SOFTWARE

KOMB: Taxonomy-oblivious Characterization of Metagenome Dynamics via K-core Decomposition

Advait Balaji[1], Nicolae Sapoval[1], R. A. Leo Elworth[1], Santiago Segarra[1,2] and Todd J. Treangen[1*]

*Correspondence: treangen@rice.edu

[1]Department of Computer Science, Rice University, 6100 Main St, 77005 Houston, Texas, USA

Full list of author information is available at the end of the article

Abstract

Background: Taxonomic classification of microbiomes has provided tremendous insight into the underlying genome dynamics of microbial communities but has relied on known microbial genomes contained in curated reference databases.

Methods: We propose K-core graph decomposition as a novel approach for tracking metagenome dynamics that is taxonomy-oblivious. K-core performs hierarchical decomposition which partitions the graph into shells containing nodes having degree at least K called K-shells, yielding $O(E+V)$ complexity.

Results: The results of the paper are two-fold: (1) KOMB can identify homologous regions efficiently in metagenomes, (2) KOMB reveals community profiles that capture intra- and inter-genome dynamics, as supported by our results on simulated, synthetic, and real data.

Software Availability: KOMB is available for use on Linux systems at https://gitlab.com/treangenlab/komb.git

Keywords: De Bruijn graph; graph-based analysis; K-core decomposition; metagenome; microbiome; unitigs

Background

Graph-based representations and analyses paved the way for several advances in computational biology over the last few decades [1–3]. This is particularly evident in the progress made in the field of genome assembly, both for isolate genome assembly [4, 5] and metagenome assembly, as well as efficient detection of structural variants [6–8] using genome graphs [9–11]. Indeed, state-of-the-art graph-based metagenome assemblers [12–15] have achieved remarkable improvements in both run-time and accuracy in recent years [16] through the use of efficient data

structures and clever heuristics. Recent examples include compact De Bruijn graph construction and traversal for assembly [17,18] as well as scaffold graphs for metagenomic samples that can generate scaffolds from contiguous overlapping sequences (contigs), which are then stitched together by using paired-end read information to obtain the complete genome [19,20]. There has also been a recent emphasis towards constructing De Bruijn graphs that encode the underlying metagenomic population information such as succinct colored De Bruijn graphs [21,22,22].

Despite recent advances, genome assembly remains challenging due to the presence of repetitive sequences and sequencing error, both of which confound graph traversal needed to generate consensus sequences [23]. This occurs in part due to the presence of repetitive sequences in the genome that tangle the assembly graph resulting in nodes with high degrees. This assembly graph tangling creates a non-trivial graph traversal problem [24]. This is further exacerbated when dealing with metagenomes as the sequences can contain intra-genomic repeats as well as inter-genomic homology. Distinguishing paralogs from orthologs, and repeats from homologs can be challenging, especially if the sample is enriched for closely-related species or strains. Assemblers or scaffolders often deal with resolving this ambiguity in two ways; either by assuming that branches in the graph were a result of base calling errors and hence collapsing the node, or by stopping the traversal to reveal a fragmented stretch of unique contiguous subsequences of the genome (unitig) [25,26]. Thus, for optimal assembly, it becomes imperative to correctly identify sequences that are part of these tangled nodes.

Another area where repetitive regions play a confounding role is in the identification of genomic variations in a large metagenomic sample. Inter-genomic homology can often link unrelated regions of different but closely related genomes. In addition to the repetitive regions, non-uniform coverage can result in an increase in false positive repeats as core genome regions of highly abundant species can be labelled as repeats [27]. Distinguishing and separating out these genomic regions with varying degrees of similarities and differences becomes a crucial step for any downstream metagenomic analysis and allows for careful tracking of genomic diversity within the sample [28–30].

A popular solution that has emerged in the literature is to identify tangled nodes caused by these different phenomena using the concept of node centrality on graphs.

More specifically, the general idea is to employ node centrality measures to separate high-similarity nodes from non-repeat nodes. Since tangled nodes have on average a larger degree than other nodes and are well-connected within the graph, it is reasonable to employ centrality measures to identify these nodes. In this context, tools like MetaCarvel [31] and Bambus2 [25] are examples of methods developed by the community grounded on the idea of centrality-based repeat detection for the specific case of betweenness centrality [32,33]. Although methods based on betweenness centrality can achieve high levels of specificity, they tend to miss out on multiple repetitive regions leading to loss in sensitivity [24]. Another fundamental drawback of betweenness centrality is its high computational complexity $O(VE)$ [34,35], where V denotes the number of nodes and E the number of edges in the graph, making impractical its implementation on large metagenomic datasets. To alleviate these concerns, a recent method employs an approximate betweenness centrality measure [36, 37] to improve the scalability of the approach. The approximate betweenness centrality relies on subsampling the nodes in the graph to estimate betweenness centralities in the complete graph. While approximate betweenness centrality is in practice an order of magnitude faster than the exact counterpart, it depends on thresholding strategies to achieve good levels of specificity. Moreover, it was later shown [24] that an ensemble approach using a random forest classifier and various features from the contig graph (including coverage, contig length, and centrality) resulted in a slight improvement over using just betweenness centrality as a measure of repeat detection.

In parallel to repeat and homolog detection, there has been an increasing need in the research community for methods that visually and quantitatively identify microbial community structures and sequence diversity among the organisms present in metagenomic samples, particularly in response to perturbations [38]. A tool that can accurately and efficiently identify and extract information from assembly, contig, or unitig graphs in an intuitive and theoretically grounded framework could help biologists understand and characterize microbial communities using repeats and homologous regions of the organisms in their samples [39–42].

In this work we present KOMB, a tool that does not rely on reference databases and can capture highly-connected repetitive regions in an entire genome or a metagenomic community. We present a novel way of achieving this using the K-

core decomposition of a unitig graph that hierarchically separates out repetitive regions into various shells that can then be used to analyze genomic variation in the sample. We show that the distribution of nodes could lead to a new methodology that describes metagenomic community structure based on sample specific signatures obtained from KOMB profiles. In Methods, found towards the end of the manuscript, we describe the pipeline of the tool, explain unitig graph construction, and elaborate on the concept of K-core decomposition. In the Results section, we provide a rigorous validation of our novel K-core decomposition tool KOMB as applied to unitig graphs constructed from simulated data as well as synthetic and real metagenomes. We demonstrate its effectiveness in identifying repetitive regions across sample types and sizes and illustrate how KOMB profiles can be used to visualize community structure. Finally, in the Discussion and Conclusions we cover the salient points and main conclusions from our study and lay out future directions of our research.

Results

We present a thorough validation of KOMB as applied to various simulated, synthetic, and real datasets. We do this through three major sets of experiments. First, we demonstrate the efficacy of the application of the K-core decomposition algorithm in genomics by testing it on simulated genomes constructed as random sequences to which we have added known repeat families. The simulated backbone sequences are constructed by appending base pairs uniformly at random until the desired length is achieved. We then simulate two families of repeats and insert them into these random backbone sequences using the multinomial distribution to determine the spacing between the repeats. We simulate two families of repeats, intra-genomic repeats that are all contained within a particular random genome, and a second family of inter-genomic repeats contained within multiple genomes. The results on the simple simulated dataset validate the theoretical results on the K-shell profiles as discussed in the Methods section and demonstrate KOMB's ability to unveil repetitive unitigs (Additional File 1, Fig S1).

Further, we also analyzed the effects of different read quality control methods that are traditionally used by biologists. Specifically, we show that read filtering via k-mer filtering techniques and read correction can significantly impact the profile

of shells. We show that error prone reads can lead to fragmentation of shells in contrast to the ideal case and can significantly impact the profile of the sample. We also reason that discarding reads with low abundance k-mers could be a better approach to prevent fragmented peaks in the KOMB profile as opposed to any read correction that may introduce noise (Additional File 1, Figure S2). Next, we repeat the simulated experiment, this time embedding repeats into real microbial genomes in lieu of a random backbone. In addition to the expected signal, this introduces some interference from the sequences that we expect to encounter in real datasets due to the presence of repeats in bacterial genomes. We show that the peaks containing unitigs bordering the inserted simulated repeats are still observed clearly with a small shift in shells at which these peaks occur (Additional File 1, Figure S3).

KOMB uses an internal unitig filter where unitigs shorter than read length are not considered for downstream analysis. Though beneficial in reducing noise while analyzing isolate genomes, this could cause loss of information in metagenomes or samples containing closely related strains or species. In such cases, the resulting de Bruijn graph is expected to be highly fragmented yielding shorter unitigs. We discuss the effect of unitig filtering in the context of species diversity through simulations on five closely related *E. coli* strains and reason that unitig filtering based on length must be turned off in order to capture the complete profile. We also show the difference in signatures obtained when we have multiple genomes in a sample that are closely related versus a sample containing more distantly related genomes. (Additional File 1, Figures S4, S5 and, S6)

Lastly, we run KOMB on real metagenomic samples to show both how the shell profile can be an indicator of the community structure present in the samples as well as its scalability to handle large metagenomic datasets. We first show the results on a synthetic metagenomic dataset [43] which allows us to identify community structure in the presence of ground truth data. We also run KOMB on real metagenomic samples from the Human Microbiome Project (HMP) and show that samples from the same body site tend to be more closely matched compared to the samples from other body sites based on their KOMB profiles. Finally, we run KOMB on approximately 1TB of longitudinal gut microbiome data to show that KOMB can help capture and visualize perturbations in microbiome communities.

KOMB validation on a simulated *E.coli* and *B.Cereus* sample

To validate KOMB on a simple simulated data model, we consider 2 genomes in a sample, *E. coli* and *B. cereus*. We embed one family of intra-genomic 400×400bp repeats in *E. coli* and one family of intra-genomic 200×400bp repeats in *B. cereus*, along with one shared family of inter-genomic 500×500bp repeats. Figure 1 shows the results on the combined sample as well as the individual genomes of *E. coli* and *B. cereus* separately. Based on the theoretical analysis (see KOMB profile in the Methods section), we expect a peak close to 1000 (for the inter-genomic repeats) and peaks close to shells 400 and 200 as well. Figure 1 shows that we do indeed see peaks close to 1000 that represent the shared simulated repeats. More interestingly we see peaks around shell 200 and 400 but also see some discernible peaks between 200-400 and 400-600. These are unitigs with two different types of repeats at their edges causing a shift beyond the expected number of shells for the intra-genomic repeats.

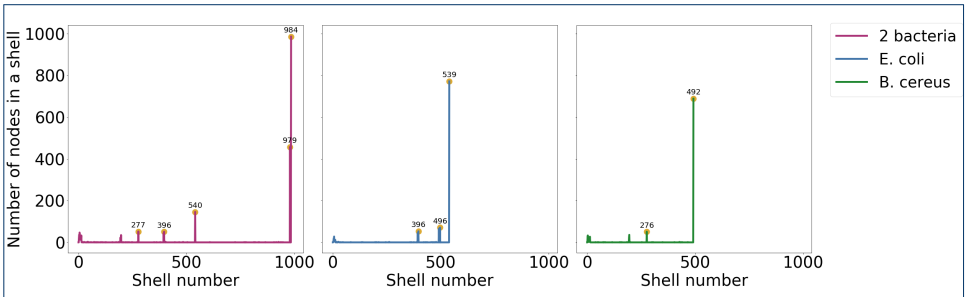

Figure 1 Combined KOMB profile of *E. coli* (intra: 400×400bp, inter 500××500bp) and *B. cereus* (intra: 200×400bp, inter 500×500bp) repeats (Left), *E. coli* single genome (Middle), and *B. cereus* single genome (Right). For the combined profile of both bacteria (L), there is a clear formation of peaks close to position 1000 (984 and 979), which indicate the inter-genomic repeats, and peaks at shell numbers 277, 396 and 540. This agrees with the theoretical model in the case of unitigs with mixed repeats at its end as discussed in Methods. We also plot the individual profiles of *E.coli* and *B.cereus*. For *E.coli* (M) we see three peaks at 396, 496 and, 539 given its higher copy number of intra-genomic repeats (400). For *B.cereus*, we observe peaks at 276 and 492 signalling its intra and inter-genomic repeats, respectively.

In order to validate the signature we receive from the above combined plot of *E. coli* and *B. cereus*, we create a ground truth dataset of repetitive unitigs by mapping back the unitigs to the reference genomes. Given that we know the position of the simulated repeats in the genome, we mark any unitigs mapping to a region overlapping the embedded repeats into three categories, either inter-genomic repeats

or one of the intra-genomic family of repeats. Figure 2 confirms that unitigs in the highest shell do indeed have inter-genomic repeats at their ends. Also, as expected, the families of intra-genomic repeats fall in peaks around 200 and 400 and the unitigs with mixed repeats at their ends form peaks in between those shells. Finally, we observe that the initial shells have no repeats hitting them. This demonstrates the ability of KOMB to delineate repeat families while being robust to background noise.

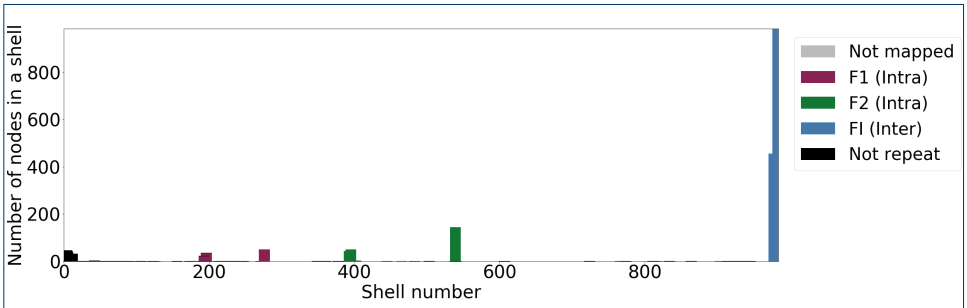

Figure 2 Validation of repeat types in *E. coli* + *B. cereus* sample via mapping unitigs back to the reference using nucmer [44]. Unitigs are labelled based on the repeats they overlap with. Based on ground truth nucmer mapping, the last shell (close to 1000) contains unitigs overlapping exclusively with inter-genomic repeats (FI) whereas the shells around 200 are overlapping *B. cereus* simulated intra-genomic repeats (F1) and shells around 400 are overlapping *E. coli* simulated intra-genomic repeats (F2). Finally, the first shells contain the background noise (colored black).

KOMB on Metagenomes

To address the question of visualization and characterization of communities within metagenomic samples we have run KOMB on a synthetic metagenomic community of 64 organisms [43] and real metagenomes obtained from the Human Microbiome Project (HMP) [45] [46]. Finally, we also show that KOMB can reveal shifts in large scale longitudinal metagenomic studies [47].

Synthetic Metagenome Dataset

We ran KOMB on the Shakya et al [43] synthetic metagenome community and carried out an in-depth analysis of the KOMB profile. The Shakya metagenomic dataset consists of 64 organisms - 16 archea and 48 bacteria. In order to validate and analyze KOMB, we also downloaded the reference genomes of all the organisms in the sample. These 64 genomes were then concatenated into a single fasta file and

used as input to nucmer for repeat finding in order to determine the ground truth (Additional File 1, Section 1.5) .

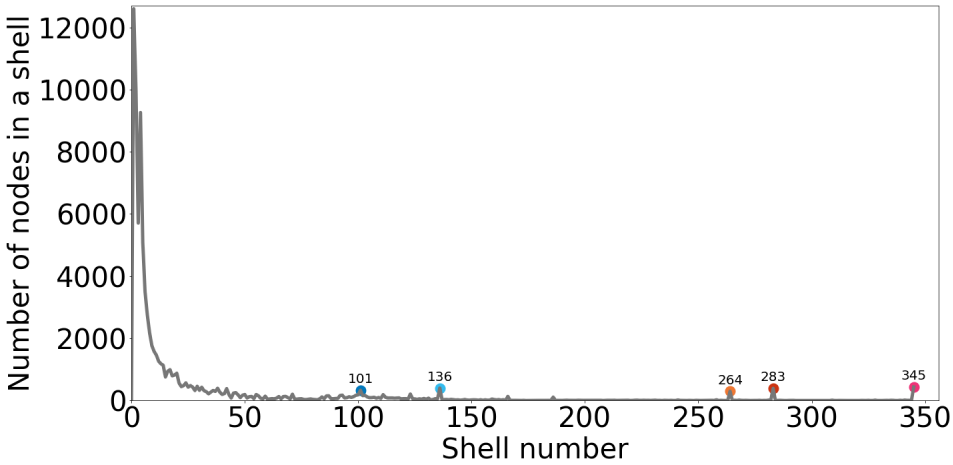

Figure 3 The KOMB profile of the Shakya et al [43] synthetic metagenome. We observe five peaks with more than 250 nodes in the shells above 50 at 101, 136, 264, 283 and 345, respectively. We mark each peak with a distinct color to use for representing respective nodes in the graph visualization in Figure 4.

Analysis of KOMB Profiles

As part of the preprocessing step in the KOMB pipeline, the paired-end reads were filtered using the k-mer filter tool from Stacks [48]. We then ran KOMB with no unitig filter to replicate a run with no prior knowledge of the community structure. Figure 3 shows the KOMB profile obtained. We observe that, similar to the case of simulated repeats with a real genome backbone, we obtain some peaks in the initial shells that represent the inherent background similarities in the genome which decay as we approach shell number 50. Post the 50th shell, we observe 5 distinct peaks in the profile (marked with colors) at shells 101, 136, 264, 283 and 345, respectively. Shell 345 is also the last shell of the profile, hence, we find consistent behaviour on the synthetic metagenome data with our simulated validations that produce a peak containing inter-genomic repeats. To further closely analyze the graph topology, we plot the largest connected component of an induced subgraph of the data. The induced subgraph is constructed such that it only contains the nodes present in shells above the 50th shell where we observe the initial peaks decay. Figure 4 shows the result of this visualization. We color each of the nodes occurring in our five

peaks of interest as dark blue (101), cyan (136), orange (264), red (283) and purple (345), and use a spring graph layout to plot the graph. We observe that shells 283, 345 and 136 form dense subgraphs whereas the 264 and 101 shells are more spread out over the connected component. This, in fact, is also a characteristic of K-core where shells can represent dense subgraphs as well as long-range connections that are important to the global structure of the graph.

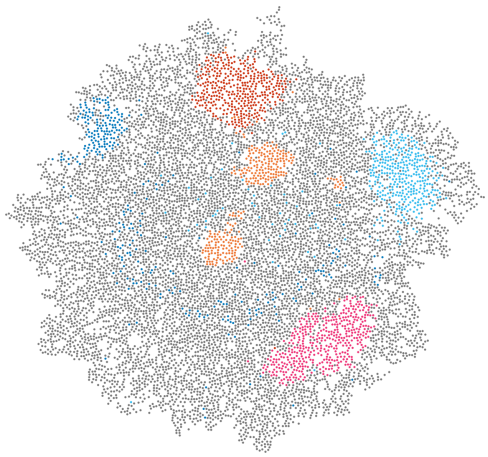

Figure 4 The unitig graph constructed from the Shakya et al synthetic metagenome data (edges removed for clarity). To effectively visualize the unitig graph, we consider only the nodes in shells above 50. The colored nodes represent peaks corresponding to Figure 3 indicating repeat unitigs. Here we tag and visualize nodes from the five peaks as dark blue (101), cyan (136), orange (264), red (283), and purple (345). The graph is constructed using sfdp tool from the Graphviz package, which uses a variant of force directed layout for placing the nodes. In force directed layouts, tightly connected clusters tend to stay together in the final representation.

We further analyze each of the repetitive unitigs in each of the peaks as well as the rest of the shells. We first plot the total number of distinct genomes hit by unitigs in each shell. This gives us information as to whether particular shells are inclined at identifying inter-genomic homologous regions and which shells capture unitigs that map predominantly to fewer organisms. In Figure 5, we see a distinctive last shell spike much like the KOMB profiles, here it indicates that the densely connected subgraph does in fact represent inter-genomic repeat unitigs. We see some similar patterns in the early shells after the 50th shell cutoff (50-161). For each of the five peaks observed in the KOMB profile we have the following number of genomes per shell, 101: 59, 136: 54, 264: 24, 286:17, 345:42. We see that the shells 264 and 286 have significantly less number of genomes per shell, indicating that the majority

of the repeats captured by that shell are more intra-genomic rather than inter-genomic in nature. Unitigs in the last shell mapped to 42 genomes (out of the total of 64) displaying a larger diversity than that of the intermediate shells and underlying KOMB's ability to capture high copy number repeat unitigs appearing across organisms in the metagenome.

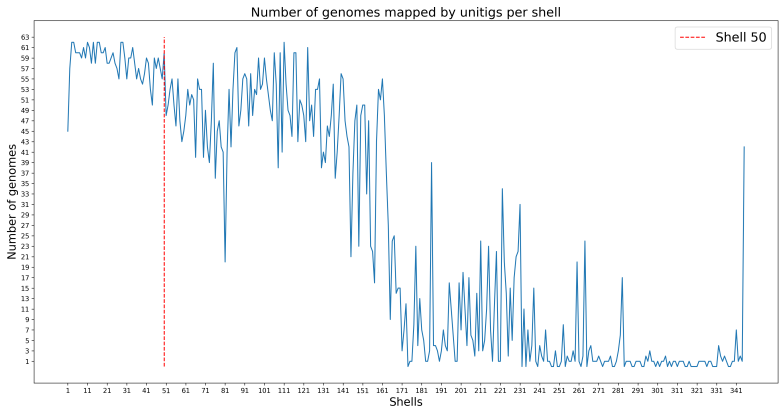

Figure 5 The number of genomes mapped by unitigs per shell. For each repeat unitig from the ground truth occurring in each shell we sum the total number of unique genomes mapped. We observe that the total number of unitigs mapped is high in the earlier shells as well as in the last shell. In the intermediate shells, we get a mixture of shells containing unitigs from multiple references and some containing unitigs from one or a small number of references. The former can be interpreted as shells capturing a more inter-genomic homology profile whereas those shells having unitigs mapped to fewer reference genomes indicate a species specific signature. For the peaks observed in Figure 3, we observe that shells 101 and 136 have unitigs mapped to 59 and 54 genomes respectively, while the number of genomes hit by unitigs in peak 264 and 286 are 24 and 17, respectively. The last shell sees a recovery of more inter-genomic signature with 42 genomes being mapped by the unitigs. The red dotted line marks shell 50. The more informative part of the plot lies to the right of this line since there is inherent noise in signals obtained from the shells before 50.

We also coin a new metric called repeat density to further analyze the copy number of repeated unitigs in each shell. We define KOMB repeat density for each shell as the copy number per genome per unitig. This is a two step calculation. First, for each repetitive unitig in the shell we sum up its copy number and divide the sum by the total number of distinct genomes it was mapped to, this gives us the copy number per genome. Second, we divide this by the total number of unitigs in the shell (repetitive and non-repetitive) which gives us a measure of how dense is the repeat information contained in a given shell. This also provides a holistic view of

how the copy number per shell normalized by the number of genomes and unitigs varies across the KOMB Profile. We observe in Figure 6 that the repeat density of the profile is higher in the higher shells, thus confirming our hypothesis that we are more likely to capture repeats accurately in the later shells where there is a stronger signal representing dense subgraphs.

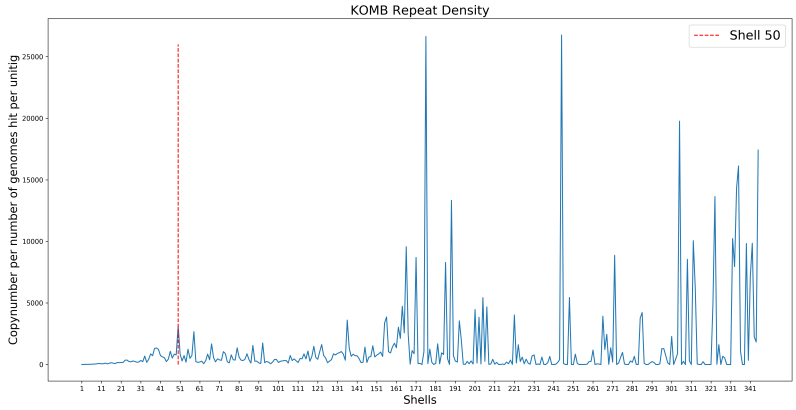

Figure 6 KOMB repeat density. For each shell we calculate the average copy number of each unitig normalized by the number of genomes. The average is calculated by dividing by the total number of unitigs (repeat + non-repeat) in the shell. This gives us a more nuanced view of how the copy number of repeats (which influences the shell number) varies across the different reference genomes it hits. We observe that the KOMB repeat density is higher in the last shells compared to initial shells, indicating a high copy number to genomes ratio as well as a higher probability to find such repeats out of all the unitigs present in the shell (low false positives)

Human Microbiome Project Samples

We have selected 4 distinct body sites among the available samples: external nares, supragingival plaque, fecal, and bucal mucosa. For each distinct site, we arbitrarily picked 4 samples, each with between 20,000,000 and 30,000,000 paired-end Illumina reads. We filtered the read sets by running k-mer filter with k-mer size 21, abundance threshold 2, and k-mer per read abundance of 80%. Thus, we only retained the reads that consist of 80% or more of 21-mers that occur at least twice in the sample. We then ran the KOMB pipeline, with k-mer size 50 used for de Bruijn graph construction. Since we are likely to encounter some closely related organisms in the samples, we have turned off unitig filtering. Thus, we have retained the unitigs that fall below read length in the graph. We then plotted the obtained profiles as stacked

violin plots presented in Figure 7. We observe that samples from different sites give rise to different profiles, as evidenced by Figure 7 zoomed in on the first 40 shells. We note that while there are outliers present for each site, the overall intra-site similarity of profiles is high. Furthermore, the inter-site comparison suggests that the profiles determined by KOMB are distinct for different sites.

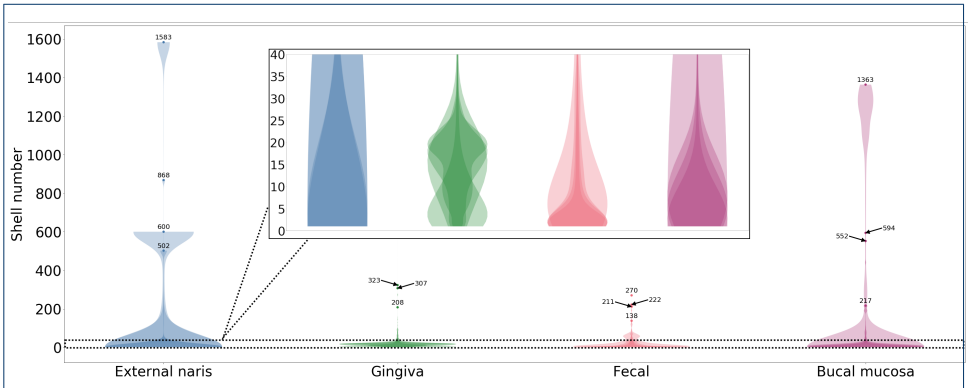

Figure 7 HMP metagenomes taken from 4 different body sites (4 samples each). Numbers indicate the last shell for each of the samples. We observe that the captured KOMB profiles are distinct for samples coming from different sites, and similar for samples originating from the same site. We also note that external naris and bucal mucosa sites have larger variances in the number of shells across the samples. Furthermore, these two sites also tend to have a much larger last shell than the gingiva and fecal samples.

Analysis on the Human Gut Microbiome

The study of the population diversity and stability of the human gut microbiome has gained increasing prominence given its impact on disease conditions and various pathologies [49–51]. Given its importance, it becomes imperative to enable large scale analysis of gut metagenomes and visualize significant shifts in community structure, particularly in cases of external perturbation like introduction of dietary changes or antibiotics. Here, we show that the KOMB profile can offer novel insights into longitudinal microbiome studies such as that of the human gut.

To demonstrate KOMB's ability to derive insights from large scale metagenomic analysis, we considered the temporal gut metagenome study by Voigt et al [47]. This study contains almost 1TB worth of human gut microbiome sequencing data collected from 7 subjects (5 male and 2 female) at different time points spread over two years. Figure 8 shows the KOMB profiles of each of 6 subjects from the initial four time points (Days 0, 2, 7 and, 60). Though we ran KOMB on the entire set of

reads in this study we exclude one male subject Halbarad from the figure because the sample at day 60 was missing. According to the study no external disruptions or sample variabilities were reported for any of the subjects during these time points. A qualitative analysis of KOMB profiles reveals two important observations. First, we observe that the general profiles of the gut microbiome closely resemble that of the fecal samples reported in Figure 7 and are very distinct from other body sites indicating KOMB's ability to consistently capture body site specific community signatures. Second, we observe a high degree of intra-sample similarity over the three time points and also observe some fundamental difference between the initial shells of the profile based on gender, which is also reported by previous studies [52] [53]. The only exception to this trend is the subject Bugkiller which showed significant variability in the early samples as compared to other male subjects Alien, Peacemaker and Scavenger which exhibited fairly consistent profiles. We reason that this deviation could be mostly due to errors or contamination in the sequences as none of the other 6 samples show such variability. To get a more quantitative understanding of the data and the effects of external disruptions on the gut microbiome we focus our attention on the subject Alien who was the only subject exposed to an antibiotic intervention and bowel cleanse procedure during the course of the study.

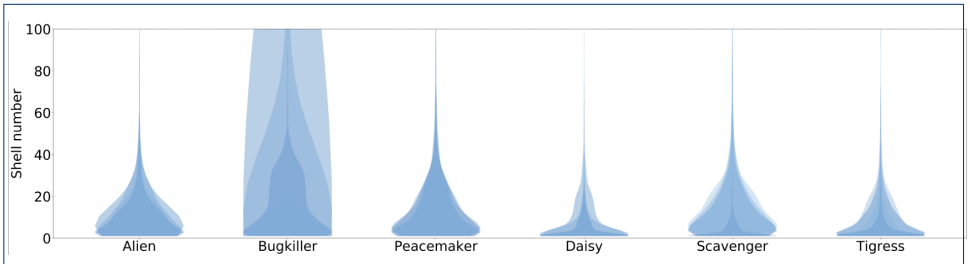

Figure 8 KOMB profile limited to the first 100 shells for the first three timepoints (Days 0, 2 and, 7) for each of the 6 subjects in the study indicated the intra and inter-profile variablity of the gut microbiome. Alien, Bugkiller, Peacemaker and Scavenger are male subjects while Daisy and Tigress are female subjects.

Figure 9 outlines the entire longitudinal trajectory of the Alien's gut microbiome over the course of 14 time points spread across two years. The KOMB profiles focus on the first 200 shells at each time point. We observe a significant compression of shells on Days 376, 377, 378, and 380 which coincides with samples taken post antibiotic intake and corresponding to a significant perturbation to the diversity and

community composition as reported in the study. This is also mirrored in the unitig count of the samples which is decreased by an order of magnitude. It is important to note here that the total number of reads in the individual time points are similar and, hence, the difference in the number of unitigs is more likely to be caused by shifts in the composition of the microbiome. We see that antibiotic intervention causes not only a reduction in the total number of shells but also alters the unitigs present in the initial shells, though this tends to recover slightly towards the end of the antibiotic cycle on Day 380. We also observe complete unitig distribution recovery in the initial shells twelve days after the last post-antibiotic sample on Day 392. Following this, the number of unitigs recovers close to earlier levels by Day 600. We observe similar but less drastic shell compression and quick recovery after bowel cleanse indicating that antibiotics cause a far greater disruption in microbiome community structure, a finding corroborated by the authors in Voigt et al [47] as well as an earlier study [54].

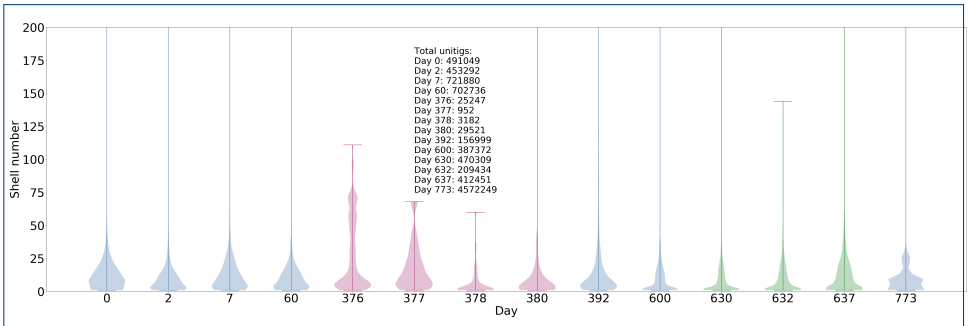

Figure 9 KOMB profile for subject "Alien" over the course of the study. Days 376, 377, 378, and 380 represent profiles during which the subject was exposed to antibiotics that caused a compression in the total shell count as well as a significant change in the node distribution of the initial shells, indicating a disruption of the microbial community. Days 630 and 632 indicate time points when the subject underwent a bowel cleanse procedure.

To further gauge if the perturbation caused was significant, we calculated the total variation of probability measure between the shell profiles (normalized to 1). Figure 10 shows the pairwise distances as calculated by the proposed measure. More precisely, for discrete probability distributions P and Q, the distance $\delta(P, Q)$ between them is computed as $\delta(P, Q) = \frac{1}{2}||P - Q||_1 = \frac{1}{2}\sum_{w\in\Omega}|P(w) - Q(w)|$, where Ω is the (discrete) sample space [55]. To get a better estimate of the difference between each probability distribution we grouped samples from three of the subjects

Alien, Bugkiller and Peacemaker according to time points, namely initial comprising Days 0, 2, 7, and 60, post-antibiotic comprising Days 376, 377, 378, and 380, and only from Alien and later comprising Days 392 (3 samples) and 773. We aimed to reason that the distance between Alien initial and Alien post-antibiotic was significantly greater than a change that could be explained merely by a difference in time duration. Indeed, we observe that Alien post-antibiotic has significantly greater pairwise distance to all other samples (Avg dist = 0.622). This also happens to be far more than the distance between samples of subjects at initial and later time points (Avg dist = 0.312). Observing samples collected from Alien, the average pairwise distance between Alien initial and other samples (excluding Alien post-antibiotic) is 0.227 and that between Alien later and other samples (excluding Alien post-antibiotic) is 0.38. The distance confirms our hypothesis that antibiotic intervention does in fact cause significant perturbation in KOMB profiles. Apart from total probability measure, we also implemented other distances between probabilities distributions such as the Earth mover's distance [56, 57] and KL Divergence [58]. Similar findings were obtained with these alternative distances; see Additional File 2, Figures S1, S2 and S3 for more details.

Performance

KOMB is written in C++ and Python. It uses the igraph C graph library [59] for the unitig construction and K-core decomposition implementations. KOMB also uses OpenMP support [60] to use multi-threading wherever available to increase the efficiency of the unitig graph construction step to ensure its scalability to a large number of metagenome samples. Table 1 shows the runtime and memory usage of KOMB on the datasets used in our study. The experiments were run on a server with 64 Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz processors having 372 GB of RAM. We observed that KOMBs memory usage and runtime largely depend on the number of reads. ABySS unitig generation is the most memory intensive step in the pipeline while read mapping using bowtie2 is the most computationally intensive step in the pipeline. We observe that in the case of Shakya and HMP there is a large memory difference despite having similar numbers of reads. We reason that this is likely due to the de Bruijn graph size and topology difference as the peak occurs during the ABySS stage. Nevertheless, we observe that KOMB can run on samples

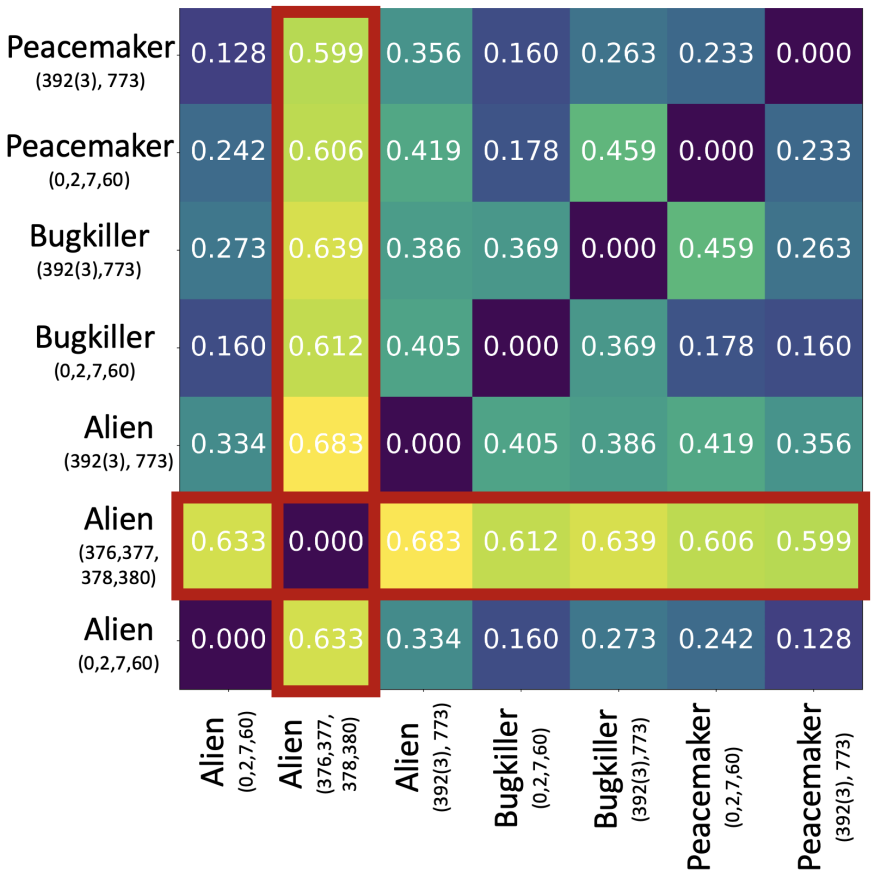

Figure 10 Heatmap showing total variation of probability measure of KOMB profiles. Each row and column represents a subject and days (four each) for which the samples are considered (in parenthesis). Day 392 had 3 samples in the dataset which are all considered here. The samples represented by Alien (Days 376, 377, 378, and, 380), also marked in red, are the ones collected during antibiotic perturbation. Higher total variation of probability denotes greater distance between two distributions. Days 0,2,7,60 correspond to the initial time points and Days 392(3) and Day 773 correspond to later time points.

with a large number of reads and can process 4 samples of HMP data in under 50 minutes and the Shakya synthetic metagenome (64 organisms) in 79 minutes. If run sequentially, the temporal gut microbiome data (70 samples, 1TB of data) can be run in approximately 2 days. As KOMB is also extremely memory efficient, one can process multiple metagenomic samples simultaneously on any modern workstation to reduce the runtime on entire datasets even further.

Table 1 Time and memory usage for KOMB. SSG: Simulated single genome; EBG: *E. coli* and *B. cereus*. EBSG: *E. coli*, *B. cereus* and *S. aureus* genomes; 5EG: Five genomes of closely-related *E. coli* strains; Shakya: Shakya et al (2013); HMP (I); individual HMP samples; HMP (A); combined HMP samples and TGM(Av); average across Temporal Gut Microbiome samples . Read filtering is treated as a pre-processing step, therefore the time and memory usage for it is not reported in this table.

Dataset	Performance metrics					
	Reads	Nodes	Edges	Wall clock	CPU time	RAM
SSG	625,000	1,336	159,060	79.46s	26m42s	1.54 GB
EBG	1,256,682	5,127	991,019	178.98s	71m50s	2.00 GB
EBSG	1,609,352	9,708	2,512,192	4m37s	132m31s	2.22 GB
5EG	3,453,508	40,769	162,606	4m12s	84m24s	2.60 GB
Shakya	53,997,046	160,083	1,767,445	79m36s	1814m43.80s	38.35 GB
HMP (I)	14,007,285	74,918	4,093,367	14m42s	211m7.2s	3.64GB
HMP (A)	56,029,140	409,370	7,496,925	47m41.95s	1995m24.6s	18.09 GB
TGM (Av)	26,520,076	776,058	7,286,158	44m41s	810m48s	20.22GB

Discussion

Identifying and visualizing homologous regions in metagenomes using current tools based on assembly graphs and contig graphs is often challenging as these graphs contain tangled intra-genomic and inter-genomic repeats. K-core decomposition can give accurate information capturing unitigs that have repeats, which can be visualized as peaks in a histogram. A peak indicates a dense subgraph of nodes in the unitig graph representing nodes connected to other homologous nodes, enabling an easy extraction for the purposes of assembly or scaffolding.

We outline the novelty of KOMB, both as a theoretical approach and as a usable tool. KOMB addresses some of the limitations of the previously used approaches based on contig graphs and betweenness centrality to identify both intra and inter-genomic repetitive structures in metagenomes. In contrast, KOMB constructs a unitig graph that captures edges within and between genomes, representing a more holistic network for homology detection. This prevents shortcomings occurring as a result of collapsing bubbles or branches by many modern assemblers, which leads to a loss of homology information among unitigs. K-core decomposition is also a natural choice to separate repeats based on their abundances as proved by our theoretical validations and is agnostic to the length of the individual repeat families. Though in our results we have shown that the background genome can have some baseline repetitiveness (low copy number), the end user can – based on the downstream applications – choose any particular shell as the cutoff to mark the unitigs as repeats, and can thus integrate KOMB into their pipeline. KOMB is also signifi-

cantly different from k-mer frequency based approaches. Though k-mer frequencies can provide general information on unique vs repetitive k-mers in a sample, KOMB more holistically captures information based off of read mapping that connects networks of similar genomic regions, which in turn represent intra and inter-genomic homology. Often, in metagenomic applications and assembly approaches, identifying contigs with highly repetitive k-mers and high coverage is a proxy for identifying repetitive contigs. KOMB, however, is an exact approach that provides information for scaffolding and exploration of the graph-based structure of the community.

Our results favorably support the utility of KOMB for the identification of homologous regions in real metagenomic samples. Though KOMB represents a promising new approach for elucidating genome dynamics within metagenomes, there still exist several challenges to develop a further understanding of how to interpret metagenomic community profiles and the separation of homologous regions in samples of varying diversity and abundance. To this end, we have classified future investigation into three separate categories. First we discuss extending our current theoretical framework to deconstruct and interpret the K-core decomposition results in a more intuitive fashion. We also discuss possible challenges that need to be addressed to interpret information on unitigs in higher shells that may not necessarily be peaks. Second, we focus on extending functionalities to a wider variety of input data, specifically long read data and other overlap graph types. Finally, we discuss possible approaches to further optimize the runtime and memory requirements.

Improving theoretical validation on metagenomes

In our validation on simulated genomes we have addressed the effects of identical simulated repeats on the K-shell profile of genomes and metagenomes. However, there exist some important limitations to our study. First, all repeats within the same repeat family were constructed to be identical. This is not necessarily the case in real genomes, since two regions can contain a few base pair differences yet be considered repeats from the biological standpoint. Though the results on synthetic and real metagenomic data containing such repeats have been promising, we are planning to extensively test KOMB with simulated homologous but not fully identical repeats in the future.

Second, we have been using multinomial distribution to space out the repeats throughout the backbone. However, in the real genomes, repeats can be less uniformly distributed with an extreme case being the tandem repeats. It is important to analyze these cases both in terms of the resulting topology of the graph, and in terms of our method's performance in these scenarios.

Third, we have considered repeats of lengths 200, 400, 500, 700 and 1000 base pairs. In a real genome, the length of a repeat can be significantly smaller or larger [61] [62], which further complicates the picture. As now some of the repeats will be causing shifts in the graph topology and manifest as increased background signal in the corresponding profile. However, other repeats will still be cleanly appearing as peaks. Deconvolution of such mixed signal in the general setting is an extremely complex problem and one that may need a combination of other graph theory and signal processing approaches. However, we aim to understand some of the simpler scenarios which have enough biological motivation. KOMB may also be prone to accumulating noisy unitigs in the higher shells as a result of being adjacent to repeat unitigs. Hence, a further filtering process within the shells would enable greater specificity of repeat unitigs [63].

One of the ways to tackle these questions will be to analyze the effects of real-world repeat patterns on the shell profiles in the simulated setting. Embedding real repeats into increasingly more complex simulated backbones, will gives us a different viewpoint on the shell profiles. It will also improve our overall understanding of the repeat induced profiles and provide a way to further deconvolve the signal obtained from metagenomic datasets.

Extending functionality

Currently, KOMB supports paired-end short reads as the input. However, we also have the capability of inputting graphs directly by using the GFA format. Graphs directly derived from the de Bruijn graph, such as the unitig overlap graph produced by SPAdes, do not have enough signal for effective KOMB processing. On the other hand, read overlap graphs obtained from long read datasets can potentially yield interesting results when processed with KOMB. Fully extending the pipeline to capture those cases and enable the effective analysis of long read datasets is one of the directions we plan to pursue in the future work.

Optimizing performance

KOMB performs highly efficient parallel graph construction and K-core decomposition. However, the memory requirements of the pipeline still calls for usage of workstations for processing metagenomic datasets. While this is customary for software working with paired-end read data, we are looking forward to supporting long read data and smaller personal devices. We plan to address this in future releases by fine tuning initial steps of the pipeline to allow low memory footprint execution. Together with compact long read sequencers, this would enable usage of KOMB as a quick profiling tool outside of the research laboratory environments.

Conclusions

In this paper, we present KOMB - an efficient and scalable tool to identify repetitive regions in metagenomes. We present a rigorous analysis of KOMB on simulated and synthetic data to capture consistent and accurate peak signatures representing repetitive unitigs. Another feature of KOMB, as shown by our validation experiments, is that the signals obtained are robust to confounding noise occurring as a result of read errors and insert size variability. This noise can be corrected to obtain near ground truth signals. We also show, through our experiments on real metagenomic samples, that KOMB profiles can be used as an indicator for sample specific signatures and diversity, with promising applications to a wide array of metagenomic analyses.

Methods

In this section, we describe the methodology behind KOMB and the various software tools and algorithms used in the pipeline. KOMB makes use of three popular bioinformatics software tools, namely k-mer filter [48] for read correction as an optional pre-processing step, ABySS [64] or SPAdes [65] for efficient de Bruijn graph creation and unitig construction, as well as Bowtie2 [66] for fast and accurate read mapping. In addition to this, our tool uses the igraph C package [59] and OpenMP [60] libraries for the K-core implementation and the fast parallel construction of the unitig graph, respectively. KOMB offers two primary operation modes. Users can either use the KOMB unitig builder pipeline which relies on ABySS [64] for de Bruijn graph construction and unitig generation or alternatively use the SPAdes unitig

generator which can output a unitig graph directly in the GFA format. We can use the GFA output directly as an input to KOMB. Using the SPAdes graph output is much faster since we avoid the graph construction step of the algorithm. However, the resulting graph only connects unitigs based on the k-mer overlap. This results in a highly compressed shell profile and weak signal for KOMB analysis. Thus, we will be using the ABySS unitig construction step in all analyses that follow. Another use for the GFA extension is that it provides users with a way to input an overlap graph or any assembly or contig graph directly into KOMB and visualize the results of the analysis. This can be particularly useful for overlap graphs constructed from long read data. For the purpose of comparing different read pre-processing methods we also use the short read correction tool Lighter [67]. The paired-end read simulator wgsim [68] is used for all simulated experiments.

Pipeline

In order to understand the workflow, we first describe a unitig graph. A unitig is a maximal consensus sequence usually obtained from traversing a de Bruijn graph. Unitigs by definition terminate at branches caused by repeats and variants, and unlike contigs, are non-overlapping. Before constructing the set of unitigs, we run the previously described k-mer filter as a preprocessing step. The first filtering step is iterating through all reads and counting occurrences of each k-mer, in our case the k-mer size is 15. A k-mer is marked as abundant if it occurs in the dataset more than twice. The next step is iterating through the reads again, and considering the k-mers present in each read separately. If less than 80% of k-mers in the read are abundant, then we discard the read. For the purposes of this work, the unitig graph refers to a graph having unitigs as its vertices and the edges being representative of adjacent or homologous unitigs. After the unitigs are obtained, in our case performed by running ABySS on the corrected reads, we follow three additional steps for careful construction of unitig graphs from short paired-end read data (Fig.11). First, all of the reads are mapped to unitigs by Bowtie2 using its sensitive global alignment module. Each read of a read pair (forward and reverse) is mapped individually and we allow for a maximum of 1000 alignments per read (this parameter can be adjusted by the user). We also trim the tail of both pairs to ensure that we get accurate alignments. The number of base pairs that we trim off the ends of the

reads is equal to the difference between the read length and the k-mer size used to construct the de Bruijn graph. As a secondary filtering step, we also filter out mapped reads without a pair as well as read pairs mapped to one unitig. This allows us to only consider reads with paired-end information and speeds up the process of unitig graph construction. Second, for each read we create a set of all unitigs that mapped to that read. For a given forward and reverse read pair, we also check if each individual read in the pair mapped to different unitigs, which would represent potentially adjacent unitigs in the genome. In this way, for a given read pair we have unitigs associated with each read, e.g., in Fig. 11 unitigs 1, 3, and 8 are associated with one read of the purple pair whereas unitig 4 is associated with the other read. We then connect all the unitigs associated with a specific read pair (nodes 1, 3, 4, and 8 for the purple read pair) where we distinguish between the notion of a vertical edge, i.e. an edge linking unitigs associated with the same read such as 1 and 3, and a horizontal edge, i.e. an edge linking unitigs mapped to different reads in the same pair such as 1 and 4.

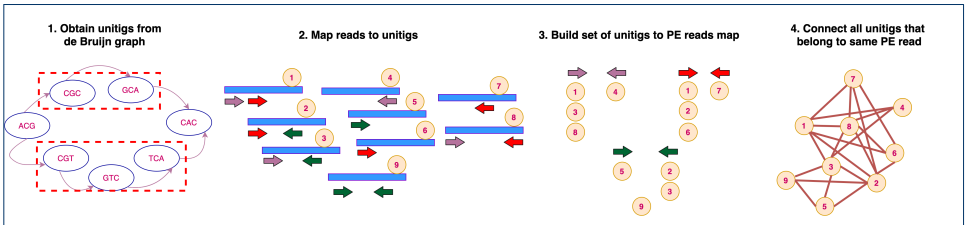

Figure 11 Construction of the unitig graph. (Step 1) unitigs are obtained from running ABySS on the filtered paired-end reads. (Step 2) We map the forward and reverse reads individually to the unitigs using Bowtie2 with the parameters *-k* (maximum alignments) set to 1000. Further, we trim the 3 and 5 ends of forward and reverse reads respectively by a length equal to the read length minus the k-mer size, using the parameter *-3* and *-5*. (Step 3) We group the unitigs that were mapped by the same read and also the unitigs that had two ends (F and R) of the same paired-end read mapped to them. This results in a final set of unitigs mapped by a given read pair (Step 4) We construct the graph with unitigs as the nodes, and connecting two nodes with an edge whenever the corresponding unitigs are in the same group after the read mapping.

K-core decomposition

K-core decomposition is a popular graph-theoretical concept used in network science to identify influential nodes in large networks [69–71]. It has been previously shown to accurately calculate node influence in Susceptible-Infected-Recovered (SIR) network models in epidemiological studies [63]. K-core decomposition partitions the

node set of a graph into layers (or shells) from more peripheral to more central nodes. More precisely, the K-core of a graph is defined as the maximal induced subgraph where every node has (induced) degree at least K. Based on this sequence of K-cores, we say that a node belongs to the K-shell if it is contained in the K-core but not in the (K+1)-core. For any given graph, one can iteratively and efficiently decompose it into shells with complexity $O(V+E)$, which is significantly faster than the computation of most exact centrality measures. This makes it effective for decomposing large and dense networks. Several implementations of the K-core decomposition have been proposed. In this work, we rely on the igraph C package [59], which implements a variation of the algorithm proposed in [72]. In contrast to centrality-based methods, the K-core algorithm identifies densely connected cliques and groups them into shells. Fig. 12 shows the decomposition of a toy graph into its K-shells. Fig. 13 shows the complete pipeline of KOMB as a flowchart.

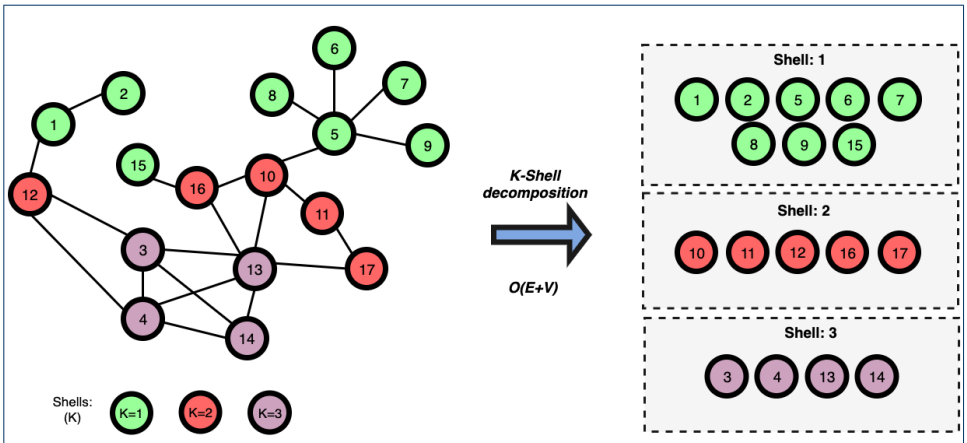

Figure 12 K-core decomposition of a graph into K-shells. The algorithm starts by considering all the vertices of degree 1. It iteratively removes those vertices and continues the execution on the resulting induced subgraph removing vertices having degree 1 after every iteration. Once the induced subgraph has no vertices of degree 1, this process stops and all discarded vertices are marked as belonging to the 1-shell (green). Then the process continues, now considering vertices of degree 2 to obtain the 2-shell (red) and, subsequently, the 3-shell (purple). The last shell is a dense subgraph of the original graph.

KOMB profile

We refer to the output of KOMB either as a KOMB profile or as the shell profile of a given sample. This is visualized as a bar plot depicting the number of nodes

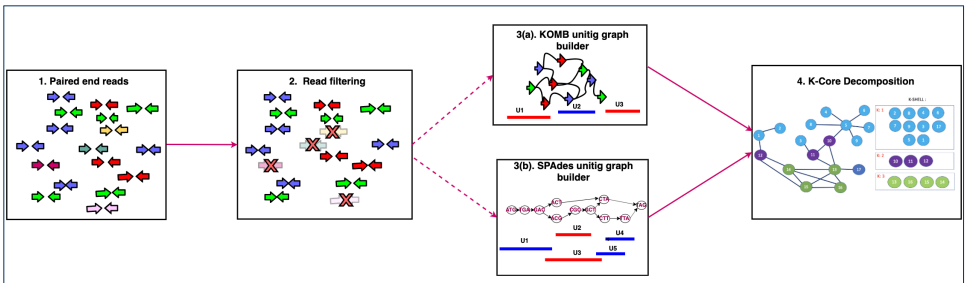

Figure 13 Workflow showing the components of the KOMB Pipeline. KOMB takes in as input a set of paired-end reads. The user can optionally filter these reads as the first step. We use the k-mer filter for read filtering for both simulated and real genomes. KOMB then constructs the de Bruijn graph and generates unitigs. This can be done with either ABySS or SPAdes, and the choice is up to the user. Note that SPAdes also generates a GFA output, which we can feed into KOMB directly for the K-core decomposition. Once we have the set of unitigs available, we construct the graph via the procedure described in Fig. 11. Finally, we perform the K-core decomposition on the unitig graph and generate text files with shell labels for unitigs.

per shell. As the read error, insert sizes, diversity, community structure, and sample sizes vary, we expect a corresponding shift in the bar plot as each of these conditions would alter the node distribution in shells. In Results, we have presented simulated experiments varying the above mentioned conditions that corroborate this hypothesis. Here, we present a theoretical analysis to calculate shifts in peaks occurring as a result of having two distinct repeat families through an example. Each shell k obtained after K-core decomposition is an induced subgraph of degree k which may or may not be disconnected. In a unitig graph, based on our construction, these would contain regions of shared homology or repetitive regions and k would depend on the abundance or copies of these shared region across the genomes in the sample. These shells containing repetitive or homologous regions tend to occur as distinct peaks at higher shells versus the rest of the background. By definition, the background contains regions more sparsely connected. Given a simulated experiment, it is possible to theoretically ascertain the shells at which we expect discernible peaks. For example, if we have a repeat R_1 with copy number K, then based on our read mapping and unitig construction steps we would expect a peak in the Kth shell. This would contain all unitigs having an overlap of k with the repetitive region, where k is the k-mer size used to generate the de Bruijn graph. The case is a little more complex when we have two families of repeats R_1 and R_2 with copy numbers K_1 and K_2 respectively. Depending on the placements of the repeats we can classify

the unitigs obtained into different classes based on the family of repeat it overlaps at its breakpoints; see Fig. 14. We have 3 possible categories of unitigs as shown in the figure depending on the repeats at the ends of the unitig. Category 2 in the figure refers to unitigs with both repeats at its ends. An important observation here is that according to our graph construction method, a node in this category will be connected to other nodes in the same category as well as all nodes in the other categories as it carries both repeats. We can estimate the expected number of unitigs in each of the categories as follows: Let N_1 be the number of unitigs overlapping the repeat R_1, N_2 be the number of unitigs overlapping the repeat R_2, and N_M be the number of unitigs overlapping both repeats. Assuming uniform probability distribution over all possible permutations of repeats in the genome, we obtain the following expected values:

$$\begin{aligned}
\mathbb{E}(N_1) &= \frac{|R_1|(|R_1|-1)}{|R_1|+|R_2|-1} \approx \frac{|R_1|^2}{|R_1|+|R_2|}, \\
\mathbb{E}(N_2) &= \frac{|R_2|(|R_2|-1)}{|R_1|+|R_2|-1} \approx \frac{|R_2|^2}{|R_1|+|R_2|}, \\
\mathbb{E}(N_M) &= \frac{|R_1||R_2|+|R_2||R_1|}{|R_1|+|R_2|-1} \approx \frac{2|R_1||R_2|}{|R_1|+|R_2|}.
\end{aligned}$$

Subsequently, in the case when the insert is larger than the length of the repeat and given enough paired-end reads, we should observe two peaks in the shell profile, namely, we will have a peak at $\mathbb{E}(N_1)+\mathbb{E}(N_M)$ and another one at $\mathbb{E}(N_2)+\mathbb{E}(N_M)$. These two shells are obtained since the unitig graph would consist of two overlapping cliques, one of size $\mathbb{E}(N_1)+\mathbb{E}(N_M)$ and another one of size $\mathbb{E}(N_2)+\mathbb{E}(N_M)$, with an overlap of size $\mathbb{E}(N_M)$ (represented in Figure 14 with red and black lines). However, notice that when the insert size is shorter than the length of the repeat, the two types of unitigs overlapping both repeats would not be connected between them in the graph (represented in Figure 14 with black lines only). This results on a shift in the position of the second shell.

Comparison to other repeat identification methods

A novel feature of our study is using unitig graphs to analyze repetitive regions in metagenomes using K-core decomposition in contrast to contig graph commonly used in previous approaches like MetaCarvel [31] and Bambus [25]. While our focus is on metagenomic repeat detection, it is worth discussing other graph based

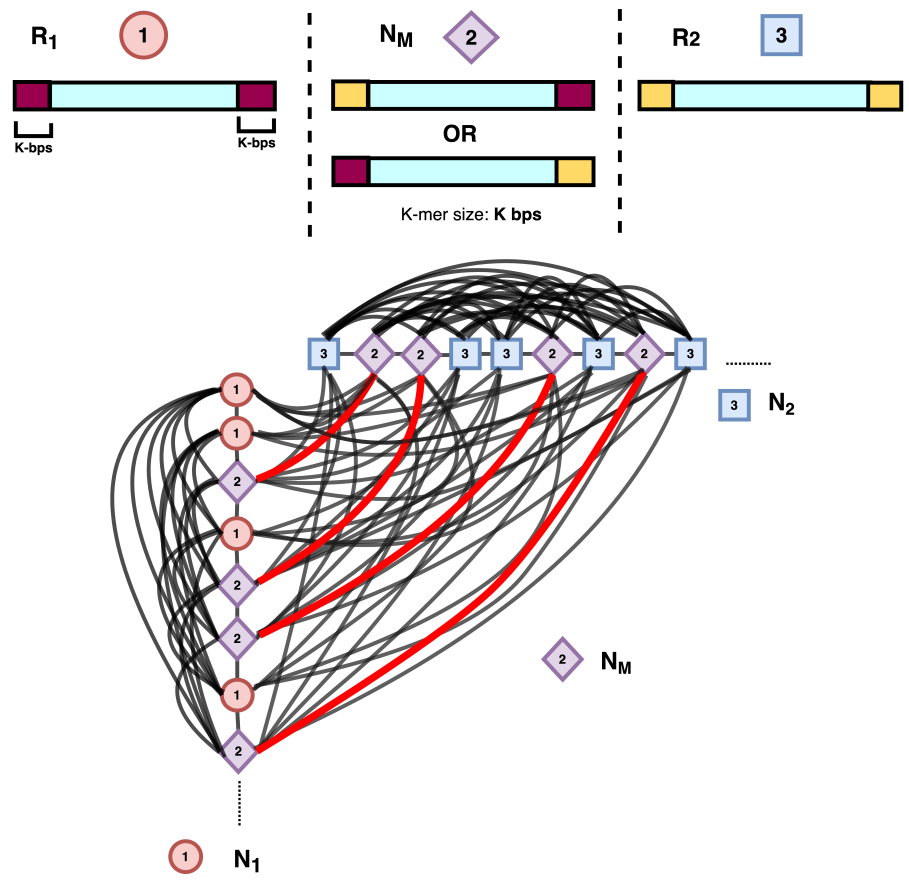

Figure 14 Types of unitigs in a genome with two repeat families and expected shell profiles in corresponding unitig graphs. The type of profile we observe depends on the relative lengths of the repeats and insert size. If the insert size is greater than the length of the repeat, the mixed repeats (N_m) will be connected to each other whereas if the insert size is smaller than the length of the repeat then it is not possible to map across the two mixed repeat unitigs and, hence, they will not be connected by an edge in the unitig graph. The black edges are present for both settings whereas the red edges are only present when the repeat length is less than the insert length.

tools that been previously applied for repeat detection in isolate genomes. A graph based hierarchical agglomerative clustering [73] approach was suggested by Novák et al [74] and used the Fruchterman and Reingold algorithm [75] to help visualize reads with similarities, but its quadratic time complexity $O(V^2 + E)$ makes it difficult to scale to large metagenomic datasets. Recently, two tools, namely, REPdenovo [76] and REPLong [77] have used underlying contig graph based structures for repeat identification. Both these methods have been applied to eukaryotic genomes to ascertain repetitive regions. REPdenovo uses abundant k-mers and assembles them to repeat contigs. It then further stitches repeat contigs into longer consensus

repeats and uses coverage based information to filter non-specific repeat contigs. An important point to note is that the formation of larger consensus repeats from raw repeat contigs is very similar to scaffolding where a directed raw contig overlap graph is constructed and then a topological sort is carried out on each strongly connected component to obtain a linear order of raw unitigs. The traversal of the graph to identify long consensus sequences is then carried out by using path finding heuristics. REPLong, on the other hand, is a more recent tool and is specific to long read data. It uses the concept of community detection in long read overlap graphs to construct repeat libraries. In addition to graph-based approaches, an alternative method to efficiently identify repeats on large genome scale datasets is by using k-mer frequency estimation, which accounts for both identical and nearly identical k-mers to identify repeats. Examples of these include ReAS [78], RepeatScout [79], WindowMasker [80], Repseek [81], Tallymer [82], RED [83], RepARK [84] at the genome level and more recently at short read level RF [85] identification D_2^R statistic [86] based on a variation of the D_2 statistic that have been previously used for sequence comparison [87–89]. K-mer frequency based approaches depend on identifying candidate k-mers that may contain repeats based on their statistical significance compared to background. Most k-mer based repeat identification tools have shown to capture a small subset of specific repeats and size, mainly either transposable elements (TE) or tandem repeats (TR). RED can detect both TE and TR with greater sensitivity in both bacterial and eukaryotic genome including the Human genomes [83]. RepARK creates de-novo repeat libraries by identifying abundant k-mers which are then assembled by a de novo genome assembly program (such as Velvet) into repeat consensus sequences. While these k-mer based tools have been shown considerable accuracy in identifying repeats, these have only been applied to assembled and un-assembled isolate genomes. Thus, their use case in metagenomic samples where repeats may be both intra and inter-genomic with varying abundances is extremely limited and remains untested. The recently introduced D_2^R statistic can be applied to metagenomes directly and is a read level mapping tool that indicates a measure of repetitiveness in a given read. This method was tested on real metagenomes and could aid the identification of CRISPR sites with high accuracy. Though indicating the presence and absence of repeats is informative, the D_2^R statistic on read level repeat information is more suited to identify

short regions consisting of clearly defined and distinct motifs. There still exists a need for a more rigorous theoretical basis that generalizes over different kinds of repeats and community diversity in metagenomes where there are far more varied and often confounding repeat structures of larger lengths that are highly sample dependent. Another potential drawback is that reads are often noisy and error prone and have some inter-sample variability which may affect its performance.

In contig graph approaches, methods based on betweenness centrality have been the preferred choice to mark repetitive contigs. This approach, though specific, has not achieved high levels of sensitivity and often tends to miss out on a lot of repetitive contigs. This served as the core basis for further investigations in this study. To the best of our knowledge, KOMB is the first tool using K-core decomposition on unitig graphs. In order to understand the advantages of our approach, it is imperative to understand topological differences captured by different methods. Most modern assemblers tend to collapse information obtained by a single read mapping to multiple unitigs. This tends to affect the vertical edges in the graph that we discussed when describing Fig. 11. This graph simplification often leads to loss of information of homologous regions present in other parts of the genome and can affect sensitivity. Moreover, as contigs contain repeat regions, paired-end data tends to reveal very little information about the presence of repeats within the contig. These structures in the contig graph tend to resemble a single node (collapsed branches) having a high degree and centrality. But the centrality threshold to mark repeats is hard to ascertain and arbitrary thresholds may lead to sub-optimal repeat detection. This is a key difference of unitig graphs in KOMB as compared to contig graphs in MetaCarvel. Contig graphs are connected only on paired end read information. Though appropriate for scaffolding, this feature precludes the successful identification of homology. In contrast, KOMB takes into consideration all unitigs mapped by the same read, preserving homology information, while also preserving positional information through paired mapping where (given sufficient insert size) links can connect two adjacent unitigs bordering the same repeat. In this way, all unitigs having repeats on their edges tend to form dense subgraphs which can be efficiently detected using K-core decomposition, yielding clear peaks at shells containing repetitive unitigs. Hence, a unitig graph can be thought of as a richer graphical representation to identify repetitive structures in metagenomes and

K-core decomposition offers the most efficient and exact method to recover these signals irrespective of the sample diversity.

Another related application based on a combination of k-mer and graph based approach to uncover genomic variants is DBGWAS [90]. DBGWAS relies on a compact de Bruijn graph representation that helps identify the connected components of the graph induced by the neighbourhoods of all significant unitigs. DBGWAS tests for the association of each variant, indicated by the presence or absence of unitig in a particular genome, against a particular set of phenotypes using a linear mixed model. It relies on the assumption that subgraphs defined by significant unitigs are a reflection of the genomic environment, and ranks such subgraphs based on their association to the phenotype. Though this work shares similarity with our unitig graph based approach, it requires draft assemblies and prior phenotypic information to capture subgraph significant unitigs. KOMB, on the contrary, requires just metagenomic reads as input and uses K-core decomposition to capture unitigs that highlight genomic diversity in a sample.

Since KOMB is a novel method that is fundamentally different from previous contig graph based or k-mer based approaches, it is difficult to perform a one to one comparison of KOMB with any of the previous methods. Specifically, the construction of unitig graph specific network signatures captured by KOMB are unique and not measured by any other previous method. In this work, through a series of meticulous validations on simulated, synthetic, and real metagenomes we demonstrate that KOMB offers a novel solution to capture underlying repetitive regions in metagenomic data.

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Additional Files

Additional file 1

Validation of KOMB on simulated repeats in single genome (random backbone), *E. coli* backbone. Effects of error and error correction approaches on KOMB profiles. Effect of unitig filter on taxonomically similar and diverse samples.

Additional file 2

Contains supporting results for distances between samples in Voigt et al.(2015) study using Earth mover's distance and KL Divergence

Availability of data and materials

All scripts, datasets, and results produced and used in this manuscript are available for download at: https://rice.box.com/v/komb-manuscript

Competing interests

The authors declare that they have no competing interests.

Funding

N.S is supported by Department of Computer Science, Rice University. A.B. and T.J.T were supported by startup funds from Rice University and the FunGCAT program from the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the Army Research Office (ARO) under Federal Award No. W911NF-17-2-0089. R.A.L.E. was supported by the FunGCAT program from the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the Army Research Office (ARO) under Federal Award No. W911NF-17-2-0089.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, ARO, or the US Government.

Authors contributions

A.B , T.J.T, S.S developed the study. A.B wrote and implemented the software, performed the validation and analyses. N.S performed the validation and analyses. R.A.L.E, S.S and T.J.T contributed to the design of the validation and the interpretation of the results. All authors wrote the paper. All authors read and approved the final manuscript.

Author details

[1]Department of Computer Science, Rice University, 6100 Main St, 77005 Houston, Texas, USA. [2]Department of Electrical and Computer Engineering, Rice University, 6100 Main St, 77005 Houston, Texas, USA.

References

1. Paten, B., Novak, A.M., Eizenga, J.M., Garrison, E.: Genome graphs and the evolution of genome inference. Genome research **27**(5), 665–676 (2017)
2. Ulyantsev, V.I., Kazakov, S.V., Dubinkina, V.B., Tyakht, A.V., Alexeev, D.G.: Metafast: fast reference-free graph-based comparison of shotgun metagenomic data. Bioinformatics **32**(18), 2760–2767 (2016)
3. Pell, J., Hintze, A., Canino-Koning, R., Howe, A., Tiedje, J.M., Brown, C.T.: Scaling metagenome sequence assembly with probabilistic de bruijn graphs. Proceedings of the National Academy of Sciences **109**(33), 13272–13277 (2012)
4. Myers, E.W.: The fragment assembly string graph. Bioinformatics **21**(suppl_2), 79–85 (2005)
5. Zerbino, D.R., Birney, E.: Velvet: algorithms for de novo short read assembly using de bruijn graphs. Genome research **18**(5), 821–829 (2008)
6. Rakocevic, G., Semenyuk, V., Lee, W.-P., Spencer, J., Browning, J., Johnson, I.J., Arsenijevic, V., Nadj, J., Ghose, K., Suciu, M.C., et al.: Fast and accurate genomic analyses using genome graphs. Technical report, Nature Publishing Group (2019)
7. Dilthey, A., Cox, C., Iqbal, Z., Nelson, M.R., McVean, G.: Improved genome inference in the mhc using a population reference graph. Nature genetics **47**(6), 682 (2015)
8. Eggertsson, H.P., Jonsson, H., Kristmundsdottir, S., Hjartarson, E., Kehr, B., Masson, G., Zink, F., Hjorleifsson, K.E., Jonasdottir, A., Jonasdottir, A., *et al.*: Graphtyper enables population-scale genotyping using pangenome graphs. Nature genetics **49**(11), 1654 (2017)
9. Garrison, E., Sirén, J., Novak, A.M., Hickey, G., Eizenga, J.M., Dawson, E.T., Jones, W., Garg, S., Markello, C., Lin, M.F., et al.: Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nature biotechnology (2018)
10. Ameur, A.: Goodbye reference, hello genome graphs. Nature biotechnology **37**(8), 866–868 (2019)
11. Kim, D., Paggi, J.M., Park, C., Bennett, C., Salzberg, S.L.: Graph-based genome alignment and genotyping with hisat2 and hisat-genotype. Nature biotechnology **37**(8), 907–915 (2019)

12. Chin, C.-S., Peluso, P., Sedlazeck, F.J., Nattestad, M., Concepcion, G.T., Clum, A., Dunn, C., O'Malley, R., Figueroa-Balderas, R., Morales-Cruz, A., *et al.*: Phased diploid genome assembly with single-molecule real-time sequencing. Nature methods **13**(12), 1050 (2016)
13. Li, D., Liu, C.-M., Luo, R., Sadakane, K., Lam, T.-W.: Megahit: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph. Bioinformatics **31**(10), 1674–1676 (2015)
14. Namiki, T., Hachiya, T., Tanaka, H., Sakakibara, Y.: Metavelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. Nucleic acids research **40**(20), 155–155 (2012)
15. Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.A.: metaspades: a new versatile metagenomic assembler. Genome research **27**(5), 824–834 (2017)
16. Vollmers, J., Wiegand, S., Kaster, A.-K.: Comparing and evaluating metagenome assembly tools from a microbiologists perspective-not only size matters! PloS one **12**(1), 0169662 (2017)
17. Chikhi, R., Limasset, A., Medvedev, P.: Compacting de bruijn graphs from sequencing data quickly and in low memory. Bioinformatics **32**(12), 201–208 (2016)
18. Minkin, I., Pham, S., Medvedev, P.: TwoPaCo: an efficient algorithm to build the compacted de Bruijn graph from many complete genomes. Bioinformatics **33**(24), 4024–4032 (2016). doi:10.1093/bioinformatics/btw609. http://oup.prod.sis.lan/bioinformatics/article-pdf/33/24/4024/25168506/btw609.pdf
19. Pop, M., Kosack, D.S., Salzberg, S.L.: Hierarchical scaffolding with bambus. Genome research **14**(1), 149–159 (2004)
20. Luo, J., Wang, J., Zhang, Z., Li, M., Wu, F.-X.: Boss: a novel scaffolding algorithm based on an optimized scaffold graph. Bioinformatics **33**(2), 169–176 (2017)
21. Almodaresi, F., Pandey, P., Patro, R.: Rainbowfish: a succinct colored de bruijn graph representation. In: 17th International Workshop on Algorithms in Bioinformatics (WABI 2017) (2017). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik
22. Muggli, M.D., Bowe, A., Noyes, N.R., Morley, P.S., Belk, K.E., Raymond, R., Gagie, T., Puglisi, S.J., Boucher, C.: Succinct colored de bruijn graphs. Bioinformatics **33**(20), 3181–3187 (2017)
23. Ghurye, J.S., Cepeda-Espinoza, V., Pop, M.: Focus: Microbiome: Metagenomic assembly: Overview, challenges and applications. The Yale journal of biology and medicine **89**(3), 353 (2016)
24. Ghurye, J., Pop, M.: Better identification of repeats in metagenomic scaffolding. In: WABI (2016)
25. Koren, S., Treangen, T.J., Pop, M.: Bambus 2: scaffolding metagenomes. Bioinformatics **27**(21), 2964–2971 (2011)
26. Treangen, T.J., Salzberg, S.L.: Repetitive dna and next-generation sequencing: computational challenges and solutions. Nature Reviews Genetics **13**(1), 36 (2012)
27. Ayling, M., Clark, M.D., Leggett, R.M.: New approaches for metagenome assembly with short reads. Briefings in Bioinformatics (2019). doi:10.1093/bib/bbz020. bbz020. http://oup.prod.sis.lan/bib/advance-article-pdf/doi/10.1093/bib/bbz020/27990743/bbz020.pdf
28. Rodriguez-R, L.M., Gunturu, S., Tiedje, J.M., Cole, J.R., Konstantinidis, K.T.: Nonpareil 3: Fast estimation of metagenomic coverage and sequence diversity. MSystems **3**(3), 00039–18 (2018)
29. Rasheed, Z., Rangwala, H., Barbará, D.: 16s rrna metagenome clustering and diversity estimation using locality sensitive hashing. BMC systems biology **7**(4), 11 (2013)
30. Ma, Z., Li, L.: Measuring metagenome diversity and similarity with hill numbers. Molecular ecology resources **18**(6), 1339–1355 (2018)
31. Ghurye, J., Treangen, T., Fedarko, M., Hervey, W.J., Pop, M.: Metacarvel: linking assembly graph motifs to biological variants. Genome biology **20**(1), 174 (2019)
32. Segarra, S., Ribeiro, A.: Stability and continuity of centrality measures in weighted graphs. TSP **64**(3), 543–555 (2016)
33. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. J. Math. Soc. **2**(1), 113–120 (1972)
34. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry, 35–41 (1977)
35. Brandes, U.: A faster algorithm for betweenness centrality. Journal of mathematical sociology **25**(2), 163–177 (2001)
36. Geisberger, R., Sanders, P., Schultes, D.: Better approximation of betweenness centrality. In: Proceedings of the

Meeting on Algorithm Engineering & Expermiments, pp. 90–100 (2008). Society for Industrial and Applied Mathematics

37. Riondato, M., Kornaropoulos, E.M.: Fast approximation of betweenness centrality through sampling. In: WSDM (2014)
38. Zhu, Z., Surujon, D., Ortiz-Marquez, J.C., Wood, S.J., Huo, W., Isberg, R.R., Bento, J., van Opijnen, T.: Entropy of a bacterial stress response is a generalizable predictor for fitness and antibiotic sensitivity. bioRxiv (2019). doi:10.1101/813709. https://www.biorxiv.org/content/early/2019/10/22/813709.full.pdf
39. Ochman, H., Lawrence, J.G., Groisman, E.A.: Lateral gene transfer and the nature of bacterial innovation. nature **405**(6784), 299 (2000)
40. Hamady, M., Knight, R.: Microbial community profiling for human microbiome projects: Tools, techniques, and challenges. Genome research **19**(7), 1141–1152 (2009)
41. Gonzalez, A., King, A., Robeson II, M.S., Song, S., Shade, A., Metcalf, J.L., Knight, R.: Characterizing microbial communities through space and time. Current opinion in biotechnology **23**(3), 431–436 (2012)
42. Gómez, P., Paterson, S., De Meester, L., Liu, X., Lenzi, L., Sharma, M., McElroy, K., Buckling, A.: Local adaptation of a bacterium is as important as its presence in structuring a natural microbial community. Nature communications **7**, 12453 (2016)
43. Shakya, M., Quince, C., Campbell, J.H., Yang, Z.K., Schadt, C.W., Podar, M.: Comparative metagenomic and rrna microbial diversity characterization using archaeal and bacterial synthetic communities. Environmental microbiology **15 6**, 1882–99 (2013)
44. Marçais, G., Delcher, A.L., Phillippy, A.M., Coston, R., Salzberg, S.L., Zimin, A.: Mummer4: a fast and versatile genome alignment system. PLoS computational biology **14**(1), 1005944 (2018)
45. Gevers, D., Knight, R., Petrosino, J.F., Huang, K., McGuire, A.L., Birren, B.W., Nelson, K.E., White, O., Methé, B.A., Huttenhower, C.: The human microbiome project: a community resource for the healthy human microbiome. PLoS biology **10**(8), 1001377 (2012)
46. Huttenhower, C., Gevers, D., Knight, R., Abubucker, S., Badger, J.H., Chinwalla, A.T., Creasy, H.H., Earl, A.M., FitzGerald, M.G., Fulton, R.S., *et al.*: Structure, function and diversity of the healthy human microbiome. nature **486**(7402), 207 (2012)
47. Voigt, A.Y., Costea, P.I., Kultima, J.R., Li, S.S., Zeller, G., Sunagawa, S., Bork, P.: Temporal and technical variability of human gut metagenomes. Genome biology **16**(1), 73 (2015)
48. Catchen, J., Hohenlohe, P.A., Bassham, S., Amores, A., Cresko, W.A.: Stacks: an analysis tool set for population genomics. Molecular ecology **22**(11), 3124–3140 (2013)
49. Ji, B.W., Sheth, R.U., Dixit, P.D., Tchourine, K., Vitkup, D.: Macroecological dynamics of gut microbiota. bioRxiv (2019). doi:10.1101/370676. https://www.biorxiv.org/content/early/2019/07/08/370676.full.pdf
50. Shreiner, A.B., Kao, J.Y., Young, V.B.: The gut microbiome in health and in disease. Current opinion in gastroenterology **31**(1), 69 (2015)
51. Treangen, T.J., Wagner, J., Burns, M.P., Villapol, S.: Traumatic brain injury in mice induces acute bacterial dysbiosis within the fecal microbiome. Frontiers in immunology **9**, 2757 (2018)
52. Santos-Marcos, J.A., Haro, C., Vega-Rojas, A., Alcala-Diaz, J.F., Molina-Abril, H., Leon-Acuña, A., Lopez-Moreno, J., Landa, B.B., Tena-Sempere, M., Perez-Martinez, P., *et al.*: Sex differences in the gut microbiota as potential determinants of gender predisposition to disease. Molecular nutrition & food research **63**(7), 1800870 (2019)
53. Fransen, F., van Beek, A.A., Borghuis, T., Meijer, B., Hugenholtz, F., van der Gaast-de Jongh, C., Savelkoul, H.F., de Jonge, M.I., Faas, M.M., Boekschoten, M.V., *et al.*: The impact of gut microbiota on gender-specific differences in immunity. Frontiers in immunology **8**, 754 (2017)
54. O'Brien, C.L., Allison, G.E., Grimpen, F., Pavli, P.: Impact of colonoscopy bowel preparation on intestinal microbiota. PloS one **8**(5) (2013)
55. Levin, D.A., Peres, Y.: Markov Chains and Mixing Times vol. 107. American Mathematical Soc., ??? (2017)
56. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), pp. 59–66 (1998). IEEE
57. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International journal of computer vision **40**(2), 99–121 (2000)

58. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
59. Csardi, G., Nepusz, T., *et al.*: The igraph software package for complex network research. InterJournal, Complex Systems **1695**(5), 1–9 (2006)
60. Dagum, L., Menon, R.: Openmp: An industry-standard api for shared-memory programming. Computing in Science & Engineering (1), 46–55 (1998)
61. Achaz, G., Rocha, E.P., Netter, P., Coissac, E.: Origin and fate of repeats in bacteria. Nucleic acids research **30**(13), 2987–2994 (2002)
62. Rocha, E., Danchin, A., Viari, A.: Analysis of long repeats in bacterial genomes reveals alternative evolutionary mechanisms in bacillus subtilis and other competent prokaryotes. Molecular biology and evolution **16**(9), 1219–1230 (1999)
63. Liu, Y., Tang, M., Zhou, T., Do, Y.: Improving the accuracy of the k-shell method by removing redundant links: From a perspective of spreading dynamics. Scientific reports **5**, 13172 (2015)
64. Jackman, S.D., Vandervalk, B.P., Mohamadi, H., Chu, J., Yeo, S., Hammond, S.A., Jahesh, G., Khan, H., Coombe, L., Warren, R.L., *et al.*: Abyss 2.0: resource-efficient assembly of large genomes using a bloom filter. Genome research **27**(5), 768–777 (2017)
65. Bankevich, A., Nurk, S., Antipov, D., Gurevich, A.A., Dvorkin, M., Kulikov, A.S., Lesin, V.M., Nikolenko, S.I., Pham, S., Prjibelski, A.D., *et al.*: Spades: a new genome assembly algorithm and its applications to single-cell sequencing. Journal of computational biology **19**(5), 455–477 (2012)
66. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with bowtie 2. Nature Methods **9**, 357–359 (2012)
67. Song, L., Florea, L., Langmead, B.: Lighter: fast and memory-efficient sequencing error correction without counting. Genome biology **15**(11), 509 (2014)
68. Li, H.: wgsim-read simulator for next generation sequencing. Github Repository (2011)
69. Alvarez-Hamelin, J.I., Dall'Asta, L., Barrat, A., Vespignani, A.: Large scale networks fingerprinting and visualization using the k-core decomposition. In: Advances in Neural Information Processing Systems, pp. 41–50 (2006)
70. Khaouid, W., Barsky, M., Srinivasan, V., Thomo, A.: K-core decomposition of large networks on a single pc. Proceedings of the VLDB Endowment **9**(1), 13–23 (2015)
71. Zhang, H., Zhao, H., Cai, W., Liu, J., Zhou, W.: Using the k-core decomposition to analyze the static structure of large-scale software systems. The Journal of Supercomputing **53**(2), 352–369 (2010)
72. Batagelj, V., Zaversnik, M.: An o (m) algorithm for cores decomposition of networks. arXiv preprint cs/0310049 (2003)
73. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Physical review E **70**(6), 066111 (2004)
74. Novák, P., Neumann, P., Macas, J.: Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. BMC bioinformatics **11**(1), 378 (2010)
75. Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. Software: Practice and experience **21**(11), 1129–1164 (1991)
76. Chu, C., Nielsen, R., Wu, Y.: Repdenovo: inferring de novo repeat motifs from short sequence reads. PloS one **11**(3), 0150719 (2016)
77. Guo, R., Li, Y.-R., He, S., Ou-Yang, L., Sun, Y., Zhu, Z.: Replong: de novo repeat identification using long read sequencing data. Bioinformatics **34**(7), 1099–1107 (2017)
78. Li, R., Ye, J., Li, S., Wang, J., Han, Y., Ye, C., Wang, J., Yang, H., Yu, J., Wong, G.K.-S., *et al.*: Reas: Recovery of ancestral sequences for transposable elements from the unassembled reads of a whole genome shotgun. PLoS computational biology **1**(4), 43 (2005)
79. Price, A.L., Jones, N.C., Pevzner, P.A.: De novo identification of repeat families in large genomes. Bioinformatics **21**(suppl_1), 351–358 (2005)
80. Morgulis, A., Gertz, E.M., Schäffer, A.A., Agarwala, R.: Windowmasker: window-based masker for sequenced genomes. Bioinformatics **22**(2), 134–141 (2005)
81. Achaz, G., Boyer, F., Rocha, E.P., Viari, A., Coissac, E.: Repseek, a tool to retrieve approximate repeats from large dna sequences. Bioinformatics **23**(1), 119–121 (2006)

82. Kurtz, S., Narechania, A., Stein, J.C., Ware, D.: A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes. BMC genomics **9**(1), 517 (2008)
83. Girgis, H.Z.: Red: an intelligent, rapid, accurate tool for detecting repeats de-novo on the genomic scale. BMC bioinformatics **16**(1), 227 (2015)
84. Koch, P., Platzer, M., Downie, B.R.: Repark—de novo creation of repeat libraries from whole-genome ngs reads. Nucleic acids research **42**(9), 80–80 (2014)
85. Misawa, K.: Rf: A method for filtering short reads with tandem repeats for genome mapping. Genomics **102**(1), 35–37 (2013)
86. Chen, S., Chen, Y., Sun, F., Waterman, M.S., Zhang, X.: A new statistic for efficient detection of repetitive sequences. Bioinformatics **35**(22), 4596–4606 (2019). doi:10.1093/bioinformatics/btz262
87. Reinert, G., Chew, D., Sun, F., Waterman, M.S.: Alignment-free sequence comparison (i): Statistics and power. Journal of Computational Biology **16**(12), 1615–1634 (2009)
88. Torney, D.C., Burks, C., Davison, D., Sirotkin, K.M.: Computation of d2: a measure of sequence dissimilarity (1990)
89. Lippert, R.A., Huang, H., Waterman, M.S.: Distributional regimes for the number of k-word matches between two random sequences. Proceedings of the National Academy of Sciences **99**(22), 13980–13989 (2002)
90. Jaillard, M., Lima, L., Tournoud, M., Mahé, P., Van Belkum, A., Lacroix, V., Jacob, L.: A fast and agnostic method for bacterial genome-wide association studies: Bridging the gap between k-mers and genetic events. PLoS genetics **14**(11), 1007758 (2018)