中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-18-001

The Spiral Assembler: An Iterative Process of NGS *De Novo* Genome Assembly with Machine-Learning for Subset Selection on Quality-Score and K-Mer Landscape

Li-An Yang , Wei-Chun Chung , Yu-Jung Chang , Shu-Hwa Chen , Chung-Yen Lin and Jan-Ming Ho

Jan. 12, 2018 || Technical Report No. TR-IIS-18-001

http://www.iis.sinica.edu.tw/page/library/TechReport/tr2018/tr18.html

The Spiral Assembler: An Iterative Process of NGS *De Novo* Genome Assembly with Machine-Learning for Subset Selection on Quality-Score and K-Mer Landscape

Li-An Yang [1], Wei-Chun Chung [1,2], Yu-Jung Chang [1,§], Shu-Hwa Chen [1], Chung-Yen Lin [1] and Jan-Ming Ho [1,2]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan

[2] Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan

[§]Corresponding author

Email addresses:

LAY: luke831215@iis.sinica.edu.tw

WCC: wcchung@iis.sinica.edu.tw

YJC: yjchang@iis.sinica.edu.tw

SHC: sophia@iis.sinica.edu.tw

CYL: cylin@iis.sinica.edu.tw

JMH: hoho@iis.sinica.edu.tw

Abstract

In this manuscript, we study the problem of selecting a subset of NGS reads for *de novo* genome assembly. In an iterative process, we develop models to score importance of each read based on XGBoost. Each read is characterized by its *k*-mer landscape, i.e., *k*-mer count at each *k*-mer window, and base quality score at each mer. We then define a fixed-length feature vector of each read as input of XGBoost. The subset selection model is developed with its performance of *de novo* assembly being tested on datasets using SPAdes assembler and QUAST evaluation. We use two Illumina datasets, *S. cerevisiae* S288c and *S. aureus* MW2, denoted as D1 and D2 respectively, to test efficacy of the subset selection model. The experiments show that after each round of subset selection, corrected N50 of *de novo* genome assembly increases for both D1 and D2. In appendix A, we present the assembly results of GAGE-B Miseq datasets using the aforementioned pipeline. In appendix B, we analyze the process of read type tagging with Burrows-Wheeler Aligner and discuss the clip issue.

Background

A typical process of *de novo* genome assembly includes pre-assembly processing (e.g., trimming of adaptors and low-quality bases, and error correction), contig assembly, and post-assembly processing (e.g., scaffolding, gap closing, and benchmarking). Previous researches targeted at improving efficiency and effectiveness of the aforementioned steps [1]. Existing research shows that an analysis after assembly has potentially improved the *de novo* assembly [2]. It also shows that selecting subsets directly from the tagging result of reads does not lead to a stable performance increasing.

The idea of subset selection is to keep as many reads which are mapped uniquely to the reference genome while avoid picking up reads which, with high probability, could not map to the reference genome. It is a heuristic to reduce the complexity of assembly. The assembly result, however, does not provide decisive evidence from the size of selected subsets, as learnt empirically from the experiments of *de novo* assembly.

With more advanced and successful Illumina sequencing techniques, machine learning emerges with various applications to exploit the growing numbers of available reads in the field of bioinformatics. Extreme Gradient Boosting (XGBoost) [3] is one of the state-of-the-art algorithms for feature selection and prediction problems. XGBoost has been vastly used in many machine learning challenges such as the Netflix prize [4] and appears as the most popular approach among the winning solutions in KDDCup 2015.

Existing biological statistics, i.e., base quality score and *k*-mer copy number (i.e., number of occurrence of the same *k*-mer subsequence in the set of reads), have been widely applied in sequence mapping [5], variant calling [6], and sequence classification [7][8]. We therefore extract these information as input features for training.

In this article, we address the problem in improving effectiveness of assembly by adding a post-assembly benchmarking procedure to assess quality of assembly and embedding intelligence into the post-assembly benchmarking procedure. We present the Spiral Assembler (TSA) as an iterative approach, consisting of intelligent relabeling, subset selection, and a *de novo* genome assembler, as shown In Figure 1. We adopt XGBoost to score the sequencing reads and propose a *multi-phase filtering* approach consisting of several binary read class probabilistic classifiers for selecting the subsets.

The selected reads are then assembled and evaluated to obtain the best evaluation result for the next iteration. We use two datasets, *S. cerevisiae* S288c denoted as D1 and *S. aureus* MW2 denoted as D2, to evaluate the assembly result of iterations. Experimental result shows that the assembly quality increases significantly for both D1 and D2 in each iteration.

Materials and Methods

Dataset

We evaluate our proposed strategy with two Illumina datasets, *Saccharomyces cerevisiae* strain S288c as D1 dataset and *Staphylococcus aureus* strain MW2 as D2 dataset. All the datasets are downloaded from the sequence read archive (SRA) in NCBI. Table 2 shows the profile of our experimental datasets, including the SRA accession number, read length, number of reads in the dataset, and length of reference genome

Implementation resources

We incorporate Burrows-Wheeler alignment (*BWA*) [9] (version 0.7.15) to report the multiplicities of reads, i.e., the number of times a read mapped to the reference. We use *SAMtools* [10] (version 0.1.19) to manipulate the results alignments in the Sequence Alignment/Map format. To assemble reads, we use the SPAdes assembler [11] (version 3.11.1) with paired-end library. We then use QUAST [12] (version 4.6.0) with MUMmer [13] (version 3.23) to evaluate the assembly result.

Read type tagging

The designated process of tagging and categorizing reads is shown in Figure 2. To ensure the cleanness of data, we first incorporate a *Trimming* step to remove adapters, vectors, or primers used in sequencing. We then screen out the reads containing *N*s (i.e., ambiguous result of base-calling) and label them with type **N**. Then rest of the reads that have no *N*s will be labeled by read mapping. The *Mapping* step of our proposed process aims at categorizing the reads into subsets to be the ground truth for model training. After the finish of this step, reads are tagged with 5 types of labels alongside with type **N**. First, for the reads that cannot be mapped to the contigs (i.e., failed to map), a type of **F** is tagged.

Till this end, the remaining untagged reads are the reads that mapped to the contigs and occurred at least once. The reads that occur on multiple locations of the contigs, called *repeats*, are labeled as type **M**. For the *unique* reads (i.e., a read occurs once on the reference), we then check if a read has *alternative hit*. Thus, type **X** is tagged to the unique reads having alternative hits, whereas type **U** is tagged to the reads, otherwise. For the sake of posterior lookup, we summarize these read types with descriptions in Table 1.

Feature extraction

We propose a strategy to generate a fixed-length feature vector to overcome the inconvenience of variable length of reads. The first part of a feature vector is the percentiles of sorted quality scores and sorted *k*-mer copy numbers in the read, both from 0% to 100% per 5%. In addition, two features are designed for extracting more information that are potentially relevant to the read classification and denoted as MeanQ, MeanKCN, which represent the mean values of the base quality scores and the *k*-mer copy numbers of the read, respectively.

Scoring scheme

We develop a scoring scheme on reads using XGBoost as our machine learning approach. For each read, we obtain the tendency or probability distribution over classes of read types, mainly unique (U) and unmapped reads (F), instead of outputting a direct binary decision on the read types. Take Table 4 for example, to start with, each of the three read goes through a probabilistic classifier and is assigned the probability of belonging to a unique read (importance score). Classification results can then be determined by self-defined tunable threshold values. Lastly, we design and base on different values of threshold to collect reads and construct subsets for assembly.

Multi-phase filtering approach

Similar to hierarchical classification, we first use the unique reads (U) classifier to choose the desired reads from the dataset. Next, we use the failed-to-map reads (F) classifier as the second-phase filter to eliminate unsuitable reads. For each classifier, we designed 5 subset selection mechanisms, each of which is based on a different threshold as eligibility criteria. Initially, the threshold of mechanism that brings in the amount of reads closest to the read type tagging is marked as reference. We then set up 4 additional points that are higher or lower than the reference threshold by a fixed margin to observe the change of the succeeding assembly result.

Subset selection for genome assembly

The subset selection procedure starts with inputting unique reads (U) using the aforementioned methods to acquire 5 assembly results respectively. Secondly, we demonstrate the combinatorial approach by introducing an F classifier to remove the unsuitable failed-to-map reads at the cost of dropping only a few useful reads. Thus, our subset selection procedure eventually accumulates 25 more assembly results.

Iterative procedures of The Spiral Assembler

After each round of assembly procedure, we target at the subset that brings in the best N50 scaffold length computed from the estimated genome size of D1 and D2. The scaffolds assembled from this subset of reads are adopted as input to rerun the procedure again, including BWA for mapping, XGBoost for subset selection, and SPAdes for assembly. We therefore generate new labels, selected subsets, and assembled scaffolds after each round of assembly with an attempt to achieve more advanced assembly results. Note that the procedure of subset selection is an unsupervised learning process in which machine learning algorithm is used to relabel each read to compensate for the potential defects of tagging results determined by the alignment to the assembled contig.

Evaluation of the iterative assembler

For comparing the assembly result, we use the original data of SRR352384 (D1) and SRR022866 (D2) as our baseline. Beginning on the first assembly that uses the full set of reads as input, we pick the best N50 scaffold length from each round of the assembly results to generate the second, third and fourth assembly results. Then we use QUAST with the reference genomes of D1 and D2 to evaluate those assemblies. As mentioned in QUAST manual, corrected contig NG50 is the NG50 [14] contig size after breaking the contigs at every misjoin and at every indel longer than 5 bases. Meanwhile, the total number of misjoin and the larger indels are defined as #mis-assembly. Both contig NG50 and #mis-assembly are very crucial to evaluate assembly quality. L99 [15] is the smallest number of contigs which cover 99% of the reference genome.

Results

Read type tagging

Moreover, the profiles of read type tags of the first assemblies (obtained by the first line of flow in Figure 1) before performing subset selection of D1 and D2 are shown in Table 3. Note that there is no reference genome in a *de novo* assembly, and thus we cannot use the reference genome for read type tagging. Take D1 for example, out of the 51 million reads, 87.05% is classified as uniquely mapped and has no alternative hits (type U) while 0.06% has alternative hits (type X). 11.51% reads are tagged as unmapped (type F), 0.02% reads are multi-mapped (type M), and 1.35% reads contain Ns (type N).

Feature distribution

Before using XGBoost for probabilistic classification, we first look at feature distribution of our data, specifically uniquely mapped (U), multi-mapped (M), and unmapped (F) reads. The distribution of U, M, and F reads with K-mer landscaping is observed in Figure 3. To outline the feature distribution, we extract the percentiles of sorted *k*-mer copy numbers (X-axis) from each type of reads in D1 and D2 and stack up the values of each percentile respectively. Y-axis then shows the mean value computed from each pile, and graph are formed by connecting these points together. All three classes are deemed separable according to the result.

Multi-phase filtering approach

The scoring graphs of the designated subset selection mechanisms, as shown in Figures 4 and 5, help clarify and integrate the idea of scoring scheme into subset selection, with

X-axis and Y-axis indicating the sorted reads and their importance score respectively. Each bar of different color represents a different subset of reads.

Assembly results

The experiment results of The Spiral Assembler for D1 and D2 datasets are listed in Table 5 and 6 respectively. Note that the subscript of each subset denotes the approximate ratio of size extracted from reads. For instance, U_{90}-F_3 implies a mechanism that extracts 90% and 3 % of total sequencing reads, using U and F classifier respectively, to form a subset that filters out failed-to-map reads from the selected unique reads.

As shown in Table 5 of D1 dataset, the read subset sizes of the 2nd, 3rd and 4th assembly are below 70% of the full set. We observed that both the N50 and corrected NG50 contig sizes are improved by more than 12% from the 1st to 2nd assembly; in detail, the N50 increases from 11,849 to 35,411 and the c. NG50 increases from 8,674 to 9,740. In the next rounds from the 2nd assembly to the 3rd and 4th assemblies in Table 5, the N50 and c. NG50 sizes continue to improve and outperform (~22%) those in the 1st assembly. When comparing the values of L99 in Table 5, we found that the number of contigs needed to assemble 99% of the genome decreases continually from 3,223 of the 1st assembly to 518 of the 4th assembly. Thus, our method has achieved significantly longer and more accurate assemblies for D1 dataset.

About the performance of our method for D1 and D2 datasets, the post-assembly analysis takes only minutes while our machine-learning approach for subset selection finish collecting reads in less than two hours. The overhead of both stages is small compared with the genome assemblies by SPAdes. Since the *i*-th assembly shown in Figure 1 will need to run genome assembler 1+(*i*-1)*r* times, where *r* = 25 in the multi-phase filtering for D1 and D2 datasets, we suggest to run our method only a few run.

Discussion and Conclusions

In this paper, we present the concept and procedure of The Spiral Assembler (TSA) with machine-learning and post-assembly analysis. Our iterative process improves efficacy of assembly as the experimental results of D1 and D2 datasets suggest. TSA has achieved significant improvements in terms of longer and more accurate assembly for D1 and D2 dataset. Additionally, we also apply our approach to the miseq datasets presented in GAGE-B [16] and show the assembly results in Appendix along with a detailed analysis of read type tagging. Here we discuss some great potential for further advancement listed as follows.

Inter-species prediction

On top of the present procedure, we also examine the usage of our developed classification model to predict probabilities on closely related genome datasets. We utilize the preferable subset size inducted from previous results to expect a predicted outcome that benefits the assembly results. Although current results fail to outperform the baseline using original dataset, this method still retains great potential for improving overall assembly quality without any extensive knowledge of the dataset in advance and therefore is worth developing.

Multi-mapped reads integration into our selection mechanism

Current approach utilizes unique reads (U) and unmapped reads (F) classifiers to make subsets of reads for assembly. In [2], Chung *et al* show that multi-mapped reads are also discovered to have an influence on the quality of genome assembly. However, these reads actually belong to the minority group among all read types, consisting of less than 1% of D1 and D2 dataset (Table 3). Therefore, the use of a multi-mapped reads classifier is not applicable given the low effectiveness of model training on an imbalanced distribution of labels. We will attempt the integration of multi-mapped reads into our subset selection mechanism in the future for genome datasets tagged with far more multi-mapped reads.

Grid search to replace heuristics of threshold tuning in the future

So far, the assignment of threshold values follows our heuristics to set up 5 points for subset selection. Although there are indeed ways to implement grid search on the task to replace the currently human-involved strategy, more comprehensive research work is required given the heavy temporal cost of genome assembly. Means to deal with parameter tuning, such as PSO [15] and genetic algorithm, will be considered for the experiments to come.

Future prospects

While TSA benefits from the scalability of distributed XGBoost library, another advantage XGBoost holds under the family of tree boosting system is that each element of the feature vector is also associated with a feature importance score under a specified type, such as frequency of use or average gain of the feature, as shown in Figure 6. One potential development is to technically select several top-ranked features and repeat the demonstrated methodology to obtain comparable or better assembly results. These selected features can also be incorporated into other machine learning algorithm or data-driven approach for alternative applications.

We plan to extend our work to examine alternative methods of selecting the sequencing read, e.g., by considering number of copies of multi-mapped reads. The post-assembly analysis might also have the potential of identifying unique reads with substitution error. Moreover, experiments on larger and more complex NGS *de novo* genome datasets, i.e. mate-pair reads, are also under investigation. We expect that more accurate contigs assembled by TSA will be useful for better scaffolding results.

Figures

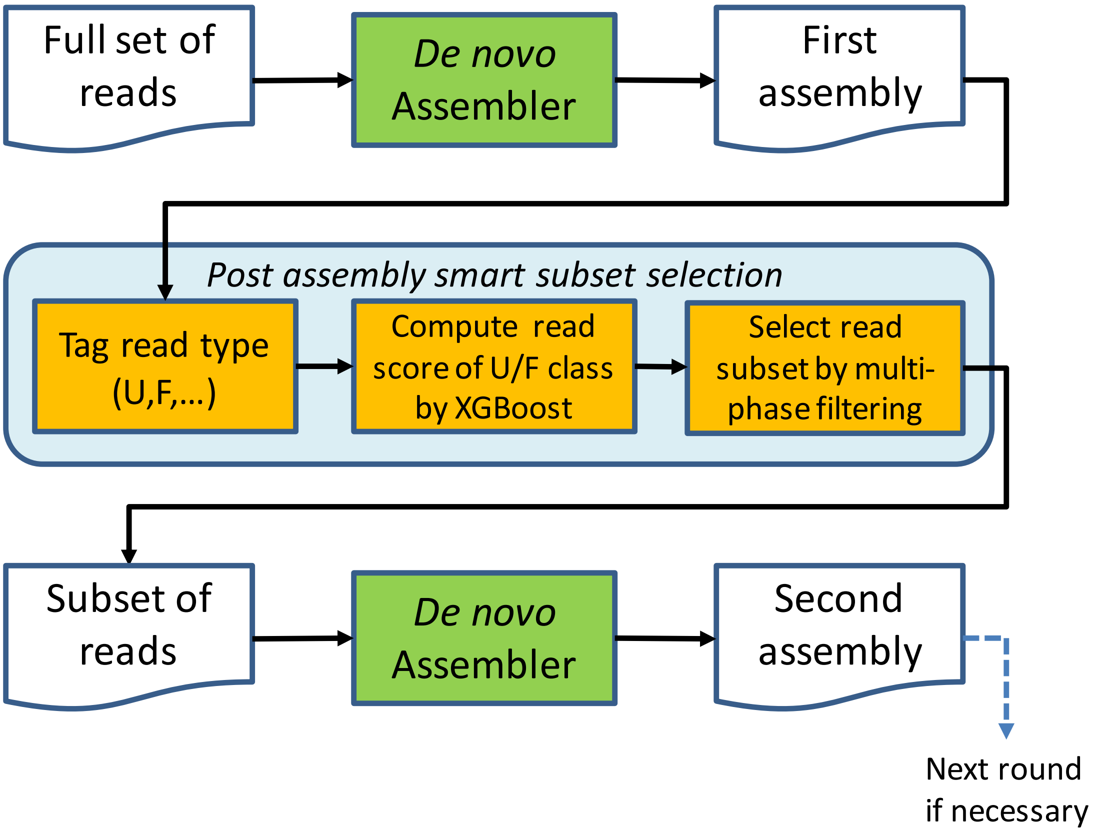


Figure 1 – Workflow of our approach

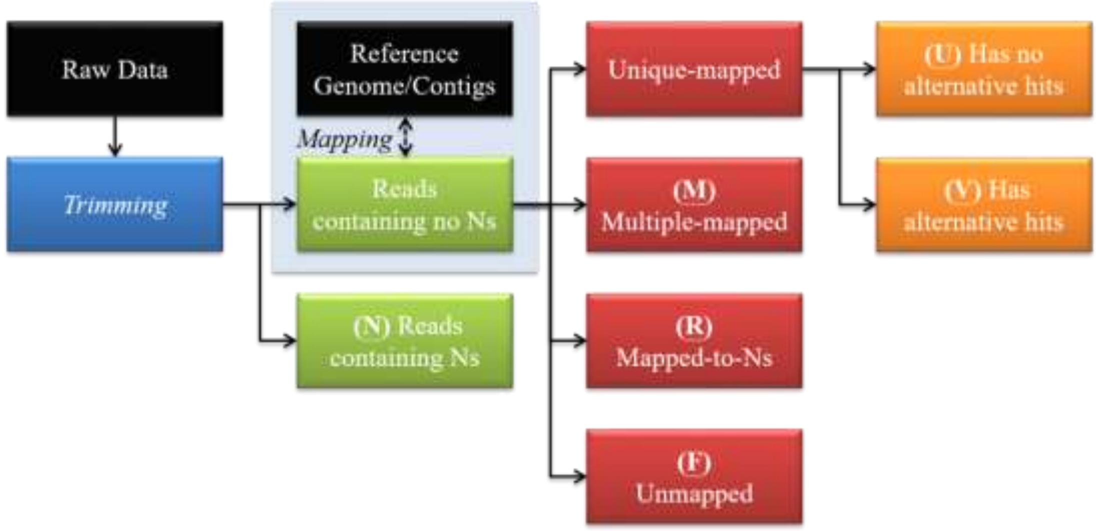
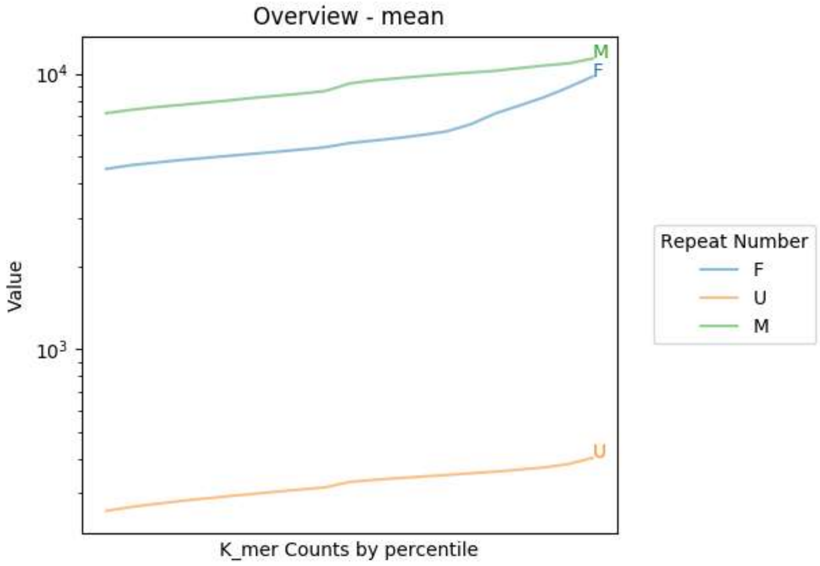


Figure 2 – Categorize the reads by read-mapping

(a) D1



(b) D2

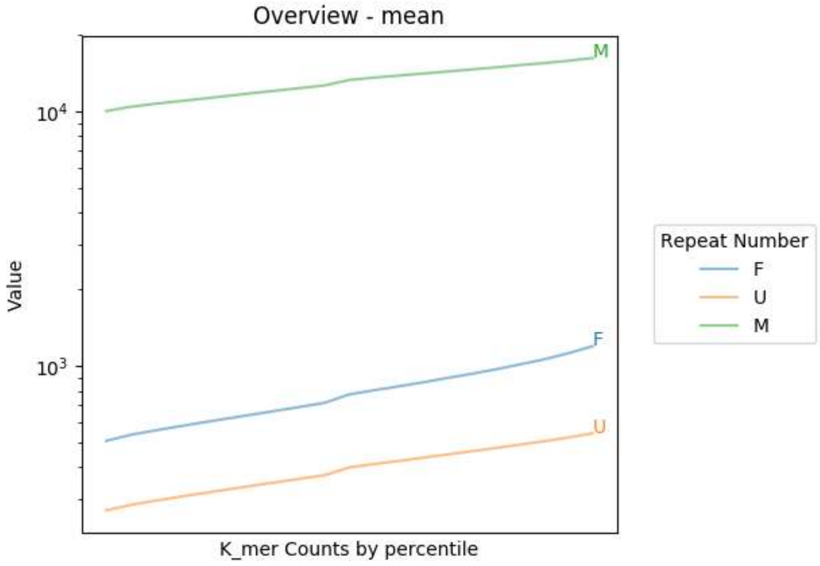


Figure 3 - Feature distribution with K-mer landscaping, separated by F, U, M classes (D1 & D2 datasets)

We first sort Kmer counts of read read in ascending order and compute the percentiles of 0%, 5%, 10%, … 100%. Then for each class we compute the mean value of Kmer counts for each percentile. X-axis is the percentiles. Y-axis is the mean value of Kmer counts for each percentile.

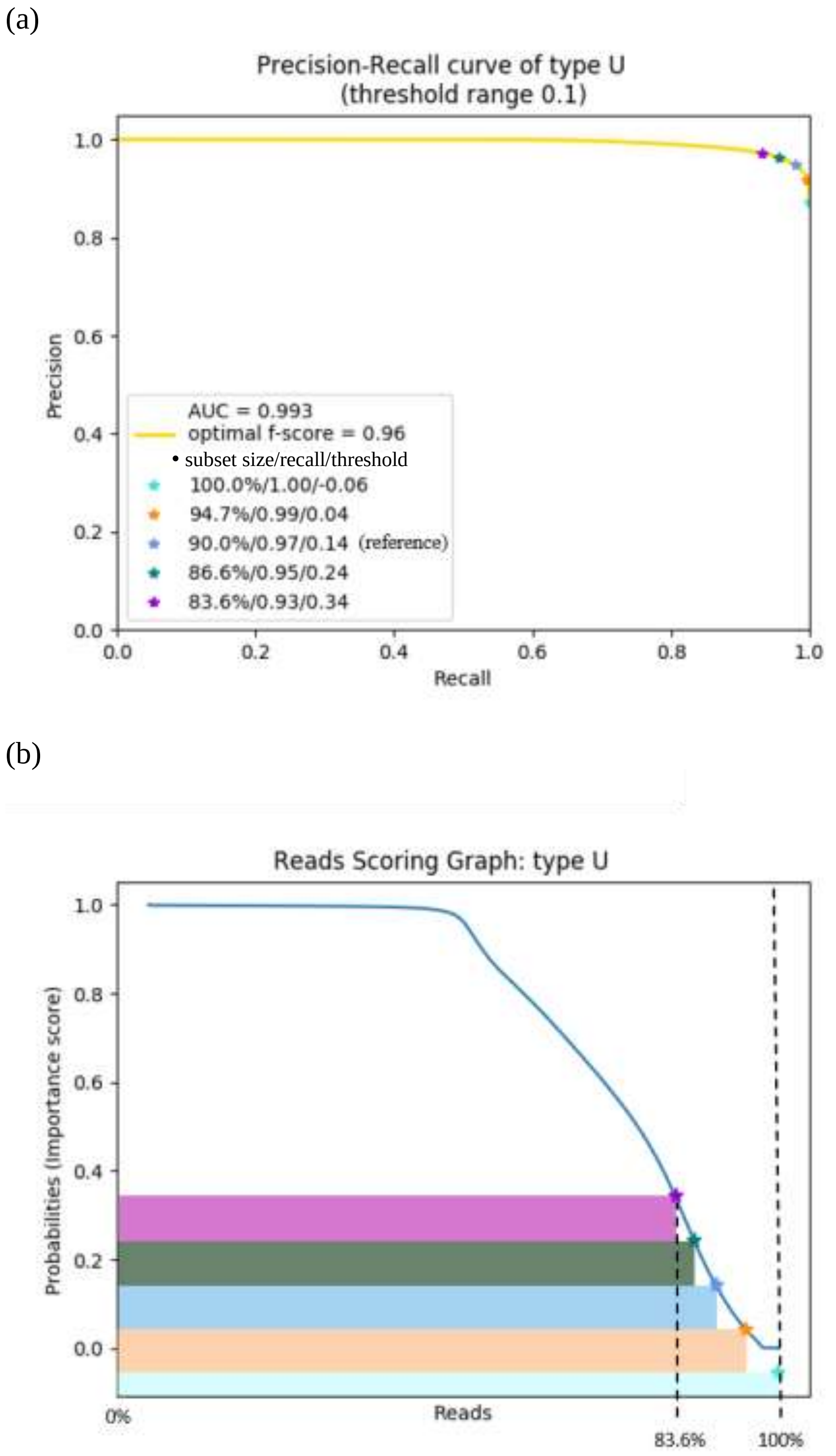
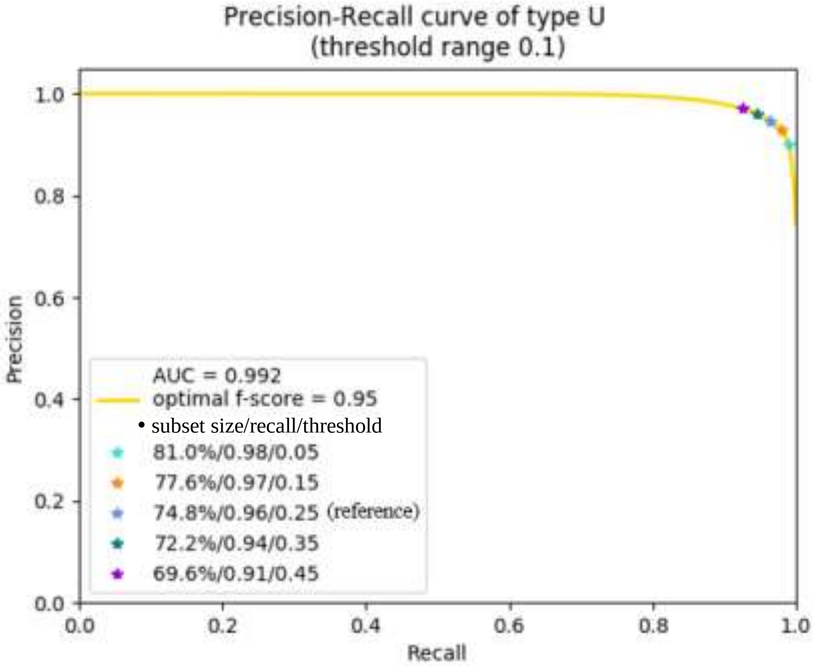


Figure 4 - Precision-Recall curves and reads scoring graph for D1 dataset

(a) Precision-Recall curve of tag U. (b) Reads scoring graph of tag U. The five points of different color on Precision-Recall curve and reads scoring graph represent five adopted selection mechanisms for subset selection. Format of the legend follows the pattern: **subset size/recall/threshold**

(a)



(b)

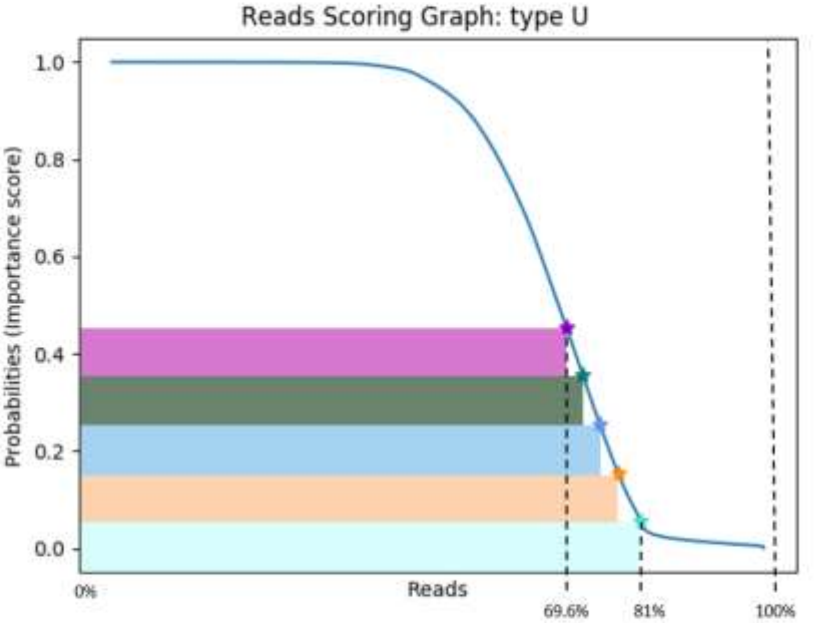


Figure 5 - Precision-Recall curves and reads scoring graph for D2 dataset

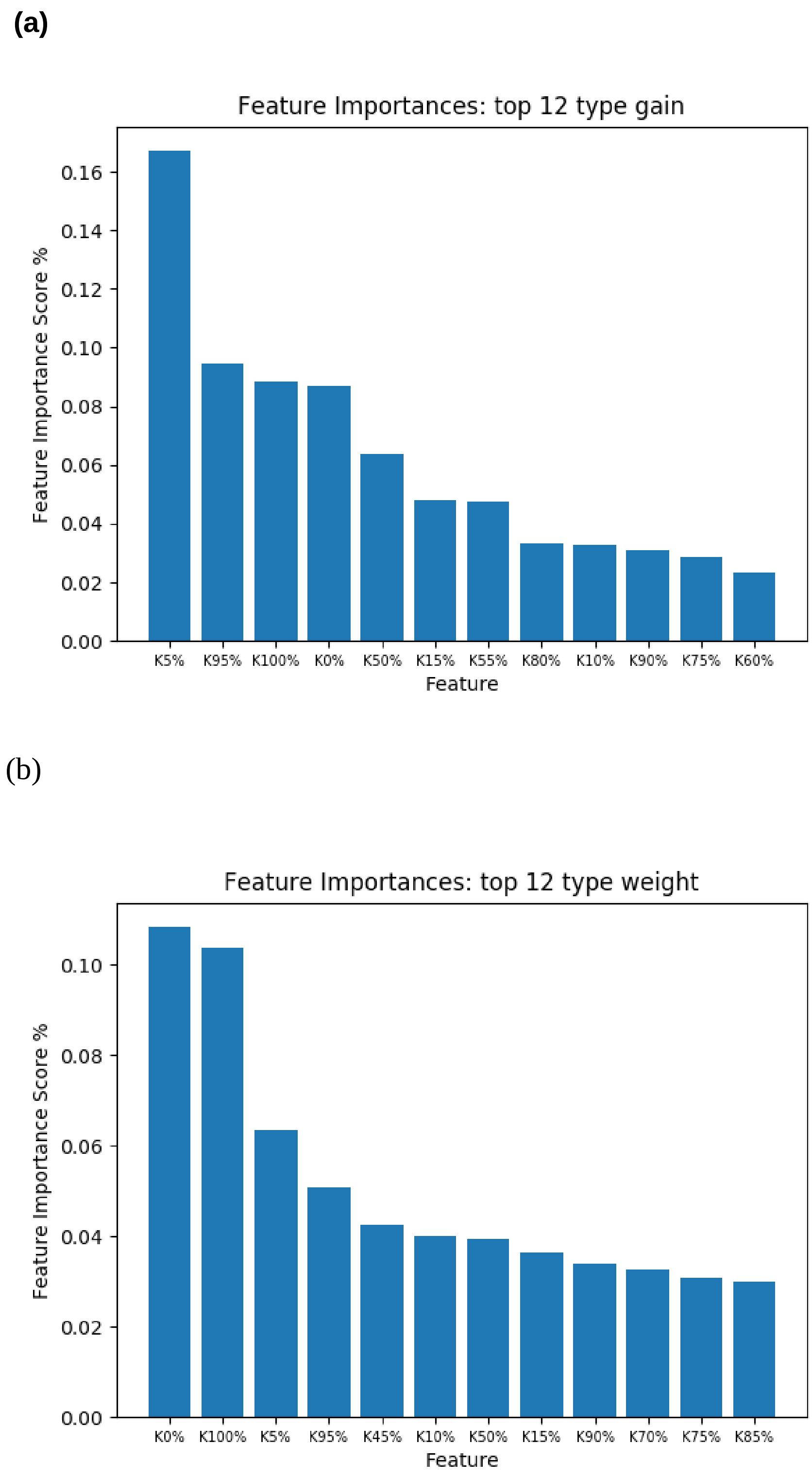


Figure 6 – Feature importance graph of XGBoost model

[1] The graph on the left rank features by their average gain of use in a decision tree

[2] The graph on the right takes the frequency of use into account.

Tables

Table 1 – Summary of read type tagging using BWA

N	Reads containing Ns
U	Unique reads with no alternative hits
F	Reads that fail to map to the contigs
M	Multi-mapped reads
X	Unique reads with no alternative hits
R	Reads that map to the contigs containing Ns

Table 2 - The sequencing datasets used in the experiments.

Dataset	D1	D2
SRA accession number	*SRR352384*	*SRR022866*
# reads	52.06 M	25M
Genome size	12.07 Mbp	2.82 Mbp
Read length	76 bp	76 bp
Reference Genome (NCBI accession number)	*S. cerevisiae* S288c (GCF_000146045.2)	*S.aureus MW2* *(GCF_000146045.2)*

Table 3 – Profile of tagged labels

# reads (Millions)	**Total**	**N**	**F**	**U**	**X**	**M**
D1	51.41 (100%)	0.69 (1.35%)	1.16 (2.26%)	49.16 (95.63%)	0.002 (0.004%)	0.39 (0.75%)
D2	25.00 (100%)	0.22 (0.88%)	5.17 (20.68%)	19.54 (78.14%)	0 (0%)	0.08 (0.3%)

Table 4 – Example of unique reads (U) probability vs threshold value

Reads No.	Threshold Probability[1]	**0.6**	**0.75**	**0.9**
R1	**0.68**	True[2]	False	False
R2	**0.76**	True	False	False
R3	**0.93**	True	True	True

[1] Probability also represents the score or tendency of belonging to the read type.

[2] True/False denotes whether or not the read is classified as a unique read.

Table 5 - The comparison of assembly results between D1-SRR352384 subset selection and the original dataset

Assembly statistics [12]	1st Assembly (Full set)	2nd Assembly U_{82}-F_4	3rd Assembly U_{77}	4rth Assembly U_{69}
Dataset size (M bp)	**51.40 (100%)**	**36.65 (71.3%)**	**33.26 (64.7%)**	**27.66 (53.8%)**
Assembly Size (M bp)	11.46	11.38	11.37	11.21
# c. scaffolds	3,191	2,378	2,129	1,937
# mis-assembly	653	1,164	1,173	1,284
Max. scaffold size (bp)	**54,541**	**64,953**	**58,016**	**76,103**
c. NG25 scaffold size (bp) [3]	**16,356**	**17,436**	**18,347**	**19,230**
c. NG50 scaffold size (bp) [3]	**8,674**	**9,740**	**10,280**	**10,575**
c. NG75 scaffold size (bp) [3]	**3,907**	**4,646**	**4,869**	**4,872**
L99	**3,223**	**1,256**	**941**	**518**
Min. scaffold size (bp)	200	201	200	203
N50 (bp) [4]	11,849	35,411	32,125	36,022

[1] The minus sign implies F classifier filters out unsuitable.

[2] The subscript of each subset denotes the approximate ratio of size extracted from reads.

[3] Here c. NGx stands for corrected NGx contig size.

[4] The contigs with the best N50 are adopted as input to rerun the procedure in the next iteration.

Table 6 - The comparison of assembly results between D2- SRR022866 subset selection and the original dataset

Assembly statistics [12]	1st Assembly (Full set)	2^{nd} Assembly All-F_{20}	3^{rd} Assembly U_{83}-F_{19}	4^{rth} Assembly U_{78}-F_{23}
Dataset size (M bp)	**25 (100%)**	**19.15 (73.4%)**	**20.31 (74.1%)**	**19.24 (69.5%)**
Assembly Size (M bp)	2.87	2.87	2.87	2.87
# c. scaffolds	78	76	74	78
# mis-assembly	17	19	17	15
Max. scaffold size (bp)	**201,558**	**201,558**	**280,578**	**201,558**
c. NG25 scaffold size (bp) [3]	**145,829**	**146,122**	**159,040**	**158,926**
c. NG50 scaffold size (bp) [3]	**75,774**	**92,095**	**100,671**	**100,206**
c. NG75 scaffold size (bp) [3]	**43,112**	**48,055**	**45,827**	**45,827**
L99	**40**	**37**	**37**	**78**
Min. scaffold size (bp)	233	229	233	233
N50 (bp) [4]	129,295	131,359	159,040	131,359

[1] The minus sign implies F classifier filters out unsuitable.

[2] The subscript of each subset denotes the approximate ratio of size extracted from reads.

[3] Here c. NGx stands for corrected NGx contig size.

[4] The contigs with the best N50 are adopted as input to rerun the procedure in the next iteration.

References

1. Sohn & Nam, “The present and future of de novo whole-genome assembly.” *Briefings in Bioinformatics*, Advance Access published October 14, 2016.
2. Wei-Chun Chung, Jan-Ming Ho, Chung-Yen Lin and D. T. Lee, "SSPA: subset selection based on post-assembly analysis for NGS data, " unpublished.
3. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.
4. Bennett, James, and Stan Lanning. "The netflix prize." Proceedings of KDD cup and workshop. Vol. 2007. 2007.
5. Li, Heng, Jue Ruan, and Richard Durbin. "Mapping short DNA sequencing reads and calling variants using mapping quality scores." *Genome research* 18.11 (2008): 1851-1858.
6. Steiner, Andreas, et al. "KvarQ: targeted and direct variant calling from fastq reads of bacterial genomes." *BMC genomics* 15.1 (2014): 881.
7. Wood, Derrick E., and Steven L. Salzberg. "Kraken: ultrafast metagenomic sequence classification using exact alignments." *Genome biology* 15.3 (2014): R46.
8. Ounit, Rachid, et al. "CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers." *BMC genomics* 16.1 (2015): 236.
9. Li, Heng, and Richard Durbin."Fast and accurate short read alignment with Burrows–Wheeler transform." *Bioinformatics* 25.14 (2009): 1754-1760.
10. Li, Heng, et al. "The sequence alignment/map format and SAMtools." *Bioinformatics* 25.16 (2009): 2078-2079.

11. Bankevich, Anton, et al. "SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing." *Journal of computational biology* 19.5 (2012): 455-477.

12. Gurevich, Alexey, et al. "QUAST: quality assessment tool for genome assemblies." *Bioinformatics* 29.8 (2013): 1072-1075.

13. Delcher, Arthur L., et al. "Alignment of whole genomes." *Nucleic acids research* 27.11 (1999): 2369-2376.

14. " N50, L50, and related statistics" https://en.wikipedia.org/wiki/N50,_L50,_and_related_statistics, accessed: 2017-08-20

15. Escalante, Hugo Jair, Manuel Montes, and Luis Enrique Sucar. "Particle swarm model selection." Journal of Machine Learning Research 10.Feb (2009): 405-440.

16. Magoc, Tanja, et al. "GAGE-B: an evaluation of genome assemblers for bacterial organisms." *Bioinformatics* 29.14 (2013): 1718-1725.

Appendix A:

Assembly Results for Subset Selection on GAGE-B

In the appendix, we present the assembly results of GAGE-B Miseq datasets. Note that the assembly process is identical to the aforementioned approach in section Materials and Methods, including pre-assembly processing (trimming), subset selection, paired-end assembly, and post-assembly processing (QUAST evaluation). The only exception is only one iteration of results is shown. Note that the notation of table A.1-A.3 are the same as Table 5 & 6. Additionally, we also indicate the number of N's per 100kbp in scaffolds.

Tables

Table A.1 - The comparison of assembly results between *R. sphaeroides* 2.4.1 subset selection and the original dataset

	1st Assembly	2nd Assembly
Subset Size	100%	93.10%
# Scaffolds	105	63
Scaffold NG50 (Kbp)	1,128,248	551,193
# c. Scaffolds	57	62
c. Scaffold Assembly Size (Mbp)	4,588,562	4,583,183
Max c. Scaffold (Kbp)	1,135,028	1,135,028
Scaffold c. NG25 (Kbp)	597,450	577,045
Scaffold c. NG50 (Kbp)	518,052	518,330
Scaffold c. NG75 (Kbp)	235,197	162,622
Scaffold LG99	20	23
# N's per 100 kbp	8.06	8.09

Table A.2 - The comparison of assembly results between *M. abscessus* 6G-0125-R subset selection and the original dataset

	1st Assembly	2nd Assembly
Subset Size	100%	78%
# Scaffolds	918	95
Scaffold NG50 (Kbp)	574,745	344,229
# c. Scaffolds	56	64
c. Scaffold Assembly Size (Mbp)	5,064,454	5,063,988
Max c. Scaffold (Kbp)	805,304	780,029
Scaffold c. NG25 (Kbp)	639,089	372,912
Scaffold c. NG50 (Kbp)	280,233	280,245
Scaffold c. NG75 (Kbp)	144,656	110,733
Scaffold LG99	16	23
# N's per 100 kbp	0	0

Table A.3 - The comparison of assembly results between *V. cholerae* CO1032 subset selection and the original dataset

	1st Assembly	2nd Assembly
Subset Size	100%	98.20%
# Scaffolds	1,863	1,818
Scaffold NG50 (Kbp)	356,090	356,090
# c. Scaffolds	100	100
c. Scaffold Assembly Size (Mbp)	3,957,272	3,957,228
Max c. Scaffold (Kbp)	737,832	737,800
Scaffold c. NG25 (Kbp)	548,525	548,525
Scaffold c. NG50 (Kbp)	227,910	227,910
Scaffold c. NG75 (Kbp)	152,234	152,234
Scaffold LG99	78	80
# N's per 100 kbp	0	0

Appendix B:

Read Type Tagging Analysis

In the appendix, we analyze the process of read type tagging with Burrows-Wheeler Aligner [9]. BWA is a software package for mapping sequences against a large reference genome. It outputs the alignment results in Sequence Alignment/Map [10] format consisting of 11 mandatory fields for essential alignment information such as mapping quality score (MAPQ), CIGAR string, and number of mismatch. Our goal is to further differentiate uniquely mapped reads into three types based on these fields.

To begin with, we discover that among unique reads, some of the alignments only contain alignment match. As shown in Figure B.1, these alignments correspond to the M operation and M operation only in the CIGAR string. For the reads without any sequence mismatch, we label them with type P as perfectly mapped. For the reads with at least one mismatch position, we label them with type S as alignments consisting of substitution error. The rest of the reads contains other operation in their CIGAR string, including clips (S or H), insertion (I), and deletion (D). They are tagged with type O. We summarize the definition in Table B.1 for convenience.

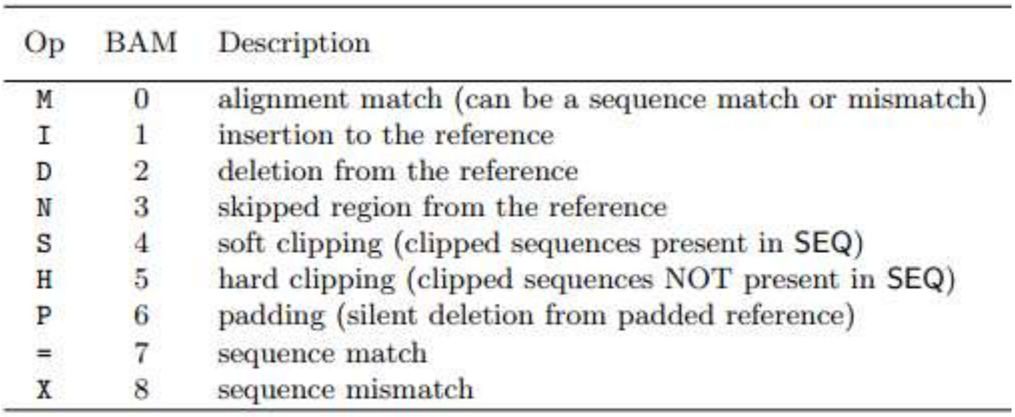
Because the assembly results of GAGE-B dataset presented in Appendix A are not as promising as those of D1 & D2, we investigate the read label distribution assigned by BWA and discover that there exist clips in the majority of sequencing reads of GAGE-B dataset. To understand the proportion of the clip in an alignment, we define clip ratio as follows:

$$\text{Clip Ratio} = \#\text{clip} \,/\, \text{read length}$$

Figure B.2 computes the distribution of clip rate for two of the GAGE-B dataset. We can see that clip rate varies between species and can consist of up to 80 percent of the read, a considerably large number beyond our expectation.

The comparison between the label distribution of D1 and *R. sphaeroides* 2.4.1 in Table B.2 allows us to infer it is the inferior quality of GAGE-B sequencing dataset which exacerbates the assembly results after subset selection. Despite little improvement on *De Novo* assembly, an analytical tool that incorporates such discoveries along with the aforementioned clip rate analysis is worth developing to examine dataset quality before assembly process in the future.

Figures



Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch

Fig. B.1 - CIGAR string table with description

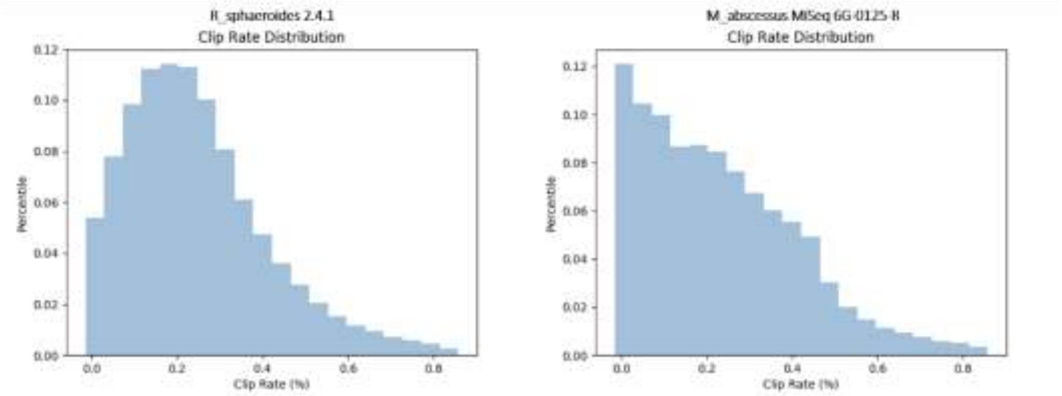


Fig. B.2 – Clip rate distribution graph of R.sphaeroides 2.4.1 & M.abscessus 6G-0125-R

Tables

Table B.1 – Summary of detailed read type tagging using BWA

N	Reads containing Ns
P	Unique reads with only alignment match (M) but no mismatch
S	Unique reads with only alignment match (M) and at least one mismatch
O	The rest of the uniquely mapped reads (contains operations other than M).
F	Reads that fail to map to the contigs
M	Multi-mapped reads
R	Reads that map to the contigs containing Ns