**VIRAMP: A web-based fast virus genome assembly pipeline**

**Abstract**

**Background**

Viral disease has been playing an important role in the history of human health. Millions of individuals are infected and died from virus like HIV, Influenza, HSV etc. As a result, the study of virus’s genetic composition becomes critical regarding to downstream pathology analysis, with further vaccine and therapeutics development. **[previous analysis method]** With the fast development and spread of high-throughput sequencing, the genome-wide variation analysis becomes possible.

Two common ways for variation analysis are: 1> Mapping against a reference genome and 2> de novo genome assembly.

Mapping against reference genome for variation analysis is frequently used in human and other complex genome organism. This strategy is suitable for the analysis of SNPs, small insertions and deletions (INDELs) and other mild mutations. However, comparing to human, viral genome is mutating at a much faster rate and the variation is also much higher. This means most of the time the viral genome we are sequencing could be too genetically distant from the reference we are comparing to. This could lead to low quality mapping, information loss, and most importantly, mapping to reference cannot used for structure variation discovery.[1]

One possible solution to the above issues is de novo assembly. Because of its reference-free strategy, de novo assembly is supposedly to solve the bias problem, as well as suitable for structural variation (SV) discovery and annotation.

There are two major strategies for de novo assembly. A more traditional one is known as *overlap-layout-consensus*. It starts by comparing the reads and overlap each other by computing the similarity (overlap step); then apply a multiple sequence alignment (MSA) to position the reads in the right order respecting to one another (layout step); in the end the MSA information is used to produce the final consensus sequence (consensus sequence). [2] This strategy has been proved to produce fairly good result through numerous early assembly projects using Sanger sequencing. However, one critical issue related with this strategy is that it is not suitable for high throughput sequencing, which result in large amount yet relatively shorter reads. With reads commonly shorter than 200bp, it is hard to deal with the repetitive regions to decide whether its real overlaps, this result in a much higher error rate. Besides, the with the huge amount of data comparing to traditional sanger sequencing, the consuming of time and computational resource in the overlap step is extremely large.

The de Bruijn graph is a completely different approach. The idea is to break the reads down into k nucleotides, or k-mers, and set the first (k-1) bases of one k-mer as prefix and last (k-1) bases as suffix, then chain the k-mers according to prefix and suffix to construct and graph and identify the most efficient path, with potential trimming and modification of the graph for error correction. In this way high redundancy problem is solved by handling k-mers rather than reads, which shorten the computational time as well as saving other computational resources. In this strategy, the k-mer size becomes the most crucial parameter. Smaller k-mer usually results in contigs of small size; while k-mers that are too large often tend to ignore repeats in the genome. In general, assemblies generated from de Bruijn graph based assemblers tend to be contigs that have smaller size comparing to the traditional overlap-layout-consensus algorithms.

Besides the technical issues regarding to different assembling algorithms, viral genome sequencing has its own features. The size of viral genomes can range from two kilobytes to two megabytes, though varies a lot, can still considered to be much smaller than the genome of other species. With today’s sequencing technology, one single run can generate the shot-gun sequence data with surprisingly high average high coverage. Take the Illumina HiSeq2500 machine as an example, one standard run can sequence about 14 billion 150bpx150pb paired-end reads, that is on average, of at least ~100X fold coverage for one sample. Taking the advantage of the fast increasing capacity of the sequencing machine, researchers tend to sequence a large number of strains of the same virus to conduct the population difference study and variation analysis. An extra advantage this high sequencing capacity brings is that now we can actually run a very deep sequencing of one single strain, so to make sure those spots in the viral genome that are hard to be sequenced are covered with adequate reads and those hard to assembled spots have enough reads guarantee a better assembly.

This results in another problem in virus assembly. Because of its high diversity in genome together with the technical difficulty in sample preparation, the resulting sequencing reads tend to be highly variable coverage against the genome, together with the potential mixture of large amount of contamination from hosts and preparation materials. Regarding to all the issues above, an efficient error correction step is necessary before applying the assembly process.

Here we present VIRAMP, a fast and efficient virus assembly pipeline dealing with high coverage yet variable coverage shot-gun sequencing datasets from virus genome. VIRAMP is a multilayered, semi- de novo assembly approach which utilize multiple de novo assemblers, integrated with a reference-guided scaffolding procedure and a k-mer based error correction model. This problem is originated from our assembly of HSV-1 (Herpes simplex virus 1), so here we validate our pipeline using \* samples of different HSV-1 strains, which represent a wide range of mutations, including snps, small INDELs, large INDELs, short sequence repeats (SSRs) etc, that might happen in virus genome. We demonstrate our approach can recover the virus genome compatible to the previous procedure, but running in a much faster and efficient way (within one hour). Also our pipeline is built upon the galaxy system, which is web-based, easy access with further assessment and visualization process build-in to help researchers without much computational background easily operate the whole process.

**Result and discussion**

Among all the human infected viruses, Herpes simplex virus 1 is one of the most successful human pathogens in terms of its global distribution, longevity host and mild symptoms. HSV-1, strain 17, which is a 152kb long genome an 108kb unique long (UL) and a 13kb unique short (US) region, with each flanked by inverted copies of large repeats (termed repeat long and repeat short regions, with length of 9.2kb and 6.6kb).

**Strategy of Viramp assembly**

**Assembly assessmemt and variation analysis**

Three different modules are implemented for the assessment of the assembly result. Quast[9]

Circos [11] has become increasingly popular in comparative genomeics visualization, since it provides a straightforward way for those structural and other variation of large datasets, which are normally hard to display, presenting in a meaningful and aesthetic way. However, since the data files and definitions are hard enough to handle by people without much computational experience, not mention the fact the software configuration itself is complicated. Here we provide an assembly visualization using circos, projecting the assembled draft genome to the aligned part of reference, creating a straightforward visualization for large structural variation (Figure). Users only need to submit a draft genome an a reference genome via web interface, requires definitely no computational skills.

**Comparing the performances of pipeline using three difference de novo assembler**

**Comparing Viramp result with other assemblers**

**Method**

**Sample preparation for Illumina sequencing**

**Viramp Assembly Pipeline**

The basic assumption for the datasets Viramp is dealing with are those shotgun sequencing viral genome data with extremely high but varied coverage (**Figure**). Previous studies often confront the problem that the coverage is too high that most single assemblers can not deal with. However, because of the fact a lot of virus, including HSV-1, contain short sequences repeat regions, which are hard to sequence, only high overall depth can guarantee enough coverage at those areas.

**Quality Trimming** Quality control of the paired-end reads was done by the seqtk toolkit[3]. By default we used the Phred algorithm provided in the toolkit to trim the low-quality bases from both ends. Reads are kept in pair and merged if pairs are kept in two separate files, so pairs are kept interlaced in one file.

**Diginorm** Further error correction and coverage reduce is done by “digital normalization”[4], a python package implementing an algorithm that reduce the sequencing depth and average the coverage variation. The final dataset coverage cutoff is set at 10x, with other configuration remains default. Because the original datasets can contain reads which one end is from the host, these reads will be eliminated due to their unstable coverage variation in the original dataset. This will result in a paired-end reads and a single-end reads fasta file.

**De novo assembly** Three de novo assemblers are integrated in the platform: Velvet (v 1.2.10)[5], VICUNA [6] and SPAdes (ver 2.5.1) [7]. All the three assemblers are based on the de Bruijn graph algorithm, which provides a relative fast performance. By default the pipeline is using velvet, and users can decide to change to other two assemblers simply by choosing from the dropdown menu before executing the pipeline. Both paired-end and unpaired-end reads resulting from diginorm step are used as input of the de novo assembler. Due to the fact that small k-mers often lead to short contigs and large k-mers on the other hand, often ignore repeats, we applied velvet five times each using k-mers 31, 41, 51, 61 and 71, each producing a contig file, we combine all the contig files into one as the input of next step, reference-guided scaffolding.

For SPAdes, --only-assembler’ option is used so we only integrate the de novo assembler rather than the whole SPAdes pipeline in the Viramp. Another option in SPAdes applied here is ‘--careful’, which tries to reduce the number of mismatch and short indels, this is recommended by the SPAdes manual. Since SPAdes uses a multisized de Bruijn graph algorithm, which requires input of multiple k-mers, the same five k-mer sizes as in Velvet are provided to the assembler. Contigs after scaffolding by SPAdes are used as input for next step.

All the options in VICUNA remain the default, ‘contig.lfv.fasta’ in the result files, which is the fasta format of consensus sequences retaining low frequent length polymorphisms are used as contig files for the input of next step.

**Reference-guided scaffolding** AMOScmp[8] is used to assemble the contigs resulting from the de novo assembler. AMOS is part of the AMOS consortium projects (v 3.1.0), and originally designed as a reference-guided assembler using an alignment-layout-consensus algorithm similar as the traditional overlap-layout-consensus assembly. There are three main steps in AMOScmp, and have been adjusted better work for contig scaffolding: Orienting contigs’ position by aligning them to the reference genome or genome of related species using MUMmer [9] aligner; layout refinement step *casm-layout* is carried out to deal with partial match, which is caused by large INDELs and structure rearrangement, random placement of repetitive reads into one of the copy location (-r option) is applied, to allow, although not recommended, the assembly of genomes with large repetitive regions (such as Hthe complete strain of HSV-1); in the end *make-consensus* step applies multiple alignment to generate consensus sequences; this step is particularly important as this will help incorporate repeats assembled with small k-mers into large contigs by large k-mers, which often eliminates existing repeats. This at some extend solves the k-mer selection problem in de Bruijn graph, paired-wise overlapping of the multiple-alignment result will provide the final one or several long consensus sequences with more accurate structural information. The minimum overlapping bases (-o option) is set to be 10. Besides the contigs from the de novo assemblers, paired-end reads file after diginorm is also used as part of the inputs to provide the mate-pair information, better helping orienting the contigs.

**Assembly assessment and variation discovery**

**Running QUAST** QUAST[10] is a quality assessment tool for genome assembly, which can evaluate draft genome assembly with multiple matrices. Here primarily use QUAST(v 2.2) to compare the assemblies against reference genome. Basic statistics such as N50, GC content, misassemblies are directly available on the webpage, and full reports are provided for download purpose.

**Running MUMmer** MUMmer package [10] is originally designed for pairwise rapid alignment of whole genomes, and now has been widely adapted in assembly evaluation and variation discovery. Here we used MUMmer (v 3.0) in multiple modules of our system, including reference-draft genome comparison, inexact and tandem repeats, and snp detection.

The core algorithm ‘nucmer’ is used for compare reference genome and draft genome with default options. Then further variation coordinates identification is applied to identify the difference so to present the SNPs and structural variation. To be noticed, for repeat identification, the draft genome is aligned against itself so to identify the alignments between different regions.

**Circos visualization** Circos is used for visualize reference-draft genome comparison, especially genomic rearrangements and large INDELs. Position based data is presented in circular way, which projects the specific regions in draft genome to the mapped reference genome region. The mapping step is done at background using MUMmer[10] and the onlyb data users need to submit from the web interface is reference genome and draft genome sequence.

**Integration with Galaxy and Amazon Cloud**

All the modules and components in the Viramp pipeline is integrated into a customized version of Galaxy[12]. Galaxy is an open sourced, web-based platform providing a web interface for most commonly used bioinformatics tools. This largely facilitates the researchers without any or much experience in programming to participate in genome analysis. Galaxy also provides a downloadable version of the Galaxy platform that allows people to alter the existing tools and plugin the tools, and run on their local server. Here in our viramp version of Galaxy, we delete most of the default tools that are not relevant to virus assembly and plugin our pipeline and visualization tools, so now biologists without much knowledge of bioinformatics can easily using the pipeline for their virus genome assembly and analysis, without any further disturbing from non-relevant tools.

Further, the whole system is hosting via the Amazon cloud, people can access simply via browser, or if one need more computational resource, one can buy extra resource from Amazon and run a separate snapshot. The viramp pipeline is available at viramp.com:8080.

**Conclusion**

Reference:

1. Yang et al. BMC Genomics: De novo assembly of highly diverse viral populations
2. Pop M et al. Briefings in Bioinformatics: Comparative genome assembly
3. Seqtk toolkit
4. Digital normalization
5. Velvet
6. VICUNA
7. SPAdes
8. AMOScmp
9. Quast
10. MUMmer
11. Circos
12. Galaxy