BIOINFORMATICS

Vol. 00 no. 00 2015
Pages 1–3

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], Atif Memon [1], 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

Summary: Existing methods for detecting mis-assemblies rely on the existence of reference genomes, which are not always available. Here, we present VALET, the first *de novo* pipeline for detecting mis-assemblies in metagenomic assemblies. VALET flags regions in the assembly that contain inconsistencies between the sequence generation process and the assembled region.

Availability: https://github.com/cmhill/VALET

Contact: cmhill@umiacs.umd.edu

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 [5] is a tool that can identify mis-assemblies and structural variants when provided with a reference genome. QUAST leverages existing methods (Plantagora [3], GeneMark.hmm [10], GlimmerHMM [12]) and quality metrics (GAGE [17]).

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. 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 [16] is a *de novo* pipeline for detecting mis-assemblies that incorporates the above constraints. As mentioned in the previous chapter, REAPR [6] 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.

Here, we detail how to modify the constraints of existing tools to allow them to work with 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

*to whom correspondence should be addressed

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 [7]. 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.

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.

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.

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

2.2 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 [13]. Regions whose coverage is greater than the third quartile plus 2.0 $\times$ 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 501 bp by default.

2.3 Insert size consistency

VALET relies on the REAPR [6] 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.

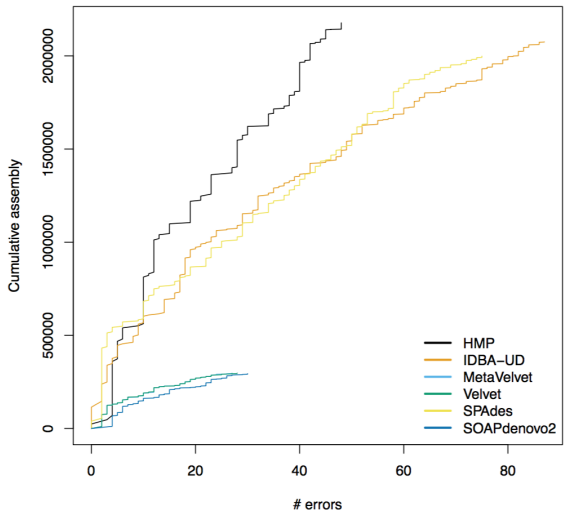

Fig. 1. FRC plot produced by VALET comparing assemblies of a Vaginal introitus sample (SRS014465) from the Human Microbiome Project [4] (using IDBA-UD [15], MetaVelvet [14], Velvet [19], SPAdes[2], and SOAPdenovo2 [11]. Point (x, y) corresponds to the number of errors incurred after processing the longest contigs until x bp is reached.

2.4 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 (50 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.

2.5 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.

2.6 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 by aligning the reads to the assembly with Bowtie 2 [8] and samtools [9]. 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 bp in length by default) that contains multiple mis-assembly signatures are marked as **suspicious**. The flagged and suspicious regions are stored in a BED file, which allows users to visualize the mis-assemblies using any genomic viewers, such as IGV [18].

2.7 Using VALET to compare assemblies of a sample from the Human Microbiome Project

Assemblies of a Vaginal introitus sample (SRS014465) from the Human Microbiome Project [4] (using IDBA-UD [15], MetaVelvet [14], Velvet [19], SPAdes[2], and SOAPdenovo2 [11]. Using VALET, we can compare the accumulated predicted error versus cumulative assembly length for each of the assemblies (Figure 1). From this plot we can observe that the HMP assembly is accumulating errors at the lowest rate compared to the other assemblers.

The suspicious regions flagged by VALET combined with the IGV support provide researchers with a powerful starting point in their mis-assembly investigation. For example, lets look at one of the suspicious regions flagged in the HMP assembly (Figure 1. VALET flags a 1.7 kbp high coverage region flanked by breakpoints. This region BLASTs [1] to *Lactobacillus amylovorus* GRL 1118 plasmid2 (CP002611.1). The right flanking region of the contig from positions 5,362 to 10,839 aligns to *Lactobacillus crispatus* ST1, strain ST1 (FN692037.1) with 98% similarity. The left flanking 3.75 kb region does not have a complete alignment with any sequence in NCBI; the closest alignment being from positions 1 to 2,599 to *Lactobacillus helveticus* strain KLDS1.8701 (CP009907.1). Researchers can now investigate further if this contig is mis-assemblies and composed of two closely-related species, or in fact, a novel species.

3 CONCLUSION

VALET is the first *de novo* pipeline for detecting mis-assemblies in metagenomic datasets. VALET allows researchers to find regions of their assemblies that are statistically inconsistent with characteristics of the sequence data generation process.

ACKNOWLEDGEMENT

Funding: Text Text Text Text Text Text Text Text.

REFERENCES

[1]Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, **25**(17), 3389–3402.

[2]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.

[3]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.

[4]Consortium, H. M. P. *et al.* (2012). Structure, function and diversity of the healthy human microbiome. *Nature*, **486**(7402), 207–214.

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

[6]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.

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

[8]Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature methods*, **9**(4), 357–359.

[9]Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., *et al.* (2009). The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**(16), 2078–2079.

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

[11]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.

[12]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.

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

[14]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.

[15]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.

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

[17]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.

[18]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.

[19]Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome research*, **18**(5), 821–829.