BIOINFORMATICS

Vol. 00 no. 00 2015
Pages 1–6

VALET: *de novo* pipeline for finding metagenomic mis-assemblies

Christopher Michael Hill [1,2,*], Jonathan Gluck [1], Victoria Cepeda [1,2], Matheiu Almedia [2], Sergey Koren [1,2], and Mihai Pop [1,2*]

[1]Department of Computer Science, University of Maryland, College Park, Maryland, 20742 USA
[2] Center for Bioinformatics and Computational Biology, University of Maryland, College Park, Maryland, 20742 USA.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text.

Results: Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text

Availability: Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text

Contact: name@bio.com

1 INTRODUCTION

Genome assembly of single organisms is made difficult due to the presence of sequencing errors and repeats. This difficulty is compounded in metagenomic samples due to the addition of varying organism abundances, intrapopulational variations, and conserved genomic regions between closely-related species. Since many downstream applications rely on these assembled genomes, it is critical that the assembly is error-free. Existing methods for finding mis-assemblies have primarily focused on single genome assembly and fall into two categories: reference-based and *de novo* evaluation.

Reference-based methods rely on having a collection of, often manually curated, reference genomes, while *de novo* methods look for inconsistencies between characteristics of the data generation process and the resulting assembly. QUAST [Gurevich *et al.*, 2013] is a tool that can identify mis-assemblies and structural variants when provided with a reference genome. QUAST leverages existing methods (Plantagora [Barthelson *et al.*, 2011], GeneMark.hmm [Lukashin and Borodovsky, 1998], GlimmerHMM [Majoros *et al.*, 2004]) and quality metrics (GAGE [Salzberg *et al.*, 2012]). QUAST uses the Plantagoras definition of a mis-assembly, i.e., a mis-assembly breakpoint is defined as a position in the assembled contigs where (1) the left flanking sequence aligns over 1kb away from right flanking sequence on the reference, or (2) the sequences overlap by over 1kb, or (3) the right flanking sequence aligns on opposite strands or different chromosomes.

De novo techniques for detecting mis-assemblies in single genomes rely on looking for inconsistencies between the sequence generation process and the resulting assembly. In other words, given a model of the sequencing process, could the sequences have been generated if the assembly was the truth (inserted into the sequencing machine)? Regions of the assembly that do not meet these assumptions are signatures of potential mis-assemblies. One assumption is that the sequence generation process is roughly uniform, i.e., sequences starting at any position have equal probability. Substantially divergent coverage may indicate mis-assembly. If the sequences are paired-end or mate-pair then additional constraints based on insert size can be used.

Amosvalidate [Phillippy *et al.*, 2008] is a *de novo* pipeline for detecting mis-assemblies that incorporates the above constraints. As mentioned in the previous chapter, REAPR [Hunt *et al.*, 2013] is a tool that leverages insert size constraints and evaluates the accuracy of the assembly using read-pair coverage. REAPR determines the fragment coverage by first independently aligning the read-pairs to the assembly. A fragment is defined as the distance from the end points of proper read-pairs (those that are determined to have correct orientation and separated by the correct distance). REAPR is able to find base-level errors by comparing the fragment coverage of a given base with the theoretical coverage.

Although all the above-mentioned tools were designed to work on single genomes, they do not function correctly on metagenomic assemblies. QUAST relies on the existence of reference genomes, which are simply not available for most metagenomic samples. Furthermore, if the *correct* reference strain is not available, then QUAST may erroneously flag correct and biologically novel sequence as mis-assembled. The *de novo* tools REAPR and amosvalidate rely on global uniform sequence coverage to flag regions. In previous chapters, we have shown that contigs within the metagenomic assemblers vary widely in abundances. Assuming uniform coverage will cause these tools to erroneously flag regions as mis-assembled. In this chapter, we detail how to modify the constraints of existing tools to allow them to work with

*to whom correspondence should be addressed

metagenomic assemblies. The result is VALET, the first *de novo* pipeline for detecting mis-assemblies in metagenomic assemblies.

2 METHODS

2.1 Types of mis-assemblies

The majority of mis-assemblies fall into two categories: (1) compression/expansion of repetitive sequence and (2) sequence rearrangements. The first category of mis-assembly results when an assembler is unable to determine the correct copy count of repeats, leading to additional or fewer copies. The second category results when an assembler erroneously links separate unique portions of the genome that lie adjacent to a repeat. The repeat acts as a bridge joining the two separate parts of the genome together. Each category of mis-assembly has its own signatures that can be used to identify potential mis-assemblies.

The sequencing process of randomly-sheared fragments follows a Poisson distribution [Lander and Waterman, 1988]. Regions within the assembly that show high variance in **depth of coverage** are a potential signature of compressed/expanded repeats, chimeric contigs, and other types of contamination.

The reads given to the assembler should be **alignable** to the resulting assembly. In practice, however, a read may fail to align for a few reasons. In metagenomic samples, an unaligned read can be from a rare, low coverage organism, and was never assembled with any other reads. A read with a large amount of errors will be unable to align within a specified similarity to the assembly. A read can be sequenced from a unfiltered contaminant or primer. If a read does not fall into one of the above categories, then it may be a sign of a potential mis-assembly.

Another signature that is used to find mis-assemblies relies on finding regions of the assembly that violate mate-pair **insert size constraints**. Certain sequencing technology allows researchers to sequence from the ends of a DNA fragment of a known insert size. Although the sequence technology can only give the raw sequence of the first couple hundred basepairs from the ends of the fragment, the distance between the ends of the sequences can be used to aid in resolving repeats, and orienting and scaffolding contigs. Regions of an assembly containing a disproportionate number of mate-pairs (reads from the same fragment) with incorrect insert sizes may be a potential mis-assembly.

VALET flags regions of the assembly based on (1) sampling depth, (2) alignability of the sequences, and (3) insert size constraints.

2.2 Estimating contig abundances using *k*-mers

An important part of most metagenomic pipelines is determining the relative abundance of each contig. The presence of repeats among different species poses a serious problem for estimating abundances. Short-read alignment tools such as Bowtie2 often randomly assign sequences that align equally well to multiple positions ignoring the relative abundances of the underlying contigs. This poses a *chicken or the egg* type problem because the sequence alignments are used to determine the contig abundances. Here we solve this problem by using the uniquely alignable sequences to establish an initial contig abundance. Then when we encounter a sequence that can align multiple locations, we randomly assign it based on the relative abundance of the corresponding contigs. We update the contig abundances and repeat the above step for a given number of iterations (30 by default). This approach is similar in spirit to that of Sailfish [Patro *et al.*, 2014] which performs alignment-free abundance estimation of RNA-seq reads.

2.3 Depth of coverage analysis

In order to find regions of unexpectedly high/low coverage, we first learn the distribution of per-base coverages across a given contig. Using this distribution, bases are marked if their coverage falls below or above a certain threshold. We set the lower cutoff as the first quartile minus 1.5 $\times$ the interquartile range (IQR), which is the difference between the first and third quartile. 1.5 $\times$ IQR is the value used by Tukeys box plot [McGill *et al.*, 1978]. Regions whose coverage is greater than the third quartile plus 1.5 $\times$ IQR are marked as high coverage.

Using the per-base coverages may result in a large number of regions erroneously marked as mis-assemblies due to the inherent noisiness of the data, so we also provide a sliding window approach to smooth out the per-base coverages. The larger the window, the fewer the regions marked as mis-assemblies. VALET uses a window size of 300 bp by default.

2.4 Insert size consistency

VALET relies on the REAPR [Hunt *et al.*, 2013] pipeline to identify mate-pair insert size inconsistencies. REAPR works by first sampling the fragment coverage across the genome to get average fragment length and depth of coverage. Using this information, REAPR scans the assembly for observed regions that differ from the expected fragment length distribution and orientations.

REAPR is designed to work with single genome assemblies, more specifically, assemblies with a global uniform coverage. Since the contig abundances can vary drastically in metagenomic assemblies, VALET first bins contigs by similar abundances and then executes the REAPR pipeline on the binned contigs.

2.5 Identifying assembly breakpoints

Possible breakpoints in the assembly are found by examining regions where a large number of partial reads are able to align. We evenly split each unaligned read into *sister* reads. The *sister* reads are then aligned independently back to the reference genome. We partition the provided assembly into bins (100 bp by default) and record which bins correspond to the sister reads. If we find a pair of bins that contain at least two different pairs of *sister* reads, then we mark it as a breakpoint location.

2.6 Comparing multiple assemblies

We visualize the quality of an assembly by recording the number of errors accumulated as we add contigs in decreasing order of length. This allows us to visually compare a set of metagenomic assemblies.

2.7 VALET pipeline

VALET takes as input a metagenomic assembly FASTA file and a collection of paired and un-paired reads (Figure 1). Assembled contigs are first filtered out based on size (2x the average read length by default). Next the abundances of contigs are calculated using our k-mer-based approach described above. Contigs undergo

an additional filtering step based on abundance (10x by default). Higher coverage and longer sequence provide a better baseline for detecting mis-assemblies.

Once filtering has finished, regions of the assembly are flagged based on the inconsistencies described above. In practice, most mis-assembly signatures have high false positive rates which can be reduced by looking at regions where multiple signatures agree. Therefore, any window of the assembly (2000 in length by default) that contains multiple mis-assembly signatures are marked as **suspicious**. The flagged and suspicious regions are stored in a GFF file, which allows users to visualize the mis-assemblies using any genomic viewers, such as IGV [Thorvaldsdóttir *et al.*, 2012].

3 RESULTS

3.1 VALET achieves high sensitivity on a simulated metagenomic community

We examine the sensitivity of VALET on a toy simulated metagenomic community consisting of four bacteria at varying abundances: *Bacteroides vulgatus* (80x), *Bacillus cereus* (60x), *Actinomyces odontolyticus* (40x), and *Acinetobacter baumannii* (20x). Approximately four million reads are simulated using WGSIM [Li, 2013] with default parameters. The dataset was assembled using IDBA-UD [Peng *et al.*, 2012], MetaVelvet [Namiki *et al.*, 2012], SPAdes [Bankevich *et al.*, 2012], and SOAPdenovo2 [Luo *et al.*, 2012]. We ran VALET on the assemblies and compared the errors found with the reference-based mis-assemblies detected by QUAST (Table 1 and Figure 2). If any part of a region flagged by VALET overlaps with a mis-assembled region reported by QUAST, we consider it a true positive (mis-assembly correctly identified by our method).

Across all assemblers, VALET detects greater than 80% of mis-assemblies detected by QUAST.IDBA-UD has the greatest N50 after breaking the assembly at regions marked by QUAST (206.7 Kbp), followed by SPAdes (128.1 Kbp), MetaVelvet (29.5 Kbp), and SOAPdenovo2 (10.8 Kbp). These rankings match those provided by VALET (Figure 2).

3.2 VALET accurately evaluates assemblies of a synthetic metagenomic community

A major challenge of evaluating assemblies of environmental datasets is that a sizeable portion of the organsisms are unknown or lack a draft genome to compare against. In silico simulations often lack the complexity and sequencing biases present in real environmental samples. Fortunately, Shakya et al. provide a *gold standard* synthetic metagenomic dataset containing a mixture of 64 organisms (16 members of Archaea and 48 organisms from 18 Bacteria phyla) with complete or high-quality draft genomes and 200-fold differences in abundances [Shakya *et al.*, 2013]. The dataset consists of 53.4 million reads (101 bp in length). Due to the greater size and complexity of this dataset compared to the previous simulation, we assemble the dataset using two recent, fast metagenomic assemblers: Omega [Haider *et al.*, 2014] and MEGAHIT[Li *et al.*, 2015]. We run VALET on the assemblies and compare the errors found with those reference-based mis-assemblies detected by QUAST (Table 3.2).

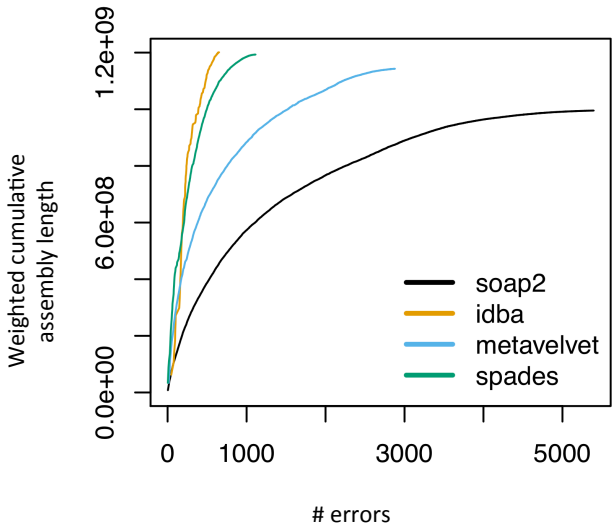

Fig. 2. FRC plot provided by VALET of a simulated mock community.

While the MEGAHIT and Omega assemblies are close in total size (192.3 Mbp vs. 194 Mbp, respectively), MEGAHIT has nearly twice as many contigs as Omega (19,145 vs. 10,284). QUAST detects far more mis-assemblies in the Omega assembly compared to MEGAHIT (56,917 vs. 770, respectively). VALET detects 34.80% and 96.10% of these mis-assemblies found by QUAST in the MEGAHIT and Omega assemblies, respectively. While Omega has a higher N50 than MEGAHIT (44.1 Kbp vs. 38.9 Kbp), after breaking at mis-assemblies, the N50 drops well below MEGAHIT's (11.9 Kbp vs. 33.5 Kbp), illustrating why the N50 metric is not always a good indicator of assembly quality. VALET is able to accurately assess the quality of the two assemblies without using the reference genomes.

We investigate the high false positive rate by examining a small number of regions flagged by VALET, but not marked by QUAST within the MEGAHIT assembly. One contig, roughly 25 Kbp in size, had a 5 Kbp region at the start of the contig marked as high coverage (Figure 3). This region was roughly 4x the median coverage of the remaining contig. Using NCBI's BLAST [**?**] and reference database, the region aligned to the organism *Nostoc* sp. PCC 7120. Upon closer inspection, this region contained 16S, 23S, and 5S rRNA genes and was found at *four* locations in *Nostoc* sp. PCC 7120. This region was only found once in the assembly, so all the sequences from the repeats aligned to this region, inflating the coverage. This noticeable and consistent increase in coverage caused VALET to mark it as mis-assembled. Unsurprisingly, QUAST did not mark this as a mis-assembly because the actual sequence within this region matched the reference.

4 DISCUSSION

In practice, VALET has high sensitivity for mis-assembly detection, but also a high false positive rate. While we can tune parameters, such as window size, to trade-off between the measures, the high

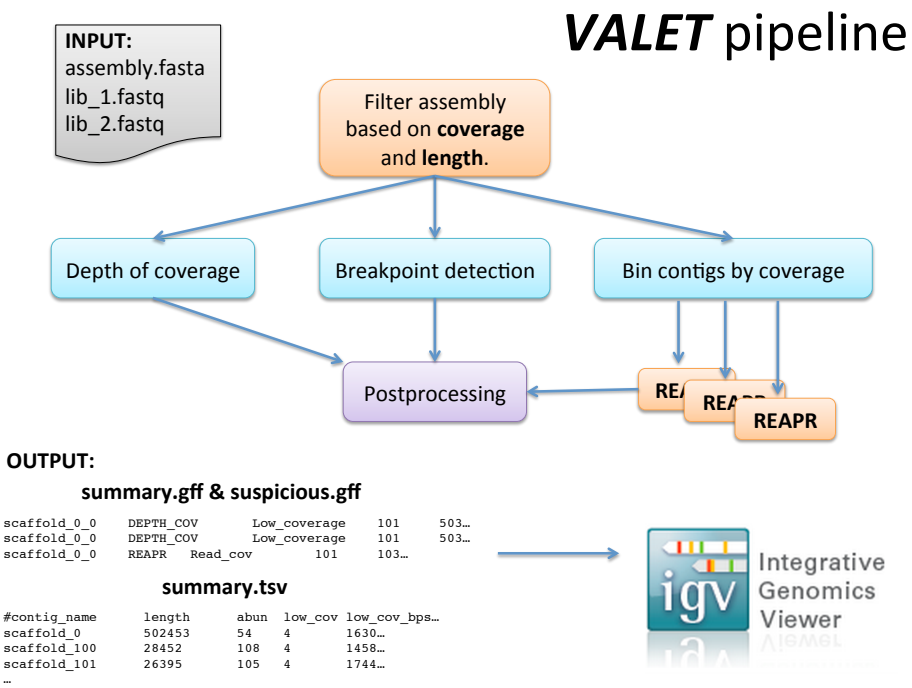

Fig. 1. Overview of the VALET pipeline.

						Mis-assembly signatures			Suspicious regions			
Assembler	Len (Mbp)	Ctgs	N50 (Kbp)	NA50	Errs	Num	Valid	Sens	Num	Valid	Sens	Mismatches per Kbp
IDBA-UD	16.5	200	208.8	206.7	36	804	36	100.00%	25	8	22.20%	23.95
MetaVelvet	16.3	1,117	29.5	29.5	21	2,802	19	90.50%	4	2	9.50%	35.52
SPAdes	16.4	330	130.9	128.1	37	1,117	31	83.80%	17	4	10.80%	22.43
Soapdenovo2	12.3	2,161	10.8	10.8	1	4,983	1	100.00%	2	0	0%	13.37

Table 1. VALET results for simulated mock community consisting of four bacteria at varying abundances: *Bacteroides vulgatus* (80x), *Bacillus cereus* (60x), *Actinomyces odontolyticus* (40x), and *Acinetobacter baumannii* (20x). Reads were assembled using the four provided assemblers. General assembly statistics include length in Mbp (Len), number of contigs (Ctgs), N50 contig size (N50), N50 of contigs after broken at mis-assemblies (NA50), number of errors detected by QUAST (Errs), number of flagged regions by VALET (Num), number of flagged regions that overlap an error found by QUAST (Valid), sensitivity (Sens), and mismatches per Kbp.

						Mis-assembly signatures			Suspicious regions			
Assembler	Len (Mbp)	Ctgs	N50 (Kbp)	NA50 (Kbp)	Errs	Num	Valid	Sens	Num	Valid	Sens	Mismatches per Kbp
MEGAHIT	192.3	19,145	38.9	33.5	770	30,377	268	34.80%	2,239	100	13.00%	92.24
Omega	194	10,284	44.1	11.9	56,917	1,425,127	55,108	96.10%	17,758	13,935	96.80%	98.55

Table 2. VALET results for assemblies of the Shakya et al. [Shakya *et al.*, 2013] dataset. General assembly statistics include length in Mbp (Len), number of contigs (Ctgs), N50 contig size (N50), N50 of contigs after broken at mis-assemblies (NA50), number of errors detected by QUAST (Errs), number of flagged regions by VALET (Num), number of flagged regions that overlap an error found by QUAST (Valid), sensitivity (Sens), and mismatches per Kbp.

false positive rate still remains fairly prevalent. Some of the false positives can be explained as the assembler deduplicates repetitive regions of the genome, e.g., the ribosomal genes. This highlights an important issue prevalent in metagenomic assemblers. In the Shakya et al. dataset, a more *correct Nostoc* sp. PCC 7120 assembly would include an additional contig consisting solely of the ribosomal genes. Then during the abundance estimation step of VALET, a quarter of the sequences would align to the original contig due to the flanking unique region and the remaining three quarters would align solely to the new contig containing the ribosomal genes. VALET would no longer mark this region in the original contig.

Assemblathon1 [**?**] has stated that assemblers have trouble with polymorphism and heterozygosity. This problem is compounded in metagenomic assemblies due to closely-related strains having uneven abundances. MetaCompass [**?**], a reference-based metagenomic assembler, was used to assemble the HMP Sample SRS024655 (retroauricular crease of a male). A 25 Kbp region was flagged as having low coverage by VALET, but not reported by QUAST

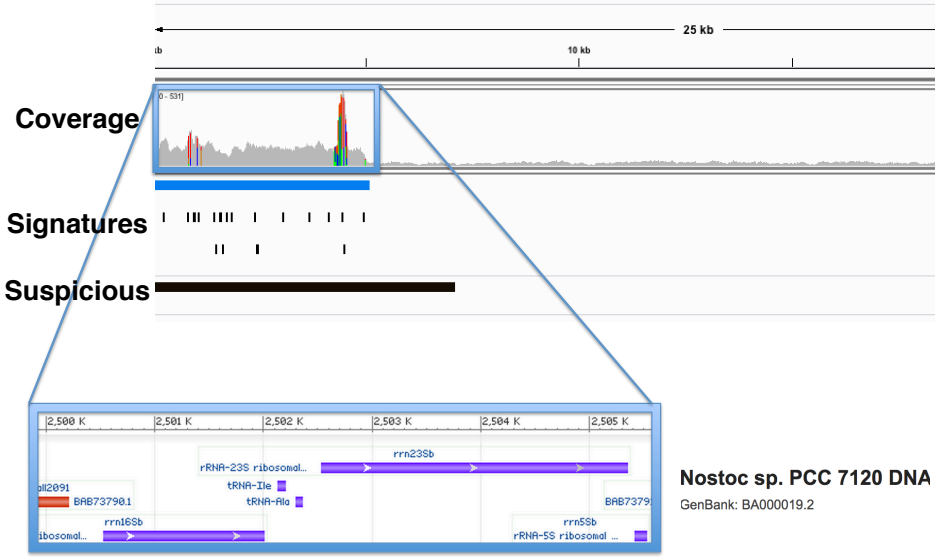

Fig. 3. A closer examination of a region flagged by VALET, but no mis-assembly reported by QUAST. This region contained 16S, 23S, and 5S rRNA genes and was found at *four* locations in the *Nostoc* sp. PCC 7120 genome.

(Figure 4). After further investigation, the 25 Kbp region belonged exclusively to the one of the reference genomes chosen by MetaCompass: *Propionibacterium acnes* KPA171202. The higher coverage flanking regions aligned to both *Propionibacterium acnes* KPA171202 and *Propionibacterium acnes* ATCC 11828. *Propionibacterium acnes* KPA171202 contains nearly 70 Kbp of insertions. Despite being found at a lower abundance, the KPA171202 strain of *Propionibacterium acnes* was chosen for the reference-guided assembly because all reads that were align to the ATCC 11828 strain also aligned to the KPA171202 strain. Since the KPA171202 strain was actually found in the dataset, QUAST detected no structural errors. A more *correct* assembly would include both complete genomes.

5 CONCLUSION

VALET is the first *de novo* pipeline for detecting mis-assemblies in metagenomic datasets. VALET searches for regions of the assembly that are statistically inconsistent with characteristics of the data generation process. VALET finds mis-assemblies on a simulated and synthetic metagenomic mock community.

ACKNOWLEDGEMENT

=

Funding: Text Text Text Text Text Text Text Text.

REFERENCES

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.* (2012). SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, **19**(5), 455–477.

Barthelson, R., McFarlin, A. J., Rounsley, S. D., and Young, S. (2011). Plantagora: modeling whole genome sequencing and assembly of plant genomes. *PLoS One*, **6**(12), 1–9.

Gurevich, A., Saveliev, V., Vyahhi, N., and Tesler, G. (2013). QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, **29**(8), 1072–1075.

Haider, B., Ahn, T.-H., Bushnell, B., Chai, J., Copeland, A., and Pan, C. (2014). Omega: an overlap-graph de novo assembler for metagenomics. *Bioinformatics*, page btu395.

Hunt, M., Kikuchi, T., Sanders, M., Newbold, C., Berriman, M., and Otto, T. D. (2013). Reapr: a universal tool for genome assembly evaluation. *Genome biology*, **14**(5), R47.

Lander, E. S. and Waterman, M. S. (1988). Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, **2**(3), 231–239.

Li, D., Liu, C.-M., Luo, R., Sadakane, K., and Lam, T.-W. (2015). MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph. *Bioinformatics*, page btv033.

Li, H. (2013). wgsim-read simulator for next generation sequencing.

Lukashin, A. V. and Borodovsky, M. (1998). GeneMark.HMM: new solutions for gene finding. *Nucleic acids research*, **26**(4), 1107–1115.

Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., He, G., Chen, Y., Pan, Q., Liu, Y., *et al.* (2012). SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience*, **1**(1), 18.

Majoros, W. H., Pertea, M., and Salzberg, S. L. (2004). TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics*, **20**(16), 2878–2879.

McGill, R., Tukey, J. W., and Larsen, W. A. (1978). Variations of box plots. *The American Statistician*, **32**(1), 12–16.

Namiki, T., Hachiya, T., Tanaka, H., and Sakakibara, Y. (2012). MetaVelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic acids research*, **40**(20), e155–e155.

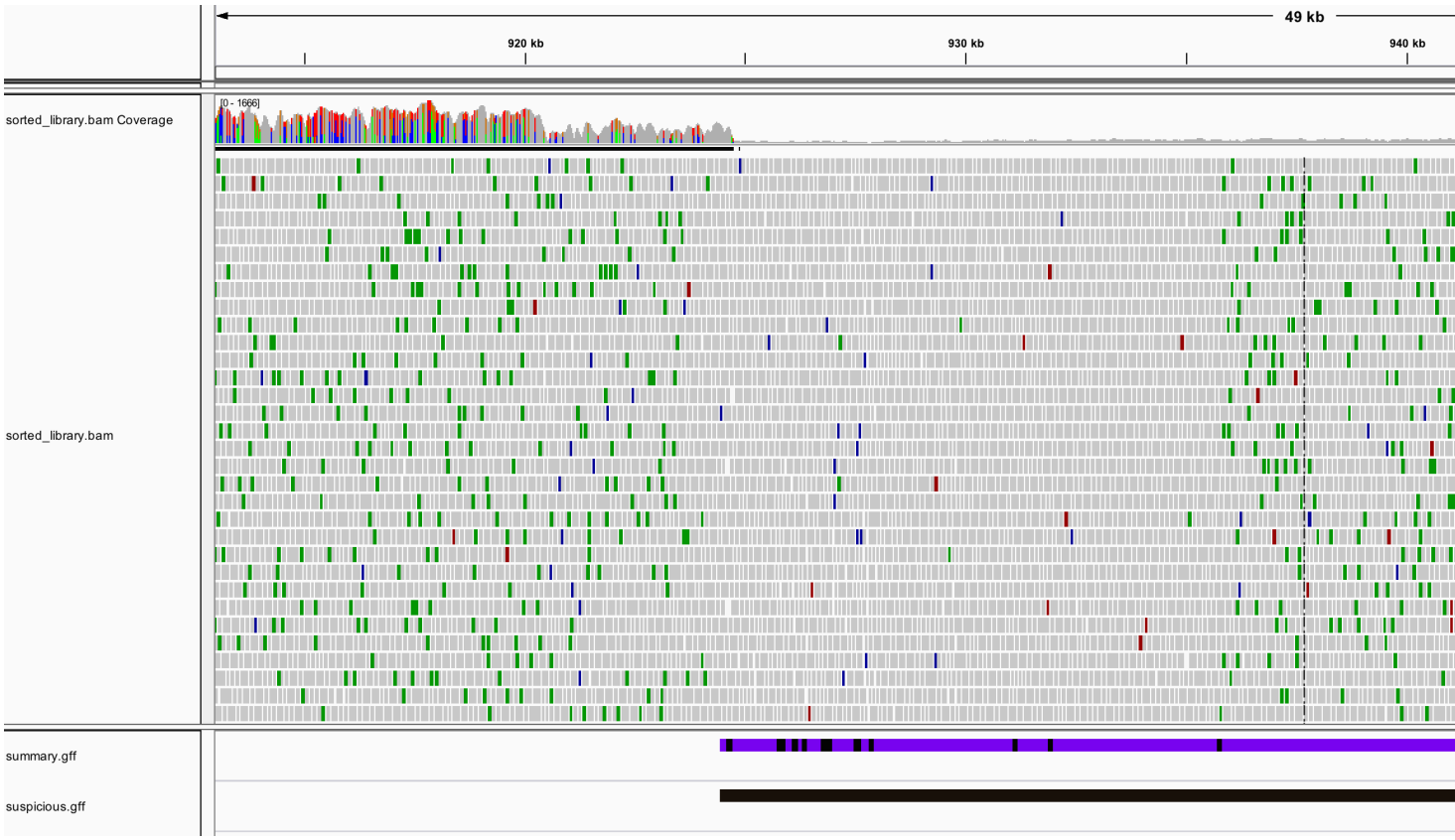

Fig. 4. A 25 Kbp low coverage region flagged by VALET, but no mis-assembly reported by QUAST. The low coverage region was due to MetaCompass selecting only a single strain of *Propionibacterium acnes* to use for assembly instead of both.

Patro, R., Mount, S. M., and Kingsford, C. (2014). Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature Biotechnology*, **32**(5), 462–464.

Peng, Y., Leung, H. C., Yiu, S.-M., and Chin, F. Y. (2012). IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, **28**(11), 1420–1428.

Phillippy, A. M., Schatz, M. C., and Pop, M. (2008). Genome assembly forensics: finding the elusive mis-assembly. *Genome Biology*, **9**(3), R55.

Salzberg, S. L., Phillippy, A. M., Zimin, A., Puiu, D., Magoc, T., Koren, S., Treangen, T. J., Schatz, M. C., Delcher, A. L., Roberts, M., *et al.* (2012). GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome research*, **22**(3), 557–567.

Shakya, M., Quince, C., Campbell, J. H., Yang, Z. K., Schadt, C. W., and Podar, M. (2013). Comparative metagenomic and rrna microbial diversity characterization using archaeal and bacterial synthetic communities. *Environmental microbiology*, **15**(6), 1882–1899.

Thorvaldsdóttir, H., Robinson, J. T., and Mesirov, J. P. (2012). Integrative genomics viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in bioinformatics*, page bbs017.