Nature Methods
Original Paper

Genome analysis

FindMyFriends: A Framework for Fast and Accurate Pangenome Analysis of Thousands of Diverse Genomes

Thomas Lin Pedersen [1,2,*], Intawat Nookaew [3,4], Maria Månsson [2] and David Wayne Ussery [3,4]

[1]Center for Biological Sequence Analysis, Department of Systems Biology, The Technical University of Denmark, Building 208, DK-2800 Lyngby, Denmark,
[2]Assays, Culture and Enzymes Division, Chr. Hansen A/S, DK-2970 Hørsholm, Denmark,
[3]Comparative Genomics Group, Biosciences Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA and
[4]Department Biomedical Informatics, College of Medicine, University of Arkansas for Medical Sciences, Little Rock, Arkansas, USA.

*To whom correspondence should be addressed.

Abstract

Pangenome analysis has suffered from a lack of scalability, making it difficult to utilize the increasing availability of microbial genome sequences to the fullest. Here we present a novel framework, FindMyFriends, for creating and analyzing pangenomes that covers thousands of genomes, on standard workstation hardware. We show the quality of the framework by comparing it to a range of popular alternatives and find that FindMyFriends are both the fastest and most accurate when it comes to grouping orthologue genes. The power of the framework is shown by creating a pangenome of 4,770 genomes spanning the full bacterial domain. The gene grouping could be completed in 99 hours running on a single processor. We show that the chromosomal neighborhood based grouping algorithm is stable, even when applied to such a heterogeneous dataset. FindMyFriends is an extensible R package published through Bioconductor and provides a strong foundation for pangenome analysis in R.

A pangenome is a grouping of genes from several different bacterial genomes according to similarity. The first study on pangenomes was conducted by Tettelin *et al.* (2005) and investigated the pangenome of eight different genomes of *Streptococcus agalactiae*. Since then, the scope of pangenome analyses has changed along with the rapid increase in genomic sequences and analyses consisting of hundreds of different genomes, including multiple species of the same genus, are now common (Leekitcharoenphon *et al.*, 2016; Land *et al.*, 2015; Jun *et al.*, 2014; Méric *et al.*, 2013; Snipen and Ussery, 2012; Kaas *et al.*, 2011). The grouping of genes across genomes has traditionally relied on comparing all sequences against each other using BLAST (Altschul *et al.*, 1990; Camacho *et al.*, 2008) and applying a clustering algorithm on the results. This is the approach of the most popular algorithms that have been developed such as OrthoMCL (Li *et al.*, 2003) and PanOCT (Fouts *et al.*, 2012) and development has mostly focused on improvements in the clustering step, ignoring the time-consuming generation of BLAST results that takes up most of the computational time. The problem with relying on data that results from BLASTing all genes against each other is that the complexity is O(nˆ2) and that BLAST is a relatively slow algorithm. Analysis of hundreds or thousands of genomes requires considerable computing hardware and time, something not available to all researchers. Recently Page *et al.* (2015) introduced the Roary algorithm that significantly cuts down on computational time by pre-clustering genes using the CD-Hit algorithm (Li and Godzik, 2006) and then BLASTing representatives for each cluster against each other, effectively applying the standard approach to a well-chosen subset of genes. They showed a complexity approaching O(n), but as the algorithm relied heavily on the data reduction provided by CD-Hit, these results are only applicable to sets of highly homogeneous genomes where the number of clusters provided by CD-Hit remains low. In cases of high heterogeneity, Roary will also run into the problem of quadratic complexity.

There has been few algorithms ignoring BLAST altogether. PanFunPro (Lukjancenko *et al.*, 2013) groups genes based on presence of functional domains, but relies heavily on the quality of the functional annotation. Sequences lacking known functional domains are grouped separately using BLAST resulting in incomparable grouping approaches within the dataset. There is also a high probability of low resolution in the grouping based on functional domains as large areas of the sequences are potentially ignored leading to grouping of distantly related sequences.

The earliest algorithms relies solemnly on gene sequence similarity and are thus unable to differentiate between homologues and paralogues, resulting in gene groups containing multiple genes from the same genomes. IONS (Seret and Baret, 2011) defined a post processing algorithm that takes chromosomal neighborhood into account in order to resolve gene groupings from other algorithms. PanOCT was the first algorithm to use chromosomal neighborhood information directly in the gene grouping and Roary uses the information in a secondary step of the algorithm. A problem with the approach employed by PanOCT is that sequence similarity is conflated with neighborhood similarity and that makes it difficult to reason about the choices made by the algorithm.

In this paper, we describe a new algorithm implemented within the FindMyFriends package available through Bioconductor (Gentleman *et al.*, 2003; Huber *et al.*, 2015). FindMyFriends uses a two-step approach that allows for huge speed gains while still providing a rigorous gene grouping. FindMyFriends itself is an extensible framework for working with pangenome data and allows researchers to utilize a long range of analyses available within R and Bioconductor directly.

Results

Overview of FindMyFriends

Figure 1 shows an overview of the steps in the FindMyFriends workflow. The standard input is .fasta files with extracted coding regions of the different genomes, either translated or untranslated. For the algorithm to know the chromosomal position of the different genes, the information should be recorded in the fasta header or provided as a separate data frame. The algorithm can proceed without this information but it will impact the quality of the results. The first step in the grouping is based on CD-Hit as in Roary, but the aims of using CD-Hit is different between the two algorithms. In Roary the preclustering will provide a grouping of the most similar genes, that will later on be further combined by BLAST. In FindMyFriends, the preclustering will provide a very coarse and broad grouping of genes that will be split into more correct groups later on. This coarse grouping is achieved by iteratively ensures that the number of gene pairs to be investigated in detail will be reduced. The second step of the grouping is thus a splitting of the groups provided by CD-Hit. The splitting is based on both the similarity of the neighboring genes, the similarity of the sequences as well as whether the genes are from the same genome. As a measure of similarity FindMyFriends employs the cosine similarity of the K-mer feature vector of each sequence. Gene groups are split by extracting cliques from a graph representation of the similarity between all genes in a group. This ensures that the final groups represents homogeneous collections of genes where all genes show high similarity with each others. This feature is not ensured by using community detection algorithms such as Markov Clustering in e.g. Li *et al.* (2003), as these algorithms will only find highly connected subgraphs rather than complete subgraphs. Following the splitting a final refinement step is performed in order to merge similar groups that failed to be part of the same preclustering group, due to the heuristic employed by CD-Hit. A detailed description of the developed algorithms is available in the Material and Methods section.

If chromosomal location is not available, this part will be disregarded in the splitting process. Still, the chromosomal neighborhood are found to be a much stronger driver for the final grouping than sequence similarity, so the quality of the end result will be impaired. Thus, if chromosomal location is available this information should be used in order to ensure the quality of the final gene grouping. FindMyFriends support additional post-processing steps such as different visualizations, detection of chromosomal regions with high plasticity as well as linking of paralogue gene groups. Further, it ties into the vast number of genomics tools provided within Bioconductor.

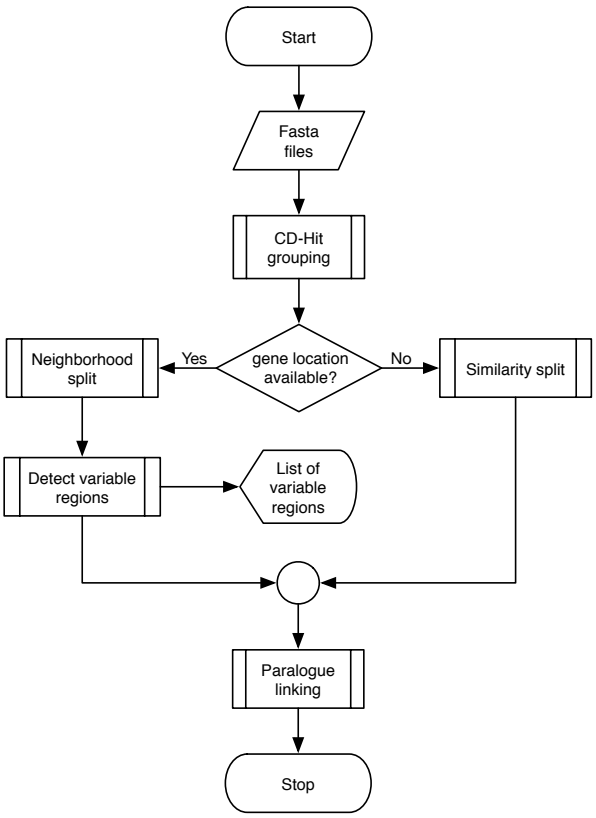

Fig. 1. Overview of the standard steps of a pangenome creation using FindMyFriends. While analysis of genes lacking chromosomal location is possible, the extra information yields better results and facilitates additional analysis. Chromosomal location should thus always be supplied if available.

Evaluating cosine similarity of K-mer features against BLAST

Berendzen *et al.* (2011) evaluated the choice of K in terms of grouping genes across species and arrived at K = 10 as an optimal choice, but their approach only took presence of a single shared word into account. By relying on the full K-mer feature vector it should be possible to reduce the word size. In order to assess the applicability of using cosine similarity of K-mers as a measure of sequence similarity, full sequences from four *Acinetobacter baumannii* genomes were BLASTed against each other as well as compared with cosine similarities for K = 3–7. The cosine similarity stabilized after K = 4 and there was no need to increase K in order to get good separation (see supplementary figure S1). Compared to both E-value and Bitscore from the BLAST comparison, cosine similarity exhibited a stronger bimodal distribution of the scores, making it much more stable to the choice of threshold value (see supplementary figure S2). Furthermore, this correlated much better with the '50/50 rule' (**?**) often used to define similarity (see supplementary figure S3). Based on this, there is no indication that using cosine similarity for sequence comparison rather than BLAST would result in degradation of the results.

Timing of FindMyFriends against other algorithms

In order to compare the true computational cost of the different algorithms against each other as well as their complexity, each algorithm was run on a set of *Acinetobacter baumannii* genomes of varying size from 5–50

genomes and timed from the onset of computation until the results were available. The results of this timing can be seen in figure 2A where it is apparent how much faster Roary and FindMyFriends are compared to the other four algorithms. Furthermore, it can be seen how both FindMyFriends and Roary approach a linear complexity.

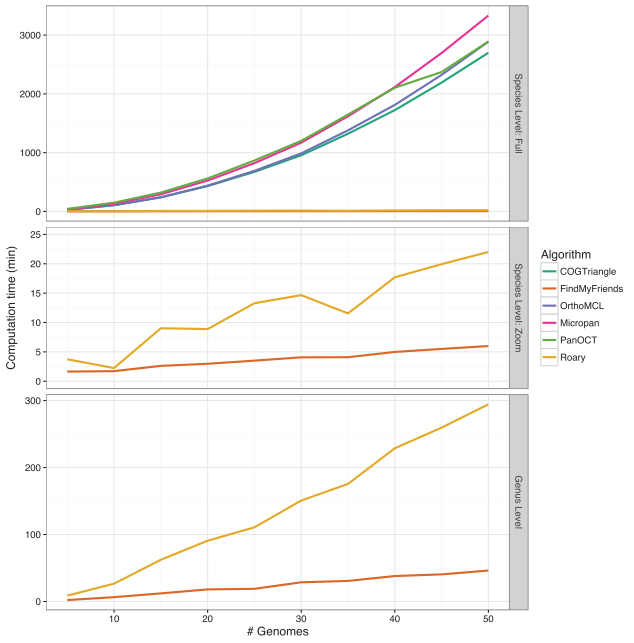

Fig. 2. Timing of 6 different algorithms as a function of the number of genomes included in the analysis. The top plot is based on 50 different Acinetobacter baumannii genomes. The drop in computational time for PanOCT after 40 genomes is due to the algorithm stopping midway because of lack of memory. The middle plot is based on the same data as the top plot but zoomed in on FindMyFriends and Roary as these operates at a very different scale. The bottom plot is based on 50 Streptococcus genomes and only shows timing of FindMyFriends and Roary.

A set of genomes from the same species provides an optimal use case for the CD-Hit pre-clustering step as the huge overlap between the genomes makes it possible to significantly reduce the number of genes to compare. In order to assess how both algorithms responded to more heterogenious data sets the same timing was performed on a set of 50 different *Streptococcus* genomes. As can be seen in figure 2B, Both algorithms responded with longer computational time, though FindMyFriends less so. Surprisingly Roary retains its linearity, as it would have been expected that an increase in heterogeneity would lead to non-linearity.

Evaluating gene grouping quality

To investigate the quality of the gene grouping provided by FindMyFriends in more detail, the same set of four *Acinetobacter baumannii* genomes as used in the PanOCT evaluation was analyzed, using both FindMyFriends, Roary, and a list of pure BLAST based tools. The overall summary of the results are listed in table 1.

FindMyFriends stands out in the total gene groups defined as well as the distribution of the different types of gene groups. FindMyFriends gives rise to an overall larger number of gene groups as well as more singletons and accessory gene groups at the expense of core gene groups. FindMyFriends is the only algorithm that intentionally tries to avoid grouping gene fragments with complete gene sequences by utilizing a hard cut-off on sequence length difference. PanOCT for instance tries to group the longest fragment arising from a frameshift event together with the complete sequences from other genomes, but as gene groups should indicate operational equivalence of the members, it makes little sense to include fragments together with functional sequences. Due to this difference it is expected that FindMyFriends will produce a higher number of gene groups and that more gene groups will be singleton and accessory.

Table 1. Overview of result from the various algorithms compared in this article.

	Singleton		*Accessory*		*Core*		*Total*	
FindMyFriends	3486	(366)	1901	(376)	1666	(1)	7053	(743)
PanOCT	2124	(31)	1365	(136)	2376	(110)	5865	(277)
Roary	2735	(213)	1494	(317)	2155	(69)	6384	(599)
COGTriangle	2212	(60)	1636	(17)	1956	(20)	5804	(97)
OrthoMCL	2134	(13)	1658	(17)	1969	(15)	5761	(45)
Micropan	2061	(32)	1453	(64)	2151	(10)	5665	(106)
Intersection	1149		814		1547		3510	

PanOCT results are from Fouts *et al.* (2012). COGtriangle and OrthoMCL are based on their respective implementations in GET_HOMOLOGUES (Contreras-Moreira and Vinuesa, 2013). Micropan is from Snipen and Liland (2015). *Singletons* are defined as gene groups containing only genes from a single genome, while *core* are gene groups containing genes from all genomes. *Accessory* is everything in between. Numbers in parentheses are number of respective groups unique to the specific algorithm.

To assess the specific differences in grouping behavior between the different algorithms, all groups from all algorithms were converted to pairs of genes, so that for example a group consisting of gene 1, 2, and 3 would be converted to the following gene pairs: 1–2, 1–3, and 2–3. This was done to make a fuzzy comparison that ensured that groups from different algorithms that were almost equal would still give rise to some similarity. Plotting the differences between gene pairs from the different algorithms (figure 3) as an upset plot (Lex *et al.*, 2014), we see that FindMyFriends is the only algorithm not giving rise to unique gene pairs (lack of bar only covering FindMyFriends). Thus, it does not take any decisions to group genes that are not supported by some of the other algorithms. This is expected as FindMyFriends is less inclined to group genes than the other algorithms.

Looking at the other end of the spectrum we see that 835 gene pairs are present in all but FindMyFriends (bar six from right in figure 3) indicating situations where the conservatism of FindMyFriends might lead to false negatives in the grouping. The gene pairs can be regarded as the most controversial decisions made by FindMyFriends, as they are not supported by any of the other algorithms. Investigating these cases in more detail we find 784 of these gene pairs differed in sequence length by more than 10 % which is the default cutoff utilized by FindMyFriends. One of the remaining 51 gene pairs had a cosine similarity below the threshold, while the remaining 50 showed high sequence similarity. The 50 gene pairs could be attributed to 26 sets of distinct gene groups where FindMyFriends disagreed with the rest. In 23 of the cases the members of of the gene groups did not share the same chromosomal neighborhood, two of the cases included frameshift events where the shortest sequence were more than 10 % shorter than the longest sequence. Only one case, corresponding to three gene pairs, showed a partial overlap of chromosomal neighborhood but the overlap appeared two coding regions downstream of the gene groups, and can thus at best be considered an edge case. Based on the most controversial decisions taken by FindMyFriends, namely its choices to not group genes that all other tested algorithms grouped, it seems that FindMyFriends are controlling the false negative rate, despite being more conservative than the other algorithms.

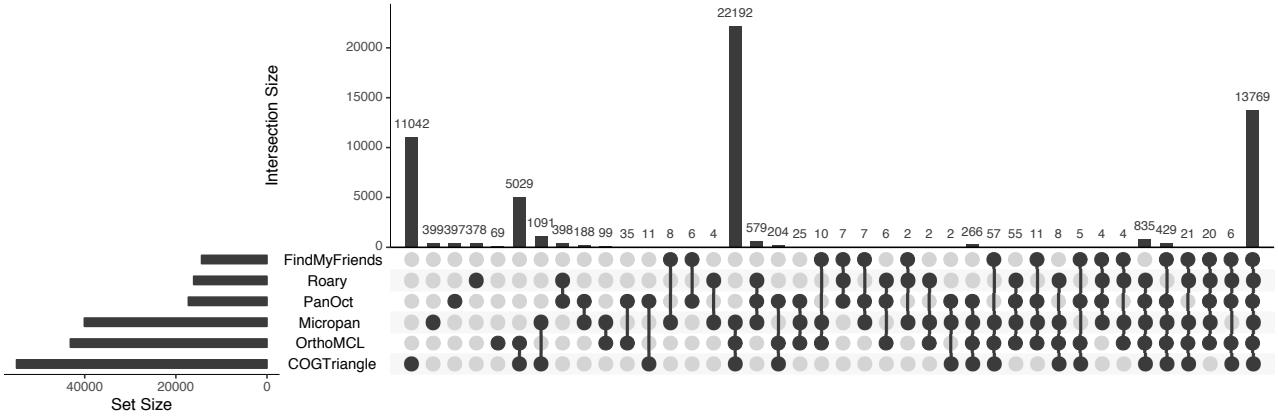

Fig. 3. Overview of intersection sizes of the different methods gene pairs. Bar height gives number of unique members to the set intersect given by the grid below the bar.

Calculating the pangenome of the bacterial domain

In order to showcase the possibilities given by FindMyFriends, the pangenome of all completed bacterial genomes stored in Genbank was calculated. The pangenome consisted of 16,141,814 genes from 4770 different genomes. Using a single processor the pangenome was calculated in 99 hours and 8 minutes. The resulting pangenome consisted of 6,887,134 orthologue gene groups. A secondary similarity based grouping of the gene groups using a 0.7 cutoff for both sequence length and similarity in CD-Hit revealed 4,372,061 homologue gene groups. Based on the fraction of genomes containing more than 1000 genes (92%) we find the average number of ratio of paralogues per genome to be ~2.3% (see supplementary figure S4). There does not seem to be any trend towards bigger genomes containing a higher fraction of paralogues. Thirteen genomes have more than 20% paralogues. All of these genomes come from either pathogens or isolates from extreme environments, which could explain their high number of paralogue genes, as gene cluster duplication is often used as an adaptive mechanism for these types of organisms.

The gene groups are split into 77% singletons and 23% accessory. The number of singletons are much higher than would normally be seen in a pangenome, but this is due to the taxon coverage. Of the 4,770 genomes 10% are the only representative for their respective genera. This results in most of their genes being labeled as singletons despite many of their genes being prevalent among their close relatives. Indeed, 30% of the singletons originate from the 10% genomes that are the only representative for their genus. This effect is even more pronounced at the species level as 73% of the singletons originates from the 30% of the genomes that are the only representatives of their species. In order to assess the quality of the gene grouping the 15,265 homologue gene groups that represented more than 100 genomes were annotated using Pfam. In around 95% of the gene groups all members shared the most common Pfam domain while only 1% of the gene groups had agreement between less than 90% of the members (see supplementary figure S5). Four percent of the gene groups had no matches in the PfamA database and could thus not be assessed.

A pangenome based phylogeny was created by minimum evolution clustering on Jaccard distances from the presence/absence matrix. The result of the clustering can be seen in figure 4 where branches are colored by the phylum they represent. The pangenome contains no core on either domain- or phylum-level and it is thus difficult to establish a correct phylogeny for these taxon-levels based on pangenome data. Still it can be seen that subsets of the different phyla are clustered together, indicating that the accessory genes can reconstruct the phylum-level taxonomy to some degree.

To see how the pangenome represents the phylogeny at a lower level we focus on the *Streptococcus* genus, which forms a distinct branch of the main tree. Figure 5 shows that the pangenome is in good correspondence with the current classification of the genus. The misplaced *St. salivarius* corresponds to the genome placed away from the main clade in figure 4. It only contains 56 genes and is thus with high certainty an erroneous sequence. The misplaced *St. constellatus* is the supspecies *pharyngis* which is also placed alone in other studies (Richards *et al.*, 2014). The last misplaced genome, according to the viridans grouping, is *St. oralis*, a member of the mitis group. Looking into the actual pangenome shows that, after *St. sp. VT 162*, the genomes that *St. oralis* has most gene groups in common with are indeed those of *St. pneumonia* and *St. mitis*. Thus, it appears that the placement in the phylogeny is an artifact of the hierarchical clustering rather than the gene grouping performed by FindMyFriends.

Discussion

We have presented a new algorithm, implemented in the FindMyFriends framework, for doing pangenome analyses of large scale bacterial genome collections. The algorithm is both fast as well as showing linear scaling with respect to the number of genomes in the pangenome. These traits makes FindMyFriends well suited to dealing with the ever increasing amount of genomic information available. Compared to other algorithms in use today, FindMyFriends employs a stricter approach to gene grouping, that is easier for the user to reason about. The resulting gene groups are ensured to contain gene cliques where all members share trait similarities passing user defined thresholds. This stricter approach to gene grouping results in a larger number of gene groups and a decrease in the number of core gene groups, but we show that the false negative rate is well controlled and that the increase in gene groups is a result of other algorithms being too prone to grouping non-orthologue genes together.

We show the power of the algorithm by calculating the pangenome of 4,770 genomes representing the complete bacterial domain. Due to the size of the pangenome, both in terms of genes and genomes as well as in terms of taxon coverage, this pangenome represents a huge task and far surpasses the size of all currently published pangenomes. Still, FindMyFriends was able to complete the gene clustering in 99 hours, running on a single workstation-class processor. Investigating the pangenome in both high and low level showed that FindMyFriends continued to show a high accuracy and that the neighborhood based gene group splitting was stable, even when subjected to highly heterogeneous genomes.

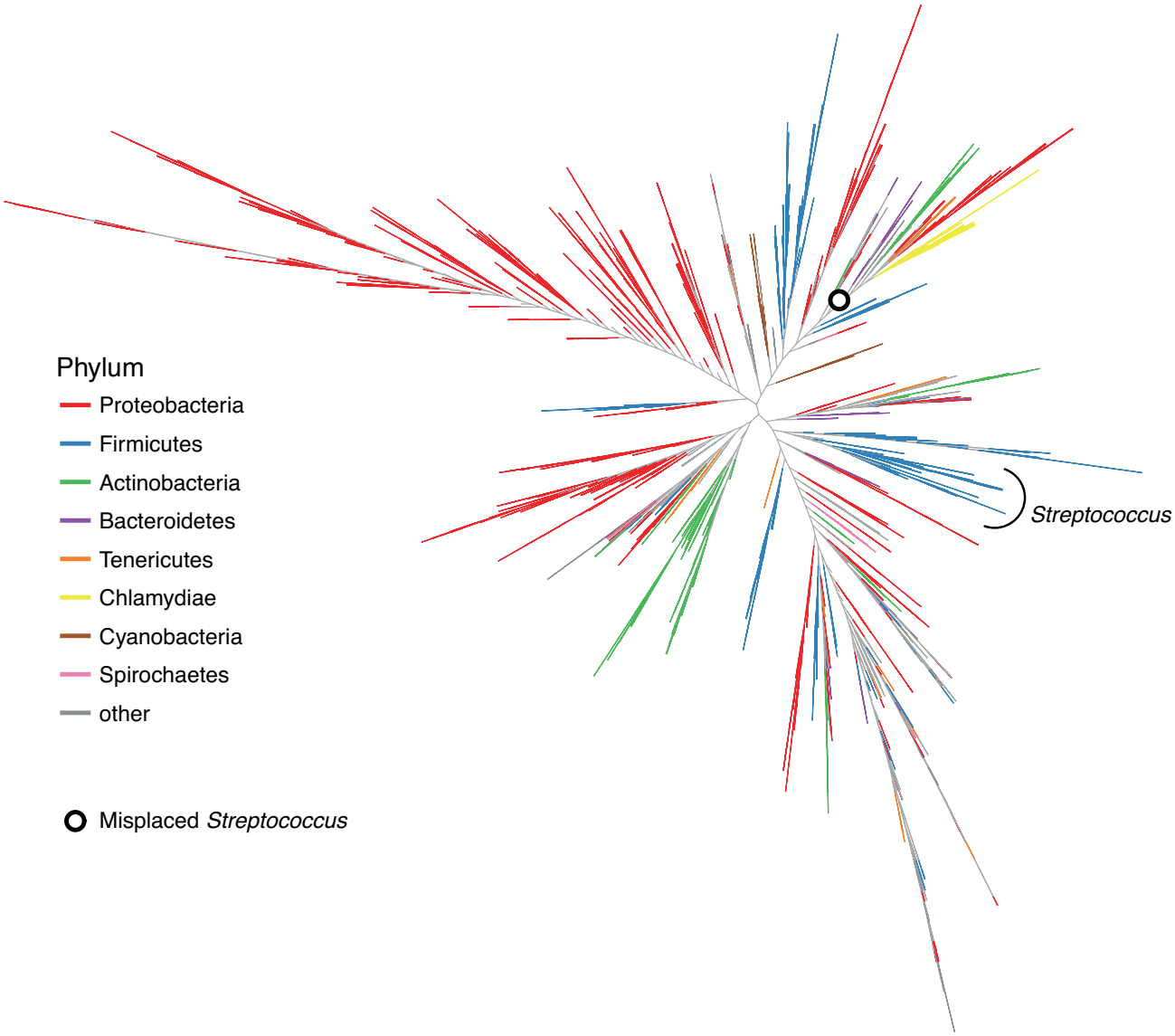

Fig. 4. Phylogenetic tree of 4,770 bacteria based the pangenome presence/absence pattern. The tree is constructed using Jaccard distance and minimum evolution clustering. Branches are colored by the phylum it contains, and the Streptococcus genus is shown explicetly.

AUTHOR CONTRIBUTIONS T.L.P. Developed and invented the methods and prepared the manuscript; I.N. Aided in the Pfam analysis of the bacterial pangenome; M.M. and D.W.U. helped with preparation of the manuscript.

COMPETING FINANCIAL INTERESTS The authors declare no competing financial interests.

Material and Methods

Genomes and Annotation

All genomes for the algorithm comparison were reannotated using Prodigal v2.6.2 (Hyatt *et al.*, 2009) using default settings. Information on the *Actinobacter baumanii* genomes used for timing and grouping comparison can be found in the supplementary material, *Streptococcus* genomes used for timing can be found in the supplementary material. The nucleotide sequences of the genomes for the bacterial domain pangenome were retrieved from NCBI assembly database (~October 2015). The ORFs were directly called from the downloaded sequences and translated into amino acid sequences using Prodigal v1.20 (Hyatt *et al.*, 2009). All of the amino acid sequences were queried through the Pfam database v24 through the pfamscan module (Finn *et al.*, 2016). The trusted cut-off was used to assign the present of a Pfam domain in the individual amino acid sequence. The complete list of identifiers for the genomes used for the full bacterial pangenome as well as all raw sequence data is available upon request. Further, the generated pangenome is available upon request.

System Requirements

FindMyFriends is written in R and C++ and is available on all systems supported by the Bioconductor project (Gentleman *et al.*, 2003; Huber *et al.*, 2015). The program has no external dependencies except for R (R Core Team, 2016) itself. All computations was performed on Amazon EC2 instances, c3.2xlarge (2.80 GHz Intel Xeon E5-2680 v2, 15 Gb RAM) for

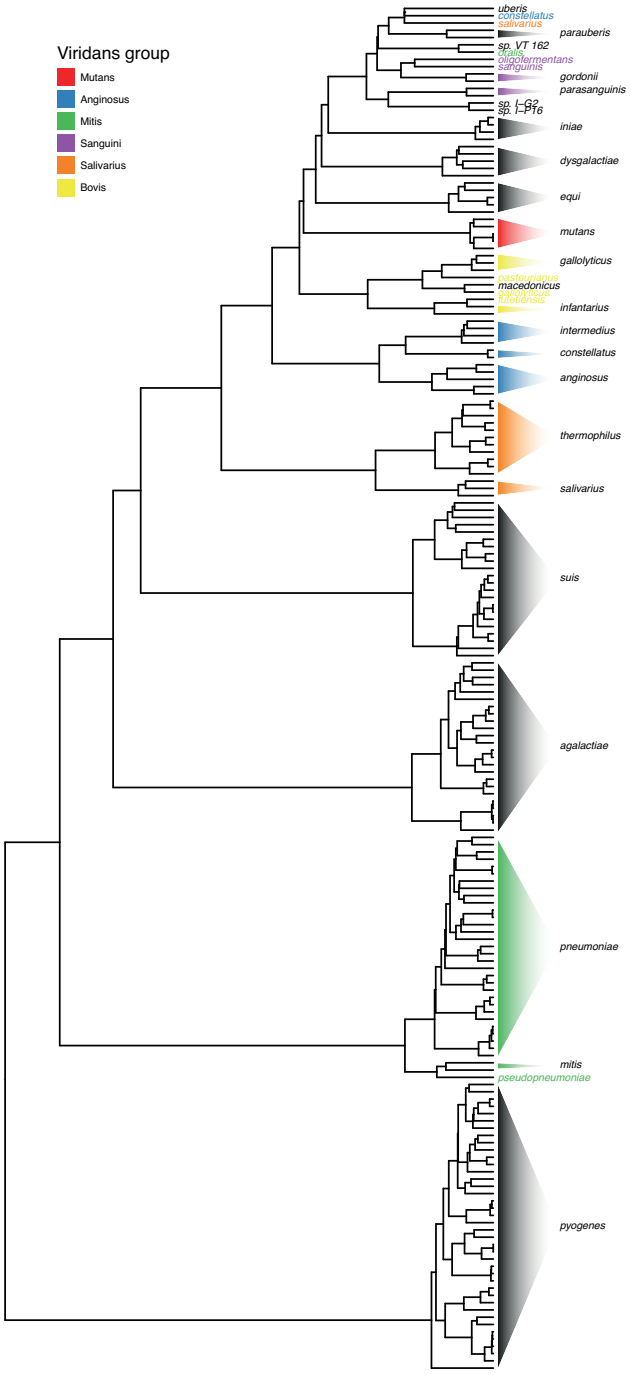

Fig. 5. Phylogenetic tree based on the Streptococcus subset of the total pangenome. The tree is constructed using Jaccard distance and Ward's minimum variance clustering. Branches are annotated by common species and colored by the viridans group the species belong to, if any.

timing and grouping comparison and r3.4xlarge (2.50 GHz Intel Xeon E5-2670 v2, 122 Gb RAM) for the bacterial pangenome analysis.

Algorithms

The initial grouping is currently using CD-Hit, but any fast approach to segmenting genes into broadly similar groups will work. The grouping proceeds by iteratively lowering the grouping threshold until the final threshold is reached. At each iteration the longest member of each group is selected as a representative for the group and groups are merged based on the clustering of the representatives by CD-Hit. The K-mer size is chosen automatically to be the highest K that supports the current threshold. The splitting of the groups provided by the above algorithm is performed as follows:

1. The groups are pre-splitted based on the chosen sequence length cutoff.
2. All groups not containing any paralogues are queued for splitting.
3. The groups in the queue are splitted based on the following.
4. The number of matching neighboring genes between each pair of genes in the group is calculated.
5. The cosine similarity of each pair of genes in the group is calculated
6. A graph with genes as vertices is created. Edges between vertices are created if the gene pair represented by the edge adheres to the following:
   - They share at least one gene in their neighborhood.
   - Their cosine similarity is above the chosen threshold.
   - Their difference in sequence length is below the chosen threshold.
   - They do not originate from the same genome.
7. The edge representing the largest number of shared neighboring genes and highest cosine similarity is chosen.
8. From the two vertices defined by the chosen edge a clique is grown by iteratively finding the strongest edge (most shared neighbors and highest cosine similarity) that fulfills the clique constraint, and adding the vertice to the clique.
9. The vertices in the final clique are removed from the graph and is assigned to a new gene group.
10. Step 4-6 is repeated until the graph is empty.
11. The unprocessed gene groups are examined and those that lie adjacent to already splitted groups are added to the queue.
12. Step 3-4 is repeated until all gene groups has been split.

Following the splitting a refinement step is performed as follows:

1. All gene groups that share a neighboring gene group either up- or downstream are detected - these are termed parallel gene groups.
2. The cosine similarities for all parallel gene groups are calculated based on the longest sequence from each group.
3. Parallel groups whose similarity is above the threshold, who adhere to the sequence length threshold, and who doesn't contain genes from the same genomes are merged.
4. Step 1-3 are repeated until convergence.

The refinement step is necessary as fast sequence clustering tools such as CD-Hit uses a heuristic to avoid comparing all sequences with each other. For a small fraction of cases similar sequences fails to be grouped, which gives rise to similar parallel groups.

Versions of Algorithms

The versions used for the comparisons are the following:
FindMyFriends: 1.2.0
PanOCT: 3.18
GET_HOMOLOGUES: 1.3
Roary: 3.2.7
Micropan: 1.0

Timing of Algorithms

All timings were done on identical Amazon EC2 instances based on the Bioconductor Amazon Machine Image `https://www.bioconductor.org/help/bioconductor-cloud-ami/`. From

the initial set of 50 *Actinobacter baumanii* genomes 10 sets of genomes with increasing size were sampled and pangenomes were calculated for each of these sets, timing the computations from the onset to completion. The scripts used to setup and run the timings for each algorithm as well as the genome members of each set can be found in the supplementary material. The process was essentially unchanged for the timing based on *Streptococcus* genomes except only FindMyFriends and Roary were timed.

Comparison of Groupings

In order to compare the gene grouping results provided by the different algorithms, the same 4 *Actinobacter baumanii* genomes as used in the PanOCT article (Fouts *et al.*, 2012) were used to create pangenomes by the different algorithms. The results from all algorithms were imported into the FindMyFriends framework for additional analysis. All gene groups from each algorithm were converted into gene pairs corresponding to edges in a complete graph based on the gene groups members as vertices. The resulting gene pairs from each algorithm were used to compare the grouping choices of each algorithm.

Phylogeny construction

Phylogeny of both the full bacterial pangenome as well as the *Streptococcus* subset are based on the Jaccard distances of the presence/absence vectors for each genome. The minimum evolution clustering used for the complete bacterial pangenome was the `fastme.ols` function from the R package ape v3.4. The Ward's minimum variance clustering used for the *Streptococcus* subset was the `hclust` function from the R package stats v3.2.4 using the *ward.D2* method.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, **215**(3), 403–410.

Berendzen, J., Bruno, W. J., Cohn, J. D., Hengartner, N. W., Kuske, C. R., McMahon, B. H., Wolinsky, M. A., and Xie, G. (2011). Rapid phylogenetic and functional classification of short genomic fragments with signature peptides. *BMC Research Notes*, **5**, 460–460.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T. L. (2008). BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421–421.

Contreras-Moreira, B. and Vinuesa, P. (2013). GET_HOMOLOGUES, a versatile software package for scalable and robust microbial pangenome analysis. *Applied and Environmental Microbiology*, **79**(24), 7696–7701.

Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., Potter, S. C., Punta, M., Qureshi, M., Sangrador-Vegas, A., Salazar, G. A., Tate, J., and Bateman, A. (2016). The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, **44**(D1), D279–D285.

Fouts, D. E., Brinkac, L., Beck, E., Inman, J., and Sutton, G. (2012). PanOCT: automated clustering of orthologs using conserved gene neighborhood for pan-genomic analysis of bacterial strains and closely related species. *Nucleic Acids Research*.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J. Y. H., and Zhang, J. (2003). Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, **5**(10), R80–R80.

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, K. D., Irizarry, R. A., Lawrence, M., Love, M. I., MacDonald, J., Obenchain, V., Oleś, A. K., Pagès, H., Reyes, A., Shannon, P., Smyth, G. K., Tenenbaum, D., Waldron, L., and Morgan, M. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, **12**(2), 115–121.

Hyatt, D., Chen, G.-L., LoCascio, P. F., Land, M. L., Larimer, F. W., and Hauser, L. J. (2009). Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, **11**(1), 119–119.

Jun, S.-R., Wassenaar, T. M., Nookaew, I., Hauser, L., Wanchai, V., Land, M., Timm, C. M., Lu, T.-Y. S., Schadt, C. W., Doktycz, M. J., Pelletier, D. A., and Ussery, D. W. (2014). Diversity of Pseudomonas Genomes, Including Populus-Associated Isolates, as Revealed by Comparative Genome Analysis. *Applied and Environmental Microbiology*, **82**(1), 375–383.

Kaas, R. S., Friis, C., Ussery, D. W., and Aarestrup, F. M. (2011). Estimating variation within the genes and inferring the phylogeny of 186 sequenced diverse Escherichia coli genomes. *BMC Genomics*, **13**, 577–577.

Land, M., Hauser, L., Jun, S.-R., Nookaew, I., Leuze, M. R., Ahn, T.-H., Karpinets, T., Lund, O., Kora, G., Wassenaar, T., Poudel, S., and Ussery, D. W. (2015). Insights from 20 years of bacterial genome sequencing. *Functional & Integrative Genomics*, **15**(2), 141–161.

Leekitcharoenphon, P., Hendriksen, R. S., Le Hello, S., Weill, F.-X., Baggesen, D. L., Jun, S.-R., Ussery, D. W., Lund, O., Crook, D. W., Wilson, D. J., and Aarestrup, F. M. (2016). Global Genomic Epidemiology of Salmonella enterica Serovar Typhimurium DT104. *Applied and Environmental Microbiology*, **82**(8), 2516–2526.

Lex, A., Gehlenborg, N., Strobelt, H., Vuillemot, R., and Pfister, H. (2014). UpSet: Visualization of Intersecting Sets. *IEEE transactions on visualization and computer graphics*, **20**(12), 1983–1992.

Li, L., Stoeckert, C. J., and Roos, D. S. (2003). OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Research*, **13**(9), 2178–2189.

Li, W. and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**(13), 1658–1659.

Lukjancenko, O., Thomsen, M. C., Voldby Larsen, M., and Ussery, D. W. (2013). PanFunPro: PAN-genome analysis based on FUNctional PROfiles. *F1000Research*.

Méric, G., Yahara, K., Mageiros, L., Pascoe, B., Maiden, M. C. J., Jolley, K. A., and Sheppard, S. K. (2013). A reference pan-genome approach to comparative bacterial genomics: identification of novel epidemiological markers in pathogenic Campylobacter. *PLoS ONE*, **9**(3), e92798–e92798.

Page, A. J., Cummins, C. A., Hunt, M., Wong, V. K., Reuter, S., Holden, M. T. G., Fookes, M., Falush, D., Keane, J. A., and Parkhill, J. (2015). Roary: rapid large-scale prokaryote pan genome analysis. *Bioinformatics*, page btv421.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 3.2.4 edition.

Richards, V. P., Palmer, S. R., Bitar, P. D. P., Qin, X., Weinstock, G. M., Highlander, S. K., Town, C. D., Burne, R. A., and Stanhope, M. J. (2014). Phylogenomics and the dynamic genome evolution of the genus Streptococcus. *Genome Biology and Evolution*, **6**(4), 741–753.

Seret, M.-L. and Baret, P. V. (2011). IONS: Identification of Orthologs by Neighborhood and Similarity-an Automated Method to Identify Orthologs in Chromosomal Regions of Common Evolutionary Ancestry and its Application to Hemiascomycetous Yeasts. *Evolutionary bioinformatics online*, **7**, 123–133.

Snipen, L. G. and Liland, K. H. (2015). micropan: an R-package for microbial pan-genomics. *BMC Bioinformatics*, **16**(1), 79.

Snipen, L. G. and Ussery, D. W. (2012). A domain sequence approach to pangenomics: applications to Escherichia coli. *F1000Research*, **1**, 19.

Tettelin, H., Masignani, V., Cieslewicz, M. J., Donati, C., Medini, D., Ward, N. L., Angiuoli, S. V., Crabtree, J., Jones, A. L., Durkin, A. S., Deboy, R. T., Davidsen, T. M., Mora, M., Scarselli, M., Margarit y Ros, I., Peterson, J. D., Hauser, C. R., Sundaram, J. P., Nelson, W. C., Madupu, R., Brinkac, L., Dodson, R. J., Rosovitz, M. J., Sullivan, S. A., Daugherty, S. C., Haft, D. H., Selengut, J., Gwinn, M. L., Zhou, L., Zafar, N., Khouri, H., Radune, D., Dimitrov, G., Watkins, K., O'Connor, K. J. B., Smith, S., Utterback, T. R., White, O., Rubens, C. E., Grandi, G., Madoff, L. C., Kasper, D. L., Telford, J. L., Wessels, M. R., Rappuoli, R., and Fraser, C. M. (2005). Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome". *Proceedings of the National Academy of Sciences of the United States of America*, **102**(39), 13950–13955.