Published online 28 August 2017 *Nucleic Acids Research, 2017, Vol. 45, No. 19* ***10989–11003***
doi: 10.1093/nar/gkx755

Separation and assembly of deep sequencing data into discrete sub-population genomes

Konstantinos Karagiannis[1,2], Vahan Simonyan[2], Konstantin Chumakov[2] and Raja Mazumder[1,3,*]

[1]Department of Biochemistry and Molecular Medicine, George Washington University Medical Center, Washington, DC 20037, USA, [2]Center for Biologics Evaluation and Research, Food and Drug Administration, Silver Spring, MD 20993, USA and [3]McCormick Genomic and Proteomic Center, George Washington University, Washington, DC 20037, USA

Received September 07, 2016; Revised August 09, 2017; Editorial Decision August 14, 2017; Accepted August 16, 2017

ABSTRACT

Sequence heterogeneity is a common characteristic of RNA viruses that is often referred to as subpopulations or quasispecies. Traditional techniques used for assembly of short sequence reads produced by deep sequencing, such as de-novo assemblers, ignore the underlying diversity. Here, we introduce a novel algorithm that simultaneously assembles discrete sequences of multiple genomes present in populations. Using in silico data we were able to detect populations at as low as 0.1% frequency with complete global genome reconstruction and in a single sample detected 16 resolved sequences with no mismatches. We also applied the algorithm to high throughput sequencing data obtained for viruses present in sewage samples and successfully detected multiple sub-populations and recombination events in these diverse mixtures. High sensitivity of the algorithm also enables genomic analysis of heterogeneous pathogen genomes from patient samples and accurate detection of intra-host diversity, enabling not just basic research in personalized medicine but also accurate diagnostics and monitoring drug therapies, which are critical in clinical and regulatory decision-making process.

INTRODUCTION

Genetic diversity of populations resulting from high mutation rates plays a key role in viral evolution. In extreme cases, mutation rate can be as high as 4×10^{-4} errors (1) per nucleotide per round of replication (RNA viruses such as poliovirus, mumps etc.). Therefore, natural virus populations consist of an extremely high number of micro-variants, often referred to as sub-populations or quasispecies (2). It is well known that such genetic diversity resulting from both point mutations and recombination events is critical for maintaining fitness of the virus (3). Evolution of viruses is driven by selection from the pre-existing universe of these variants, in response to changing replication conditions and various pressures such as the immune system, drug treatment, switching to another host, etc. Therefore, identification of population heterogeneity is of critical importance for treatment design (4,5) and pathogen surveillance (6).

Traditional approaches, such as Sanger sequencing (7), are not capable of addressing the sub-population issue because they are geared towards sequencing DNA from homogenous and pure samples. High-throughput (deep) sequencing technologies (HTS) (8,9) that produce highly redundant (massively parallel) sequencing information are already used in clinical diagnostics (10) and can provide the necessary level of detail for sub-population genome delineation. However, specialized analysis is required to interpret properly the genetic diversity. In recent years, few algorithms have been designed to use HTS information, but accurate population reconstruction and frequency estimation is still somewhat intangible (11–15).

Available tools address the problem using statistical approaches, applying error correction filters to the reads, developing de novo assembles for diverse samples (16), building graphs from overlapping reads, or utilizing expectation maximization algorithms to reconstruct either local or global sequences (17–22). The accuracy of reconstruction is affected by the heterogeneity of intra-host viral population. Abundance of conserved genomic regions that extend significantly beyond the maximal read length restricts the full-genome assembly of highly heterogeneous populations (11). Nevertheless, none of the current algorithms can efficiently process current HTS data produced in deep-sequencing experiments. More specifically, ShoRAH (22) performs multiple sequencing alignment and clustering of limited number of reads (up to tens of thousands) and calls haplotypes based on the centers of the clusters. ViSpA (21) and QuRe (18) were designed for reads generated by pyrosequencing

[*]To whom correspondence should be addressed. Tel: +1 202 994 5004; Email: mazumder@gwu.edu

technologies with techniques to address insertions, deletions and errors in homopolymeric regions and therefore cannot adequately handle current HTS data. Finally, PredictHaplo (19) and HaploClique (17) have been tested with technologies other than pyrosequencing, including MiSeq paired end reads but still like ShoRAH have limitations on the number of reads it can analyze.

Here, we describe a novel deterministic algorithm, called Hexahedron, based on HTS data that can reconstruct local and global sequences and determine their relative frequency at a much larger scale than what was manageable before. We also offer a novel visualization technique that comprehensively represents the dynamic nature of the results with a simple interactive interface. The graph contains, per position depth of coverage, information for each reconstructed sequence and the first (bifurcation) and last (merging) position that differentiates a sequence and its closest sequence from which it has been derived. Annotations of the provided references, if included in the input set, will also be plotted in the same coordinate system as the reconstructed sequences. Finally, the interface allows sequences to be combined with each other following the paths that connect different bifurcation and merging positions as well as their consensuses and path compositions to be extracted. The source code of Hexahedron is publicly available at https://github.com/kkaragiannis/hexahedron and a web-based implementation at https://hive.biochemistry.gwu.edu/hexahedron/.

MATERIALS AND METHODS

Algorithm

The algorithm starts by constructing a frame (window) with a dynamic size (Additional File 1). The frame is placed in the 'leftmost' position of the reference (5′-end) and its size is set at twice the length of the first alignment. Then, all the alignments starting within this frame are scanned and a SNV profile is constructed. The profile extends beyond the original frame and occupies twice the size of the longest alignment considered so far. At the end of this step, the frame is scanned for mutations above the given threshold and a new sub-population genome profile is bifurcated based on the first mutation. The alignments are then reassigned to the appropriate branches based on the mutation that triggered this bifurcation event. This process continues recursively until all frames are clean of mutations exceeding the threshold. In the next step, the extended profile of the frame serves as an overlap region with the next step, which is used to determine to which frame each alignment belongs. During the reconstruction, the information about the alignments and contribution of each reference in any given position for any given frame is maintained. This leads to resolved profiles, where each one also represents a similarity plot based on the reference selected during the alignment step. The complexity of the algorithm is $O(nk)$, where n is the number of short reads and k the number of mutations that trigger bifurcations.

Let $\boldsymbol{R}$ denote the reference sequences we used and $\boldsymbol{M}$ the mutual alignment between these sequences. Let $R_{i,j}$ be the jth position of the ith reference so m_j be the jth position of the mutual alignment. Similarly, let $S_{i,j}$ be the jth short read aligned to $\boldsymbol{R_i}$ reference sequence and $S_{i,j}(p)$ be the reference coordinate that the pth position of the $S_{i,j}$ short read was aligned to. Let also $\boldsymbol{S_j}$ be the jth short read mapped to $\boldsymbol{M}$. Finally, let $\boldsymbol{C_i}$ be the ith contig, $C_{i,j}$ the jth position of $\boldsymbol{C_i}$ and $B(p)$ the basecall for any position.

Initially, we map all reads aligned against all references to the mutual alignment coordinates, so that:

$$S_j(p) = M\left(S_{i,j}\ (p)\right) = M\left(R_{ij}\ \right)$$

where $M(R_{i,j})$ is a fixed look up table of the specific $\boldsymbol{M}$ that returns the coordinate of the mutual alignment that corresponds to $R_{i,j}$.

Then all short reads are sorted based on their aligned position on increasing order so that $S_i(p) \geq\ S_j(p)\ \forall\ i\ \geq\ j$. The sorting method used is quicksort. Let $\boldsymbol{W_i}$ be the matrix where the ith contig will be constructed into and $W_{i,j}$ the jth position of $\boldsymbol{W_i}$. Hence, $B(W_{i,j})[b]$ $(\forall\ b\ \in\ \{0,\ 1,\ 2,\ 3,\ 4,\ 5\})$ is the total number of Adenines, Cytosines, Guanines and Thymines; insertions and deletions are mapped in $W_{i,j}$ $(0 \rightarrow\ \ A,\ 1 \rightarrow C,\ 2 \rightarrow G,\ 3 \rightarrow T,\ 4 \rightarrow insertion,$ $5 \rightarrow deletion)$. As a result, during the first step of the algorithm:

$$B\left(W_{0,l}\ \right)\ [b] =\ B\left(W_{0,l}\ \right)[b] +\ B\left(S_j\ \ (p)\right)\ \forall j\ :\ 0\ <\ S_j\ (0)\ <\ L$$

where $l\ = S_j\ (p)\ mod\ 2L$ and $L\ =\ Length(\boldsymbol{W}_0)$. However, the length of $\boldsymbol{W}_0$ is dynamically allocated during this step, and self-adjusted to the maximum length of the short reads that have been scanned so far.

$$Length\ \left(\boldsymbol{W}_0\ \right) =\ max\left(Length\left(S_i\right)\right)\ \forall i\ :\ 0\ <\ j\ <\ i$$

After all S_j that fall under the range of the first step have been scanned and the initial contig has been constructed, we check for mutations that exceed the frequency threshold (ft). So:

$$F_l\ [b] =\ \frac{B\left(W_{0,l}\right)[b]}{\max\left(B\left(W_{0,l}\right)[b]\right)}\ \forall j\ :\ 0\ <\ l\ <\ L\ and\ p\ :\in\ N_0\ and\ p\ :\ 0\ \leq\ p\ \leq\ 5$$

Therefore, we consider the bifurcation position I (or l in the context of the $\boldsymbol{W_i}$) where:

$$I\ =\ min\left(l\right)\ \forall l\ :\ \ F_l\left[b\right] \geq\ \ ft$$

After a bifurcation position has been detected, we construct a new matrix $\boldsymbol{W}_{i+1}$ and iterate the contig construction process using a subset of all reads mapped through the position p and containing the bifurcating mutation. Now j is in the range where $0\ <\ S_j(0)\ <\ L$. Another difference is that now we have to decide the matrix the S_j will contribute to. The bifurcation and construction steps are iterated until no bifurcation position is found.

With the exception of the first step of the algorithm, where there is only one contig, a decision is made before assignation of a short read to a contig. There are different points in the algorithm where contig voting is applied. The first is after bifurcation; the decision is between contig i where the bifurcation position was detected and contig n the newly created one. The decision is made based on the base of the bifurcated position. So,

$$B\left(W_{i,l}\right)\ [b] =\ B\left(W_{i,l}\right)[b] + B\left(S_j\left(p\right)\right)\ \forall j\ :\ 0 < S_j\left(0\right) < I\ and\ B\ \left(S_j\left(p\ =\ I\right)\right) =\ B(c_i,\ I)$$
$$B\left(W_{n,l}\right)[b]\ =\ B\left(W_{n,l}\right)[b] + B\left(S_j\left(p\right)\right)\ \forall j\ :\ 0 < S_j\left(0\right) < I\ and\ B\left(S_j\left(p\ =\ I\right)\right)\ =\ B(c_n,\ I)$$

The second scenario is after the end of the first step of algorithm, where there are an arbitrary number of frames.

In this case:

$$B(W_{c,l})[b] = B(W_{c,l})[b] + B\left(S_j(p)\right) \forall j : m_i < S_j(0) < m_i + L$$

where $m_i = size_{step} * step$ and $c = min(Dist(\boldsymbol{S_j}, \boldsymbol{C_k})) \forall k \in N_0$. Where $Dist(\boldsymbol{S_j}, \boldsymbol{C_k})$ is the Hamming distance between the $\boldsymbol{S_j}$ read and the $\boldsymbol{C_k}$ contig.

Before each step, all contigs are scanned to ensure that they are well supported by mutated positions against all other contigs. As the algorithm proceeds, all contigs unsupported for a region greater than the size of the step are merged together. Similarly, contigs that drop the coverage inside the current frame to zero - are excluded from participation in the construction process. So,

$$B\left(W_{i,l}\right)[b] = B\left(W_{i,l}\right)[b] + B\left(W_{j,l}\right)[b] \forall l : m_i < S_j(0) < m_i + L \, and \, \forall i, j \in N_0$$

where $Dist(C_{i,l}, C_{j,l}) = 0$ and $Dist(C_{i,l}, C_{j,l})$ is the Hamming distance between the $\boldsymbol{C_i}$ read and the $\boldsymbol{C_i}$ contig for position l.

The algorithm iterates until the last alignment is considered on the rightmost frame. The result is a collection of arrays, each representing a contig $\boldsymbol{C_i}$. Position j of the ith contig is an array $\boldsymbol{P} = \boldsymbol{C_{i,j}}$ where $\boldsymbol{P} = [P_0, P_1, \ldots, P_r \ldots, P_k]$ and P_r is the number of alignments that support the contig $\boldsymbol{C_i}$ and have been aligned to reference $\boldsymbol{R_r}$.

Validation

In order to test the algorithm, we generated in silico short reads using Sabin2 as a template sequence (Supplementary Table S1). Although Sabin1, Sabin2 and Sabin3 (GenBank (23) accession numbers AY184219, AY184220 and AY184221 respectively) are three strains that could be used to produce a heterogeneous sample of three sequences the goal was to generate samples of arbitrary number of population. As a result, a sequence was used to generate ten mutant strains, each with a genetic distance described in the second column of Supplementary Table S1. From these sequences, we randomly generated different number of short reads creating different relative frequencies for each haplotype. The length of the reads and the random noise introduced in each group of short reads is also described in the table. All short reads were merged into one file. After creating the in silico heterogeneous sample, we aligned and profiled them against the original Sabin2 sequence. The profile revealed a large number of mutations and the two distinct relative frequencies created two baselines of mutations at the level of 10% and the second at the 30% (Supplementary Figure S1a). After applying our algorithm with 1% mutation threshold we were able to reconstruct ten sequences with no mutations (Supplementary Figure S1B)

Contigs <2000 bp were ignored during all validation processes. For each experiment reconstructed sequences were extracted and aligned against all original sequences using hexagon (24). Correct hits are reported as true positives (TP). False positives (FP) are the excess hits to the original sequences (anything that overlaps with the longest contig). False negatives (FN) are considered the original sequences with no hits. Furthermore, for true positives, the number of mismatches is reported as an additional measurement of accuracy. True and false positives were used to calculate the precision and recall of each test.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FP}$$

Another measurement that was used is the harmonic mean of precision and recall, called F-score or F_1.

$$F_1 = 2 * \frac{1}{\frac{1}{Recall} + \frac{1}{Precision}} = \frac{2 * TP}{2 * TP + FP + FN}$$

The algorithm returns the depth of coverage per position and the frequency of each sequence was calculated based on the average depth of coverage of each haplotype. In order to measure the accuracy of the predicted relative frequencies, we use the Kullback–Leibler divergence between the predicted (P) frequency distribution and the true (T) distribution. Zero data points, which have resulted from false positives predictions, were disregarded.

$$D_{KL}(T \| P) = \sum_{i \in l} T(i) \, log \frac{T(i)}{P(i)}$$

We validated using a sample of trivalent poliovirus vaccine made from a mixture of three serotypes of attenuated Sabin strains of poliovirus (25). Sabin1, Sabin2 and Sabin3 sequences (GenBank accession numbers AY184219, AY184220 and AY184221 respectively) were used for Hexagon alignment with default parameters. This produced a total of 60 million alignments, which were then analyzed by the algorithm using 1% as mutation threshold. The three sequences were fully reconstructed and four more contigs of <150 bp were also reported (Supplementary Figure S2A). We predicted Sabin1 with relative frequency 93.58% and 17 mismatches, which correspond to 0.22% of the positions. Sabin2 was reconstructed with 0 mismatches despite the low frequency of 1.2%. Finally, the identified Sabin3 sequence had only 3hree mismatched positions, an equivalent of 0.04% of the total length and 5.14% frequency (Supplementary Figure S2B and Additional File 6). Very deep sequencing further supports the results because even for the lowest frequency Sabin2 virus the depth of coverage exceeds an 8000x. What appears as decrease in the depth of coverage in Figure 2C is due to the gaps introduced by the mutual alignment. The normalized representation of the graph highlights this effect, where relative frequency of a sequence increases locally to 100% in regions where these gaps occur.

Datasets

The data sets supporting the results of this article and an implementation of the algorithm are available in HIVE at: https://hive.biochemistry.gwu.edu/review/Hexahedron%20publication

Artificial datasets were generated using a native HTS simulator. Sequence Sabin2 was used as a template to generate 10 more sequences, each representing a different population in the heterogeneous sample. Mutations were introduced at every 50 positions with a starting position such that no mutation was overlapping with another on a different sequence. Each of these sequences was used as a template to generate NGS reads with the characteristics included in Supplementary Table S1. Artificially generated reads were merged into a single fastq file and aligned against the original Sabin2 sequence. Quality scores were all set to 30.

For the sensitivity analysis, six datasets were generated using Sabin2 as a template, consisting of 1 million short reads each. Initially two new sequences were created by introducing random mutations to Sabin2. Each sequence had 5% mutated position from the original sequence and ∼9% differences from each other (Supplementary Table S3). The two sequences were used to create these six samples by generating short reads of 250bp at different ratios with relative frequency ranging from 50% to 0.1% (Supplementary Table S2).

In order to produce different levels of genetic distance, a set of 100 sequences were generate by randomly mutating Sabin2. The greater genetic distant was set to 50% and the space between the two most distant sequences was divided into steps of 0.5% genetic distance. The relative abundance of each of the 100 points was calculated using the power distribution.

$$f(x) = \lambda e^{-\lambda x},\ 0 < x < 10,\ \lambda = 0.5$$

Six datasets were generated each using an increasing number of sequences as templates chosen in such a way that each consecutive pair of references is equidistant with the previous one (Supplementary Table S4). Distances between sequences for each sample are described in Supplementary Tables S5–S10.

Comparative analysis

The tools included in the comparison were QuRe_v0.99971, ShoRAH, ViSpA and PredictHaplo1.0. Additionally, ViQuaS1.3 and HaploClique were included in the comparisons but failed to run successfully in our environment and no results are reported. All tools were executed sequentially against all samples, with a specific time limit described in Table 3 and Supplementary Table S12. All results were validated for the accuracy of the sequence reconstruction and abundance estimation. Trees were created from the sequences each tool predicted for each sample. Reconstructed sequences from each computation were aligned together with the original sequences of the corresponding sample using MUSCLE from MEGA7 (26) with the default arguments. Trees were generated in MEGA7 using the neighbor-joining method (27) with default arguments. In order to associate the predicted sequences with the original ones, each tree was traversed using the *B** algorithm starting from one original sequence, looking for the first predicted sequence and removing the latter from the available leaves. The operation was repeated until all leaves of original sequences were matched to a leaf of predicted sequence or until no other leaves were available. False positives were considered either sequences that were not matched with original ones or that had mismatches to the associated original more than a given threshold. The trees were drawn using the ETE toolkit. In order to measure the accuracy of the predicted abundances, we used Jensen–Shannon divergence between the predicted (*P*) frequency distribution and the true (*T*) distribution.

$$D_{JSD}(T\,||\,P) = \frac{D_{KL}(T||M)}{2} + \frac{D_{KL}(P||M)}{2}$$

where $M = (T + P)/2$

The tools that successfully produced results for the samples SCL1–4 were validated by aligning the reconstructed sequences back to the original ones (Additional File 7). For those results where no predicted sequence aligned against any original, we performed a more thorough analysis using the genome comparator available in HIVE. 100 000 random reads were generated from the predicted sequences of each run and mixed with 100 000 random reads generated by the original sequences. The reads were mapped against both predicted and original sequences and chord graphs were generated. Reads aligned to two different sequences were used as evidence of the similarity between the regions of these sequences (Supplementary Figure S3). Only Hexahedron was able to predict sequences associated to original sequences. Further validation was performed using the same techniques applied on the specificity analysis. Recall, precision, % of mismatches and Kullback–Leibler divergence were identical to the SP4 sample of the specificity analysis.

Hexahedron workflow

The implementation of the algorithm is part of HIVE platform that can be publicly accessed through this URL (https://hive.biochemistry.gwu.edu/review/Hexahedron%20publication). After logging in, using guest's credential, the user may start by uploading the HTS reads through HIVE's web interface. Necessary input for the algorithm is an alignment of the sample against a number of references. Tutorial on performing alignment computations is also available through the same portal (see Additional File 8). Should multiple reference sequences have been selected for the alignment, an additional mutual alignment of the reference sequence is required before executing Hexahedron. A link to Hexahedron is available through the web page of the HTS sample alignment as an option of subsequent computations ('what's next' section). Hexahedron webpage will display the alignment that will be used as an input and will request for the ID of multiple alignment process. Additional arguments include the mutation threshold, above which bifurcations are triggered, and a flag that will mark the HTS sample as pair-end reads. The source code is also publicly available at https://github.com/kkaragiannis/hexahedron and a web-based implementation can be found at https://hive.biochemistry.gwu.edu/hexahedron/.

Architecture and computational environment

All data displayed on the website as well as any data or references used in the analysis are stored in the High-performance Integrated Virtual Environment (HIVE) server (http://hive.biochemistry.gwu.edu). The results are accessible online and available for downloading. Users can use the algorithm as a next step after the alignment or browse the publicly available pre-existing results. HIVE provided computational infrastructure for storage and analysis for this project. Comparison of Hexahedron with other tools and performance tests were done on CentOS installed on an Intel 2 Quad core 2.26 GHz with 24GB of RAM system.

RESULTS

Hexahedron is a deterministic algorithm that extensively explores all mutations above a given threshold in all available genomes in a sample. Each position is treated individually so that multiple nucleotide variations are processed as a series of single nucleotide variations (SNVs). An SNV is defined within the context of comparing a particular sequence read with the consensus sequence of multiple references; thus, sub-populations that have consistent groups of mutated nucleotides will be detected as correlated groups of SNV patterns in that specific sub-population or quasispecies genome. The input of the algorithm is the result of the alignment of the sequencing reads against a group of related references. Although the algorithm can accept a single reference, it is beneficial to use a multiple sequence alignment of related genomes to avoid the bias of reference selection. A comprehensive set of multiple references covering the entire range of sequence variations provides a better scaffold for reference assisted *de novo* assembly of the heterogeneous genomes in the sample. Allowing multiple references also provides the advantage of aligning more reads that would, otherwise, be too distant to match a single sequence and therefore be omitted from analysis (Table 1, Supplementary Figures S4–S12). Each read is aligned with the highest scoring reference sequence and then the result is re-mapped to a common frame generated by a multiple alignment of the reference sequences to each other. This provides the best scoring alignment to the frame of the common coordinate system. The algorithm proceeds in a step-wise manner following a 5′ to 3′ directionality of the references. In every step the variant calling profile along the reference frame is constructed and mutations trigger branching of the variant calling profile and re-allocation of the alignments. Besides requiring a collection of alignments, there are two additional user defined parameters: the frequency of occurrence threshold, above which a mutation is considered valid, and optionally the mutual alignment of reference sequences in case of alignment against multiple references (Figure 1).

Validation using *in silico* reads

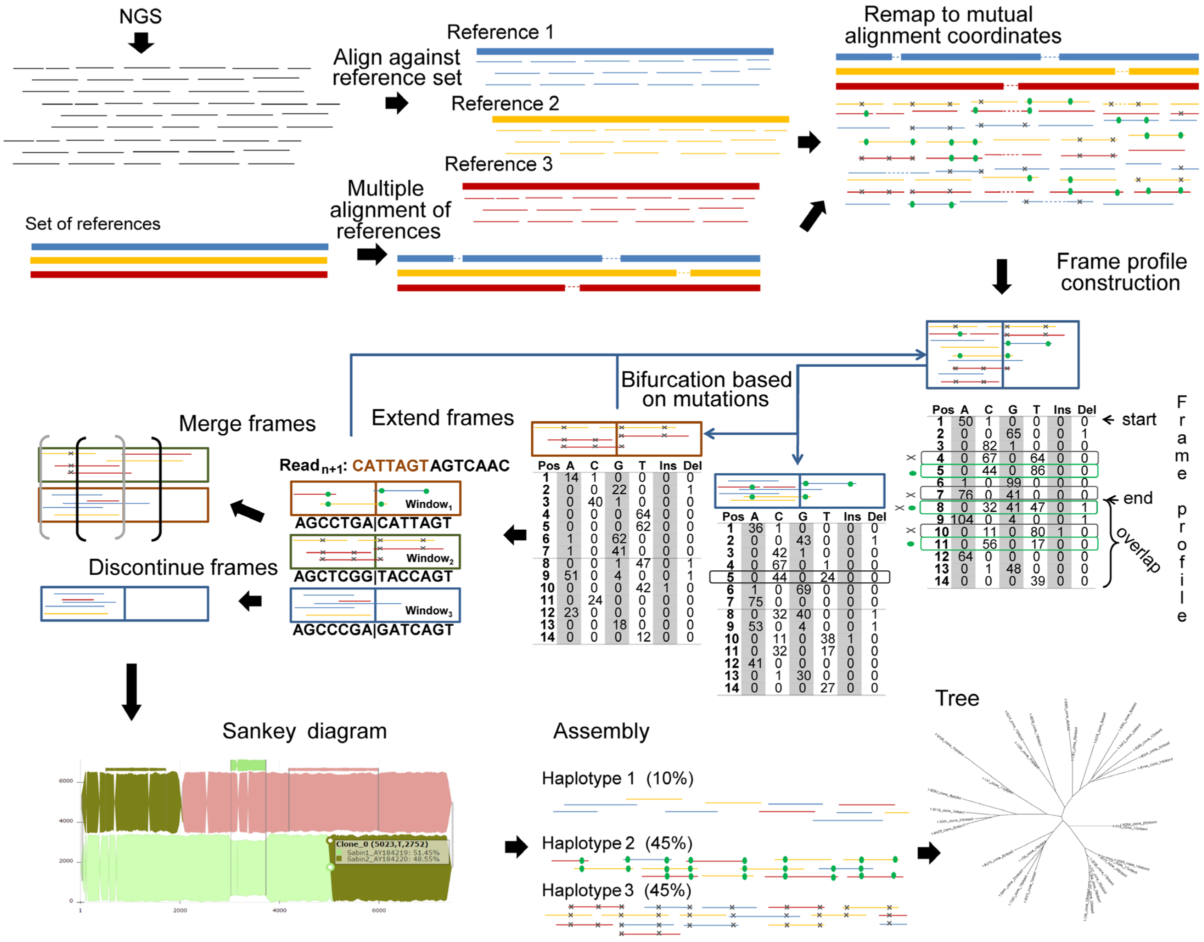
The algorithm's performance was validated using simulated *in-silico* data and determined the sensitivity and specificity boundaries. In addition, we have applied the algorithm to real samples and analyzed the results using the novel interactive visualization, similar to Sankey diagrams but developed specifically for this purpose. Each band represents a separate reconstructed sequence and the width of the flow is represents the depth of the coverage for each position along the x-axis. It comprehensively represents where a genomic contig is detected relative to the common coordinate system, the coverage of that contig and finally where and how the contig ends. Bifurcation and merging events are represented by grey line. Additionally, different colors inside each contig describe the similarity at a given position based on the references to which the reads, considered for the position, have been aligned. In the Sankey diagram, all trajectories following the bifurcation and merging events are possible assemblies in the variant spectrum.

First, the validity of the concept was tested by generating a mixture of reads derived from ten sequences. Each sequence was created from the same 7800 bp template sequence by introducing mutations randomly at a 10% rate. We generated 10 000 short reads of 150 bp length for eight of the sequences individually and 60 000 for each of the remaining two. Compared to a variant detection analysis, which identifies two populations one at frequency 5% and one at 30%, the assembler identified all of them (Supplementary Figure S1). In all experiments, reads were aligned using the Hexagon aligner (24). The quality of the results was measured based on precision, recall and Kullback–Leibler (28) divergence.

Sensitivity. To determine the sensitivity of the algorithm, two sequences 20% different from each other were produced, split into short reads of 250 bp and combined together in six different proportions (Supplementary Tables S2 and S3). The six samples, named SN1–6, were aligned against the reference sequences used to generate the two original sequences, and processed by the algorithm. One percent mutation threshold was selected for the first 3 samples, 0.3% for the next 2 and 0.07% for the last. Threshold was set so that it lies above the expected 0.3% noise level, unless we specifically wanted to detect populations below the noise level. The assembled sequences were then aligned back to the original references and those that were aligned only against their original sequence were reported as true positives (TP).

Filtering short contigs also decreased false positives generated by phased noise. All datasets resulted in fully reconstructed sequences with 100% recall (Table 2 and Supplementary Figure S13). After filtering all contigs shorter than 2000 bp, precision was also 100% for all samples. It remained 100% even when contigs greater than 1000 bp were allowed with the exception of the 0.1% dataset, where 24 false positives (FP) were reported having a great impact in the precision. It is understandable that the noise, which in this case is three times higher than the bifurcation threshold, was phased allowing contigs to extend up to 1500 bp. No mismatches were detected between the true positives and the original sequences confirming sequence reconstruction accuracy. The distribution of the predicted sequences was identical to the actual distribution with divergence at the level of 10^{-5} (Table 2).

Specificity. To determine the specificity, the closest genetic distance the algorithm can separate sequences, six more datasets, named SP1–6, were generated (Supplementary Table S4) with an increasing number of sequences (2, 4, 8, 16, 32 and 64) derived from Sabin 2 poliovirus (29) (GenBank accession: AY184220.1). Each sample exhibited the same diversity, such that the two marginal sequences of each sample had almost the same distance across datasets, resulting in sequences that were genetically very close to each other (Supplementary Tables S5–S10). Furthermore, the relative frequency of each sequence was calculated based on the ex-



Pos	A	C	G	T	Ins	Del
1	50	1	0	0	0	0
2	0	0	65	0	0	1
3	0	82	1	0	0	0
4	0	67	0	64	0	0
5	0	44	0	86	0	0
6	1	0	99	0	0	0
7	76	0	41	0	0	0
8	0	32	41	47	0	1
9	104	0	4	0	0	1
10	0	11	0	80	1	0
11	0	56	0	17	0	0
12	64	0	0	0	0	0
13	0	1	48	0	0	0
14	0	0	0	39	0	0

Pos	A	C	G	T	Ins	Del
1	14	1	0	0	0	0
2	0	0	22	0	0	1
3	0	40	1	0	0	0
4	0	0	0	64	0	0
5	0	0	0	62	0	0
6	1	0	62	0	0	0
7	1	0	41	0	0	0
8	0	0	1	47	0	1
9	51	0	4	0	0	1
10	0	0	0	42	1	0
11	0	24	0	0	0	0
12	23	0	0	0	0	0
13	0	0	18	0	0	0
14	0	0	0	12	0	0

Pos	A	C	G	T	Ins	Del
1	36	1	0	0	0	0
2	0	0	43	0	0	1
3	0	42	1	0	0	0
4	0	67	0	1	0	0
5	0	44	0	24	0	0
6	1	0	69	0	0	0
7	75	0	0	0	0	0
8	0	32	40	0	0	1
9	53	0	4	0	0	1
10	0	11	0	38	1	0
11	0	32	0	17	0	0
12	41	0	0	0	0	0
13	0	1	30	0	0	0
14	0	0	0	27	0	0

Figure 1. Hexahedron algorithm overview. The algorithm creates bifurcations in every point mutation that exceeds a user specified threshold and maintains the contig as long as possible. It takes advantage of the existing alignment data and correlates distant mutations, reported by an alignment algorithm possibly because of the selection of a distant reference genome. Hexahedron makes no statistical assumption; it rather extracts the information from the overlapping alignments in a step wise, contig assembly, fashion, using coordinates of the aligned reads mapped to the mutual alignment of the references.

Table 1. Impact of increased number of references on sequence assembly

# of References	Mismatches allowed (%)	Unaligned reads (%)	Precision (%)	Recall (%)
1	5	79.5	100	37.5
1	15	25	100	87.5
3	5	24.8	100	87.5
3	15	0.1	100	100
10	5	0	100	100
500	15	0	100	100

ponential distribution resulting in sub-populations of frequency as low as 0.05% (see Methods).

Analysis of the samples with 2, 4, 8 and 16 sequences (with frequencies as low as 0.25%) led to global reconstruction of all sequences (Table 2). The recall and the precision for these samples was 100% with no FP contigs detected longer than 300bp (Supplementary Figures S14A–D and S15A–D). All globally reconstructed sequences were aligned against the original sequence and found to have <0.5% mismatches (Additional File 5). This occurred in low-coverage regions such as the 5′ and 3′ ends. After trimming these regions out of the predicted sequences, the number of mismatched position decreased to <0.04% for these samples (data not shown). The distribution of frequencies of the predicted sequences for these samples was identical to the true distribution with KL divergence 10^{-5}. Recall and precision decreased for samples with 32 and 64 sequences (Table 2). Twenty-seven fully resolved sequences were generated for the mixture of 27 sequences and 39 for the mixture of 64 (Supplementary Figures S14E–F and S15E–F). Prediction of relative frequencies was also affected in these dense and highly heterogeneous samples. This was observed mainly for the low frequency haplotypes, while those with higher frequency resulted in a more accurate prediction (Additional File 5). This may be because a large number of the generated sequences were present below the noise level.

Table 2. Sequence reconstruction accuracy and Kullback-Leibler divergence between known and predicted frequency distribution

Sample	Recall (%)	Precision (%)	Av. Mismatches (%)	KL-Divergence
Sensitivity				
SN1	100	100	0.0067	$6.8\ 10^{-6}$
SN2	100	100	0.0067	0.00043
SN3	100	100	0.0067	0.00024
SN4	100	100	0.0067	$5.6\ 10^{-5}$
SN5	100	100	0.0067	$3.2\ 10^{-5}$
SN6	100	100	0.0067	$5.7\ 10^{-5}$
Specificity				
SP1	100	100	0.0605	$4.5\ 10^{-06}$
SP2	100	100	0	$3.4\ 10^{-06}$
SP3	100	100	0.1519	$4.8\ 10^{-05}$
SP4	100	100	0.0439	$2.5\ 10^{-5}$
SP5	84.375	84.375	0.6934	0.00786
SP6	60.9375	50.64935	1.4272	0.03951

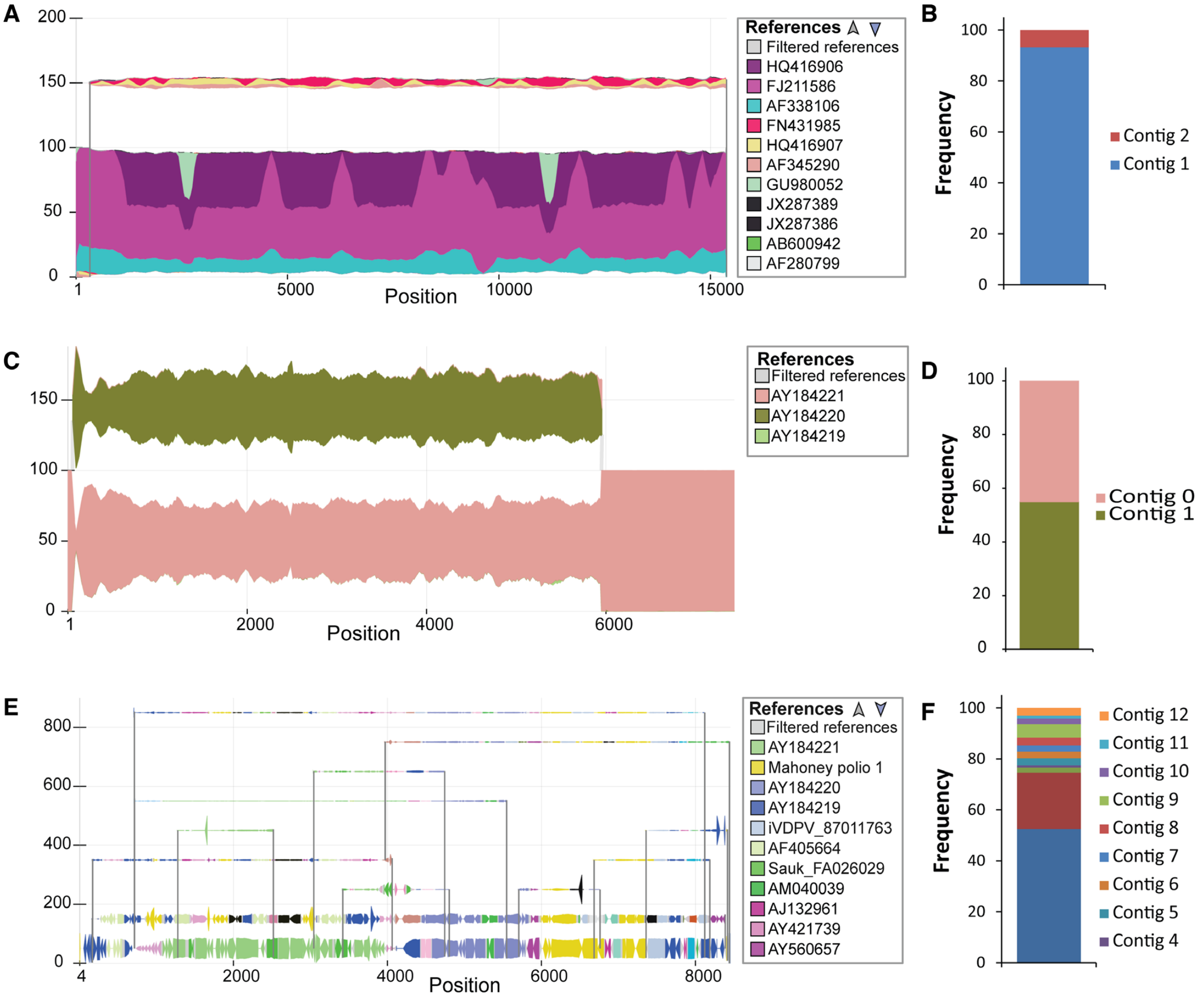


Figure 2. Populations assembly results of real sample. (**A**) A mixture of mumps was resolved into exactly two sequence with a frequencies of 94% and 6% (**B**). (**C**) Sample isolated from concentrated sewage water in RD cell culture. Only two sequences longer than 1000 bp were reconstructed. The two sequences were identical in the 3′ end indicated by the shorter reconstructed merged to the dominant one at position 4325. (**D**) Relative frequency of the two sequences at 88.67% and 11.23%. (**E**) Another sample isolated from sewage concentrate using RD cell culture was analyzed using our software. A more complex picture reveals three fully reconstructed sequence and a number of smaller contigs that depict the dynamic nature of the virus. (**F**) Frequency distribution of identified populations in the environmental sample.

Speed performance. To measure the speed performance of the algorithm (Supplementary Figure S16), we constructed different sets of samples changing all pairs of the following characteristics: the noise, the number of sequences and the distance between sequences. Noise was set to 0.3%, number of sequences to 4, distance between them to 2%, number of short reads to 100 000 and the read length to 200 as defaults and the mutation threshold to 1%. The algorithm was shown to perform well even on deep sequencing data and most of the computations finished within seconds (Supplementary Figure S16). The slowest computations, finishing within 1 h, were the ones with the greatest number of aligned reads and when sensitivity was set below the noise baseline.

Validation using experimental data for viral samples with subpopulations

In addition to the above simulation study, the algorithm was applied to real HTS datasets obtained for a live attenuated vaccine strain and for an environmental (sewage) sample. All samples were aligned using hexagon; when necessary, multiple alignment of selected references was performed using MAFFT (30).

Mumps virus. Mumps is an RNA virus with a 15 000 bp genome that encodes nine proteins. Jeryl Lynn strain used in live Mumps virus vaccine was shown by conventional sequencing of plaque-purified clones to contain two distinct virus sequences (31). After aligning the paired end reads to a comprehensive database of genomic sequences of 54 strains of mumps, using hexagon with default parameters, 688 000 hits were recorded and used for analysis. Two globally reconstructed sequences were detected using 1% mutation threshold (Figure 2A). The predicted frequencies of the reconstructed sequences were 93.18% and 6.82% (Figure 2B), consistent with previous estimate based on quantitative PCR (31). Consensus sequences of these two substrains were identical to those determined by conventional sequencing of plaque-purified clones.

Environmental isolate of poliovirus (example 1). The same analysis was applied to a sewage sample that was previously found by conventional virological analysis to contain vaccine poliovirus along with another non-polio enterovirus. The 55 million paired end reads were aligned against Sabin1, Sabin2, and Sabin3 sequences (GenBank accession numbers AY184219, AY184220 and AY184221 respectively) resulting in 48.26 million alignments. The reads produced hits to all of the references and the alignments were used as input for our algorithm with a 1% mutation threshold. Two sequences were reconstructed, one globally and another of 5948 nucleotides long (Figure 2C and Additional File 6). The globally reconstructed sequence was also the dominant one and had a predicted frequency of 54.75%; the shorter sequence had a predicted frequency of 45.25% (Figure 2D and Additional File 6). The shorter sequence bifurcates from the dominant sequence at position 53 and merges back at position 5981 of the common coordinate system. Hence, the sample contains a mixture of two recombinant viruses that differ only in this range by 2513 mutations (Additional File 6). Comparison of the reconstructed consensus sequences showed that one component of the mixture was a Sabin 2–Sabin 3 recombinant, while another was close to Echovirus 11.

Environmental isolate of poliovirus (example 2). The same analysis was applied to another virus isolated from sewage. The 2 million paired end reads were aligned against a comprehensive set of 500 enteroviruses resulting in 3 million alignments. The reads aligned produced hits to 315 of the references and used as an input for our algorithm with a 1% mutation threshold. The large number of references did not affect the efficiency of the algorithm; it did, however, affect the mutual alignment frame, where we observed an increased number of gaps (Figure 2E). The algorithm identified two major fully reconstructed variants, and several minor variants that represented recombinants of the first two with different inserts with length between 1000 and 5000 bp. The two fully reconstructed sequences have predicted frequencies of 64% and 26%, and are identified as recombinant vaccine-derived polioviruses of serotypes 1 and 3. (Figure 2D and Additional File 6). In the set of minor variants, frequencies below the 1% mutation threshold were detected (0.98%), demonstrating the flexibility of the algorithm.

Comparative studies

Hexahedron was also compared against the current state-of-the-art sequence reconstruction algorithms compatible with the available computational environment (see Materials and Methods): QuRe (18), ViSpA (21) ShoRAH (32) and PredictHaplo (33,34). Initially, 20 samples, name SCS1–20, were generated varying in sequence length of the populations with 500, 1000, 2000 and 5000 base pairs and the number of the *in silico* generated short reads with 1000, 5000, 10 000, 50 000 and 100 000 short reads. All samples were simulated to consist of four populations each one at 25% prevalence. The sequences were derived from Sabin 2 poliovirus (see Materials and Methods) with an increasing distance from the original template of 1% positions on average (Table 3).

Reconstruction performance. The reconstructed haplotypes were compared to the original sequences and were called True Positive (*TP*) or False Positive (*FP*) depending on whether the percentage of mismatches exceeds one of the following thresholds: 0%, 0.01%, 0.05%, 0.1%, 0.5%, 1% and 5%. Hexahedron produced the least amount of *FP* and False Negative (*FN*) with few exceptions, only against PredictHaplo (Supplementary Figures S17 and S18). All tools failed to reconstruct any haplotype with less than 0.5% mismatches for samples with average depth of coverage less than 5x per population (Table 3). ShoRAH, QuRe and ViSpA failed to reconstruct any haplotype with less than 1% mismatches for any of the samples with the only exception of PredictHaplo that successfully reconstructed two sequences for the sample of the 1000 short reads and 500 bp long sequences. In contrast, for samples with more than 10 000 reads and considering a 0.1% mutation threshold Hexahedron produced consistently the least amount of FP and FN and as a result it performed better against all algorithms in terms of precision (Supplementary Figure S19), recall (Supplementary Figure S20) and F-score (Figure 3).

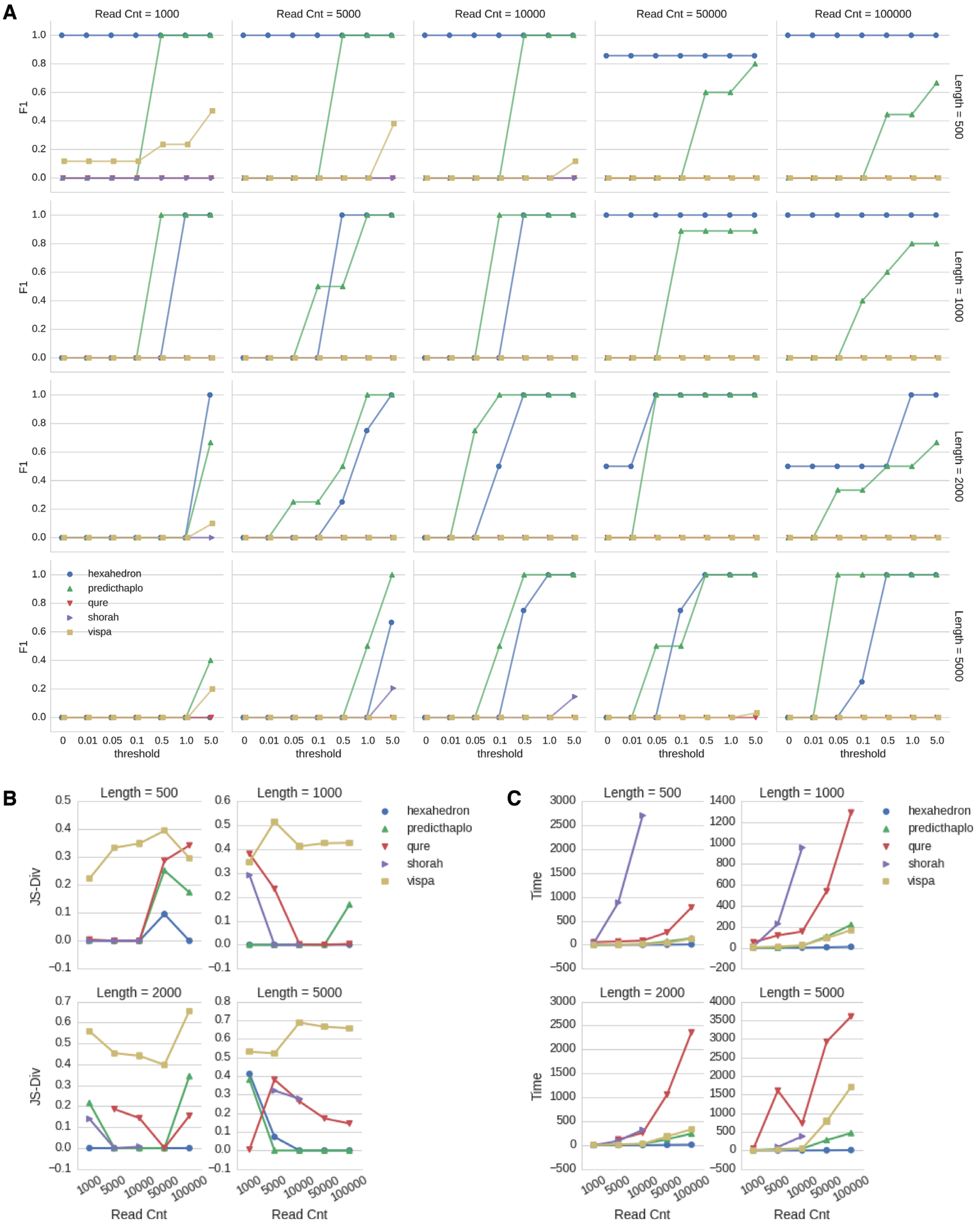


Figure 3. Comparison studies results. Hexahedron, PredictHaplo, QuRe, ShoRah and ViSpA were compared using 20 *in silico* samples of four populations, each one of 25% abundance. (**A**) *F*-scores. Each column of charts represents a different number of short reads in the samples and each row a different sequence length of the populations. For each computation True Positive (*TP*) are called based on the number of mismatches between the reconstructed and the original sequences. Vertical axes with the label F1 represent the *F*-score and horizontal axes represent the mismatch threshold below which the sequences are declared *TP*. *F*-scores are reported based on mismatch thresholds 0%, 0.01%, 0.05%, 0.1%, 0.5%, 1% and 5%.Missing data points are due to the failure to obtain results within the period specified for each sample. (**B**) Jensen–Shannon divergence between the predicted and the original frequencies. (**C**) Speed performance of the tools reported in minutes. The results indicate an time increases exponential as the number of short reads increases for PredictHaplo, QuRe, ShoRah and ViSpA.

Table 3. Summary of *in silico* samples generated for comparison studies

Name	Length (bp)	# of short reads	Av. Depth/haplotype	WC limit (h)	# of haplotypes
SCS1	500	1000	125	1	4
SCS2	500	5000	625	2	4
SCS3	500	10000	1250	3	4
SCS4	500	50000	6250	6	4
SCS5	500	100000	12500	12	4
SCS6	1000	1000	62.5	1	4
SCS7	1000	5000	312.5	2	4
SCS8	1000	10000	625	3	4
SCS9	1000	50000	3125	6	4
SCS10	1000	100000	6250	12	4
SCS11	2000	1000	31.25	1	4
SCS12	2000	5000	156.25	2	4
SCS13	2000	10000	312.5	3	4
SCS14	2000	50000	1562.5	6	4
SCS15	2000	100000	3125	12	4
SCS16	5000	1000	12.5	1	4
SCS17	5000	5000	62.5	2	4
SCS18	5000	10000	125	3	4
SCS19	5000	50000	625	6	4
SCS20	5000	100000	1250	12	4
SCL1	~7200	100000	2417–926 937	12	16
SCL2	~7200	500000	483.5–185 387.5	24	16
SCL3	~7200	1000000	241.7–92 693.7	48	16
SCL4	~7200	5000000	48.3–18 538.8	96	16

Samples with the SCS prefix have four haplotypes with uniform abundance distribution and samples with SCL prefix have 16 haplotypes with power distribution. Column 'WC limit' describes the Wall Clock time, in hours, given to tools to run each sample.

Using the F-score to compare the algorithms, PredictHaplo performed better in samples SCS7, SCS8 and SCS19 only when considering 0.05% as mismatch threshold, in samples SCS13 and SCS20 with 0.05% and 0.1% mismatches and in SCS12 for thresholds between 0.01% and 1. Furthermore, PredictHaplo did not reconstruct perfectly any sequence, while Hexahedron produced all sequences without mismatches in sample SCS1, SCS2, SCS3, SCS5, SCS9 and SCS10 and perfectly reconstructed at least one sequence in samples SCS4, SCS14 and SCS15.

Phylogeny of reconstructed sequences. In addition to diagnostic testing, neighbor-joining relatedness trees were constructed for each sample including the original and all sequences predicted by all tools (Figure 4 and Supplementary Figure S21). The tree of sample SCS5 is represented in Figure 4A and is in concordance with the *F*-scores reported in Figure 3A. Hexahedron predicted exactly four sequences while PredictHaplo reconstructed one more than expected with 1% abundance, possibly due to noise and high depth of coverage (12500 per sequence). Despite similar F-scores in sample SCS11, Figure 4B shows that, PredictHaplo predicted only two sequences, each one with almost 50% abundance, while Hexahedron predicted exactly four with ~25% abundance each. Consequently, this difference was also highlighted when we measured the accuracy of the detected abundances of the reconstructed sequences. For this purpose, we measured the Jensen-Shannon divergence between the predicted and the original distribution. Each original sequence was paired with the closest reconstructed one without considering any mismatch threshold. PredictHaplo predicted more accurately the frequency distribution than Hexahedron only in samples SCS16 and SCS17 while the latter outperformed all the tools in samples with 100000 reads with the only exception of sample SCS20, where PredictHaplo achieved the same accuracy. In sample SCS16, Hexahedron reconstructed seven sequences and PredictHaplo reconstructed one instead of four, while in SCS17 Hexahedron reconstructed five sequences and PredictHaplo four resulting in more accurate frequency distribution prediction.

Speed performance. In terms of speed, Hexahedron was found to be the fastest for all samples (Figure 3c). In fact, all other tools displayed a polynomial time complexity with respect to the number of short reads with the only exception of samples SCS17 and SCS18, where QuRe spent more time to process the sample with smaller number of reads (SCS17). Notably, QuRe predicted one sequence for SCS17 and two for SCS18. Perhaps the depth of coverage in SCS17 forced QuRe to consider more reconstructed sequences per sliding window, but this was not sufficient to expand them into global sequences providing the worst-case scenario for the algorithm. Conversely, SCS18 with sufficient depth of coverage allowed QuRe to reconstruct two sequences, removing a number of variants from the single pool of the four populations and reducing the computational load.

Large datasets. In order to further compare the performance of the tools on datasets more representative of current HTS technologies, we generated four more datasets, named SCL1–4, with 16 populations and abundance following a power distribution, similar to sample SP4 (Table 3 and Supplementary Table S12). The number of reads of the samples is ranging from 100,000 to 5,000,000. Each tool was given up to 4 days of wall clock time to conclude the computations (Table 3). Within this timeframe, only Hexahedron processed successfully all samples and PredictHaplo

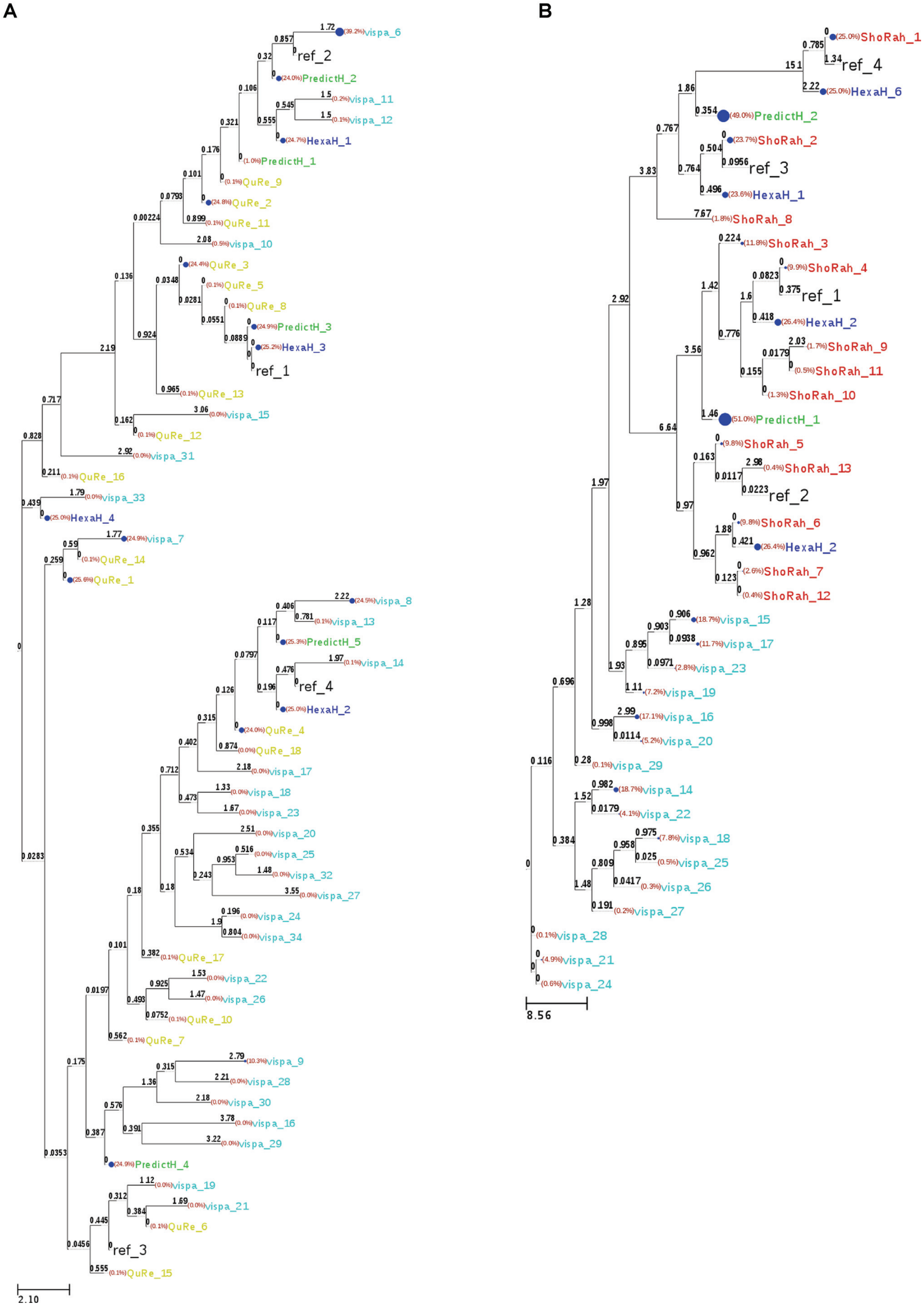


Figure 4. Neighbor-joining trees of original and reconstructed sequences from all tools. ShoRah leaves are colored red, PredictHaplo green, Hexahedron blue, QuRe yellow, ViSpa light blue and original sequences are colored black. Blue circles next to leaves represent the abundance predicted for the corresponding sequence. (**A**) Tree of sequences generated from sample SCS5. Two of the PredictHaplo reconstructed sequences are assigned to nodes closer to their original sequences compared to Hexahedron but PredictHaplo has reconstructed 5 sequences instead of five. ShoRah did not conclude the computation within the given time frame. (**B**) Sample SCS11. QuRe did not produce results.

successfully finished only samples SCL1, SCL2 and SCL3. None of the sequences reconstructed by PredictHaplo was identified as a TP even considering mismatch threshold as high as 20% (Supplementary Table S13, Figure S3 and Additional File 3). Hence, the results were not analyzed further. Hexahedron was able to reconstruct successfully all haplotypes (Additional File 7). We also used the sample described in the 'Mumps virus' section as a semi-empirical dataset to compare the tools. Hexahedron, was the only tool to successfully resolve the sequences. PredictHaplo partially reconstructed 22 sequences, each 5400 bp, instead of the two 15 000 bp sequences. The rest of the tools did not produce any results within 24 h.

Recombination dataset. Finally, we compared the tools on datasets with recombinants of the original sequences. Two paired-end samples, name SCR1 and SCR2 were generated with 40 000 reads each and a 0.3% noise. Both samples consist of three *in silico* progenitor sequences derived from Sabin 2. Sample SCR1 consists of additionally one recombinant between the first two progenitor sequences. Sample SCR2 consists of the same sequences as in SCR1 with two more recombinants, also between the first two progenitor sequences but with recombination events in different positions (Supplementary Tables S14–S16). As described in the first example of the environmental samples (Figure 2C), Hexahedron detects recombination events by providing the merging and bifurcation positions. As a result, recombinants of resolved sequences are represented as contigs of a single position, serving as links, which bifurcate from one sequence and merge to a different one (Supplementary Figure S22). The global sequences can be obtained by following the path starting from the link and continuing on both sides. Hexahedron resolved all three progenitor sequences in both samples and detected one link in SCR1 and three links in SCR2. QuRe did not produce any results. ViSpA predicted more sequences than the original ones but none with <0.5% mismatches. ShoRAH predicted 32 sequences in SCR1 where three of them were <0.5% distant from the original sequences and two in sample SCR2 but none successfully matched the original ones. PredictHaplo reconstructed four sequences for sample SCR1 but one did not match any of the original sequences; it also reconstructed four sequences for SCR2 without any of them being a successful prediction. On the other hand, Hexahedron predicted successfully all sequences in both samples (Table 4, Supplementary Figure S23). Similarly, Hexahedron outperformed all the tools in terms of frequency prediction based on Jensen-Shannon divergence between the predicted frequencies of all reconstructed sequences and the frequencies of the original sequences. It is also worth mentioning that we run the rest of the tools against the sample described in the section 'Environmental isolate of poliovirus (sample 2)', which also contains a recombinant and Hexahedron was the only tool to successfully produce results in <24 h.

DISCUSSION

Deep sequencing is a powerful tool for the analysis of heterogeneous populations that are present in most specimens derived from natural sources (environment, clinical isolates) and those that emerge during treatment of viral diseases and cancer. Creation of new experimental and mathematical techniques, as well as optimization of the existing protocols will play a crucial role in discovering and interpreting the biological impact of the sequence heterogeneity. It could be used for the surveillance for viral pathogens as well as monitoring the emergence of drug-resistant variants during the treatment of infectious diseases and cancer. This study presents a new algorithm that identifies discrete populations in heterogeneous samples based on mutation patterns in the genetic profile. Due to the nature of the algorithm, the only limiting factor is the size of the step, which is defined by the length of the alignments. Thus, the performance of the algorithm improves as the sequencing technologies advance and are able to produce longer reads with greater accuracy. *In silico* experiments with longer reads (Additional File2 S1-S4) demonstrated an increase in the accuracy of the sequence reconstruction; however short reads produced by current technologies appear to be already sufficient to separate sub-populations. Inaccurately reconstructed sequences and mismatched nucleotides were mostly observed in short contigs. One explanation is that the noise was partially phased with the mutation patterns that identified the sequence. This might cause the noise to be included in reconstructed sequences (particularly in the low frequency populations) or trigger the bifurcation of the contigs, which in turn accumulates the defining mutations of the sequence.

It is true that the noise will always constitute a limitation and similarly to any technology defines the level of sensitivity and accuracy of this algorithm. Recent high-throughput sequencers can reduce the noise to as low as 0.1% (35) enabling the detection of sequences of very low abundance. Another defining limitation of this algorithm is the representation of sequence space from the selected reference. The algorithm accepts alignments as an input, thus it inherits limitations introduced by the alignment methods. It is evident that sections of the sequence space that have not been represented sufficiently by the selected reference (s) will have a major impact on the accuracy of the reconstructed sequences and the predicted frequencies. For instance, a large insertion, longer that the read length, which is not included in the provided reference (s) will be missed by the alignment step. As a result, the algorithm will not accurately reconstruct the sequence. Furthermore, insertions of sequences that exhibit homology with other regions of the reference (s) can produce false positives. Insertions longer than the read length might cause reads to be assigned to the homology region represented by the reference. These alignments in combination with the alignments of the reads that are correctly assigned to this region can produce mutations and subsequently trigger a bifurcation event. The user must be aware of both the intrinsic and the parametric limitations of the algorithm. The noise and the length of the reads are considered intrinsic limitations and there is little the user can do to overcome them. The insufficiently represented sequence space is a limitation that is imposed by the input or the parameters provided to the algorithm. Hexahedron provides the option to use multiple references to represent the sample as comprehensively as possible in order to alleviate this problem. It is true that in case of absence of good reference sequence, *de novo* assembly tools are indispensable

Table 4. Summary of results from Hexahedron, PredictHaplo, QuRe, ViSpa and ShoRah against samples SCR1 and SCR2

Sample	# of haplotypes	ShoRAH	ViSpA	QuRe	PredictHaplo	Hexahedron
Number of reconstructed sequences						
SCR1	4	34	8	NA	4	4
SCR2	6	2	14	NA	4	6
Number of successfully predicted sequences (<0.01% mismatches)						
SCR1	4	0	0	NA	0	0
SCR2	6	0	0	NA	0	0
Number of successfully predicted sequences (<0.05% mismatches)						
SCR1	4	0	0	NA	0	3
SCR2	6	0	0	NA	0	5
Number of successfully predicted sequences (<0.1% mismatches)						
SCR1	4	0	0	NA	3	4
SCR2	6	0	0	NA	1	5
Number of successfully predicted sequences (<0.5% mismatches)						
SCR1	4	3	0	NA	3	4
SCR2	6	0	0	NA	1	6
JS Divergence						
SCR1	4	0.341	0.125	NA	0.0927	0.061
SCR2	6	0.411	0.312	NA	0.25	0.078

The Jensen-Shannon divergence is calculated based on all predicted sequences.

and can be used to create the reference sequence that will be used to re analyze the sample using Hexahedron. Alternatively, longer reads produced by HTS technologies such as PacBio can also be used as reference sequences to guide the assembly step of the algorithm.

Notably, the number of the selected references does not necessarily affect the sensitivity of the analysis. A large set of homologous references, such as the comprehensive list of enteroviruses used for the environmental samples, versus a small set of references that sufficiently maps all short reads will produce the same results even if the original sequences are absent in the reference set (Table 1). The density, with which the genetic space needs to be covered, is specified by the parameters of the alignment. In fact, comprehensive reference sets allow for identification of more distant and diverse populations without increasing the ambiguity with flexible alignments. Importantly, the information of the aligned reference is maintained and displayed for each reconstructed contig at the end of the process. At this point, the density with which this space has been covered by the selected references is indicated by the number of references contributing at every position. This is not a recombination analysis, where the identified references are part of recombination events. The graph is offering an overview of the closest similarities with references across the reconstructed sequences and serves as an indicator for a further, more detailed, recombination analysis. It is true though that careful selection of references can lead to robust results in terms of recombination events. This recombination analysis refers to detection of recombination events of the reference sequences that produced each of the reconstructed sequences in the sample. A different type is the recombination of two or more resolved sequences from the same sample. An important consideration is the fact that Hexahedron is not resolving these recombination events of the reconstructed sequences through a statistical model. Discovery of such events is achieved by detecting alignments with ends assigned on different already resolved contigs. Such alignments support the bifurcation and merging events that can be used to identify recombinants. As described in the 'Recombination dataset' section, Hexahedron identifies the recombination positions and does not globally infer the recombinant sequence. As a result, recombinant sequences that resulted from more than one recombination events are not easy to be reconstructed. The algorithm provides the option to extract the sequences following all permutations of the Sankey paths but this can increase type 1 errors. It is in our future plans to introduce a post-computational step for statistical inference of such recombinants.

The algorithm combined with the Sankey diagram of the reconstructed contigs gives a comprehensive representation of the genotypic cloud that describes highly diverse, viral populations. This study proposes a novel approach that can capture the mutant spectrum of evolving diverse populations in an exceptional accuracy and an unparalleled speed. Both characteristics will eventually be vital in clinical research where antiviral efficacy and recent combinatorial treatments have already proved (36,37) to be affected by escape mutants. Recent studies on vertically HIV infected children after failed nevirapine prophylaxis (NVP) revealed the existence of linked multiclass drug resistant mutations using single genome sequencing (38). Hence, our approach is of critical significance and can help to address the challenge of discovering broad quasispecies spectrum that single genome sequencing is unable to do due to its low throughput. From a broad basic research perspective, it will enhance the arsenal of supersensitive genotyping methods allowing the evaluation of superinfection, major variants and minor variants. Finally, we anticipate that this approach could be extremely useful for virological surveillance that is vitally important for timely identification of emerging pathogens and development of rational countermeasures.

DATA AVAILABILITY

The data sets supporting the results of this article and an implementation of the algorithm are available in HIVE at: https://hive.biochemistry.gwu.edu/review/Hexahedron%20publication.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thank M. Laassri who generated the deep sequence reads for Mumps and Sabin vaccine, S. Rubin who provided us with the mumps sample and N. Gulzar who tested the interface of the publicly accessible implementation of the algorithm. Konstantinos Karagiannis is a predoctoral student in the Biochemistry and Systems Biology Program of the Institute for the Biomedical Sciences at the George Washington University. This work is from a dissertation to be presented to the above program in partial fulfillment of the requirements for the Ph.D. degree.

Publication costs are covered by The McCormick Genomic and Proteomic Center (MGPC).

Authors' contribution: K.K. conceived, designed developed, implemented, debugged, tested the algorithm and the visualization methods, conceived, designed, performed and collected the data of the *in silico* experiments and wrote the manuscript. V.S. participated in initial design and mentored in algorithmic concepts. K.C. participated in initial design provided valuable input on biological significance, evaluated the algorithm; provided biological sequences and analyzed using the tool. R.M. provided guidance to K.K. and was involved in streamlining the analysis method for routine analysis of high-throughput sequence data. All authors read, approved and participated in preparation of the final manuscript.

FUNDING

This project is supported in part by the George Washington University graduate student stipend and the Research Participation Program at the Center of Biologics Evaluation and Research (CBER) administered by the Oak Ridge Institute for Science and Eduction (ORISE) through an interagency agreement between the US Department of Education (USDOE) and the US Food and Drug Administration (USFDA).

Conflict of interest statement. None declared.

REFERENCES

1. Sanjuan,R., Nebot,M.R., Chirico,N., Mansky,L.M. and Belshaw,R. (2010) Viral mutation rates. *J. Virol.*, **84**, 9733–9748.
2. Holmes,E.C. (2010) The RNA virus quasispecies: fact or fiction? *J. Mol. Biol.*, **400**, 271–273.
3. Arenas,M., Lorenzo-Redondo,R. and Lopez-Galindez,C. (2015) Influence of mutation and recombination on HIV-1 in vitro fitness recovery. *Mol. Phylogenet. Evol.*, **94**, 264–270.
4. Lundgren,J.D., Babiker,A.G., Gordin,F.M., Borges,A.H. and Neaton,J.D. (2013) When to start antiretroviral therapy: the need for an evidence base during early HIV infection. *BMC Med.*, **11**, 148.
5. Korber,B., Gaschen,B., Yusim,K., Thakallapally,R., Kesmir,C. and Detours,V. (2001) Evolutionary and immunological implications of contemporary HIV-1 variation. *Br. Med. Bull.*, **58**, 19–42.
6. Rossi,L.M., Escobar-Gutierrez,A. and Rahal,P. (2015) Advanced molecular surveillance of hepatitis C virus. *Viruses*, **7**, 1153–1188.
7. Sanger,F. and Coulson,A.R. (1975) A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J. Mol. Biol.*, **94**, 441–448.
8. Eriksson,N., Pachter,L., Mitsuya,Y., Rhee,S.Y., Wang,C., Gharizadeh,B., Ronaghi,M., Shafer,R.W. and Beerenwinkel,N. (2008) Viral population estimation using pyrosequencing. *PLoS Computat. Biol.*, **4**, e1000074.
9. Willerth,S.M., Pedro,H.A., Pachter,L., Humeau,L.M., Arkin,A.P. and Schaffer,D.V. (2010) Development of a low bias method for characterizing viral populations using next generation sequencing technology. *PLoS ONE*, **5**, e13564.
10. Metzker,M.L. (2010) Sequencing technologies - the next generation. *Nat. Rev. Genet.*, **11**, 31–46.
11. Zagordi,O., Daumer,M., Beisel,C. and Beerenwinkel,N. (2012) Read length versus depth of coverage for viral quasispecies reconstruction. *PLoS ONE*, **7**, e47046.
12. Prosperi,M.C., Yin,L., Nolan,D.J., Lowe,A.D., Goodenow,M.M. and Salemi,M. (2013) Empirical validation of viral quasispecies assembly algorithms: state-of-the-art and challenges. *Sci. Rep.*, **3**, 2837.
13. Yang,X., Charlebois,P., Macalalad,A., Henn,M.R. and Zody,M.C. (2013) V-Phaser 2: variant inference for viral populations. *BMC Genomics*, **14**, 674.
14. Mangul,S., Wu,N.C., Mancuso,N., Zelikovsky,A., Sun,R. and Eskin,E. (2014) Accurate viral population assembly from ultra-deep sequencing data. *Bioinformatics*, **30**, i329–i337.
15. Schirmer,M., Sloan,W.T. and Quince,C. (2014) Benchmarking of viral haplotype reconstruction programmes: an overview of the capacities and limitations of currently available programmes. *Brief Bioinform.*, **15**, 431–442.
16. Yang,X., Charlebois,P., Gnerre,S., Coole,M.G., Lennon,N.J., Levin,J.Z., Qu,J., Ryan,E.M., Zody,M.C. and Henn,M.R. (2012) De novo assembly of highly diverse viral populations. *BMC Genomics*, **13**, 475.
17. Topfer,A., Marschall,T., Bull,R.A., Luciani,F., Schonhuth,A. and Beerenwinkel,N. (2014) Viral quasispecies assembly via maximal clique enumeration. *PLoS Computat. Biol.*, **10**, e1003515.
18. Prosperi,M.C. and Salemi,M. (2012) QuRe: software for viral quasispecies reconstruction from next-generation sequencing data. *Bioinformatics*, **28**, 132–133.
19. Prosperi,M.C., Prosperi,L., Bruselles,A., Abbate,I., Rozera,G., Vincenti,D., Solmone,M.C., Capobianchi,M.R. and Ulivi,G. (2011) Combinatorial analysis and algorithms for quasispecies reconstruction using next-generation sequencing. *BMC Bioinformatics*, **12**, 5.
20. Beerenwinkel,N. and Zagordi,O. (2011) Ultra-deep sequencing for the analysis of viral populations. *Curr. Opin. Virol.*, **1**, 413–418.
21. Astrovskaya,I., Tork,B., Mangul,S., Westbrooks,K., Mandoiu,I., Balfe,P. and Zelikovsky,A. (2011) Inferring viral quasispecies spectra from 454 pyrosequencing reads. *BMC Bioinformatics*, **12**(Suppl. 6), S1.
22. Zagordi,O., Klein,R., Daumer,M. and Beerenwinkel,N. (2010) Error correction of next-generation sequencing data and reliable estimation of HIV quasispecies. *Nucleic Acids Res.*, **38**, 7400–7409.
23. Benson,D.A., Cavanaugh,M., Clark,K., Karsch-Mizrachi,I., Lipman,D.J., Ostell,J. and Sayers,E.W. (2013) GenBank. *Nucleic Acids Res.*, **41**, D36–D42.
24. Santana-Quintero,L., Dingerdissen,H., Thierry-Mieg,J., Mazumder,R. and Simonyan,V. (2014) HIVE-hexagon: high-performance, parallelized sequence alignment for next-generation sequencing data analysis. *PLoS ONE*, **9**, e99033.
25. Sabin,A.B., Ramos-Alvarez,M., Alvarez-Amezquita,J., Pelon,W., Michaels,R.H., Spigland,I., Koch,M.A., Barnes,J.M. and Rhim,J.S. (1960) Live, orally given poliovirus vaccine. Effects of rapid mass immunization on population under conditions of massive enteric infection with other viruses. *JAMA*, **173**, 1521–1526.
26. Kumar,S., Stecher,G. and Tamura,K. (2016) MEGA7: Molecular Evolutionary Genetics Analysis Version 7.0 for Bigger Datasets. *Mol. Biol. Evol.*, **33**, 1870–1874.
27. Saitou,N. and Nei,M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**, 406–425.
28. Kullback,S. (1987) The Kullback-Leibler distance. *Am. Stat.*, **41**, 340–340.
29. Rezapkin,G.V., Fan,L., Asher,D.M., Fibi,M.R., Dragunsky,E.M. and Chumakov,K.M. (1999) Mutations in Sabin 2 strain of poliovirus and stability of attenuation phenotype. *Virology*, **258**, 152–160.

30. Katoh,K. and Standley,D.M. (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.*, **30**, 772–780.
31. Afzal,M.A., Pickford,A.R., Forsey,T., Heath,A.B. and Minor,P.D. (1993) The Jeryl Lynn vaccine strain of mumps virus is a mixture of two distinct isolates. *J. Gen. Virol.*, **74**, 917–920.
32. Zagordi,O., Bhattacharya,A., Eriksson,N. and Beerenwinkel,N. (2011) ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC Bioinformatics*, **12**, 119.
33. Prabhakaran,S., Rey,M., Zagordi,O., Beerenwinkel,N. and Roth,V. (2014) HIV Haplotype Inference Using a Propagating Dirichlet Process Mixture Model. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **11**, 182–191.
34. Prabhakaran,S., Rey,M., Zagordi,O., Beerenwinkel,N. and Roth,V. (2013) HIV Haplotype Inference Using a Propagating Dirichlet Process Mixture Model. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, EA017B9F-EC09-4077-85AB-DED3AF538C48.
35. Goodwin,S., McPherson,J.D. and McCombie,W.R. (2016) Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.*, **17**, 333–351.
36. Behera,A.K., Basu,S. and Cherian,S.S. (2015) Molecular mechanism of the enhanced viral fitness contributed by secondary mutations in the hemagglutinin protein of oseltamivir resistant H1N1 influenza viruses: modeling studies of antibody and receptor binding. *Gene*, **557**, 19–27.
37. Pfeiffer,J.K. and Kirkegaard,K. (2003) A single mutation in poliovirus RNA-dependent RNA polymerase confers resistance to mutagenic nucleotide analogs via increased fidelity. *Proc. Natl. Acad. Sci. U.S.A.*, **100**, 7289–7294.
38. Lange,C.M., Hue,S., Violari,A., Cotton,M., Gibb,D., Babiker,A., Otwombe,K., Panchia,R., Dobbels,E., Jean-Philippe,P. *et al.* (2015) Single genome analysis for the detection of linked multiclass drug resistance mutations in HIV-1-infected children after failure of protease inhibitor-based first-line therapy. *J. Acquir. Immune Defic. Syndr.*, **69**, 138–144.