PanTA: An ultra-fast method for constructing large and growing microbial pangenomes

Duc Quang Le[1,2, †], Tien Anh Nguyen[1,3, †], Tam Thi Nguyen[4], Son Hoang Nguyen[1], Van Hoan Do[5], Canh Hao Nguyen[6], Huong Thanh Phung[3], Tho Huu Ho[7,8] , Vo Sy Nam[9,10], Trang Nguyen[1] , Hoang Anh Nguyen[1], and Minh Duc Cao[1, *]

[1]AMROMICS Working Group; [2]Faculty of IT, National University of Civil Engineering, Hanoi, Vietnam; [3]Faculty of Biotechnology, Hanoi University of Pharmacy, Hanoi, Vietnam; [4]Oxford University Clinical Research Unit, Hanoi, Vietnam; [5]Center for Applied Mathematics and Informatics, Le Quy Don Technical University; [6]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan; [7]Department of Medical Microbiology, The 103 Military Hospital, Vietnam Military Medical University, Vietnam; [8]Department of Genomics & Cytogenetics, Institute of Biomedicine & Pharmacy, Vietnam Military Medical University, Vietnam; [9]Center for Biomedical Informatics, Vingroup Big Data Institute, Hanoi, Vietnam; [10]College of Engineering and Computer Science, VinUniversity, Hanoi, Vietnam

[†]These authors contributed equally to this work.

[*]To whom correspondence should be addressed. E-mails: minhduc.cao@gmail.com

Abstract: Pangenome analysis has become indispensable in bacterial genomics due to the high variability of gene content between isolates within a clade. While many computational methods exist for constructing the pangenome from a bacterial genome collection, speed and scalability still remain an issue for the fast-growing genomic collections. Here, we present PanTA, a efficient method to build and analyze pangenomes of bacteria strains. We show that PanTA exhibits an unprecedented 10 times speed up and 2 times more memory efficient over the current state of the art methods. More importantly, PanTA enables the progressive pangenome construction where new samples are added into an existing pangenome without the need of rebuilding the accumulated collection from the scratch. The progressive building of pangenomes can further reduce the memory requirements by half. We demonstrate that PanTA can build the pangenome of the *Escherichia coli* species from the entire collection of over 28000 high quality genomes collected from the RefSeq database. Crucially, the whole analysis is performed on a modest laptop computer within two days, highlighting the scalability and practicality of PanTA.

Background

Prokaryotic genomes are known for enormous intraspecific variability owing to great variation events such as horizontal gene transfers, differential gene losses and gene duplication [1]. The pangenome concept that refers to the entire set of gene families of a given clade is hence introduced as a methodology to investigate the diversity of bacterial genomes [2]. Core genome, the set of essential gene families presenting in the majority of isolates in the clade, is considered to characterize the clade and to capture the phylogenetic information. On the other hand, the set of genes presenting in a smaller number of the organisms, namely accessory genome, provides a view of the variation and evolutionary trajectories [3]. Since the introduction in 2005, pangenome analysis has been an dispensable tool in microbial genomics studies[4] and have generated novel biological insights in bacterial population structures [5, 6], genetic diversity [7] and niche adaptation [8]. In responses to the recent rise of antimicrobial resistance, pangenome studies have also been successfully applied into inferring the evolution of lineages of pandemic causing pathogens and identifying lineage-specific genetic features [9, 10], investigating genetic signatures associated with antimicrobial resistance [11], pan-reactome analyses [12], and therapeutic development including vaccine design [13] and novel drug discovery [14, 15].

To meet the need of pangenome analysis, a plethora of computational methods have been developed to construct the pangenome of a collection of prokaryotic genomes with notable examples include PGAP [16], PanOCT [17], Roary [18], BPGA [19], panX [20], MetaPGN [21], PIRATE [22] and Panaroo [23]. The core of the pangenome inference is the clustering of gene sequences into gene families. This step is typically performed by first estimating the similarity between gene sequences by a homology search tool such as CD-HIT [24], BLASTP [25] and diamond [26] followed by a clustering method such as the commonly used Markov Clustering algorithm (MCL) algorithm [27]. The clustering step is also the most computational intensive of the pipeline. The gene families are further fine-tuned by the identification of paralogous genes with either a graph- or tree-based approach. The resulting gene clusters are then classified into core or accessory genes based on their prevalence in the collection.

Advances in high-throughut sequencing technologies have recently enabled the exponential growth of microbial genomics data in public databases and in research laboratories around the world. The Genbank database stores hundreds of thousands of genomes for common bacterial species, and the numbers are fast growing. While these resources contain the rich information of the population genomics, pangenome analysis has not been able to scaled with the volume of data. Most existing pangenome inference methods take days and require excessive amounts of memory that are typically beyond the capacity of a standard computer in order to construct the pangenome of a few thousand isolates. In addition, the genomic databases are growing by nature, accumulating genomes of isolates collected and sequenced at different time points. There

currently exists no efficient method to update an existing pangenome when new relevant genomes become available. In such cases, the pangenome of the accumulated collection has to be constructed from scratch over and over, leading to the additional burden of computational resources.

In order to address these considerations, we have developed PanTA, an efficient pangenome construction method to keep up with the scale of the current and future genomics data sources. With vigorous computational experiments, we demonstrate that PanTA shows an unprecedented 10-fold reduction of running time and 2-fold reduction of memory usage comparing with current state of the arts on building the pangenomes of large collections. Importantly, PanTA allows performing pangenome analysis progressively where new samples can be added into an existing pangenome without the need recomputing the accumulated pangenome from the scratch. The progressive mode can also be applied to construct large pangenomes as batches of genomes are gradually added into the growing pangenome. We show that the progressive mode can further reduce PanTA memory usage by half without affecting running time. Finally, we demonstrate the utility and practicality of PanTA by constructing the pangenome of the entire set of high quality *Escherichia coli* genomes deposited to RefSeq database to date on a laptop computer.

Results

Overview of PanTA pipeline

PanTA is developed with the aim to build the pangenome of a large collection of genomes, and to add a set of new genomes to an existing pangenome without rebuilding the accumulated pangenome from the scratch. The workflow of PanTA pipeline is summarized in Figure 1A. PanTA takes as input a list of genomes in gff3 format (a file containing gene annotations in gff format followed by genome assembly in fasta format). Alternatively, the genome can be represented by a pair of separate files for genome annotation and genome sequence. PanTA then extracts the protein coding regions as specified by the annotation, and translates these to protein sequences. In the process, it verifies and filters out coding regions that incorrectly annotated or that can potentially introduce noise into the clustering step and downstream analyses.

The core of the pipeline is the clustering of all genes in the collection into gene clusters, that represent the gene families in the collection. PanTA employs a similar clustering strategy as most recent pangenome methods such as Roary [18], PIRATE [22] and Panaroo [23]. It first runs CD-HIT [24] to group similar protein sequences together, and essentially reduces the set of all protein sequences to a smaller set of representative sequences from the groups. The default identity threshold for CD-HIT grouping is 98% and the value can be adjusted by users. PanTA then performs an all-against-all alignment of the representative sequences with diamond [26] or optionally BLASTP [25]. The resulting pairwise alignments are filtered to retain those that pass certain thresholds of sequence identity (default at 70%), alignment length ratios, and length difference ratio. These alignments are inputted into Markov clustering (MCL) [27] that clusters the representative sequences into homologous groups of genes. Each protein sequence is then assigned into the gene cluster its representative sequence belongs to.

PanTA can run in progressive mode where it adds new genomes into an existing pangenome without the rebuilding the pangenome from the scratch. In this mode, PanTA uses CD-HIT-2D, a tool in CD-HIT suite [24] to match new protein sequences extracted from the new samples to the representative sequences from the existing groups. The protein sequences that are matched are assigned into the existing groups and by proxy, to existing gene clusters while unmatched sequences are subject to CD-HIT to create new groups (Figure 1A). Similarly, during the all-against-all alignment step, PanTA first performs alignment of the representative sequences of the new groups against the representative sequences of the existing groups. It also runs all-against-all alignment of the new representative sequences (Figure 1B). The two sets of alignments after filtering are combined and are subject to MCL clustering. With the strategy, PanTA reduces the number of sequences in the grouping and alignment steps which are the most resource intensive steps of the whole pipeline and hence significantly speeds up the process.

Finally, PanTA performs post-processing steps where paralog genes are identified and split, and the genes in each clusters are multiple-aligned. PanTA employs the conserved gen neighbourhood (CGN) approach as described in [18]. Sequences of each gene cluster are aligned using MAFFT [28] at both DNA and protein levels. PanTA then generates output reports following the standard set out by Roary, including a spreadsheet detailing the presence and absence of each gene in each isolate and a summary statistics of the pangenome.

PanTA is significantly more efficient than existing pangenome inference methods

We evaluated the performance of PanTA and compared it with that of existing pan-genome construction methods on collections of bacteria genomes. We sourced the genomes of isolates from three bacterial species *Streptococcus pneumoniae*, *Pseudomonas aeruginosa* and *Klebsiella pneumoniae* that are known for harboring resistance to multiple drugs. These three species were chosen to cover a range of genome sizes and CG content as well as both gram positive and gram negative. We selected 600 *S. pneumoniae*, 800 *P. aeruginosa* and 1500 *K. pneumoniae* isolates to create three datasets, named Sp600, Pa800 and Kp1500, respectively (Table 1). We downloaded their genome assemblies from RefSeq database [29], and ran Prokka [30] to produce the gene annotations of these genomes in gff3 format. The gffs files were then used as the input to the pangenome construction process.

We applied PanTA and existing methods including Roary [18], PIRATE [22] and Panaroo [23] to the three genome collections. Other pangenome construction methods such as panX [20] and COGSoft [31] were reported to be prohibitively expensive to be applied to thousands of genomes [23] and were excluded from the comparison. We ran all pangenome inference methods with the default and recommended parameters. As these methods have different levels of rigorousness in splitting paralogs, we ran

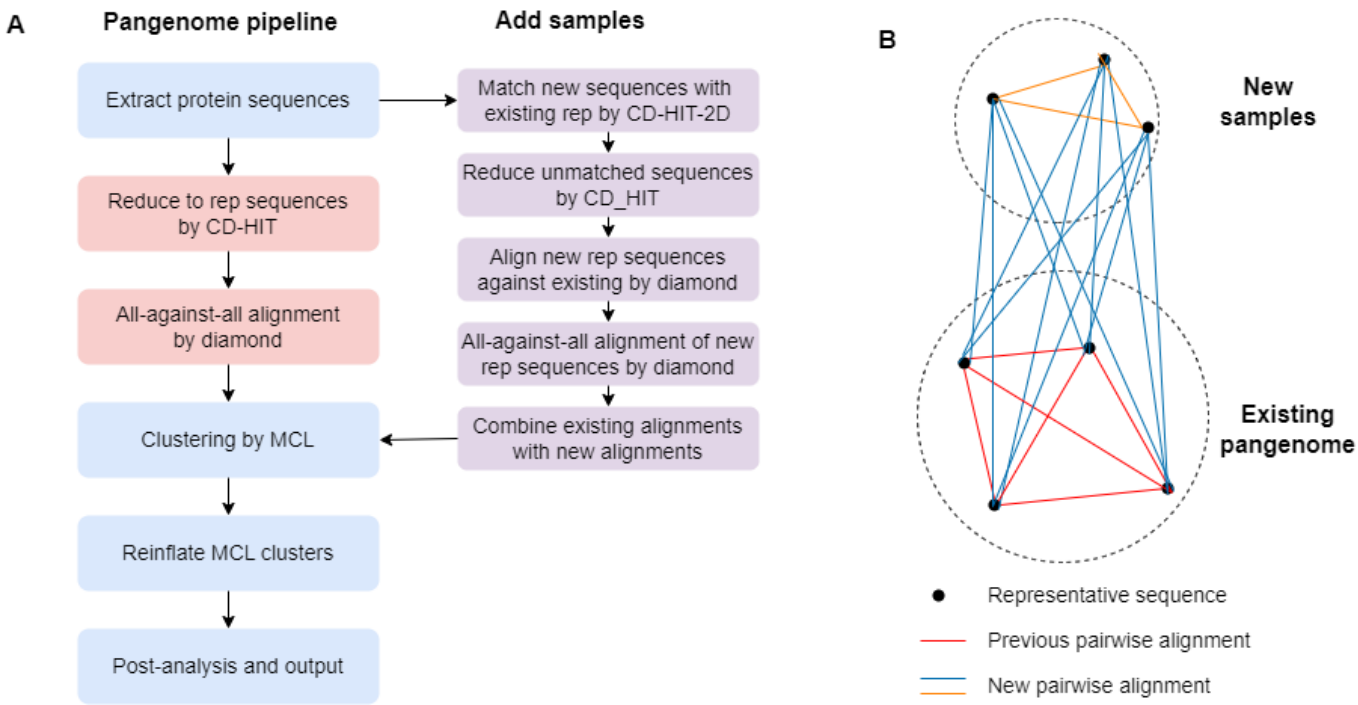


Figure 1: The schematic depict of PanTA workflow. **A.** The flowchart of PanTA pipeline in both single and progressive modes. In single model, the gene clustering process involves the reduction of protein sequences to representative gene sequences using CD-HIT, the all-against-all alignment of the representative sequences by diamond, and the MCL clustering. In progressive mode, new protein sequences are first matched with the existing representative sequences and only unmatched sequences are reduced to form new groups. Pairwise alignments are performed only between new representative sequences against existing representative sequences and among new representative sequences. **B.** Pairwise alignments of representative sequences during progressive mode. Only pairwise alignments of new representative sequences against themselves and against existing representative sequences are needed.

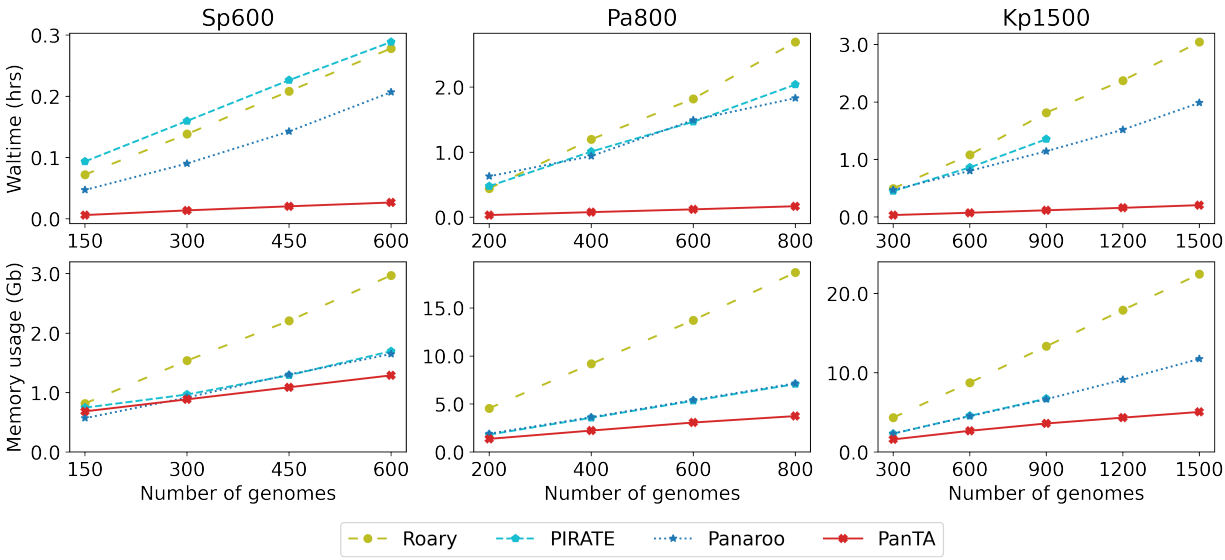
Dataset	Species	Number of genomes	Genome size	Ave. gene number	CG content	gram
Sp600	*S. pneumoniae*	600	2.0Mb	2.0k	40%	positive
Pa800	*P. aeruginosa*	800	6.1Mb	6.0k	67%	negative
Kp1500	*K. pneumoniae*	1500	5.4Mb	5.1k	57%	negative

Table 1: Characteristics of the three datasets to evaluate pangenome construction methods

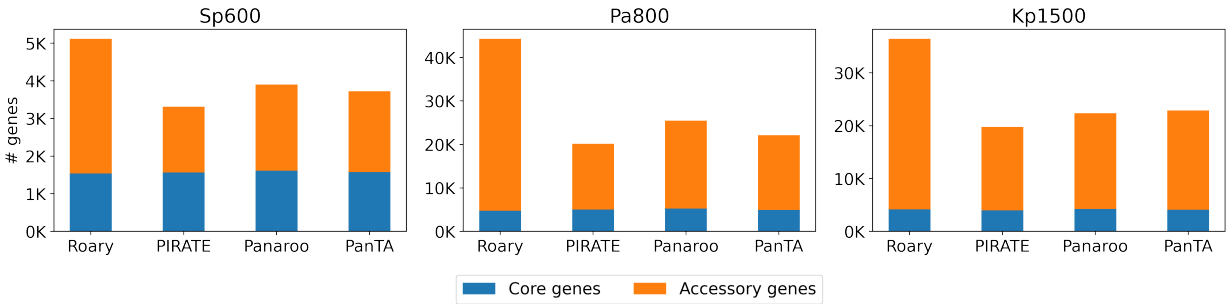
them without split paralog options. We excluded running alignment of genes clusters as these tools eventually call a multiple alignment method such as Mafft [28] for this. In order to access the performance of the methods on differing input sizes we ran them on the subsets of these collections. All computational experiments were run on a laptop computer equipped with a 20 hyper-thread CPU (Intel core I7-1280P) and 64Gb of memory running Ubuntu Linux 22.0. We ran these experiments on 20 threads and recorded the wall-time and the peak memory usage of all the runs.

Figure 2a shows the computational resources in wall-time and peak memory against the size of the genome collection for the competing pangenome inference methods on the three datasets. We noted that PIRATE did not complete constructing the pangenomes for the sets of 1200 and 1500 *K. pneumoniae* genomes and hence the results for these runs are not included in the figure. We observed that all methods exhibited a linear increase in time and memory usage against the input size. Generally, Panaroo was faster than Roary by about 30%-40% while PIRATE's wall-times were somewhere between these two methods. Strikingly, we found PanTA was significantly faster than the competing methods, about 14 times faster than Roary, and 10 times faster than the next best method Panaroo (Figure 2a, top panel). Specifically, it took 0.168 hours and 0.207 hours to build the pangenomes for 800 *P. aeruginosa* and 1500 *K. pneumoniae* genomes, respectively, while the corresponding times for Roary were 2.70 hours and 3.04 hours, and for Panaroo were 1.83 and 1.98 hours.

In terms of memory usage, Roary required the most amounts of memory, consuming at least two times more than all other methods. Pirate and Panaroo had similar memory usage patterns. In small data sets such as that with 300 or fewer *S. pneumoniae* genomes (less than 60,000 genes in total), PanTA needed similar amounts of memory as Panaroo and PIRATE. However, as the dataset size increased, PanTA required only a small fraction of that needed by Panaroo and Pirate (Figure 2a, bottom panel). For example, for 1500 *K. pneumoniae* genomes (about 8 million genes), PanTA needed only 5.07Gb of memory



(a) Wall-times and memory usages of the competing methods



(b) The number of core-genes and accessory genes in the pangenomes constructed by the competing methods

Figure 2: The performance of PanTA and existing methods on the three data collections

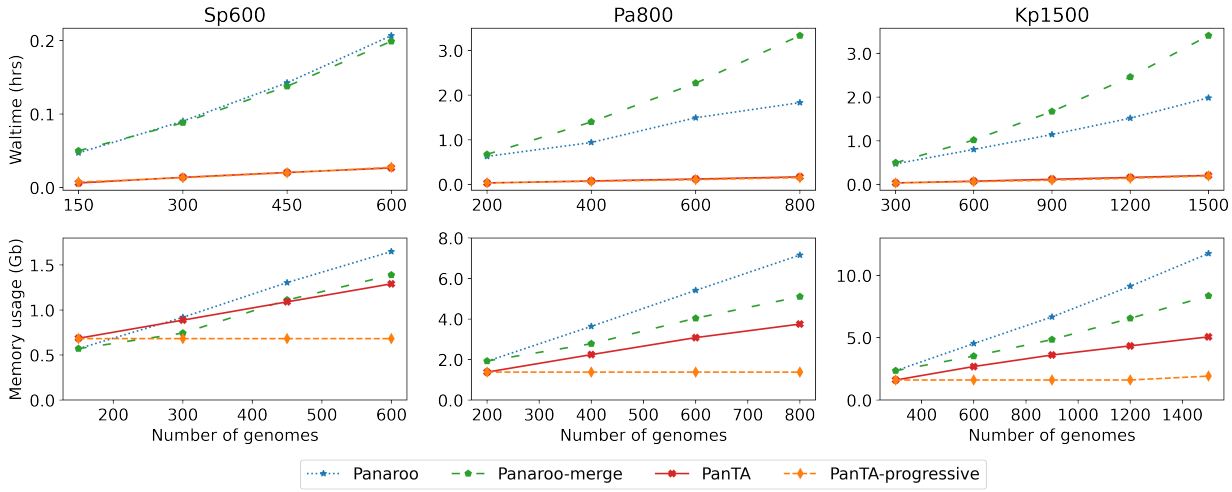
which is less than half of Panaroo's peak memory (11.8Gb) and a quarter of Roary (22.4Gb).

Figure 2b presents the numbers of core genes and of accessory genes in the pangenomes inferred by the completing methods. Note that for the Kp1500 dataset, because PIRATE did not complete constructing the pangenomes beyond 900 genomes, we show the statistics from the pangenome constructed from the 900 genomes for a fair comparison of all four methods. We observed that the pangenomes produced by Panaroo and PanTA contained consistent number of gene families as the result of the same sequence identity threshold (70%). Roary, which used a higher threshold (95%) resulted in much higher gene clusters in its inferred pangenomes. On the other hand, PIRATE applied a series of thresholds ranging from 50% to 95% giving rise to the smallest number of gene clusters. All the methods however inferred the similar numbers of core genes, in that PanTA pangenomes reported within 5% of core genes with the corresponding pangenomes produced by the other methods.

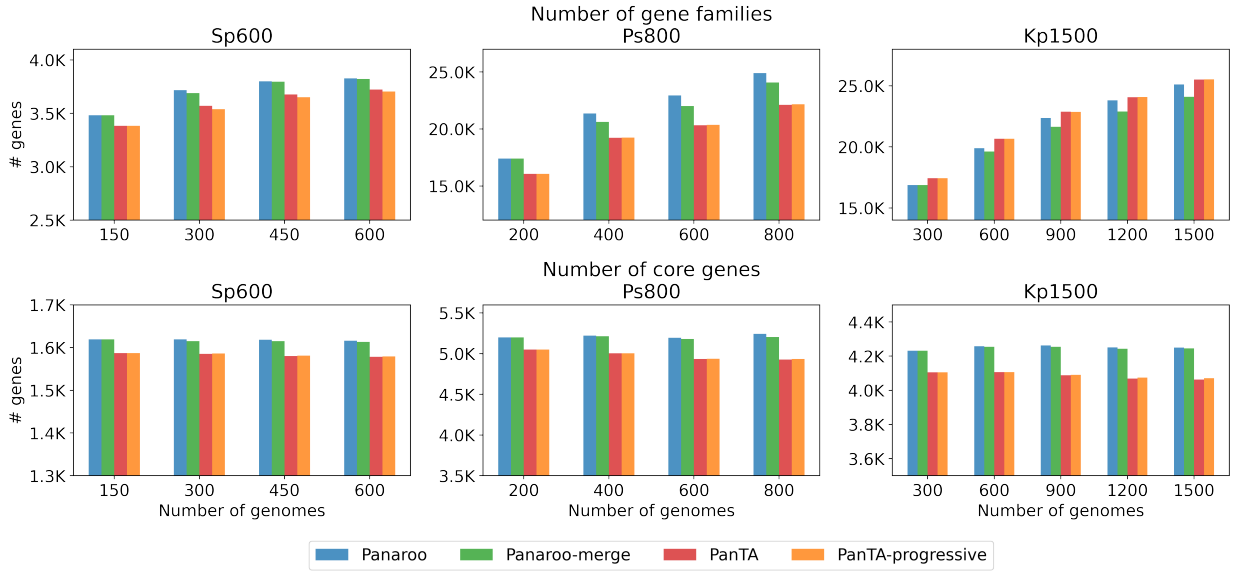
PanTA progressively builds pangenome

We next evaluated the performance of PanTA in progressive mode where new samples are added into an existing pangenome without the need of rebuilding the pangenome from the scratch. For each of the aforementioned datasets, we ran PanTA to construct the pangenome of the smallest partition, and progressively added the genomes of the subsequent partitions into the pangenome. We noticed Panaroo also offered a similar functionality, namely Panaroo-merge, that merges the pangenomes of multiple collections together. For comparison, we ran Panaroo on each partition of the dataset, and then applied Panaroo-merge to merge the partition collections together. In these experiments, we collected the wall-times for each pangenome as the sum of the wall-time of each step, and the peak memory usage as the maximum amount of memory at each step. Figure 3 presents the computational resources consumed by both methods on the three datasets. We also included the resources needed by both methods when computing the pangenomes from the scratch as part of the comparison. As presented in the Figure, Panaroo-merge improved memory usage by 20% over the Panaroo at the cost of 70% longer running time. PanTA in progressive mode took similar running times to PanTA in single mode, and further reduced memory usage by half. All in all, PanTA in progressive mode consumed only 25% and 15% of the amount of memory required by Panaroo and Panaroo-merge, respectively, while was 10 and 17 times faster.

We assessed the consistency of the pangenomes constructed by each of the methods in single mode and in progressive/merge modes. Figure 3b presents the number of gene families (in top panel) and the number of core genes in the pangenomes. As indicated, the numbers of



(a) Computational resources in progressive



(b) The number of core-genes and accessory genes in the pangenomes constructed by the competing methods

Figure 3: The performance of PanTA and existing methods on the three data collections

gene clusters representing inferred gene families increase with the numbers of genomes added into the pangenomes by both methods and in both running modes. We found that the numbers of gene families in both configurations for Panaroo diverged the genome collection increased. Specifically, in the set of 1500 *K. pneumoniae* genomes, they differed by above 4%. On the other hand, PanTA produced pangenomes within 0.1% different from each other, indicating the stability of the clustering method. The number of core genomes were consistent (less than 0.1% divergence) for both methods (Figure 3b, bottom panel).

Building the pangenome of a growing genome collection

The primary goal of PanTA is to analyze and manage the large and fast-growing collections of microbial genomes. We demonstrate this utility by applying PanTA to a realistic collection of bacterial genomes that is growing over time. To this end, we collected all *Escherichia coli* genomes that are deposited into the RefSeq database [29] during the three years 2020, 2021 and 2022. *E. coli* is one of the most well-studied model prokaryotic organisms and is known for its genotypic diversity and pathogenic for both human and animals [32]. Some *E. coli* strains have recently drawn attention from public health for causing outbreaks of multidrug-resistant [33]. After removing outliers, we obtained a dataset of 12,560 genomes (Methods). To illustrate the growing nature of the dataset, we grouped the samples in the quarters in which they were released. Table 2 shows the breakdown of the samples.

We ran PanTA to build the initial pan-genome of the genomes collected in the first quarter. We then progressively add genomes from subsequent quarters into the collection. For comparison with PanTA in single mode, we also ran PanTA on the accumulated data at each quarter. As shown in Figure 4, PanTA in progressive mode needed only 16.6Gb of memory to construct the pangenome for 12000 *E. coli* genomes while the single mode consumed 30.1Gb of memory. Both modes exhibited similar running times, around 6.5 hours.

Quarter	#isolates	#isolates accum.
Q1-2020	534	534
Q2-2020	713	1247
Q3-2020	830	2077
Q4-2020	1109	3186
Q1-2021	1166	4352
Q2-2021	694	5046
Q3-2021	1411	6457
Q4-2021	866	7,323
Q1-2022	1645	8,968
Q2-2022	1233	10,201
Q3-2022	1214	11,415
Q4-2022	1145	12,560

Table 2: Number of *E. coli* samples deposited into RefSeq database between 2020-2022 by quarter

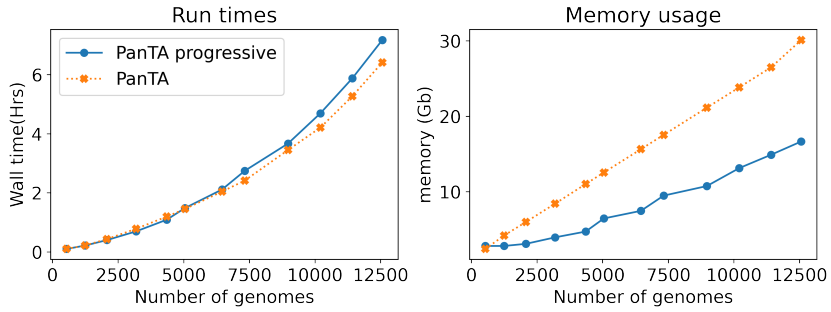


Figure 4: Time and memory required for constructing the pangenome for *E. coli*

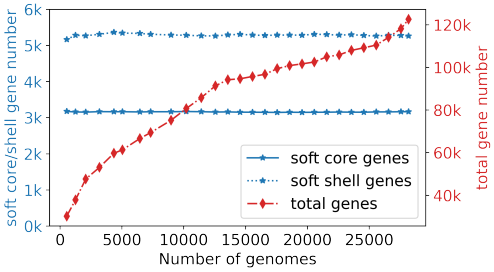


Figure 5: The numbers of soft core genes, soft shell genes and total genes across different number of genomes. Soft core genes are those that

The pangenome of E. coli species

Encouraged by the scalability of PanTA, we proceeded to build the pangenome for the entire set of *E. coli* genomes from the RefSeq database. We downloaded all *E. coli* genomes that were released prior to 2020, and after filtering outliers, we obtained 15,625 genomes in addition to the previous collected set (Methods). We divided these genomes into batches of maximum 1000 genomes each, and iteratively added these batches into the *E. coli* pangenome with PanTA-progressive. In effect, we constructed the pangenome of all 28,275 high quality *E. coli* genomes from the RefSeq database. Strikingly, the pangenome of *E. coli* species was inferred on a laptop computer with the total time of 47.3 hours, including the time to build the pangenome from the three recent years. The peak memory recorded during the pangenome construction was 39.9Gb.

The pangenome of the *E. coli* species built from the large and diverse collection of genomes enabled the analysis of the dynamics and diversity of the species. We revisited the analysis on by Tantoso *et al.* [32] and observed that while the number of total gene clusters increases sharply to over 120,000 genes on 28,000 genomes, the number of soft core genes (genes present in at least 95% of the isolates) remains at appropriately 3170 genes (Figure 5). This observation is consistent with previous analyses of the *E. coli* pangenome [34, 35]. We further examined the number of soft shell genes, which are defined as genes presents in at least 15% of the isolates. Interestingly, we found the number of soft shell genes also saturates at around 5280 genes, further validating the hypothesis proposed by [32, 35] that the soft core genome of *E. coli* can be defined with a generation threshold of up to 95%.

Discussion

Bacteria are among the most diverse life forms on earth, evidence by the high level of variability of gene content across strains in a species. It is therefore not possible to use the genome of a single isolate as a reference genome to represent a clade. Pangenome analysis offers an alternative approach where all gene families of the clade constitute the pangenome that representing the total diversity of the clade.

Most computational methods for pangenome construction usually apply clustering of gene sequences. These methods in most cases run multiple times of CD-HIT clustering on different level of sequence similarity in order to achieve stability of clustering. In developing PanTA, we use only one round of CD-HIT clustering and yet we obtain the near identical pangenomes with existing tools on the same sequence identity threshold. PanTA is shown to be 10 times faster than requires less than half of the memory consumed by the current state of the arts.

The bacterial genome collections are growing by nature as more and more genomes are routinely sequenced in laboratories and in research and medical settings around the whole. PanTA addresses the complexity of rebuilding pangenomes by providing the progressive mode where new genomes are added into an existing pangenome. By utilizing the group membership information of the existing clustering, PanTA needs to compare the genes in the new genomes with existing groups and thereby are significantly faster than rebuilding the pangenomes. Interesting, we found that building the pangenome progressively from batches of genomes takes a similar amount of time to building from the whole collection, and at the same time, reduces the memory requirements by half, making PanTA suitable for the practical use.

The scalability of PanTA is demonstrated by the ability to construct the pangenome for a bacterial species from the entire set of *E. coli* genomes from RegSeq database on a laptop computer in a unprecedented 2-day time. PanTA can construct the pangenome progressively in when new samples are added into the collection, without recomputing the accumulated collection from the scratch. This further saves times and memory, and is practically suitable for analysis of the large growing collections of bacteria in the sequencing ages.

Methods

Pan-genome pipeline

PanTA accepts input genomes in GFF3 files which store gene annotations in gff format followed by the genome assembly in fasta format. This format is the output from Prokka [30] and has been popularized for pangenome analysis started by Roary [18]. Each genome is associated with a unique ID which can be input by user or otherwise generated by PanTA. The ID of each contig in the genome, and each annotated coding is also required to be unique. Coding sequences are extracted and are translated into protein sequences. Coding sequences that are less than 120 nucleotides in length or lack both start and stop codon will be excluded. Protein sequences having more than 5% of unknown amino acid will also be removed. Next, a fast pre-clustering is performed using CD-HIT [24] with the identity threshold of 98%. The CD-HIT's representative sequences are compared all-against-all by diamond [26] or BLASTP [25]. The e-value threshold is set to 10e-6 by default. To reduce the all-against-all alignment time, the list of representative sequences are chucked up into smaller files to perform matching in parallel. The identified matches are filtered to retain those with sequence identity over a threshold (default at 70%). The diamond result is then put into MCL[27], which uses a normalised bit score for clustering with inflation value being 1.5. Finally, removed sequences in CD-HIT step are merged back into MCL clusters.

Add samples pipeline

Firstly, protein sequences of new samples are compared and matched with CD-HIT's representative sequences from the previous collection. This is performed by CD-HIT-2D with the identity threshold of 98%. The protein sequences that are matched to a representative sequence is assigned to the represented group. The unmatched sequences are clusterred by CD-HIT to create new groups with new representative sequences. The new representative sequences are then subject to all-against-all alignment by the alignment method of choice, *i.e.*, diamond or BLASTP. The new representative sequences are also aligned against the existing representative sequences. The two sets of alignments are then filtered according the criteria, and are then combined with the existing set of alignments in the pangenome. Finally, MCL is applied to the combined set of alignments as above.

Annotating clusters

For each cluster, PanTA maintains a list of all the gene names and gene products of all genes in the cluster. It also keeps gene length statistics such as the number of genes, the minimum length, the maximum length and the average gene length in the cluster. The cluster is assigned a name taken from one of the genes that are annotated. The gene product for the cluster is the concatenation of all the gene products of the gene members. PanTA also picks the longest gene sequence as to be the representative sequence for the cluster.

Post-processing and output

PanTA presents the pangenome following the standard set out by Roary. Specifically, the presence and absense of genes in each sample are presented in a csv format file and a Rtab format file. Upon users' request, PanTA performs multiple alignment of all gene clusters on both In addition, PanTA stores the existing all-against-all alignments and the existing CD-HIT groupings for subsequent analyses.

Data collection for the E. coli dataset

The set of genomes available on RefSeq database was downloaded from https://ftp.ncbi.nlm.nih.gov/genomes/refseq/assembly_summary_refseq.txt (accessed Feb 22 2023). We selected only genomes of samples belonging to *E. coli* species. The genome sequence (fna file) and genome annotation (gff file) for each sample were downloaded and combined to generate a gff3 format file. Coding sequences that were shorter than 120bp or contained non-canonical nucleotides were removed. To remove outliers, we inspected the histograms of genome sizes, number of genes and N50 statistics and selected genomes that were between 4.2Mb and 5.9Mb long, contained 4200 and 5500 genes and having N50 statistics of 50kb or higher. These genomes were grouped into quarters based on their release dates.

Availability of source code and requirements

- Project name: PanTA
- Project home page: e.g. https://github.com/amromics/panta
- Operating system(s): Platform independent
- Programming language: Python
- License: BSD

Declarations

Competing Interests

The author(s) declare that they have no competing interests

Funding

This work has been supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA11.

References

[1] McInerney, J.O., McNally, A., O'Connell, M.J.: Why prokaryotes have pangenomes. Nature Microbiology **2**(4), 17040 (2017). doi:10.1038/nmicrobiol.2017.40

[2] 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.M., 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., Fraser, C.M.: Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae : Implications for the microbial "pan-genome". Proceedings of the National Academy of Sciences **102**(39), 13950–13955 (2005). doi:10.1073/pnas.0506758102

[3] Cummins, E.A., Hall, R.J., Connor, C., McInerney, J.O., McNally, A.: Distinct evolutionary trajectories in the Escherichia coli pangenome occur within sequence types. Microbial Genomics **8**(11), 1–13 (2022). doi:10.1099/mgen.0.000903

[4] Kim, Y., Gu, C., Kim, H.U., Lee, S.Y.: Current status of pan-genome analysis for pathogenic bacteria. Current Opinion in Biotechnology **63**, 54–62 (2020). doi:10.1016/j.copbio.2019.12.001

[5] Pinto, M., González-Díaz, A., Machado, M.P., Duarte, S., Vieira, L., Carriço, J.A., Marti, S., Bajanca-Lavado, M.P., Gomes, J.P.: Insights into the population structure and pan-genome of Haemophilus influenzae. Infection, Genetics and Evolution **67**, 126–135 (2019). doi:10.1016/j.meegid.2018.10.025

[6] Freschi, L., Vincent, A.T., Jeukens, J., Emond-Rheault, J.-G., Kukavica-Ibrulj, I., Dupont, M.-J., Charette, S.J., Boyle, B., Levesque, R.C.: The Pseudomonas aeruginosa Pan-Genome Provides New Insights on Its Population Structure, Horizontal Gene Transfer, and Pathogenicity. Genome Biology and Evolution **11**(1), 109–120 (2019). doi:10.1093/gbe/evy259

[7] Cai, H., McLimans, C.J., Beyer, J.E., Krumholz, L.R., Hambright, K.D.: Microcystis pangenome reveals cryptic diversity within and across morphospecies. Science Advances **9**(2), 1–11 (2023). doi:10.1126/sciadv.add3783

[8] Lu, Q.-F., Cao, D.-M., Su, L.-L., Li, S.-B., Ye, G.-B., Zhu, X.-Y., Wang, J.-P.: Genus-Wide Comparative Genomics Analysis of Neisseria to Identify New Genes Associated with Pathogenicity and Niche Adaptation of Neisseria Pathogens. International Journal of Genomics **2019**, 1–19 (2019). doi:10.1155/2019/6015730

[9] Domman, D., Quilici, M.-L., Dorman, M.J., Njamkepo, E., Mutreja, A., Mather, A.E., Delgado, G., Morales-Espinosa, R., Grimont, P.A.D., Lizárraga-Partida, M.L., Bouchier, C., Aanensen, D.M., Kuri-Morales, P., Tarr, C.L., Dougan, G., Parkhill, J., Campos, J., Cravioto, A., Weill, F.-X., Thomson, N.R.: Integrated view of Vibrio cholerae in the Americas. Science **358**(6364), 789–793 (2017). doi:10.1126/science.aao2136

[10] Chung The, H., Karkey, A., Pham Thanh, D., Boinett, C.J., Cain, A.K., Ellington, M.J., Baker, K.S., Dongol, S., Thompson, C., Harris, S.R., Jombart, T., Le Thi Phuong, T., Tran Do Hoang, N., Ha Thanh, T., Shretha, S., Joshi, S., Basnyat, B., Thwaites, G., Thomson, N.R., Rabaa, M.A., Baker, S.: A high-resolution genomic analysis of multidrug-resistant hospital outbreaks of Klebsiella pneumoniae. EMBO Molecular Medicine **7**(3), 227–239 (2015). doi:10.15252/emmm.201404767

[11] Kavvas, E.S., Catoiu, E., Mih, N., Yurkovich, J.T., Seif, Y., Dillon, N., Heckmann, D., Anand, A., Yang, L., Nizet, V., Monk, J.M., Palsson, B.O.: Machine learning and structural analysis of Mycobacterium tuberculosis pan-genome identifies genetic signatures of antibiotic resistance. Nature Communications **9**(1), 4306 (2018). doi:10.1038/s41467-018-06634-y

[12] Genome-scale metabolic reconstructions of multiple Salmonella strains reveal serovar-specific metabolic traits. Nature Communications **9**(1), 3771 (2018). doi:10.1038/s41467-018-06112-5

[13] Zeng, L., Wang, D., Hu, N., Zhu, Q., Chen, K., Dong, K., Zhang, Y., Yao, Y., Guo, X., Chang, Y.-F., Zhu, Y.: A Novel Pan-Genome Reverse Vaccinology Approach Employing a Negative-Selection Strategy for Screening Surface-Exposed Antigens against leptospirosis. Frontiers in Microbiology **8** (2017). doi:10.3389/fmicb.2017.00396

[14] Doron, S., Melamed, S., Ofir, G., Leavitt, A., Lopatina, A., Keren, M., Amitai, G., Sorek, R.: Systematic discovery of antiphage defense systems in the microbial pangenome. Science **359**(6379) (2018). doi:10.1126/science.aar4120

[15] Bhardwaj, T., Somvanshi, P.: Pan-genome analysis of Clostridium botulinum reveals unique targets for drug development. Gene **623**, 48–62 (2017). doi:10.1016/j.gene.2017.04.019

[16] Zhao, Y., Wu, J., Yang, J., Sun, S., Xiao, J., Yu, J.: PGAP: pan-genomes analysis pipeline. Bioinformatics **28**(3), 416–418 (2012). doi:10.1093/bioinformatics/btr655

[17] Fouts, D.E., Brinkac, L., Beck, E., Inman, J., Sutton, G. Nucleic Acids Research **40**(22), 172–172 (2012). doi:10.1093/nar/gks757

[18] Page, A.J., Cummins, C.A., Hunt, M., Wong, V.K., Reuter, S., Holden, M.T.G., Fookes, M.,

Falush, D., Keane, J.A., Parkhill, J.: Roary: rapid large-scale prokaryote pan genome analysis. Bioinformatics **31**(22), 3691–3693 (2015). doi:10.1093/bioinformatics/btv421

[19] Chaudhari, N.M., Gupta, V.K., Dutta, C.: BPGA-an ultra-fast pan-genome analysis pipeline. Scientific Reports **6**(1), 24373 (2016). doi:10.1038/srep24373

[20] Ding, W., Baumdicker, F., Neher, R.A.: panX: pan-genome analysis and exploration. Nucleic Acids Research **46**(1), 5–5 (2018). doi:10.1093/nar/gkx977

[21] Peng, Y., Tang, S., Wang, D., Zhong, H., Jia, H., Cai, X., Zhang, Z., Xiao, M., Yang, H., Wang, J., Kristiansen, K., Xu, X., Li, J.: MetaPGN: a pipeline for construction and graphical visualization of annotated pangenome networks. GigaScience **7**(11), 1–11 (2018). doi:10.1093/gigascience/giy121

[22] Bayliss, S.C., Thorpe, H.A., Coyle, N.M., Sheppard, S.K., Feil, E.J.: PIRATE: A fast and scalable pangenomics toolbox for clustering diverged orthologues in bacteria. GigaScience **8**(10), 1–9 (2019). doi:10.1093/gigascience/giz119

[23] Tonkin-Hill, G., MacAlasdair, N., Ruis, C., Weimann, A., Horesh, G., Lees, J.A., Gladstone, R.A., Lo, S., Beaudoin, C., Floto, R.A., Frost, S.D.W., Corander, J., Bentley, S.D., Parkhill, J.: Producing polished prokaryotic pangenomes with the Panaroo pipeline. Genome Biology **21**(1), 180 (2020). doi:10.1186/s13059-020-02090-4

[24] Li, W., Godzik, A.: Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics **22**(13), 1658–1659 (2006). doi:10.1093/bioinformatics/btl158

[25] Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., Madden, T.L.: BLAST+: architecture and applications. BMC Bioinformatics **10**(1), 421 (2009). doi:10.1186/1471-2105-10-421

[26] Buchfink, B., Xie, C., Huson, D.H.: Fast and sensitive protein alignment using DIAMOND. Nature Methods **12**(1), 59–60 (2015). doi:10.1038/nmeth.3176

[27] Enright, A.J.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research **30**(7), 1575–1584 (2002). doi:10.1093/nar/30.7.1575

[28] Nakamura, T., Yamada, K.D., Tomii, K., Katoh, K.: Parallelization of MAFFT for large-scale multiple sequence alignments. Bioinformatics **34**(14), 2490–2492 (2018). doi:10.1093/bioinformatics/bty121

[29] Li, W., O'Neill, K.R., Haft, D.H., DiCuccio, M., Chetvernin, V., Badretdin, A., Coulouris, G., Chitsaz, F., Derbyshire, M.K., Durkin, A.S., Gonzales, N.R., Gwadz, M., Lanczycki, C.J., Song, J.S., Thanki, N., Wang, J., Yamashita, R.A., Yang, M., Zheng, C., Marchler-Bauer, A., Thibaud-Nissen, F.: RefSeq: expanding the Prokaryotic Genome Annotation Pipeline reach with protein family model curation. Nucleic Acids Research **49**(D1), 1020–1028 (2021). doi:10.1093/nar/gkaa1105

[30] Seemann, T.: Prokka: rapid prokaryotic genome annotation. Bioinformatics **30**(14), 2068–2069 (2014). doi:10.1093/bioinformatics/btu153

[31] Kristensen, D.M., Kannan, L., Coleman, M.K., Wolf, Y.I., Sorokin, A., Koonin, E.V., Mushegian, A.: A low-polynomial algorithm for assembling clusters of orthologous groups from intergenomic symmetric best matches. Bioinformatics **26**(12), 1481–1487 (2010). doi:10.1093/bioinformatics/btq229

[32] Tantoso, E., Eisenhaber, B., Kirsch, M., Shitov, V., Zhao, Z., Eisenhaber, F.: To kill or to be killed: pangenome analysis of Escherichia coli strains reveals a tailocin specific for pandemic ST131. BMC Biology **20**(1), 146 (2022). doi:10.1186/s12915-022-01347-7

[33] Forde, B.M., Zowawi, H.M., Harris, P.N.A., Roberts, L., Ibrahim, E., Shaikh, N., Deshmukh, A., Sid Ahmed, M.A., Al Maslamani, M., Cottrell, K., Trembizki, E., Sundac, L., Yu, H.H., Li, J., Schembri, M.A., Whiley, D.M., Paterson, D.L., Beatson, S.A.: Discovery of mcr-1 -Mediated Colistin Resistance in a Highly Virulent Escherichia coli Lineage. mSphere **3**(5), 1–11 (2018). doi:10.1128/mSphere.00486-18

[34] Kaas, R.S., Friis, C., Ussery, D.W., Aarestrup, F.M.: Estimating variation within the genes and inferring the phylogeny of 186 sequenced diverse Escherichia coli genomes. BMC Genomics **13**(1), 577 (2012). doi:10.1186/1471-2164-13-577

[35] Gordienko, E.N., Kazanov, M.D., Gelfand, M.S.: Evolution of Pan-Genomes of Escherichia coli, Shigella spp., and Salmonella enterica. Journal of Bacteriology **195**(12), 2786–2792 (2013). doi:10.1128/JB.02285-12