viaDBG : Inference of viral quasispecies with a paired de Bruijn graph

Borja Freire [1], Susana Ladra [1], Jose Paramá [1], and Leena Salmela [2]

[1]Universidade da Coruña
[2]University of Helsinki

February 2020

Contents

Introduction

Viral quasispecies problem motivation

- Viral quasispecies are population of closely related strains emerged from RNA viruses with high mutation rate.
- The higher mutation rate the larger number of closely related strains.
- Each mutation produces his own haplotypes.
- It is important to capture the whole set of strains because different strains might have different responses to the available drugs and treatments.

Introduction

Viral quasispecies problem

- The viral quasispecies assembly problem asks to characterize the quasispecies present in a sample from high-throughput sequencing data.
- There are two base hypotheses that relax the problem :
  - All the genomes are totally covered in the sample.
  - The coverage of the genomes is expected to be larger than in common assembly problems.
- There are two major challenges :
  - The presence of similar haplotypes in the data makes it difficult to separate the reads to different haplotype sequences.
  - Viral samples are typically sequenced to a much deeper coverage than e.g samples for genomic or metagenomic sequencing.

Methods

Reference based and de-novo methods

- Current methods available for assembling viral quasispecies are either reference-based or *de novo*.
- Reference-based methods :
  - Reference-guided methods are based on using one or several strains to guide the assembly problem.
  - Some examples : HaploClique, ViQuaS or PredictHaplo.
  - The main problem of these methods is that the reference used might be obsolete due the high mutation ratio.
- *de novo* methods :
  - They are reference free.
  - Some examples : SAVAGE, PeHaplo or MLEHaplo.

Methods

Overlap and de Bruijn graphs

- De Bruijn graphs :
  - Faster.
  - Less accurate.
  - SOAPdenovo2, SGA & metaSPAdes (for metagenomic but also useful on viral quasispecies).
- Overlap graphs :
  - Slower.
  - More accurate.
  - SAVAGE, PeHaplo & HaploClique.

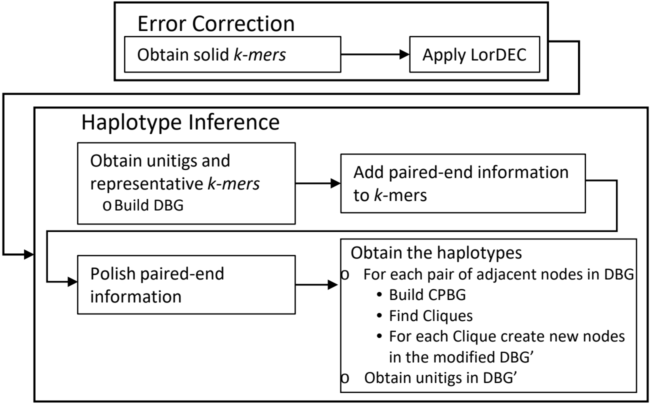
Methods

Overlap and de Bruijn graphs

- De Bruijn graphs :
  - Faster.
  - Less accurate.
  - SOAPdenovo2, SGA & metaSPAdes (for metagenomic but also useful on viral quasispecies).
- Overlap graphs :
  - Slower.
  - More accurate.
  - SAVAGE, PeHaplo & HaploClique.

viaDBG - Overview

Pipeline



viaDBG - Error Correction

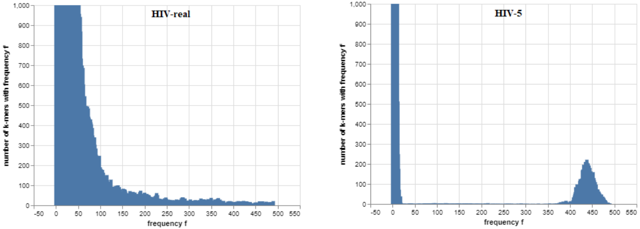
Obtain solid k-mers

- What is a solid k-mer ? Solid k-mers commonly refer to k-mers that are likely to be part of the real genomic information.
- There are several methods to obtain these k-mers such as :
  - Parametrical statistical methods - based on the mix of different distribution like Gaussian or Poisson.
  - Non-parametrical statistical methods - based on features provided by the sample like k-mer frequency, gradient information and so on.

viaDBG - Error Correction

viaDBG solid k-mers

- viaDBG uses the histogram of k-mer in the sample (Non-parametrical statistical method).
- The idea behind the selection is to find a point t where frequencies reach a stable state.



- The stability is measured using a window, but surprisingly we obtained from several tests that the windows size does not have a high impact over the final result.

viaDBG - Error Correction

Apply LoRDEC

- LoRDEC is a “well-known” hybrid reads corrector for third generation sequencing (TGS) reads.
- Steps (simplified version) :
  - Classify k-mers from the TGS as solid or not solid based on the k-mer frequency.
  - Building of a de Bruijn graph from short reads.
  - Between solid k-mers with non-solid gap between them look for a path in the de Bruijn graph.
  - Complete de reads by using this paths.
- Repeat iteratively by selecting a higher k-mer size for each iteration.

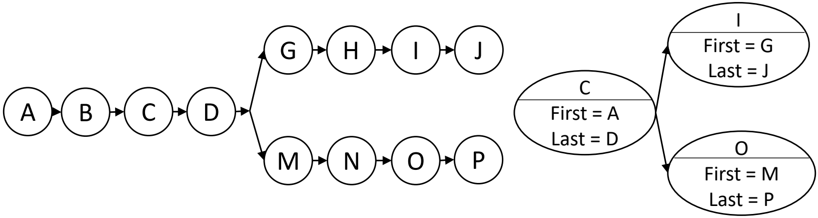
viaDBG - Haplotype inference

Obtain representative k-mers

- What is a representative k-mer ? In our case, it is the k-mer in the middle of a unitig.
- The use of representative k-mers covers two main problems :
  - Efficiency - by working only with representatives, we create a more succinct graph representation (this is exactly the same idea under the succinct de Bruijn graph)
  - Effectiveness - by using representatives, we are reducing the impact of the $\pm\Delta$ (variability of the paired end distance).

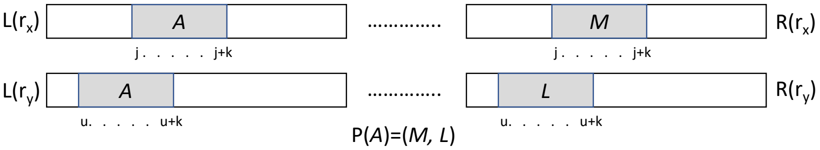
viaDBG - Haplotype inference

Obtain representative k-mers



viaDBG - Haplotype inference

Add paired-end information to k-mers



viaDBG - Haplotype inference

Polish paired-end information

- The polishing method removes outliers with large variance in the insert size.
- Challenge - remove outliers without removing low abundance strains.
- The idea behind the polishing can be summarise as :

$$\text{f'(A,M)} = \min \begin{cases} \text{f}(A,M) + |\{S \mid \text{f}(A,S) \geq 1 \text{ and } d(M,S) < \text{max-path-len}\}| \\ \text{max-threshold} \end{cases}$$

Where f(A,M) is the number of times A and M has been associated as left and right k-mers, and d(M,S) is the distance between M and S.

viaDBG - Obtain the haplotypes

Cliques Paired de Bruijn Graph

- For each pair of adjacent nodes of the DBG, viaDBG builds one *Cliques Paired de Bruijn Graph*, henceforth CPBG.
- What is a CPBG ? The nodes of the CPBG are the paired k-mers of the two considered nodes and edges connect paired k-mers if they are connected in the DBG by a short path. Furthermore, nodes have labelled the number of times the k-mer has been associated with the left k-mer.

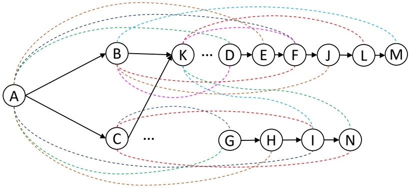
viaDBG - Obtain the haplotypes

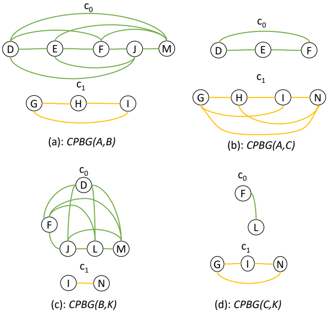
Cliques Paired de Bruijn Graph

- The next step is to find the maximal cliques in the CPBG. Conceptually, cliques on the graph are sets of k-mers that belong to the same haplotypic sequence.
- The obtained cliques must be polished because some of them come from erroneous k-mers, wrong relations (from shared regions between strains) and/or repetitive sections.

viaDBG - Obtain the haplotypes

Cliques Paired de Bruijn Graph (easy example)

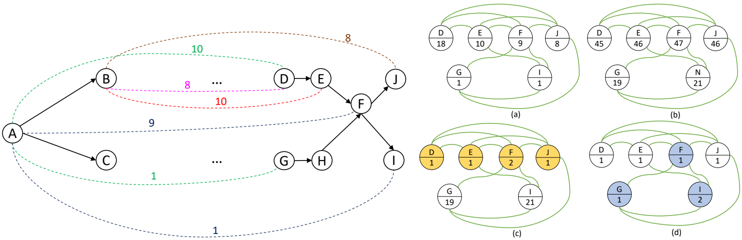




(a): *CPBG(A,B)*
(b): *CPBG(A,C)*
(c): *CPBG(B,K)*
(d): *CPBG(C,K)*

viaDBG - Obtain the haplotypes

Cliques Paired de Bruijn Graph (complete example)



viaDBG - Obtain the haplotypes

Building the new de Bruijn graph

- Given A and B, two nodes of the de Bruijn graph and C a set of maximal cliques from the CPBG of A and B.
- For each clique $c_x \in C$:
  - If c_x has nodes of $P(A)$ and $P(B)$, where $P(X)$ is the paired-end information for node X then the nodes $A_{P_A \cap c_x}$ and $B_{P_B \cap c_x}$ are added to the new de Bruijn graph, henceforth DBG'.
  - When we should not create new nodes ? If $A_{P_A \cap c_x}$ or $B_{P_B \cap c_x}$ already belongs to the DBG'.
- Finally, contigs are obtained as unitigs in this new graph.

Results

Datasets

	Virus Type	Genome Length (bp)	Average Coverage	Num. Strains	Abun- dance	Diver- gence
HIV-real	HIV-1	9487–9719	20000x	5	10–30%	1–6%
HIV-5	HIV-1	9487–9719	20000x	5	5–28%	1–6%
ZIKV-3	ZIKV	10251–10269	20000x	3	16–60%	3–10%
ZIKV-15	ZIKV	10251–10269	20000x	15	1–13%	1–12%
HCV-10	HCV-1a	9273–9311	20000x	10	5–19%	6–9%

Results

data set	method	% genome	N50	misass-emblies	% mis-matches	elap time (min)	memory (GB)
HIV-real	viaDBG*	87.25%	1813	0	0.197	4.48	3.74
	SAVAGE	91.79%	611	0	0.684	218.30	49.12
	PEHaplo**	91.43%	1262	0	0.074	7.56	3.48
	SPAdes	20.15%	660	1	2.091	12.74	5.52
	metaSPAdes	83.10%	1432	3	9.291	9.06	4.29
HIV-5	viaDBG	97.50%	8046	2	0.151	5.01	2.89
	SAVAGE	98.22%	6001	3	0.014	204.40	26.11
	PEHaplo	78.59%	9328	2	0.690	23.93	4.86
	SPAdes	90.91%	5097	2	0.051	3.31	4.12
	metaSPAdes	35.87%	6385	6	5.322	3.86	2.99
ZIKV-15	viaDBG	86.06%	1759	0	0.002	18.26	3.71
	SAVAGE	82.72%	1632	0	0.002	352.98	9.03
	PEHaplo	-	-	-	-	-	-
	SPAdes	38.97%	2063	0	0.147	6.17	4.42
	metaSPAdes	16.03%	3863	0	2.273	4.49	3.19

Results

Summariy

- Effectiveness - the three viral quasispecies methods are comparables in terms of accuracy.
  - In real data overlap methods retrieve a bit more % genome while de N50 is lower for both of them.
  - In simulated data, SAVAGE and viaDBG have the best performance, while PeHaplo is bellow. Actually, when the number of strains is 15 (ZIKV case), PeHaplo was not able to finish.
- Memory and time efficiency - de Bruijn methods are by far the best in all cases. Only PeHaplo seems to be a real rival because it reduces a lot the number of reads by removing duplicates. However, most cases is slower than viaDBG, SPAdes and/or metaSPAdes.

Conclusions

- Viral samples generally contain several haplotypes, each haplotype with its own frequency, namely each viral genome has its own level of abundance.
- General purpose and metagenomic assemblers are not able to retrieve the viral genomes in the sample as shown in the experimental evaluation.
- The results also show that :
  - viaDBG is able to retrieve competitive results in comparison with state-of-the-art methods such as SAVAGE and PEHaplo.
  - viaDBG is much faster than SAVAGE and also than PEHaplo in most cases.

Future work

- We will plan to reduce the memory footprint of viaDBG by taking full advantage of compacted de Bruijn graphs - opening this way the path into new problems such as metagenomics.
- Another line of improvement is to enhance the current parallelisation of viaDBG by taking into account some relevant issues such as disk accesses, thread synchronization, and data interchanges.

viaDBG : Inference of viral quasispecies with a paired de Bruijn graph

Borja Freire [1], Susana Ladra [1], Jose Paramá [1], and Leena Salmela [2]

[1]Universidade da Coruña
[2]University of Helsinki

February 2020