow 1s$ denotes that all sequences from the D cluster are assigned to the same S cluster. $1d \rightarrow Ns$ denotes that sequences in a single D cluster are assigned to $N$ distinct S clusters with $N \geq 1$. Similarly, (panel B) $1s \rightarrow Nd$ denotes that sequences in a single S cluster are assigned to $N$ distinct D clusters with $N \geq 1$.

Figure 6. Comparison between DAB and SB clusters using DAB as reference. Each bar represents the relative frequency of one DAB cluster containing sequences assigned to $\{1, 2, \ldots, 5\}$ and 6 or more SB clusters. Axis labels follow notation in Figure 5.
Figure 7. Comparison between DAB and SB clusters using SB as reference. Each bar represents the relative frequency of one SB cluster containing sequences assigned to \{1, 2, \cdots, 5\} and 6 or more DAB clusters. Axis labels follow notation in Figure 5.

This is not surprising considering that such generic functions are usually associated with a high sequence diversity. Conversely, ABC transporters are found in multiple DAB clusters. However, many of them are grouped into a single SB cluster with ATPase domain containing proteins (\(1s \rightarrow Nd\) case).

We observed distinct architectures with one of two very similar domains, the GDSL-like Lipase/Acylhydrolase and the GDSL-like Lipase/Acylhydrolase family domain (PF00657 and PF13472 respectively) and those architectures were often seen clustered using a SB approach. However, architectures containing both domains were also identified, pointing to a degree of functional difference as a result of convergent or divergent evolution. Still, the corresponding sequences remain similar enough as to be indistinguishable when a SB approach is used.

For SB clustering we also observed the case of identical protein sequences not clustered together, probably because of the tie breaking implementation when BBH are scored.
In all cases we found the size of both the pan- and the core-genome to be larger when a SB approach is used to identify gene clusters and SB approaches lead to a larger number of singletons than DAB ones. This indicates that DAB clusters are assigned to several SB clusters, many of them consisting of just one protein.

When going from species to phylum level, the ratio between the number of DAB and SB singletons changes from 0.48 and 0.41 (for \textit{H. pylori} and \textit{L. monocytogenes} respectively) to 0.19 and 0.40 when considering organisms of a higher taxonomic level (Corynebacteria and Cyanobacteria respectively).

We investigated the predicted size of the pan-genome upon addition of new sequences. Heaps’ law regression can be used to estimate whether the pan-genome is open or closed\cite{Heaps1968} through the fitting of the decay parameter $\alpha$; $\alpha < 1$ indicates openness of the pan-genome (indicating that possibly many clusters remain to be identified within the considered set of sequences), while $\alpha > 1$ indicates a closed one; the $\alpha$ values are given in Table 3. In all cases the pan-genome is predicted to be open; however, $\alpha$ values obtained using DAB clusters ($\alpha_{\text{DAB}}$) are systematically closer to one than the $\alpha_{\text{SB}}$ obtained with the standard sequence similarity approach.

DAB comparison across multiple taxa
DAB clusters can be labeled by their domain architecture and since this is a formal description of functional equivalence, results of independently obtained analyses can be combined. Figure 8 shows

![Figure 8. Large scale functional comparison of species.](image)

Principal component analysis of functional similarities of 446 genomes based on the presence/absence of domain architectures on the corresponding genomes. The variance explained by the first two components is indicated on axes labels.

<table>
<thead>
<tr>
<th>Group</th>
<th>$\alpha_{\text{DAB}}$</th>
<th>$\alpha_{\text{SB}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{H. pylori}</td>
<td>0.95</td>
<td>0.42</td>
</tr>
<tr>
<td>\textit{L. monocytogenes}</td>
<td>0.77</td>
<td>0.50</td>
</tr>
<tr>
<td>\textit{Bacillus}</td>
<td>0.93</td>
<td>0.59</td>
</tr>
<tr>
<td>\textit{Pseudomonas}</td>
<td>0.94</td>
<td>0.61</td>
</tr>
<tr>
<td>\textit{Streptococcus}</td>
<td>0.87</td>
<td>0.72</td>
</tr>
<tr>
<td>\textit{Enterobacteriaceae}</td>
<td>0.99</td>
<td>0.74</td>
</tr>
<tr>
<td>\textit{Cyanobacteria}</td>
<td>0.64</td>
<td>0.58</td>
</tr>
<tr>
<td>\textit{Corynebacteriales}</td>
<td>0.88</td>
<td>0.52</td>
</tr>
</tbody>
</table>

$\alpha < 1$ indicates an open pan-genome
the results of a principal component analysis of the combined DAB clusters for selected genomes from seven taxa. The first two components account for a relatively low explained variance (29%) still grouping of genomes from the same taxa is apparent. High functional similarity among genomes of the same species (H. pylori and L. monocytogenes) is reflected by the compact clustering, while phylogenetically more distant genomes appear scattered in the functional space defined by the principal components.

Discussion

We have shown that domain architecture-based methods can be used as an effective approach to identify clusters of functionally equivalent proteins, leading to results similar to those obtained by classical methods based on sequence similarity. The DAB approach takes advantage of the large computational effort that has already been devoted to the identification and definition of protein domains in dedicated databases such as Pfam. Protein domain models are built using large scale sequence comparisons which is an extremely computationally intensive task. However, once the domain models are defined, mining a sequence for domain occurrences is much less demanding task. Indeed, the task with the higher computational load (the definition of the domains) is performed only once and results can be stored and re-used for further analysis. This provides an effective scalable approach for large scale functional comparisons which by and large is independent of phylogenetic distances between species. The chosen set of domain models and the database used as a reference greatly impact the results. Highly redundant databases, such as InterPro increase the granularity of the obtained clusters, biasing downwards and upwards the size of the pan- and core-genome sizes, respectively as shown in Table 1. The changes in the size of core- or pan-genome and their relationship with the evolutionary distance does not depend on the particular method considered and decrease of the ratio between core- and pan-genome sizes is observed when phylogenetic distance increases.

In many cases a one-to-one correspondence could be established between DAB and SB clusters indicating that often the sequence can be used as a proxy for function. At first this may seem a trivial result but it has a profound implication: domain model databases (in this case Pfam) contain enough information, encoded by known domain models, to represent the quasi totality of biological functions, hence they would tend to occupy a similar niche or, at least, more similar than when comparing species from different taxa. As the origin of the horizontally acquired genes may vary for each organism an SB approach will correctly recognize the heterologous origin of the corresponding sequences and those will be assigned to singletons. However, the probabilistic hidden Markov models used for domain recognition are better at recognizing the functional similarity of the considered sequences and clusters them together.

Another indication of the relative impact of horizontal and vertical gene acquisition events is provided by the openness or closeness of the genome. Values for the decay parameter \( \alpha \) in Table 3 indicate a relatively large impact of horizontal gene transfer. Within the considered taxa we observed \( \alpha_{\text{DAB}} > \alpha_{\text{SB}} \), meaning that the sequence diversity is larger than the functional diversity: upon addition of new genomes to the sample the rate of addition of new sequence clusters appears higher than the rate of addition of new functions.

Conclusions

As protein domain databases have evolved to the point where DAB and SB approaches produce similar results in closely related organisms, the DAB approach provides a fast and efficient alternative to SB methods to identify groups of functionally equivalent/related proteins for comparative genome analysis. The lower computational cost of DAB approaches makes them the better choice for large scale comparisons involving hundreds of genomes. Highly redundant databases, such as InterPro, are best suited for domain based protein annotation, but are not effective for DAB
clustering if the goal is to identify clusters of functionally equivalent proteins. Currently Pfam is for this task a better alternative.

Differences between DAB and SB approaches increase when the goal is to study bacterial groups spanning wider evolutionary distances. The functional pan-genome is more closed in comparison to the sequence based pan-genome. Both methods have a distinct approach towards horizontally transferred genes, and the DAB approach has the potential to detect functional equivalence even when sequence similarities are low.

Complementing the standardly applied sequence similarity methods with a DAB approach pinpoints potential functional protein adaptations that may add to the overall fitness.

Data availability
List of genomes used for the analysis at different phylogenetic levels. The genomes are grouped per taxonomic lineage used in this study.

Bacillus
GCA_000523045 Bacillus subtilis BEST7003
GCA_000782835 Bacillus subtilis
GCA_000832885 Bacillus thuringiensis str. Al Hakam
GCA_000473425 Bacillus infantis NRRL B-14911
GCA_000832585 Bacillus anthracis
GCA_000590445 Bacillus pumilus
GCA_000831065 Bacillus bombyepticus
GCA_000833275 Bacillus anthracis str. Turkey32
GCA_000952895 Bacillus sp.
GCA_00029365 Bacillus sp. JS
GCA_000143605 Bacillus cereus biovar anthracis str. CI
GCA_000186745 Bacillus subtilis BSn5
GCA_000987825 Bacillus methylotrophicus
GCA_000706725 Bacillus lehensis G1
GCA_000815145 Bacillus sp. Pc3
GCA_000496285 Bacillus toyonensis BCT-7112
GCA_000742895 Bacillus mycoides
GCA_000169195 Bacillus coagulans 36D1
GCA_000835145 Bacillus amyloliquefaciens KHG19
GCA_000321395 Bacillus subtilis subsp. subtilis str. BSP1
GCA_000009045 Bacillus subtilis subsp. subtilis str. 168
GCA_000293765 Bacillus subtilis QB928
GCA_000025805 Bacillus megaterium DSM 319
GCA_000747345 Bacillus sp. X1(2014)
GCA_000833005 Bacillus amyloliquefaciens
GCA_000400885 Bacillus paralicheniformis ATCC 9945a
GCA_000742895 Bacillus anthracis str. Vollum
GCA_000829195 Bacillus sp. OxB-1
GCA_000800825 Bacillus sp. WP8
GCA_000706705 Bacillus subtilis subsp. subtilis str. OH 131.1
GCA_000338735 Bacillus subtilis XF-1
GCA_000832445 Bacillus anthracis
GCA_000747335 Bacillus anthracis
GCA_000008505 Bacillus thuringiensis serovar konkukian str. 97-27
GCA_000195515 Bacillus amyloliquefaciens TA208
GCA_000209795 Bacillus subtilis subsp. natto BEST195
GCA_000017425 Bacillus cytotoxicus NVH 391-98
GCA_000877815 Bacillus sp. YP1
GCA_000177235 Bacillus cellulosilyticus DSM 2522
GCA_000344745 Bacillus subtilis subsp. subtilis 6051- HGW
GCA_000227485 Bacillus subtilis subsp. subtilis str. RO-NN-1
GCA_000494835 Bacillus amyloliquefaciens CC178
GCA_000011145 Bacillus halodurans C-125
GCA_000724485 Bacillus methanolicus MGA3
GCA_000018825 Bacillus weihenstephanensis KBAB4
GCA_000005825 Bacillus pseudefimurus OF4
GCA_000178855 Bacillus pumilus SAFR-032
GCA_000583065 Bacillus methylotrophicus Trigo-Cor1448
GCA_000349795 Bacillus subtilis subsp. subtilis str. BAB-1
GCA_000306745 Bacillus thuringiensis Bt407
GCA_000011645 Bacillus licheniformis DSM 13 = ATCC 14580
GCA_000497485 Bacillus subtilis PY79
GCA_000009825 Bacillus clausii KSM-K16
GCA_000227465 Bacillus subtilis subsp. spizizenii TU-B-10
GCA_000971925 Bacillus subtilis KCTC 1028
GCA_000972245 Bacillus endophyticus
GCA_000242895 Bacillus sp. INLA3E
GCA_000832485 Bacillus thuringiensis
GCA_000830075 Bacillus atropheae
GCA_000146565 Bacillus subtilis subsp. spizizenii str. W23

Corynebacteriales
GCA_000016005 Mycobacterium sp. JLS
GCA_000758405 Mycobacterium abscessus subsp. bolletii
GCA_000283295 Mycobacterium smegmatis str. MC2 155
GCA_001021045 Corynebacterium testudinoris
GCA_000341345 Corynebacterium halotolerans YIM 70093 = DSM 44683
GCA_000525655 Corynebacterium falsenii DSM 44353
GCA_000255195 Corynebacterium diptheriae HC04
GCA_000523235 Nocardia nova SH22a
GCA_000026685 Mycobacterium leprae Br4923
GCA_000980815 Corynebacterium camporealeensis
GCA_000328565 Mycobacterium sp. JS623
GCA_000015405 Mycobacterium sp. KMS
GCA_000987865 [Brevibacterium] flavum
GCA_001020985 Corynebacterium mustelae
GCA_001021065 Corynebacterium utequai
GCA_000177535 Corynebacterium resistens DSM 45100
Cyanobacteria

GCA_000011305 Corynebacterium efficiensYS-314
GCA_000035265 Mycobacterium avium subsp. paratuberculosis
GCA_000073945 Corynebacterium imitanus
GCA_0000831265 Mycobacterium kansasii662
GCA_0000819445 Corynebacterium humireducensNBRC106098=DSM45392
GCA_0000770235 Mycobacterium avium subsp. avium
GCA_0000980835 Corynebacterium kutscheri
GCA_0000102255 Corynebacterium glutamicumR
GCA_000590555 Corynebacterium argentoratenseDSM44202
GCA_000247715 Gordonia polyisopenivoransVAH2
GCA_000416365 Mycobacteriumsp.VKMAc-1817D
GCA_000418365 CorynebacteriumterpenotubidumY-11
GCA_000092225 TsukamurellapaurometabolamDSM20162
GCA_000442645 Corynebacterium marisDSM45190
GCA_000277125 Mycobacterium intracellulareATCC13950
GCA_000196695 Rhodococcus equi103S
GCA_000828995 Mycobacterium tuberculosisstr. Kurono
GCA_000066605 Corynebacterium jeikeiumK411
GCA_000022905 Corynebacterium aurimucosum
GCA_001021025 Corynebacterium epidermidicans
GCA_000010105 Rhodococcus erythropolisPR4
GCA_000092825 SegniliparusrotundusDSM44985
GCA_000758245 Mycobacterium bovis
GCA_000184435 MycobacteriumgilvumSpyr1
GCA_000282905 Mycobacterium avium subsp. hominisuisTH135
GCA_000214175 Amycolicucocus sulbflavus DQS3-9A1
GCA_000769635 Corynebacterium ulcerans
GCA_000626675 Corynebacterium glyciniphilum AJ3170
GCA_001026945 Corynebacterium pseudotuberculosis
GCA_000026445 Mycobacterium lilandii128FXT
GCA_000013925 Mycobacterium ulceransAg99
GCA_000954115 Rhodococcus sp.B7740
GCA_0000143885 Gordonia sp. KTR9
GCA_000014565 Rhodococcusjostii RHA1
GCA_000173939 Corynebacterium variabileDSM44702
GCA_000732945 Corynebacterium atypicum
GCA_000723425 Mycobacterium marinumE11
GCA_000230895 Mycobacterium rhodesiaeNBB3
GCA_000344785 Corynebacterium callunaeDSM 20147
GCA_000010805 Rhodococcus opacusB4
GCA_0009982715 Rhodococcus aethereovirans
GCA_000298095 Mycobacterium indicuspranii MTCC9506
GCA_000833575 Corynebacterium singulare
GCA_000231455 Corynebacterium kroppenstedtiiDSM 44385

Cyanobacteria

GCA_000317085 Synechococcus sp. PCC 7502
GCA_000011385 Gloeobacter violaceus PCC 7421

GCA_000014585 Synechococcus sp. CC9311
GCA_000012465 Prochlorococcus marinus str. NATL2A
GCA_000737535 Synechococcus sp. KORDI-100
GCA_000013205 Synechococcus sp. JA-3-3Ab
GCA_000021825 Cyanobethe sp. PCC 7424
GCA_000063505 Synechococcus sp. WH 7803
GCA_000022045 Cyanobethe sp. PCC 7425
GCA_000316575 Calothrix sp. PCC 7507
GCA_000316685 Synechococcus sp. PCC 6312
GCA_000012505 Synechococcus sp. CC9902
GCA_000317475 Oscillatoria nigro-viridis PCC 7112
GCA_000063525 Synechococcus sp. RCC307
GCA_000317695 Anaebena cylindrica PCC 7122
GCA_000014265 Trichodesmium erythraeum IMS101
GCA_000817325 Synechococcus sp. UTEX 2973
GCA_000737575 Synechococcus sp. KORDI-49
GCA_000317125 Chroococcidiopsis thermalis PCC 7203
GCA_000017845 Cyanobethe sp. ATCC 51142
GCA_000020025 Nostoc punctiforme PCC 73102
GCA_000018105 Acaryochloris marina MBC11017
GCA_000757865 Prochlorococcus sp. MIT 0801
GCA_000317045 Geitlerinema sp. PCC 7407
GCA_000012625 Synechococcus sp. CC9605
GCA_000737595 Synechococcus sp. KORDI-52
GCA_000317635 Halothece sp. PCC 7418
GCA_000025125 Candidatus Atelocyanobacterium thalassaisolate ALOHA
GCA_000010625 Microcystis aeruginosanIES-843
GCA_000317065 Pseudanabaenasp. PCC 7367
GCA_000312705 Anabaenasp. 90
GCA_000316515 Cyanobium gracile PCC 6307
GCA_000316605 Leptolyngbyasp. PCC 7376
GCA_000317025 Pleurocapsasp. PCC 7327
GCA_000009705 Nostocsp. PCC 7120
GCA_000013225 Synechococcus sp. JA-2-3B’(2-13)
GCA_000757845 Prochlorococcus sp. MIT 0604
GCA_000317515 Microcoleus sp. PCC 7113
GCA_000734895 Calothrixsp. 336/3
GCA_000007925 Prochlorococcus marinussubsp. marinuss str. CCMP1375
GCA_000021805 Cyanobethe sp. PCC 8801
GCA_000019485 Synechococcus sp. PCC 7002
GCA_000317655 Cyanobacterium stanieri PCC 7202
GCA_000316625 Nostocsp. PCC 7107
GCA_000011465 Prochlorococcus marinus subsp. pastoris str. CCMP1986
GCA_000316665 Rivulariasp. PCC 7116
GCA_000317105 Oscillatoria acuminate PCC 6304
GCA_000317435 Calothrixsp. PCC 6303
GCA_000317555 Gloeocapsasp. PCC 7428
Enterobacteriaceae

GCA_000259175 Providencia stuartii MRSN 2154
GCA_000214805 Serratia sp. AS13
GCA_000330865 Serratia marcescens FG194
GCA_001010285 Photorhabdus temperata subsp. thracensis
GCA_000364725 Candidatus Moranella endobia PCVAL
GCA_000251255 Buchnera aphidicola subsp. USDA (Myzus persicae)
GCA_000174055 Candidatus Sodalis pierantonius subsp. SOPE
GCA_0001012005 Shigella dysenteriae Sd197
GCA_000196475 Photobacterium phosphorum
GCA_000750295 Salmonella enterica subsp. enterica serovar Enteritidis
GCA_000007885 Yersinia pestis biovar Microtus str. 91001
GCA_000739495 Klebsiella pneumoniae
GCA_000252995 Salmonella bongori NCTC 12419
GCA_000270125 Pantoea ananatis AJ13355
GCA_000215745 Enterobacter aerogenes KCTC 2190
GCA_000092525 Shigella sonnei S046
GCA_000020865 Edwardsiella tarda EIB202
GCA_000023545 Dickeya dadantii Ech703
GCA_000238975 Serratia symbiotica subsp. ‘Cinara cedri’
GCA_000975245 Serratia liquefaciens
GCA_000006645 Yersinia pestis KIM10+
GCA_000224675 Enterobacter asburiae LF7a
GCA_000007405 Shigella flexneri 2a subsp. 2457T
GCA_001022275 Citrobacter freundii
GCA_000963575 Klebsiella michiganensis
GCA_000504545 Cronobacter sakazakii CMCC 45402
GCA_000012025 Shigella boydii Sb227
GCA_000814125 Enterobacter cloacae
GCA_000987925 Yersinia enterocolitica
GCA_000117455 Candidatus Blochmannia pennsylvanicus subsp. BPEN
GCA_000255535 Rahmella aquatilis HX2
GCA_000952955 Escherichia coli
GCA_000695995 Serratia sp. FS14
GCA_000648515 Citrobacter freundii CFNIH1
GCA_001022295 Klebsiella oxytoca
GCA_000147055 Dickeya dadantii 3937

Helicobacter pylori

GCA_000148855 Helicobacter pylori SJM180
GCA_000021165 Helicobacter pylori G27
GCA_000185245 Helicobacter pylori SouthAfrica7
GCA_000093185 Helicobacter pylori v225d
GCA_000277365 Helicobacter pylori Shi417
GCA_000498315 Helicobacter pylori BM012A
GCA_000270065 Helicobacter pylori F57
GCA_000392455 Helicobacter pylori UM032
GCA_000277385 Helicobacter pylori Shi169
GCA_000085255 Helicobacter pylori 26695
GCA_000270045 Helicobacter pylori F32
GCA_000148915 Helicobacter pylori Sat464
GCA_000185225 Helicobacter pylori Lithuania75
GCA_000600045 Helicobacter pylori oki102
GCA_000600205 Helicobacter pylori oki828
GCA_000192335 Helicobacter pylori 2018
GCA_000827025 Helicobacter pylori
GCA_000590775 Helicobacter pylori SouthAfrica20
GCA_000270025 Helicobacter pylori F30
GCA_000148665 Helicobacter pylori 908
Listeria monocytogenes

Pseudomonas

GCA_000392515 Helicobacter pylori UM037
GCA_000392475 Helicobacter pylori UM299
GCA_000262655 Helicobacter pylori XZZ74
GCA_000008785 Helicobacter pylori J99
GCA_000685745 Helicobacter pylori
GCA_000185205 Helicobacter pylori Gambia94/24
GCA_000826985 Helicobacter pylori 26695-1
GCA_000315955 Helicobacter pylori Aklavik117
GCA_000498335 Helicobacter pylori BM012S
GCA_000277405 Helicobacter pylori Shi112
GCA_000224535 Helicobacter pylori Puno120
GCA_000317875 Helicobacter pylori Aklavik86
GCA_000600185 Helicobacter pylori oki673
GCA_000196755 Helicobacter pylori B8
GCA_000439295 Helicobacter pylori UM298
GCA_000348885 Helicobacter pylori OK310
GCA_000307795 Helicobacter pylori 26695
GCA_000013245 Helicobacter pylori HPAG1
GCA_000392535 Helicobacter pylori UM066
GCA_000185185 Helicobacter pylori India7
GCA_000213135 Helicobacter pylori 83
GCA_000685705 Helicobacter pylori
GCA_000224575 Helicobacter pylori SNT49
GCA_000600085 Helicobacter pylori oki112
GCA_000023805 Helicobacter pylori 52
GCA_000348865 Helicobacter pylori OK113
GCA_000259235 Helicobacter pylori HUP-B14
GCA_000020245 Helicobacter pylori Shi470
GCA_000270005 Helicobacter pylori F16
GCA_000192315 Helicobacter pylori 2017
GCA_000685665 Helicobacter pylori
GCA_000600165 Helicobacter pylori oki422
GCA_000255955 Helicobacter pylori ELS37
GCA_000021465 Helicobacter pylori P12
GCA_000600145 Helicobacter pylori oki154
GCA_000224555 Helicobacter pylori Puno135
GCA_000011725 Helicobacter pylori 51
GCA_000148895 Helicobacter pylori Cuz20
GCA_000817025 Helicobacter pylori
GCA_000178935 Helicobacter pylori 35A

Listeria monocytogenes

GCA_000438745 Listeria monocytogenes
GCA_000438705 Listeria monocytogenes
GCA_001027125 Listeria monocytogenes
GCA_000438725 Listeria monocytogenes
GCA_000197755 Listeria monocytogenes
GCA_001027245 Listeria monocytogenes
GCA_001027085 Listeria monocytogenes
GCA_001005925 Listeria monocytogenes
GCA_000746625 Listeria monocytogenes
GCA_000382925 Listeria monocytogenes
GCA_000438665 Listeria monocytogenes
GCA_000438625 Listeria monocytogenes
GCA_000382925 Listeria monocytogenes
GCA_001027165 Listeria monocytogenes
GCA_000438605 Listeria monocytogenes
GCA_0000438585 Listeria monocytogenes
GCA_000808055 Listeria monocytogenes
GCA_000950775 Listeria monocytogenes
GCA_001027065 Listeria monocytogenes
GCA_000600165 Listeria monocytogenes
GCA_001005925 Listeria monocytogenes
GCA_000438685 Listeria monocytogenes
GCA_001005985 Listeria monocytogenes
GCA_000438625 Listeria monocytogenes
GCA_000681515 Listeria monocytogenes
GCA_000438645 Listeria monocytogenes
GCA_000210815 Listeria monocytogenes

Pseudomonas

GCA_000829885 Pseudomonas aeruginosa
GCA_000510285 Pseudomonas monteilii SB3078
GCA_000988485 Pseudomonas syringae pv. syringae B301D
GCA_000013785 Pseudomonas stutzeri A1501
GCA_000759535 Pseudomonas cremoricolorata
GCA_000953455 Pseudomonas pseudoalcaligenes
GCA_000981825 Pseudomonas aeruginosa
GCA_000661915 Pseudomonas stutzeri
GCA_000508205 Pseudomonas sp. TKP
GCA_000014625 Pseudomonas aeruginosa UCBPP-PA14
GCA_000019445 Pseudomonas putida W619
GCA_000316175 Pseudomonas sp. UW4
GCA_000498975 Pseudomonas mossellii SJ10
GCA_000473745 Pseudomonas aeruginosa VRPFA04
GCA_000691565 Pseudomonas putida
GCA_000730425 Pseudomonas fluorescens
GCA_000007805 Pseudomonas syringae pv. tomato str. DC3000
GCA_000349845 Pseudomonas denitrificans ATCC 13867
GCA_000026105 Pseudomonas entomophila L48
GCA_000689415 Pseudomonas knackmussii B13
GCA_000325725 Pseudomonas putida HB3267
GCA_000412695 Pseudomonas resinovorans NBRC 106553
GCA_000831585 Pseudomonas plecoglossicida
GCA_000756775 Pseudomonas sp. 20_BN
GCA_000590475 Pseudomonas stutzeri
GCA_000829255 Pseudomonas aeruginosa
**Streptococcus**

*GCA\_000211015* Streptococcus pneumoniae SPN034183  
*GCA\_000210975* Streptococcus pneumoniae ATCC BAA-2069  
*GCA\_000203195* Streptococcus galolyticus subsp. galolyticus ATCC BAA-2069  
*GCA\_0001020185* Streptococcus pyogenes  
*GCA\_000253155* Streptococcus oralis U05  
*GCA\_000696505* Streptococcus equi subsp. zooepidemicus CY  
*GCA\_000463355* Streptococcus intermedius B196  
*GCA\_000698888* Streptococcus thermophilus ASCC 1275  
*GCA\_000014205* Streptococcus sanguinis SK36  
*GCA\_000007045* Streptococcus pneumoniae R6  
*GCA\_000306805* Streptococcus intermedius JTH08  
*GCA\_000196595* Streptococcus pneumoniae TCH8431/19A

*GCA\_000262145* Streptococcus parasanguinis FW213  
*GCA\_001026925* Streptococcus agalactiae  
*GCA\_000251085* Streptococcus pneumoniae ST556  
*GCA\_000019025* Streptococcus pneumoniae Taiwan19F-14  
*GCA\_000211055* Streptococcus pneumoniae SPN994039  
*GCA\_000688775* Streptococcus sp. VT 162  
*GCA\_000231905* Streptococcus suis D12  
*GCA\_000026665* Streptococcus pneumoniae ATCC 700669  
*GCA\_000283635* Streptococcus macedonicus ACA-DC 198  
*GCA\_000014365* Streptococcus pneumoniae D39  
*GCA\_000019265* Streptococcus pneumoniae Hungary19A-6  
*GCA\_000299015* Streptococcus pneumoniae gamPN0373  
*GCA\_000019985* Streptococcus pneumoniae CGSP14  
*GCA\_000463395* Streptococcus constellatus subsp. pharyngis C232  
*GCA\_000187935* Streptococcus parauberis NCFD 2020  
*GCA\_000253315* Streptococcus salivarius JIM8777  
*GCA\_000427055* Streptococcus pneumoniae ILRI112  
*GCA\_000246835* Streptococcus infantarius subsp. infantarius CJ18  
*GCA\_000427075* Streptococcus pneumoniae ILRI005  
*GCA\_000007465* Streptococcus mutans UA159  
*GCA\_000831165* Streptococcus anginosus  
*GCA\_000147095* Streptococcus pneumoniae 670-6B  
*GCA\_000817005* Streptococcus pneumoniae  
*GCA\_000180515* Streptococcus pneumoniae SPNA45  
*GCA\_000441535* Streptococcus lutetiensis 033  
*GCA\_000210955* Streptococcus pneumoniae OXC141  
*GCA\_000009545* Streptococcus uberis 0140J  
*GCA\_000648555* Streptococcus iniae  
*GCA\_000027165* Streptococcus mitis B6  
*GCA\_000189895* Streptococcus pneumoniae JJA  
*GCA\_000270165* Streptococcus pasteurianus ATCC 43144  
*GCA\_000479315* Streptococcus sp. I-P16  
*GCA\_000478925* Streptococcus anginosus subsp. whileyi MAS624  
*GCA\_000199825* Streptococcus pneumoniae G54  
*GCA\_000017005* Streptococcus gordonii str. Challis substr. CH1  
*GCA\_000479335* Streptococcus sp. L-G2  
*GCA\_000385925* Streptococcus oligofermentans AS 1.3089  
*GCA\_000210935* Streptococcus pneumoniae INV200  
*GCA\_000211035* Streptococcus pneumoniae SPN994038  
*GCA\_000221985* Streptococcus pseudopneumoniae IS7493  
*GCA\_000006885* Streptococcus pneumoniae TIGR4  
*GCA\_000018965* Streptococcus pneumoniae 70585  
*GCA\_000348705* Streptococcus pneumoniae PCS8235  
*GCA\_000210995* Streptococcus pneumoniae SPN034156  
*GCA\_000231925* Streptococcus suis ST1  
*GCA\_000019005* Streptococcus pneumoniae P1031  
*GCA\_000188715* Streptococcus dysgalactiae subsp. equisimilis ATCC 12394  
*GCA\_000026585* Streptococcus equi subsp. equi 4047
Author contributions
JJK, MSD, ES, PJS participated in the set-up of the research. JJK and MSD were responsible for the analysis. JJK, ES, PJS, MSD and VdS wrote the manuscript. All authors critically revised the manuscript.

Competing interests
No competing interests were disclosed.

Grant information
This work was partly supported by the European Union’s Horizon 2020 research and innovation programme (EmPower-Putida, Contract No. 635536, granted to Vitor A P Martins dos Santos).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Supplementary material

Figure S1. Persistence of sequence similarity based (SSB) clusters. Cluster persistence is defined as the relative number of genomes with at least one protein assigned to the cluster. The plots show frequency of SSB clusters according to their persistence. Publicly available and complete genome sequences assigned to each taxon were selected so that phylogenetic diversity within the taxon was preserved, as described in materials and methods. 60 distinct genome sequences were considered for each taxon shown.
Figure S2. Persistence of domain architecture based (DAB) clusters. The plots show frequency of DAB clusters according to their persistence.
Figure S3. Comparison between DAB and SSB clusters using SSB as reference. Each bar represents the relative frequency of one SSB cluster containing sequences assigned to 1, 2, ..., 5 and 6 or more DAB clusters.
Figure S4. Comparison between DAB and SSB clusters using DAB as reference. Each bar represents the relative frequency of one DAB cluster containing sequences assigned to 1, 2, …, 5 and 6 or more SSB clusters.
References

The limitations of global sequence similarity based methods to identify proteins that perform similar functions are well-known. Thus, the approach described in this manuscript of using domain-based clustering of orthologous groups (DAB) represents an exciting and very welcome addition to the field. Or at least it will when it is fully developed, although this manuscript has not convinced me that it outperforms other methods at its current level of development, and I have several substantial reservations about some of its content:

- As the first reviewer also mentioned, methods such as CDART and DELTA-BLAST (published in 2002 and 2012, respectively) have been available for many years. The latter even seems to aim to perform the exact same function as DAB, by considering domain architectures. How is DAB different or better? I suspect that DAB may have greater accuracy since it uses HMMs rather than PSSMs, but this remains to be shown, and DELTA-BLAST is far easier for a user to run, since it is available as a webserver.

- The comparison performed in this manuscript appears to fall prey to a straw man argument. In some cases, but not all, re-writing the relevant sections of the manuscript would help to avoid any misconceptions in this regard.

a) The issue of replacing a $O(n^2)$ cost with a $O(n)$ one upon addition of a new genome was dealt with over 15 years ago, so the statement "On the other hand, addition of a new genome using an SB approach require a new set of all-against-all sequence comparisons which come at a $O(n^2)$ computational cost" is false - at least as it is currently written. It is true that building groups of orthologs do require an initial $O(n^2)$ computational cost, but once those orthologous groups are formed, methods such as COGNITOR (first published in the year 2000) work extremely quickly and efficiently to assign genes in newly-sequenced genomes to existing groups. In fact, COGNITOR works in the exact same manner in which DAB uses pre-computed domain databases to achieve the much lower $O(n)$ cost, although in COGNITOR’s case it searches against a pre-computed database of orthologous groups (of which there are far fewer than domains, so with a smaller "n" it would actually be faster than DAB).
In should be noted that despite DAB’s somewhat higher cost, it has the theoretical potential to achieve better accuracy than COGNITOR (at least in some cases) since as a global sequence similarity approach, the latter does not explicitly consider domain architecture. At least not in an automated fashion - doing so would require manual curation of its results, which is often done by careful researchers, but is not a process that is scalable to handle the ever-decreasing cost and ever-increasing amounts of genomic data. Although since a comparison with COGNITOR was not included in the manuscript, either in terms of speed or accuracy, it is unknown how much more useful DAB would be in practice.

b) Even the initial $O(n^2)$ cost does not have to be terribly burdensome, since the SIMAP method pre-computes and stores BLAST results between all pairs of sequenced genomes anyway, and then uses efficient database retrieval methods to report the stored results. When a new genome is added, $O(n)$ new comparisons have to be made - for a total accumulated cost of $O(n^2)$, although with the work spread out over many years - and these in turn are useful for many other purposes, thus mitigating the construction costs. For instance, the EGGNOG database uses this method to build groups of orthologs.

c) Why was only a single SB method chosen to be a representative for this entire class of approaches? Multiple forms of DAP were tested, whereas the only SB method used for comparison was one that uses a strict e-value cutoff of 1e-5, in the form of OrthaGogue and the OrthoMCL method. Also, why was the latter chosen to be this single representative? The latter approach was designed (nearly a decade and a half ago) for eukaryotic organisms, and while it has been applied more recently to bacteria as well, it is by no means the only - or even necessarily the best - approach for prokaryotic genomes. One advantage that it has is that it is completely automated, and thus is "easy" for people to use (even if, as this manuscript points out, horribly slow due to the $O(n^2)$ procedure that it uses). On the other hand, methods like CDART and COGTRIANGLES are all also automated (the latter of which uses no arbitrary e-value cutoff - that is, the results are robust to e-values over an immense range such as 1e-5, 1, 10, or even well beyond that on up to 100, or even 1000), and some pre-computed databases (such as COGs, representing the protein families present in the last common ancestor of all cellular life several billions of years ago) even take advantage of further manual validation, and from which pre-computed groups can be identified in newly-sequenced genomes by the fully automated and even easier approaches such as DELTA-BLAST and COGNITOR. Is it at least possible that the poorer performance of SB methods in comparison to DAB as shown in the current manuscript is due to the choice of this particular SB method? I for one would have loved to see a comparison against the new release of the COGs database last year, since due to its being manually curated it acts as a sort of "Gold Standard" that can be tested against, with perhaps the EGGNOG groups being used as a more realistic measure of what a purely automated method can do without human supervision. Likely, DAB would fall somewhere in-between, and which would benefit the community of researchers who want to do comparative genomics of prokaryotic organisms to have a fully automated method that was demonstrated to surpass the existing fully automated methods. As it now stands though, DAB has only been shown to surpass OrthoMCL, which is not hard to do at all. Indeed, as seventh paragraph of the Discussion section (starting "Two of the most prominent...") states, unlike DAB, the SB methods were not able to cluster together the proteins with functional similarity but little sequence identity, especially across wider taxonomic ranges - which of course is what would be expected from a SB method that uses an e-value cutoff of 1e-5.

d) Above and beyond the choice of SB method, it also seems that there may have been a bug in its implementation. The statement "For SB clustering we also observed the case of identical protein sequences not clustered together, probably because of the tie breaking implementation when BBH are scored." However, this was not supposed to happen, due to the within-species reciprocal BBH procedure that is used. In contrast, the tie breaking refers to between-species comparisons, but as shown in Figure 1
of the OrthoMCL paper (http://www.ncbi.nlm.nih.gov/pubmed/12952885), these two sources of information were supposed to have been combined together to form the final orthologous groups. If the proteins were highly similar (e.g., 99%) then perhaps a tie-breaking could be explained, but for 100% identical proteins - e.g., produced by a tandem duplication event - then they should have been collected into the group. One possibility is that this particular SB method simply was not designed to handle the large numbers of extremely closely-related genome assemblies that are available today, since at the time, very few instances of multiple genomic assemblies were available for the same species. If this explanation was demonstrated to be the reason why these identical proteins were not clustered together, that would be another reason for a user to choose to use DAB over this particular SB method. In any case (bug, design flaw, or something else), this event could greatly contribute to explaining some of the results that were observed whereby this single SB method found so many more singletons than DAB with Pfam - i.e., fixing the bug, or using some other SB method, may move many of those singletons into clusters. Although it would not explain why DAB with InterPro found even more singletons than this SB method?

- **DAB has a lot of potential, but its limitations need to be made more clear:**
  a) Why and how is the matrix of domain architecture binarized? Specifically, what if multiple copies of a domain are present? And does order matter - such as the architectures shown in Figure 2 of "A+B" and "B+A"? So, would "B+A+A" be a different architecture? And, as another reviewer also pointed out, what about "complicated" domain topologies where domains are interrupted by the insertion of another domain? Another major aspect of partial topologies is if DAB only recognizes some but not all of a newly-discovered architecture. E.g., a protein with architecture A+B+C+D, where A is known but B, C and D domains are not yet known. How would this be handled by DAB? Would it be reduced to appear merely as a single-domain "A" architecture? If so, how could that be distinguished from an architecture such as A+Z, which would also be reduced to appear just as a single-domain A? It seems like global sequence similarity methods might be more useful in those particular scenarios? i.e., if all the above domains were the same length, and a coverage threshold was used, then A+B+C+D could not be put into the same group as A+Z and A. Therefore, DAB seems primarily useful to either quickly extend known architectural types into a newly sequenced genome, but at the cost of not being able to work with unknown types.

b) For newly sequenced genomes that are not yet well-characterized enough to have all of their domains present in the domain databases, DAB can be severely handicapped in comparison to global sequence similarity methods that do not have this limitation. In particular, Table 1 shows that up to nearly a fifth of the *H. pylori* and Cornebacteriales genomes are not able to be assigned to domain families. Even these numbers are merely lower-bound estimates, since brand-new architectures are expected to be discovered constantly, and yet these may incorporate at least one element that is known - such as the aforementioned A+B+C+D architecture, where only the A domain is represented in Pfam, but B and C and D are unknown. And yet it seems likely that even the fact that these domains are unknown would go unrecognized by the DAB approach - unless a factor is added to look for large segments of a gene that do not have matches in the databases of known domains. Therefore, the cost of DAB not being able to work with unknown architectural types might be quite high indeed. Worse, the exact value of that cost is also likewise unknown, and yet it would seem to be the single crucial piece of information that is most sorely needed in order to answer the question: does the benefits of DAB outweigh its costs?

- **If the goal is to bring together groups of proteins that have functional equivalence, then why was the only comparison that was done performed against the presence/absence membership of SB orthology approaches? Would it not have been better to actually measure the functional consistency observed within the SB groups, and within the DAB groups, in order to show that the latter was higher than the former? Many other methods that purport to improve upon the state-of-the-art orthology prediction process do just that - for instance Figure 4 of**
http://www.ncbi.nlm.nih.gov/pubmed/19148271 shows several comparisons with similarity of GO terms, enzyme nomenclature (EC), gene expression, and syntenic local neighborhood tests, with 12 different methods of orthology prediction. While neighborhood conservation is irrelevant for the issue of functional equivalence, the former three (or at least GO terms) would help to answer whether DAB is truly better than SB at the task of measuring functional equivalence. It would also help to answer whether this improved functional equivalence would be outweighed by the costs of being unable to handle unknown domain architectures, especially for highly divergent new genomes. If not, DAB may still be useful to check the consistency of existing orthologous groups in terms of their architecture, at least when domain architectures are expected to be completely known in advance - e.g., microevolutionary variations within a species where mutational events may disrupt a protein's function - but for other tasks such as the discovery of a new phyla of cellular life that contains radically different domain architectures, global similarity methods may be preferable instead.

Finally, some minor points concerning Figure 2:

1. the vertical arrows seem to be pointing the wrong direction - a gene sequence undoubtedly contains more information content than a mere functional description. e.g., if I were to give you a GO code for molecular function, or biological process, then I could not tell you whether the original gene sequence is closer to one type of bacteria vs another type; but if I had the original gene sequence, then I could answer that question as well as many more.

2. I did not see a description of how amino acid coordinates are used anywhere else in the manuscript, either in DAB itself or in the comparison? In short, what does "Structure" have to do with anything, other than the general theoretical flow of "sequence begets structure which begets function"? If the purpose of Figure 2 is to describe the flowchart of DAB specifically though, it should focus only on the relevant elements. I suppose Structure could have meant how the sequence alignment was made, but if that were true, then DAB would only work for domain families for which a structure is available, instead of those for which only genomic or individual gene sequence has been provided.

3. The ordering also seems unclear - wouldn't BBHs inform HMM domains, which then in turn inform domain architectures? Or if starting with BBHs, then how could architectures possibly be known prior to knowing the domains themselves? Or if it should be read from top to bottom as shown, how exactly does one start with Function (e.g., a GO term) and then, somehow via Structure, thereby arrive at a Sequence alignment? Specifically, is a Pfam entry a "Function", from which the Sequence alignment is downloaded? Or are Function and the Sequence alignment both part of the starting Pfam entry (and then again, what does any of that have to do with Structure)? From which domains are found (but aren't Pfam entries domains to begin with?), and then BBHs are made from the domain architectures? (an extremely different way of doing the BBH procedure, which is normally done via Sequence alignments). In any case, as pointed out by other reviewers, the methodology used by DAB is not clearly explained in this figure, nor in the manuscript text.

Also, the last paragraph of the Discussion uses the word "closeness", but I think "closedness" was intended.

**Competing Interests:** No competing interests were disclosed.
I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 09 Nov 2016

Jasper Koehorst, Wageningen University and Research, Stippeneng, The Netherlands

1. As the first reviewer also mentioned, methods such as CDART and DELTA-BLAST (published in 2002 and 2012, respectively) have been available for many years. The latter even seems to aim to perform the exact same function as DAB, by considering domain architectures. How is DAB different or better? I suspect that DAB may have greater accuracy since it uses HMMs rather than PSSMs, but this remains to be shown, and DELTA-BLAST is far easier for a user to run, since it is available as a webserver.

Following the suggestions made by the other reviewers we have added a paragraph in the Introduction regarding domain architectures, comparison of domain architectures and their use for sequence search. We have also discussed on how these have been included in domain databases and on the preservation of domain architectures at high phylogenetic distances.

We agree that most likely HMMs outperform PSSMs, however as the reviewer says, that is a topic that would required a dedicated investigation. Here our goal was to use domain architectures for functional comparative genomics, and we agree that a similar approach could be implemented using PSSM.

Regarding usability, we have used SAPP (semantic annotation platform with provenance) for genome analysis and annotation. SAPP is able to store the results in the RDF data model, that can be then queried using SPARQL. This tool is available with a web interface and is available at http://semantics.systemsbiology.nl/

2. The comparison performed in this manuscript appears to fall prey to a straw man argument. In some cases, but not all, re-writing the relevant sections of the manuscript would help to avoid any misconceptions in this regard.

The issue of replacing a O(n²) cost with a O(n) one upon addition of a new genome was dealt with over 15 years ago, so the statement "On the other hand, addition of a new genome using an SB approach require a new set of all-against-all sequence comparisons which come at a O(n²) computational cost" is false - at least as it is currently written.

We have amended the above mentioned sentence to:

On the other hand, addition of a new genome using an SB approach require a new set of all-against-all sequence comparisons which come at a $O(n^2)$ computational cost. However, approaches has been proposed to overcome this shortcomings of SB methods, such as COGNI TOR which reduces the computational to $O(n)$ by using pre-computed databases.
3. It is true that building groups of orthologs do require an initial $O(n^2)$ computational cost, but once those orthologous groups are formed, methods such as COGNITOR (first published in the year 2000) work extremely quickly and efficiently to assign genes in newly-sequenced genomes to existing groups. In fact, COGNITOR works in the exact same manner in which DAB uses pre-computed domain databases to achieve the much lower $O(n)$ cost, although in COGNITOR's case it searches against a pre-computed database of orthologous groups (of which there are far fewer than domains, so with a smaller "n" it would actually be faster than DAB).

In should be noted that despite DAB's somewhat higher cost, it has the theoretical potential to achieve better accuracy than COGNITOR (at least in some cases) since as a global sequence similarity approach, the latter does not explicitly consider domain architecture. At least not in an automated fashion - doing so would require manual curation of its results, which is often done by careful researchers, but is not a process that is scalable to handle the ever-decreasing cost and ever-increasing amounts of genomic data.

**We have commented on the analogy between DAB and COGNITOR.**

In this respect, the DAB approach is similar in to the approach implemented in COGNITOR, by searching against existing databases of domains architectures.

4. Although since a comparison with COGNITOR was not included in the manuscript, either in terms of speed or accuracy, it is unknown how much more useful DAB would be in practice.

**The focus of the paper was not to propose a comparative analysis of different methods but rather to present and contextualize the use of domain architecture for comparative genomics. However, we want to stress that we are not claiming that DA methods are superior to SB but that are an efficient and scalable alternative.**

5. Even the initial $O(n^2)$ cost does not have to be terribly burdensome, since the SIMAP method pre-computes and stores BLAST results between all pairs of sequenced genomes anyway, and then uses efficient database retrieval methods to report the stored results. When a new genome is added, $O(n)$ new comparisons have to be made - for a total accumulated cost of $O(n^2)$, although with the work spread out over many years - and these in turn are useful for many other purposes, thus mitigating the construction costs. For instance, the EGGNOG database uses this method to build groups of orthologs.

Why was only a single SB method chosen to be a representative for this entire class of approaches? Multiple forms of DAB were tested, whereas the only SB method used for comparison was one that uses a strict e-value cutoff of 1e-5, in the form of OrthaGogue and the OrthoMCL method. Also, why was the latter chosen to be this single representative?

**We have added the following to the discussion:**

To assess whether DAB results were consistent with those of SB methods we chosen. OrthaGogue as a representative of the latter class. Several tools such as COGNITOR and MultiPARANOID are available that implement different algorithm solutions to the task of identifying homologous sequences; however, despite different implementation, they all rely on sequence similarity as a proxy for
functional equivalence. Here we considered SB methods as a golden standard for functional comparative genomics, especially when organisms within close evolutionary proximity are considered. Our aim was to investigate whether using HMMs instead of sequence similarity would yield similar results, thereby justifying their use for large scale functional genome comparisons. Regarding domain architectures, we have explored different alternatives, as we have seen that the chosen database or set of reference domains plays a critical role, an example is the low coverage of TIGRFAM preventing obtention of reasonable clusters.

6. The latter approach was designed (nearly a decade and a half ago) for eukaryotic organisms, and while it has been applied more recently to bacteria as well, it is by no means the only - or even necessarily the best - approach for prokaryotic genomes.

One advantage that it has is that it is completely automated, and thus is "easy" for people to use (even if, as this manuscript points out, horribly slow due to the O(n^2) procedure that it uses).

On the other hand, methods like CDART and COGTRIANGLES are all also automated (the latter of which uses no arbitrary e-value cutoff - that is, the results are robust to e-values over an immense range such as 1e-5, 1, 10, or even well beyond that on up to 100, or even 1000), and some pre-computed databases (such as COGs, representing the protein families present in the last common ancestor of all cellular life several billions of years ago) even take advantage of further manual validation, and from which pre-computed groups can be identified in newly-sequenced genomes by the fully automated and even easier approaches such as DELTA-BLAST and COGNITOR. Is it at least possible that the poorer performance of SB methods in comparison to DAB as shown in the current manuscript is due to the choice of this particular SB method? I for one would have loved to see a comparison against the new release of the COGs database last year, since due to its being manually curated it acts as a sort of "Gold Standard" that can be tested against, with perhaps the EGGNOG groups being used as a more realistic measure of what a purely automated method can do without human supervision. Likely, DAB would fall somewhere in-between, and which would benefit the community of researchers who want to do comparative genomics of prokaryotic organisms to have a fully automated method that was demonstrated to surpass the existing fully automated methods. As it now stands though, DAB has only been shown to surpass OrthoMCL, which is not hard to do at all. Indeed, as seventh paragraph of the Discussion section (starting "Two of the most prominent...") states, unlike DAB, the SB methods were not able to cluster together the proteins with functional similarity but little sequence identity, especially across wider taxonomic ranges - which of course is what would be expected from a SB method that uses an e-value cutoff of 1e-5.

Above and beyond the choice of SB method, it also seems that there may have been a bug in its implementation. The statement "For SB clustering we also observed the case of identical protein sequences not clustered together, probably because of the tie breaking implementation when BBH are scored." However, this was not supposed to happen, due to the within-species reciprocal BBH procedure that is used. In contrast, the tie breaking refers to between-species comparisons, but as shown in Figure 1 of the OrthoMCL paper (http://www.ncbi.nlm.nih.gov/pubmed/12952885), these two sources of information were supposed to have been combined together to form the final orthologous groups. If the
proteins were highly similar (e.g., 99%) then perhaps a tie-breaking could be explained, but for 100% identical proteins - e.g., produced by a tandem duplication event - then they should have been collected into the group. One possibility is that this particular SB method simply was not designed to handle the large numbers of extremely closely-related genome assemblies that are available today, since at the time, very few instances of multiple genomic assemblies were available for the same species. If this explanation was demonstrated to be the reason why these identical proteins were not clustered together, that would be another reason for a user to choose to use DAB over this particular SB method. In any case (bug, design flaw, or something else), this event could greatly contribute to explaining some of the results that were observed whereby this single SB method found so many more singletons than DAB with Pfam - i.e., fixing the bug, or using some other SB method, may move many of those singletons into clusters. Although it would not explain why DAB with InterPro found even more singletons than this SB method?

We have added a paragraph in the discussion regarding why the InterPro hierarchy has to be taken into account, also we mention this in the conclusion section. The hierarchical structure produces an increase in the domain multiplicity as many are related to each other. As a result, an artificial variability in the DA is introduced leading to a higher number of singletons.

7. DAB has a lot of potential, but its limitations need to be made more clear.

We have added a new section to the Discussion: Limitations of DAB approaches

8. Why and how is the matrix of domain architecture binarized? Specifically, what if multiple copies of a domain are present?

We understand that our phrasing may have caused some confusion and we apologize for unclarity. The matrix of domain architectures is only binarized (presence/absence) to compute the PCA shown in Fig. 8, not to compare DAB and SB clustering.

We have rephrased this in the Materials and Methods section:
...a binarized presence-absence matrix was obtained and used solely for principal component analysis.

9. […] does order matter - such as the architectures shown in Figure 2 of "A+B" and "B+A"? So, would "B+A+A" be a different architecture? And, as another reviewer also pointed out, what about "complicated" domain topologies where domains are interrupted by the insertion of another domain? Another major aspect of partial topologies is if DAB only recognizes some but not all of a newly-discovered architecture. E.g., a protein with architecture A+B+C+D, where A is known but B, C and D domains are not yet known. How would this be handled by DAB? Would it be reduced to appear merely as a single-domain "A" architecture? If so, how could that be distinguished from an architecture such as A+Z, which would also be reduced to appear just as a single-domain A? It seems like global sequence similarity methods might be more useful in those particular scenarios? i.e., if all the above domains were the same length, and a coverage threshold was used, then A+B+C+D could not be put into the same group as A+Z and A. Therefore, DAB seems primarily useful to either quickly extend known architectural types into a newly sequenced genome, but at the cost of not being able to work
with unknown types.

For newly sequenced genomes that are not yet well-characterized enough to have all of their domains present in the domain databases, DAB can be severely handicapped in comparison to global sequence similarity methods that do not have this limitation. In particular, Table 1 shows that up to nearly a fifth of the H. pylori and Cornebacteriales genomes are not able to be assigned to domain families. Even these numbers are merely lower-bound estimates, since brand-new architectures are expected to be discovered constantly, and yet these may incorporate at least one element that is known - such as the aforementioned A+B+C+D architecture, where only the A domain is represented in Pfam, but B and C and D are unknown. And yet it seems likely that even the fact that these domains are unknown would go unrecognized by the DAB approach - unless a factor is added to look for large segments of a gene that do not have matches in the databases of known domains. Therefore, the cost of DAB not being able to work with unknown architectural types might be quite high indeed. Worse, the exact value of that cost is also likewise unknown, and yet it would seem to be the single crucial piece of information that is most sorely needed in order to answer the question: does the benefits of DAB outweigh its costs?

The reviewer raises a very interesting point regarding how extensive available knowledge on protein domains is. The high agreement between the results of DAB and SB methods is only possible because databases of protein domains have enough information. Still, we believe many domains remain to be identified and in the scenarios the reviewer mentions DAB methods will be limited. We have added the following to the Discussion section, under the “Limitations of DAB approaches” header.

Still around 15% of the genome coding content corresponds to sequences with no identified protein domains. DAB approaches can be complemented with SB methods to consider these sequences or even protein sequences with low domain coverage, possible indicating the location of protein domains yet to be identified.

We have extended the paragraph in the Materials and Methods where domain architectures are defined to further emphasize that N- C- terminal domain order is an inherent part of domain architecture definition.

Labels indicating N-C terminal order of identified domains were assigned to each protein using the starting position of the domains: the same labels were assigned to proteins sharing the same domain architecture.

In the Introduction we have added a paragraph regarding the use of protein domain architecture in protein annotations and we have included references to previous works showing that domain order is often key for the function of the protein and that domain duplications/insertions can also alter the function of the protein.

Moreover, a similar point on how domain architectures were defined and how the hierarchical relationships between protein domains, families and clans has been raised by R. Finn and a paragraph has been added in the Discussion (see answer to R. Finn’s comments).
10. If the goal is to bring together groups of proteins that have functional equivalence, then why was the only comparison that was done performed against the presence/absence membership of SB orthology approaches? Would it not have been better to actually measure the functional consistency observed within the SB groups, and within the DAB groups, in order to show that the latter was higher than the former? Many other methods that purport to improve upon the state-of-the-art orthology prediction process do just that - for instance Figure 4 of http://www.ncbi.nlm.nih.gov/pubmed/19148271 shows several comparisons with similarity of GO terms, enzyme nomenclature (EC), gene expression, and syntenic local neighborhood tests, with 12 different methods of orthology prediction. While neighborhood conservation is irrelevant for the issue of functional equivalence, the former three (or at least GO terms) would help to answer whether DAB is truly better than SB at the task of measuring functional equivalence. It would also help to answer whether this improved functional equivalence would be outweighed by the costs of being unable to handle unknown domain architectures, especially for highly divergent new genomes. If not, DAB may still be useful to check the consistency of existing orthologous groups in terms of their architecture, at least when domain architectures are expected to be completely known in advance - e.g., microevolutionary variations within a species where mutational events may disrupt a protein's function - but for other tasks such as the discovery of a new phyla of cellular life that contains radically different domain architectures, global similarity methods may be preferable instead.

We have added the following section dedicated to limitations of DAB methods:

We have shown that domain architecture-based methods can be used as an effective approach to identify clusters of functionally equivalent proteins, leading to results similar to those obtained by classical methods based on sequence similarity. However, whether DAB methods are more accurate than SB methods to assess functional equivalence will require further analysis. In this light, results of functional conservation for both approaches could be compared in terms of GO similarity and/or EC number.

The performance of DAB methods may be sub-optimal when dealing with newly sequenced genomes that are not yet well-characterized enough to have all of their domains present in domain databases, since DAB methods will be unable to handle unknown architectural types.

Around 15% of the genome coding content corresponds to sequences with no identified protein domains. DAB approaches can be complemented with SB methods to consider these sequences or even protein sequences with low domain coverage, possible indicating the location of protein domains yet to be identified. Since DAB methods rely on the constant upgrading of public resources like UniProt and Pfam databases, an initial assessment of domain coverage appears as a sine qua non condition for application of these methods.

DAB approaches could be used to assess the consistency of existing orthologous groups in terms of their domain architectures, at least when domain architectures are expected to be completely known in advance (for instance in the case of...
micro-evolutionary variations within a species where mutational events may disrupt a protein's function). For other purposes, such as the discovery of a new phyla of cellular life that contains radically different domain architectures, global similarity methods may be preferred.

11. Finally, some minor points concerning Figure 2:
The vertical arrows seem to be pointing the wrong direction - a gene sequence undoubtedly contains more information content than a mere functional description. e.g., if I were to give you a GO code for molecular function, or biological process, then I could not tell you whether the original gene sequence is closer to one type of bacteria vs another type; but if I had the original gene sequence, then I could answer that question as well as many more.

I did not see a description of how amino acid coordinates are used anywhere else in the manuscript, either in DAB itself or in the comparison? In short, what does "Structure" have to do with anything, other than the general theoretical flow of "sequence begets structure which begets function"? If the purpose of Figure 2 is to describe the flowchart of DAB specifically though, it should focus only on the relevant elements. I suppose Structure could have meant how the sequence alignment was made, but if that were true, then DAB would only work for domain families for which a structure is available, instead of those for which only genomic or individual gene sequence has been provided.

The ordering also seems unclear - wouldn't BBHs inform HMM domains, which then in turn inform domain architectures? Or if starting with BBHs, then how could architectures possibly be known prior to knowing the domains themselves? Or if it should be read from top to bottom as shown, how exactly does one start with Function (e.g., a GO term) and then, somehow via Structure, thereby arrive at a Sequence alignment? Specifically, is a Pfam entry a "Function", from which the Sequence alignment is downloaded? Or are Function and the Sequence alignment both part of the starting Pfam entry (and then again, what does any of that have to do with Structure)? From which domains are found (but aren't Pfam entries domains to begin with?), and then BBHs are made from the domain architectures? (an extremely different way of doing the BBH procedure, which is normally done via Sequence alignments). In any case, as pointed out by other reviewers, the methodology used by DAB is not clearly explained in this figure, nor in the manuscript text.

We have edited the Figure for clarity incorporating the reviewer's suggestions.

12. Also, the last paragraph of the Discussion uses the word "closeness", but I think "closedness" was intended.

The typo has been amended.

**Competing Interests:** No competing interests were disclosed.
The article by Koehorst et al. describes a comparison of two approaches for clustering genomes sequences for the purpose of performing comparative genomics. The principle behind the two approaches, sequenced based clustering and domain based clustering, is described well in the introduction. The motivation of the article is clear and well founded. However, the details provided about how domain assignments are actually performed and handled throughout the experiment generated so many questions, these have clouded the validity of any conclusions.

1. How was InterPro used to assign a domain architecture? As the database presents a hierarchy of protein families and domains, unlike Pfam and TIGRFAM, there are numerous overlaps between the entries. Some of these are trivial C-terminal to N-terminal overlaps, while others are complex arrangements that cannot be simply represented as described. If three overlapping domains from InterPro are in the same hierarchy, which domain is used? If all member databases are used, this will account for the explosion of clusters in the InterPro based-clustering seen in Table 1. If InterPro accessions are used (e.g. as seen in the condensed view of a sequence on the InterPro website) then numbers are surprising.

2. How were Family vs Domain “types” handled from InterPro or Pfam? In InterPro, type families tend to be near full length protein families. In Pfam, they represent a more heterogeneous bag of entries that are yet to be established as a ‘domain’.

3. Pfam has a notion of related families, termed clans. Here the entries may not be intended to represent functionally distinct domains, but rather can represent a collection of families representing a continuum of evolution. How are entries belonging to a clan handled? How would the results differ if entries in one clan were treated as a single entity, for example, all P-loop NTPases as CL0023? How does this influence the sequence cluster to domain architecture relationships (schematically shown in Figure 5).

4. Why was the N-terminal starting position used to assess position of the domain? What is the effect of choosing the mid-point?

5. Both Pfam and TIGRFAM use HMMER version 3, which uses local-local alignment algorithm. How are partial hits to an HMM handled? Would two partial domain matches that occur due to an insertion between two halves of a domain be treated differently (see Triant and Pearson, 2015)?

Other comments:

1. The use of domain architectures as an approach for accelerating sequence searching is not that novel, for example, CD-ART has been available for many years. Domain architecture views have been present in most domain databases (e.g. Pfam, SMART, Prosite) for over a decade, and used
in genomic contexts. A more extensive overview of the use of domain architectures in the field is desirable.

2. The composite graphs presented in Figures 6, 7 and supplementary figures use different scales, so make the graphs hard to compare.

3. When the domain based clusters are compared to the sequence based clusters, it would be interesting to understand whether the number of domains that makes up the domain architecture influences the correlations to the sequence based clusters. Do single domain architectures predominated the 1:1 clusters?

4. Many readers may be unaware of the thresholds employed in InterProScan relate to the individual databases, so greater clarity is required.

5. Why is the versioned InterProScan described as a semantic wrapper?

References

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 09 Nov 2016

Jasper Koehorst, Wageningen University and Research, Stippeneng, The Netherlands

Thank you for the review, we have responded to your comments below:

1. How was InterPro used to assign a domain architecture? As the database presents a hierarchy of protein families and domains, unlike Pfam and TIGRFAM, there are numerous overlaps between the entries. Some of these are trivial C-terminal to N-terminal overlaps, while others are complex arrangements that cannot be simply represented as described. If three overlapping domains from InterPro are in the same hierarchy, which domain is used? If all member databases are used, this will account for the explosion of clusters in the InterPro based-clustering seen in Table 1. If InterPro accessions are used (e.g. as seen in the condensed view of a sequence on the InterPro website) then numbers are surprising.

   All member databases in InterPro were used. We partly took into account trivial N-terminal overlaps by alphabetically ordering the domains when distances between starting position were <3 amino acids. After analysing the results, we agree that this was not enough and this is the most likely cause of the explosion of this of clusters. As the reviewer suggests, taking the the full hierarchy of protein families and domains within InterPro would be required for comparative genome analysis based on domain architectures.
We have now better explained the selection criteria in the Materials and Methods section:

The positions (start and end on the protein sequence) of domains having Pfam, TIGRFAMs and InterPro identifiers were extracted through SPARQL querying of the graph database and domain architectures were retrieved for each protein individually. InterPro aggregates protein domain signatures from different databases. Here no pruning for redundancies has been done. Identification of domains was done using the intrinsic InterPro cut-off that represents in each case the e-values and the scoring systems of the member databases. The domain starting position was used to assess relative position in the case of overlapping domains; alphabetic ordering was used to order domains with the same starting position or when the distance between the starting position of overlapping domains was <3 amino acids. Labels indicating N-C terminal order of identified domains were assigned to each protein in such a way that the same labels were assigned to proteins sharing the same domain architecture.

We have commented on this point in the discussion, where the use of InterPro is addressed.

This paragraph now reads:

The chosen set of domain models and the database used as a reference greatly impact the results. InterPro aggregates protein domain signatures from different databases, which leads to redundancy of the domain models. This redundancy causes overlaps between the entries and an increase of the granularity of the clusters retrieved: this can bias downwards the size of the pan-genome and upwards the size of the core-genome, as shown in Table 1. In InterPro this redundancy is taken into account by implementing a hierarchy of protein families and domains. The entries at the top of these hierarchies correspond to broad families or domains that share higher level structure and/or function; the entries at the bottom correspond to specific functional subfamilies or structural/functional subclasses of domains \cite{mitchell_interpro_2015}. Using InterPro for DAB clustering would require taking into account the hierarchy of protein families and domains: however, this would pose challenges of its own and would require discrimination of the functional equivalence of different signatures within the same hierarchy.

We have also added the following to the conclusion

To enable DAB approaches for highly structured databases, such as InterPro, the hierarchy of protein families and domains within has to be explicitly considered.

2. How were Family vs Domain “types” handled from InterPro or Pfam? In InterPro, type families tend to be near full length protein families. In Pfam, they represent a more heterogeneous bag of entries that are yet to be established as a ‘domain’.

No distinction has been introduced as there don’t seem to be general rules that
apply to all cases. In the discussion section a paragraph has been added on the
effects of the structure of the databases.

3. Pfam has a notion of related families, termed clans. Here the entries may not be intended to
represent functionally distinct domains, but rather can represent a collection of families
representing a continuum of evolution. How are entries belonging to a clan handled? How
would the results differ if entries in one clan were treated as a single entity, for example, all
P-loop NTPases as CL0023? How does this influence the sequence cluster to domain
architecture relationships (schematicly shown in Figure 5).

The reviewer raises here an interesting point that we have now discussed. The
following has been added to the first paragraph of the discussion section.

Another source of redundancy are functionally equivalent domains from distantly
related sequences. Pfam represents this notion through related families, termed
clans, where relationship may be defined by similarity of sequence, structure or
profile-HMM. Clans might contain functionally equivalent domains, however it is not
clear whether this is always the case as the criteria for clan definition includes
functional similarity but not functional equivalence. Members of a clan have
diverging sequences and very often SB approaches would recognize the
evolutionary distance between the sequences and group them in different clusters.
If we were to assume that members of a clan are functionally equivalent and collect
them in the same DA cluster, we will have a higher number of cases where a single
DA cluster is split in multiple sequence clusters 1d→Ns. Also there would be higher
number of cases of sequence clusters with the same DA but no exactly matching
the DA clusters (1s→1d cases).

4. Why was the N-terminal starting position used to assess position of the domain?

The following line has been rewritten in the Methods section

Labels indicating N-C terminal order of identified domains were assigned to each
protein using the starting position of the domains: the same labels were assigned to
proteins sharing the same domain architecture.

5. What is the effect of choosing the mid-point?

We have commented on this in Results and Discussion. The following paragraph
has been added:

The starting position of the domains was used to generate labels indicating N-C
terminal order of identified domains. The labels were used only for clustering as
proteins sharing the same labels were assigned to the same clusters. Choosing
instead the mid-point or the C-terminal position could affect the labeling but it not
the obtained clusters.

6. Both Pfam and TIGRFAM use HMMER version 3, which uses local-local alignment
algorithm. How are partial hits to an HMM handled? Would two partial domain matches that
occur due to an insertion between two halves of a domain be treated differently (see Triant
and Pearson, 2015)?

In the discussion we have added a subsection on the limitations on DAB approaches. There we have added the following:

Partial domain hits might arise as a result of alignment, annotation and sequence assembly artifacts (cite Triant et al.). To reduce the number of partial domain hits additional pruning could be implemented to distinguish these cases. However, this is an open problem that requires caution as it could influence the functional capacity of an organism and clustering approaches using DA.

7. The use of domain architectures as an approach for accelerating sequence searching is not that novel, for example, CD-ART has been available for many years. Domain architecture views have been present in most domain databases (e.g. Pfam, SMART, Prosite) for over a decade, and used in genomic contexts. A more extensive overview of the use of domain architectures in the field is desirable.

We have added the paragraph in the introduction regarding domain architectures, comparison of domain architectures and their use for sequence search. We have also discussed on how these have been included in domain databases and, as also suggested by the first reviewer, on the preservation of domain architectures at high phylogenetic distances.

The following paragraph has been added to the introduction:

Domain architectures have been shown to be preserved at large phylogenetic distances both in prokaryotes and eukaryotes (Koonin 2002, Kummerfeld 2009). This lead to the use of protein domain architectures to classify and identify evolutionarily related proteins and to detect homologs even across evolutionarily distant species (Bjorklund 2005, Fong 2007, Song 2007, Lee 2009). Structural information encoded in domain architectures has also been deployed to accelerate sequence search methods and to provide better homology detection. Examples are CDART (Geer 2002) which finds homologous proteins across significant evolutionary distances using domain profiles rather than direct sequence similarity, or DeltaBlast (Boratyn 2012) where a database of pre-constructed position-specific score matrix is queried before searching a protein-sequence database. Considering protein domain content, order, recurrence and position has been shown to increase the accuracy of protein function prediction (Messih 2012) and has led to the development of tools for protein functional annotation, such as UniProt-DAAC (DougaFn 2016) which uses domain architecture comparison and classification for the automatic functional annotation of large protein sets. The systematic assessment and use of domain architectures is enabled by databases containing protein domain information such as UniProt (Uniprot Consortium 2015), Pfam (Finn 2016), TIGRFAMs (Haft 2003) and InterPro (Mitchell 2015), SMART (Letunic 2015) and PROSITE (Sigrist 2012), that also provide graphical view of domain architectures.
8. The composite graphs presented in Figures 6, 7 and supplementary figures use different scales, so make the graphs hard to compare.

**Figures 6 and 7 have been combined (also supplementary figures).**

9. When the domain based clusters are compared to the sequence based clusters, it would be interesting to understand whether the number of domains that makes up the domain architecture influences the correlations to the sequence based clusters. Do single domain architectures predominated the 1:1 clusters?

*We have looked into this and single domain architectures predominated the 1:1 clusters. A table has been added to the text (Table 3).*

10. Many readers may be unaware of the thresholds employed in InterProScan relate to the individual databases, so greater clarity is required.

*This point was also raised by A. Rosato. We have further explained the selected thresholds in the material and methods. Identification of domains was done using the intrinsic InterPro cut-off that represents in each case the e-values and the scoring systems of the member databases.*

11. Why is the versioned InterProScan described as a semantic wrapper?

*This line has been re-written, now it is explained that the versioned InterProScan stores the output in the RDF data model.*

**Competing Interests:** No competing interests were disclosed.
architectures, respectively. The first method is in principle more accurate and has higher coverage, whereas the second method is significantly faster and thus more suitable to cope with the explosion of genome information.

The authors demonstrate that domain-based methods provide results that are well in line with alignment-based methods. Consequently, their speed advantage does not compromise accuracy. In addition, the authors suggest that the Pfam database works better than InterPro for the present clustering purpose.

This article can benefit from some improvements:

- It is not clear to me why the labels within the colored boxes representing domains of Figure 1 differ in the top panel (Domain architectures) and the bottom panel (Domains).

- The new genome annotations generated by the authors should be made available to allow others to reproduce their calculations. It would be useful to have some data on the overall difference with respect to the original annotation.

- There are no details on the parameters used for domain identification such as E-value cut-offs. The latter has a strong impact on the number of singletons (1). It would be even more useful if the authors provided VMs with the complete setup for the entire procedure (from reannotation to clustering).

- The header SB is misaligned in Table 1. Why did the authors report the fraction of proteins containing at least one InterPro domain when the rest of the analysis is based on Pfam domains?

- I find the section "Comparison of DAB and SB clusters" difficult to read. In part this is due to the fact that the authors in the text describe actual numbers while Figures 6 and 7 report percentages. In particular, why should the "horizontal acquisition of the gene" reduce the sequence similarity score (i.e. increase the E-value of the blastp alignment)? Furthermore, preservation of domain architecture at high phylogenetic distances has been extensively analyzed in the literature. References should be added.

- It could be useful to combine Figures 6 and 7 to have a synoptic view.

- Table 1 shows that InterPro domains provide pangenomes that are not only always larger than the pangenomes obtained from Pfam domains but sometimes even larger than SB-derived pangenomes (e.g. *H. pylori* or Cyanobacteria). How is this possible?

- The low value of alpha in the Heaps regression for *L. monocytogenes* afforded by the DAB is striking and should be analyzed in more detail.

- The line break after "transfer events" in the second paragraph of the introduction is not needed.

- In the Supplementary material, SSB should SB

References
Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Jasper Koehorst, Wageningen University and Research, Stippeneng, The Netherlands

1. It is not clear to me why the labels within the colored boxes representing domains of Figure 1 differ in the top panel (Domain architectures) and the bottom panel (Domains)

In the older version the labels in the top referred to the domain names whereas the labels on the bottom contained the PFAM identifiers. The figure has been changed so that only one set of labels is presented.

2. The new genome annotations generated by the authors should be made available to allow others to reproduce their calculations. It would be useful to have some data on the overall difference with respect to the original annotation.

The reviewer raises a very interesting topic that has been the focus of a different study. We have performed a detailed analysis of the differences between the original and the de novo annotation in a set of 432 Pseudomonas genomes. In that case, an average difference of 153 genes per genome was detected. Differences in annotations were observed at all functional levels (EC numbers, GO terms and protein domains). The magnitude of the differences correlated with the date the original annotation. The manuscript is currently under review and we will include the reference as soon as it is published.

The SAPP annotation framework used to generate these files can be found at http://semantics.systemsbiology.nl/. Extensive documentation is available at http://sapp.readthedocs.io. A section (reproducibility) has been added indicating the workflow to reproduce the analysis here presented. We have included how annotations are compared.

3. There are no details on the parameters used for domain identification such as E-value cut-offs. The latter has a strong impact on the number of singletons (1).

We agree that the choice of the E-value cut off plays a critical role on domain detection and greatly impacts the size of the core-genome. However, as reported in InterPro: “The signatures contained within InterPro are produced in different ways by different member databases, so their E-values and/or scoring systems cannot be meaningfully compared” therefore we have selected the intrinsic cutoff within InterPro [Mitchel et al 2015]. This has been mentioned in the Material and Methods section:

Identification of domains was done using the intrinsic InterPro cut-off that represents in each case the E-values and the scoring systems of the member databases (Mitchel 2015).
4. It would be even more useful if the authors provided VMs with the complete setup for the entire procedure (from reannotation to clustering)

   The SAPP annotation framework used to generate these files can be found at http://semantics.systemsbiology.nl/. Extensive documentation is available at: http://sapp.readthedocs.io. A section has been added indicating the workflow to reproduce the analysis here presented.

5. The header SB is misaligned in Table 1. Why did the authors report the fraction of proteins containing at least one InterPro domain when the rest of the analysis is based on Pfam domains?

   We have modified Table 1 and included an additional column with the fraction of proteins containing at least one Pfam domain.

6. I find the section "Comparison of DAB and SB clusters" difficult to read. In part this is due to the fact that the authors in the text describe actual numbers while Figures 6 and 7 report percentages. In particular, why should the "horizontal acquisition of the gene" reduce the sequence similarity score (i.e. increase the E-value of the blastp alignment)?

   We have rephrased the sentence on horizontal gene acquisition, it now reads: Similarly, there are 399 1s → 1d clusters. Each of these cases represent a sequence cluster where all the sequences share the same domain architecture, but other sequences exist with the same architecture that have not been included in the cluster due to a too low similarity score. The low similarity between sequences with the same domain architecture could be due to a horizontal acquisition of the gene or to a fast protein evolution at the sequence level. Genes acquired from high phylogenetic distances could greatly vary in sequence while presenting the same domain architecture.

7. Furthermore, preservation of domain architecture at high phylogenetic distances has been extensively analyzed in the literature. References should be added.

   The following paragraph has been added to the introduction:
   Domain architectures have been shown to be preserved at large phylogenetic distances both in prokaryotes and eukaryotes (Koonin 2002, Kummerfeld 2009). This lead to the use of protein domain architectures to classify and identify evolutionarily related proteins and to detect homologs even across evolutionarily distant species (Bjorklund 2005, Fong 2007, Song 2007, Lee 2009). Structural information encoded in domain architectures has also been deployed to accelerate sequence search methods and to provide better homology detection. Examples are CDART (Geer 2002) which finds homologous proteins across significant evolutionary distances using domain profiles rather than direct sequence similarity, or DeltaBlast (Boratyn 2012) where a database of pre-constructed a position-specific score matrix is queried before searching a protein-sequence
database. Considering protein domain content, order, recurrence and position has been shown to increase the accuracy of protein function prediction (Messih 2012) and has led to the development of tools for protein functional annotation, such as UniProt-DAAC (Dougan 2016) which uses domain architecture comparison and classification for the automatic functional annotation of large protein sets. The systematic assessment and use of domain architectures is enabled by databases containing protein domain information such as UniProt (Uniprot Consortium 2015), Pfam (Finn 2016), TIGRFAMs (Haft 2003) and InterPro (Mitchell 2015), SMART (Letunic 2015) and PROSITE (Sigrist 2012), that also provide graphical view of domain architectures.

8. It could be useful to combine Figures 6 and 7 to have a synoptic view

Figures 6 and 7 (and supplementary figures) have been combined.

9. Table 1 shows that InterPro domains provide pangenomes that are not only always larger than the pangenomes obtained from Pfam domains but sometimes even larger than SB-derived pangenomes (e.g. H. pylori or Cyanobacteria). How is this possible?

InterPro aggregates protein domain signatures from different databases, which leads to redundancy of the domain models. This redundancy causes overlaps between the entries and an increase of the granularity of the clusters retrieved: this can bias downwards the size of the pan-genome and upwards the size of the core-genome, as shown in Table 1.

10. The low value of alpha in the Heaps regression for L. monocytogenes afforded by the DAB is striking and should be analyzed in more detail

We thank the reviewer for this very interesting observation. We have investigated the low value of alpha in this case and the following paragraph has been added

The alpha DAB value retrieved for L. monocytogenes is strikingly low. Heaps law regression relies on the selected genomes providing a uniform sampling of selected taxon, here species. Analysis of the domain content of the selected genomes shows a divergent behaviour of strain LA111 (genome id GCA\_000382925-1). This behaviour is clear in Figure 7 (PCA), where GCA\_000382925-1 appears as an outlier of the L.monocytogenes group. Removal of these outlier leads to alpha DAB=1.04 and alpha SB=0.64, which emphasizes the need for uniform sampling prior to Heaps regression analysis.

11. The line break after "transfer events" in the second paragraph of the introduction is not needed

The line break has been removed.

12. In the Supplementary material, SSB should SB

This typo has been fixed.
**Competing Interests:** No competing interests were disclosed.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com