Supplementary Document A : Computational settings

The models were implemented on a 3.70GHz Intel i7-8700K processor, 2 NVIDIA GeForce GTX 1080Ti computer graphics cards and 32GB RAM. Randomness of the initial weights of the auto-encoder may cause the neural network to remain in a local minimum during training. To overcome this, we run GAEseq multiple times and choose the result with the lowest MEC score. Given a SNP fragment matrix, we run GAEseq 200 times, train 200 models and get 200 reconstructed haplotype matrix candidates; the algorithm selects the candidate corresponding to the lowest MEC score automatically. In the GAEseq software, users can specify how many times to run the algorithm; the software will run automatically as opposed to manually running GAEseq multiple times.

As for hyperparameters, the number of graph convolutional layers in the auto-encoder was set to 2 (in particular, one read-nodes-to-SNP-nodes layer and one SNP-nodes-to-read-nodes layer); dimension of messages reduces linearly during message passing between the layers. For example, if the number of reads is m, the haplotype length is n, the ploidy is k and we use 2 graph convolutional layers and a dense layer, the dimensions of the weight and bias matrix of the first and second layer are $n \times \lceil (n - \frac{n-k}{3}) \rceil$ and $\lceil (n - \frac{n-k}{3}) \rceil \times \lceil (n - 2 \times \frac{n-k}{3}) \rceil$, respectively. The dimensions of the weight and bias matrix of the denser layer are set to $\lceil (n - 2 \times \frac{n-k}{3}) \rceil \times k$ and $m \times k$, respectively. We use the Adam optimizer (Diederik 2015), set the step size to 0.0001, and set all other parameters to their default values in Tensorflow. We also use Xavier initialization (Xavier 2010) with default settings and the number of epoches set to 100. In our studies, we found that an architecture with two graph convolutional layers achieves significantly better results than state-of-the-art haplotype assembly methods. To ensure the model generalizes well to unobserved entries, we added dropout regularization to message passing between the layers; for each layer, the dropout probability is 0.1. For all experiments on viral quasispecies reconstruction, the initial number of clusters k_0 is set to 2.

Supplementary Document B : Determining the number of components in a viral quasispecies

When reconstructing haplotypes sampled by a collection of sequencing reads, GAEseq requires as input the number of haplotypes, k. While the ploidy of an individual organism in the haplotype assembly problem is known a priori, cardinality of a viral community needs to be estimated. To determine k, we examine the improvement rate of the MEC score defined as

$$\text{MECimpr}(k) = \frac{\text{MEC}(k) - \text{MEC}(k+1)}{\text{MEC}(k)}. \qquad (1)$$

Recall that the MEC score is defined as the smallest number of the observed entries in R that need to be altered such that the resulting data is consistent with having originated from k distinct haplotypes. The score decreases monotonically with k; however, once k reaches the actual number of components, the improvement rate of the MEC score (MECimpr) saturates. To find the saturation point, we compare MECimpr with a pre-defined threshold. Following (Ahn 2018), the number of components is determined via binary search. Specifically, starting from an initial k_0, the number of components is updated as $k_\tau \leftarrow 2k_{\tau-1}$ until $\text{MECimpr}(k_\tau) \leq \eta$; at this point, the number of components starts to decrease as $k_{\tau+1} \leftarrow \lfloor (k_\tau + \max\{1, k_i\})/2 \rfloor$ where $\{i \in \{1, \cdots, \tau-1\} : k_i \leq k_\tau\}$. Once $\text{MECimpr}(k_\tau) > \eta$, the number of components increases again as $k_{\tau+1} \leftarrow \lfloor (k_\tau + \min k_i)/2 \rfloor$ where $\{i \in \{1, \cdots, \tau-1\} : k_i > k_\tau\}$. If $k_\tau = k_{\tau-1}$, the search procedure stops by assigning $k_{\tau+1} \leftarrow k_\tau + 1$ which is the estimated number of strains. The recommended choice of the threshold η is discussed in (Ahn 2017) where the estimation of the number of components via MECimpr was demonstrated to be robust with respect to the choice of the threshold.

Supplementary Document C : Performance comparison on biallelic Solanum Tuberosum semi-experimental data

The semi-experimental data is obtained by simulating mutations, shotgun sequencing procedure, read alignment and SNP calling steps in an experiment on a single individual *Solanum Tuberosum*. In particular, we use *Haplogenerator* (Motazedi 2018) to generate haplotypes by introducing independent mutations that follow the lognormal distribution of a randomly selected genome region from *Solanum Tuberosum* chromosome 5 (Potato Genome Sequencing Consortium 2011) of length 5000 bp. The mean distance between neighboring SNPs and the standard deviation (SD) are set to 21 bp and 27 bp, respectively, as previously suggested by (Motazedi 2018). Due to *Haplogenerator*'s limitations, we constrain mutations to transitions and do not consider transversions (i.e., mutations are constrained to be between A and C and between G and T). 2×250 bp-long Illumina's MiSeq reads of inner distance 50 bp and standard deviation 10 bp are generated to uniformly sample haplotypes using *ART* software (Huang 2012) with default setting. Following this step, the generated reads are aligned to the reference genome using the BWA-MEM algorithm (Li 2009); the reads having mapping quality score lower than 60 or being shorter than 70 bp are discarded. SNPs are called if, at any given site, the abundance of a minor allele exceeds a predetermined threshold; the SNP fragment matrix is formed by collecting all such heterozygous sites. Seven different sets of semi-experimental data obtained by sampling at varying coverage (10×, 15×, 20×, 25×, 30×, 35× and 40×) are generated; each set consists of 10 samples. We first generate genome regions of length 5000 bp by partitioning the *Solanum Tuberosum* chromosome 5 and then randomly select 70 among them (generated haplotypes and reads are different for each sample). The sequencing error rate is automatically set by the built-in quality profiles of *ART* inferred from large amounts of recalibrated sequencing data (Huang 2012). Table 1 shows the performance comparison of GAEseq, AltHap, HapCompass and H-PoP on biallelic Solanum Tuberosum semi-experimental data.

Table 1: Performance comparison of GAEseq, AltHap, HapCompass and H-PoP on biallelic Solanum Tuberosum semi-experimental data.

Coverage		MEC Mean	SD	CPR Mean	SD
10	GAEseq	**18.500**	**4.552**	**0.848**	0.074
	HapCompass	100.300	43.584	0.769	**0.039**
	H-PoP	19.700	25.254	0.803	0.086
	AltHap	64.100	32.953	0.727	0.072
15	GAEseq	**8.200**	**4.686**	**0.822**	0.048
	HapCompass	100.700	66.150	0.763	**0.046**
	H-PoP	28.700	32.667	0.783	0.066
	AltHap	59.100	28.125	0.709	0.054
20	GAEseq	**16.800**	**15.873**	**0.862**	0.062
	HapCompass	95.600	53.883	0.795	**0.047**
	H-PoP	30.500	37.023	0.791	0.078
	AltHap	82.100	56.658	0.737	0.068
25	GAEseq	**8.400**	**4.719**	**0.831**	0.081
	HapCompass	124.800	132.156	0.810	0.063
	H-PoP	33.800	47.434	0.798	**0.046**
	AltHap	92.600	83.649	0.756	0.068
30	GAEseq	**27.200**	**19.887**	**0.914**	**0.033**
	HapCompass	306.800	187.934	0.796	0.081
	H-PoP	34.200	32.798	0.879	0.088
	AltHap	263.000	499.659	0.762	0.133
35	GAEseq	**10.700**	**3.234**	**0.857**	0.087
	HapCompass	217.400	174.135	0.775	**0.072**
	H-PoP	41.700	53.971	0.823	0.094
	AltHap	164.000	101.583	0.754	0.093
40	GAEseq	**16.400**	**7.333**	**0.835**	**0.034**
	HapCompass	208.000	176.699	0.833	0.070
	H-PoP	30.4	28.487	0.823	0.102
	AltHap	195.8	281.641	0.762	0.084

Supplementary Document D : Performance comparison on simulated biallelic diploid data and polyallelic triploid and tetraploid data.

To further test GAEseq, we evaluate its performance on synthetic data. Once again we use *Haplogenerator* (Motazedi 2018) to generate haplotypes of a randomly synthesized reference genome of length 5000 bp. The mean distance between neighboring SNPs and the standard deviation (SD) are set to 5 bp and 3 bp respectively, creating haplotype blocks of length about 500. All the possible mutations were allowed and set to be equally likely, leading to not only biallelic but also polyallelic SNPs in the synthesized haplotype data. Illumina's MiSeq read generation, read alignment and SNP calling procedures are implemented following the same procedure as in the case of semi-experimental data from Section 3.1. The data synthesized in this fashion consists of 24 different sets, each with 10 samples, as we explore different ploidy (k = 2, 3 and 4) and sequencing coverage (5×, 10×, 15×, 20×, 25×, 30×, 35× and 40×).

For the diploid synthetic data sets, we represent an allele by 0 if it coincides with the corresponding reference allele and by 1 if it is an alternative allele. SNP positions with only alternative alleles are removed. In addition to H-PoP, HapCompass and AltHap, we also compare GAEseq with HapCUT2 (Edge 2017); by design, use of HapCUT2 is limited to haplotype assembly of diploids. The metrics of performance are the previously introduced MEC score and CPR. Table 2 shows the mean and standard deviation of the MEC score and CPR for diploid data. The results are evaluated over 10 samples for each combination of ploidy and coverage. GAEseq achieves the lowest average MEC score and the lowest standard deviation of the MEC score for almost all coverage settings; its performance is followed by those of H-PoP, HapCompass, HapCut2 and AltHap. The average CPR achieved by GAEseq is very close to 1 for all coverage settings, indicating that GAEseq is able to near-perfectly reconstruct haplotypes of diploid species even when the coverage is very low; its performance is followed by those of H-PoP, HapCut2, HapCompass and AltHap. When the coverage is 20×, the average CPR achieved by GAEseq is 100% while it is approximately 98.9%, 97.2%, 96.1% and 74.3% for H-PoP, HapCut2, HapCompass and AltHap, respectively.

For the polyploid synthetic data sets, both H-PoP and HapCompass are restricted to reconstruction of biallelic haplotypes and are not applicable to the assembly of polyallelic ones. Furthermore, recall that HapCUT2 can only be ap-

Table 2: Performance comparison of GAEseq, HapCut2, HapCompass, H-PoP and AltHap on simulated biallelic diploid data.

Coverage		MEC Mean	SD	CPR Mean	SD
5	GAEseq	**23.300**	**4.165**	**0.996**	**0.002**
	HapCUT2	110.500	23.922	0.975	0.006
	HapCompass	87.500	25.903	0.965	0.010
	H-Pop	40.000	30.551	0.989	0.011
	AltHap	884.200	659.565	0.699	0.204
10	GAEseq	**30.700**	**6.667**	**0.999**	**0.001**
	HapCUT2	213.600	63.132	0.980	0.005
	HapCompass	159.600	58.329	0.974	0.005
	H-Pop	34.600	6.736	0.997	0.004
	AltHap	583.900	948.344	0.796	0.218
15	GAEseq	**47.800**	**8.587**	**0.999**	**0.001**
	HapCUT2	339.800	59.066	0.978	0.003
	HapCompass	268.300	67.003	0.971	0.005
	H-Pop	47.900	9.539	0.998	0.002
	AltHap	342.900	379.213	0.852	0.169
20	GAEseq	**70.900**	**10.754**	**1.000**	**0.001**
	HapCUT2	519.400	57.386	0.972	0.010
	HapCompass	408.000	81.067	0.961	0.018
	H-Pop	129.700	191.788	0.989	0.030
	AltHap	668.400	579.261	0.787	0.201
25	GAEseq	**85.200**	**16.130**	**1.000**	**0.001**
	HapCUT2	613.000	157.786	0.977	0.006
	HapCompass	460.700	97.637	0.968	0.007
	H-Pop	85.700	17.192	0.998	0.003
	AltHap	1151.600	649.058	0.743	0.150
30	GAEseq	**97.800**	8.954	**1.000**	**0.000**
	HapCUT2	685.300	180.714	0.979	0.006
	HapCompass	591.600	150.400	0.968	0.009
	H-Pop	98.000	**8.743**	0.999	0.001
	AltHap	554.000	612.292	0.871	0.185
35	GAEseq	**107.300**	8.138	**1.000**	**0.001**
	HapCUT2	827.600	202.643	0.978	0.006
	H-Pop	702.200	180.647	0.968	0.007
	H-Pop	107.900	**8.006**	0.999	0.001
	AltHap	668.800	730.814	0.891	0.146
40	GAEseq	**124.000**	10.499	**1.000**	**0.001**
	HapCUT2	1015.400	219.442	0.977	0.006
	HapCompass	896.500	204.603	0.965	0.008
	H-Pop	124.500	**10.277**	0.999	**0.001**
	AltHap	1073.300	1099.181	0.847	0.184

plied to diploid haplotypes. We therefore limit performance comparison of GAEseq on polyploid synthetic data to only AltHap; Tables 3 and 4 illustrate the mean and standard deviation of the MEC score and CPR for triploid and tetraploid data, respectively. The results are evaluated over 10 samples for each combination of ploidy and coverage. As can be seen in these tables, GAEseq outperforms AltHap for all ploidy and coverage settings. As shown in Table 3, GAEseq performs well on triploid data, achieving 92% average CPR and relatively small standard deviation even for the low coverage of $5\times$; at the same time, performance of AltHap deteriorates rapidly with increased ploidy, achieving 72% average CPR while GAEseq achieves 98.2% at coverage $30\times$. As illustrated in Table 4, in applications to tetraploid data the performance of GAEseq starts to gracefully deteriorate – when the coverage is $10\times$, GAEseq achieves average CPR of approximately 80% while in the same scenario AltHap achieves average CPR of approximately 65%. When the coverage is increased to $40\times$, GAEseq achieves average CPR of approximately 87.8% while AltHap achieves average CPR of approximately 76.2%.

Table 3: Performance comparison of GAEseq and AltHap on simulated polyallelic triploid data.

Coverage		MEC Mean	SD	CPR Mean	SD
5	GAEseq	**103.400**	**51.379**	**0.920**	**0.047**
	AltHap	1908.500	237.324	0.559	0.059
10	GAEseq	**112.800**	**45.917**	**0.958**	**0.037**
	AltHap	1769.300	948.754	0.760	0.091
15	GAEseq	**165.800**	**106.999**	**0.945**	**0.073**
	AltHap	1058.100	864.563	0.796	0.123
20	GAEseq	**241.300**	**159.657**	**0.959**	**0.047**
	AltHap	1287.100	578.507	0.682	0.070
25	GAEseq	**314.900**	**158.326**	**0.934**	**0.070**
	AltHap	1430.200	757.482	0.775	0.093
30	GAEseq	**292.400**	**203.242**	**0.974**	**0.040**
	AltHap	2133.200	1082.576	0.729	0.109
35	GAEseq	**306.200**	**196.918**	**0.982**	**0.037**
	AltHap	2928.700	869.617	0.723	0.075
40	GAEseq	**502.200**	**247.380**	**0.922**	**0.088**
	AltHap	2943.600	1113.480	0.737	0.104

Table 4: Performance comparison of GAEseq and AltHap on simulated polyallelic tetraploid data.

Coverage		MEC Mean	SD	CPR Mean	SD
5	GAEseq	**266.700**	**46.371**	**0.739**	**0.041**
	AltHap	2641.700	410.159	0.544	0.056
10	GAEseq	**415.100**	**74.608**	**0.800**	**0.051**
	AltHap	2807.200	938.668	0.658	0.075
15	GAEseq	**592.200**	**112.282**	**0.798**	**0.054**
	AltHap	2742.500	1055.672	0.718	0.081
20	GAEseq	**628.900**	**245.841**	**0.843**	**0.047**
	AltHap	1929.700	1008.766	0.729	0.063
25	GAEseq	**881.900**	**189.987**	**0.845**	**0.058**
	AltHap	1987.100	1091.893	0.779	0.084
30	GAEseq	**944.100**	**182.440**	**0.848**	**0.041**
	AltHap	2265.200	1277.366	0.759	0.051
35	GAEseq	**815.900**	**295.195**	**0.866**	**0.063**
	AltHap	3906.400	1131.654	0.747	0.056
40	GAEseq	**949.500**	**319.238**	**0.878**	**0.046**
	AltHap	3775.300	1036.702	0.762	0.075

Supplementary Document E : Performance comparison on real Solanum Tuberosum data

We further test the performance of GAEseq on real potato data (accession SRR6173308) at *Solanum Tuberosum* chromosome 5 (Potato Genome Sequencing Consortium 2011). The 10 samples of real potato data are generated by first randomly selecting 10 genome regions of length varying from 5032 to 7573 and then aligning the Illumina HiSeq 2000 paired-end reads to the selected genome regions. After the read alignment step using the BWA-MEM algorithm (Li 2009), the SNP calling step is implemented to create the SNP fragment matrix. Reads having mapping quality score lower than 60 or shorter than 70 bp are discarded. Since the ground truth haplotypes are not available for this dataset, we only evaluate the performance of GAEseq and the competing methods in terms of the MEC score. Table 5 compares the performance of GAEseq, AltHap, HapCompass and H-PoP averaged over 10 selected regions of the real Solanum Tuberosum data. As can be seen from the table, GAEseq outperforms all the competing schemes in terms of both the average MEC score and its standard deviation, achieving 379.8 average MEC score. GAEseq is followed by H-PoP and AltHap while HapCompass achieves the highest average MEC score.

Table 5: Performance comparison of GAEseq, AltHap, HapCompss and H-PoP on the real Solanum Tuberosum data.

	MEC Mean	SD
GAEseq	**379.8**	**271.61**
HapCompass	2726	2393.7
H-PoP	409.5	282.24
AltHap	742.1	469.5

Supplementary Document F : Further results on reconstruction of HIV viral communities

Table 6 shows the gene-wise reconstruction results on the real HIV-1 data that include inferred frequencies (omitted from Table 2 in the main paper for brevity).

We further evaluate the performance of GAEseq on the 4036bp long gag-pol region. Following (Ahn 2018), we divide the gag-pol region into overlapping blocks, reconstruct the viral components in each block independently, and combine the results to reconstruct the full region of interest. Specifically, the region is divided into a sequence of blocks of length 500bp where the consecutive blocks overlap by 250bp. We run GAEseq to perform reconstruction of viral components in each of the total 18 blocks and merge the results to retrieve the entire region of interest. Particularly, the mismatches between strains reconstructed on two consecutive blocks in the overlapping region are corrected based on majority voting using reads that are covering the mismatched positions and are assigned to the aligned strains. Following this procedure, GAEseq perfectly reconstructed all of 5 HIV-1 strains in the gag-pol region, achieving 100% Reconstruction Rate for all 5 strains and Predicted Proportion of 1 on 355241 remained paired-end reads. The frequencies of 5 HIV-1 strains are estimated as $15.21\%, 19.34\%, 25.56\%, 27.61\%$ and 12.27% by counting the proportion of reads assigned to the same strain; these results are consistent with the frequencies estimated by aBayesQR and TenSQR softwares.

Table 6: Performance comparisons of GAEseq, TenSQR, aBayesQR and PredictHap on a real HIV-1 5-virus-mix data.

		p17	p24	p2-p6	PR	RT	RNase	int	vif	vpr	vpu	gp120	gp41	nef
GAEseq	PredProp	**1**	1	**1**	**1**	**1.2**	**1**	**1**	**1**	**1**	1.2	1	1	1
	CPR_{HXB2}	**100(20.5)**	99.4(17.1)	**100(21)**	**100(30.9)**	**100(12.1)**	**100(9.6)**	**100(13.6)**	**100(10.4)**	**100(6.6)**	100(34.3)	96.2(8.7)	96.7(2.8)	100(6.6)
	$CPR_{89.6}$	**100(18.8)**	99.4(21.8)	**100(20)**	**100(18)**	**100(18.2)**	**100(20.9)**	**100(18.2)**	**100(20.4)**	**100(20.3)**	99.2(10.5)	99.4(24.1)	100(25.6)	98.2(22.9)
	CPR_{JR-SCF}	**100(30.9)**	100(31.5)	**100(27)**	**100(21.6)**	**100(23.5)**	**100(20.2)**	**100(21.5)**	**100(29.8)**	**100(34.6)**	100(36.4)	99.9(33.0)	100(27)	99.3(19.7)
	CPR_{NL4-3}	**100(17.4)**	100(18.3)	**100(14.1)**	**100(19.7)**	**100(30.6)**	**100(33.1)**	**100(37.1)**	**100(32.8)**	**100(30.6)**	100(7.9)	100(29.6)	100(34.5)	99.8(39.1)
	CPR_{YU2}	**100(12.3)**	100(11.2)	**100(18)**	**100(9.9)**	**100(11.9)**	**100(16.2)**	**100(9.6)**	**100(6.6)**	**100(7.9)**	100(10.2)	99.6(4.6)	100(10)	98.1(11.7)
PredictHap	PredProp	**1**	0.6	**1**	**1**	**1**	0.8	0.8	0.8	**1**	0.8	0.8	0.8	0.8
	CPR_{HXB2}	**100(17.8)**	0(0)	**100(18.7)**	**100(15.2)**	**100(12.2)**	98.9(25.4)	100(12.1)	100(17.7)	**100(10.2)**	93.2(10.8)	0(0)	0(0)	0(0)
	$CPR_{89.6}$	**100(19.9)**	100(46.4)	**100(21.7)**	**100(22.2)**	**100(19.4)**	100(18.2)	99.8(27.6)	100(20.9)	**100(22.1)**	0(0)	97.8(20.7)	100(26.7)	98.8(20.7)
	CPR_{JR-SCF}	**100(31.9)**	100(21.8)	**100(30.3)**	**100(26.9)**	**100(23.4)**	100(23.2)	100(22.3)	100(24.9)	**100(23.7)**	100(34.1)	99.7(42.7)	100(28.9)	100(23.2)
	CPR_{NL4-3}	**100(17)**	99.1(31.8)	**100(16.4)**	**100(20.9)**	**100(30.2)**	100(33.2)	100(38.1)	100(36.6)	**100(35.5)**	100(47.1)	100(28.6)	100(32.7)	100(39.3)
	CPR_{YU2}	**100(13.4)**	0(0)	**100(12.9)**	**100(14.8)**	**100(14.7)**	0(0)	0(0)	0(0)	**100(8.5)**	100(7.9)	98.6(7.9)	100(11.7)	100(16.9)
TenSQR	PredProp	**1**	1.6	**1**	1	1.4	**1**	**1**	**1**	**1**	1.6	2.2	1.2	0.8
	CPR_{HXB2}	**100(18.7)**	98.9(13.1)	**100()17.4**	100(9.9)	99.2(12.1)	**100(9.2)**	**100(8.1)**	**100(9.6)**	**100(7.2)**	92.8(5.9)	96.0(18)	99.0(11.5)	0(0)
	$CPR_{89.6}$	**100(18.4)**	100(19.6)	**100(20.1)**	100(17.2)	98.0(13.5)	**100(17.2)**	**100(16.7)**	**100(25)**	**100(19.3)**	94.0(15)	97.2(10.3)	100(27.8)	95.7(26)
	CPR_{JR-SCF}	**100(33.8)**	100(33)	**100(33.6)**	100(21.7)	100(20.7)	**100(24.6)**	**100(23.3)**	**100(20.5)**	**100(20.3)**	100(31.4)	98.3(33.5)	97.7(18.8)	99.8(19)
	CPR_{NL4-3}	**100(17)**	99.3(19.7)	**100(17.2)**	100(21.4)	99.5(26.7)	**100(37.7)**	**100(41.2)**	**100(38.4)**	**100(46.2)**	100(38.8)	99.8(9.2)	99.5(23.2)	99.7(42.7)
	CPR_{YU2}	**100(12.1)**	99.3(14.6)	**100(7.7)**	99.7(29.8)	99.7(14.5)	**100(11.4)**	**100(10.7)**	**100(6.5)**	**100(7.1)**	100(4.1)	94.9(10.5)	100(10.2)	98.6(12.3)
aBayesQR	PredProp	1	1	**1**	**1**	1	**1**	1	1	**1.2**	1	0.8	0.8	1.2
	CPR_{HXB2}	**100(16.3)**	99.4(21.1)	**100(22.2)**	**100(12.5)**	98.5(24.3)	**100(16.1)**	99.9(9.7)	100(9.2)	**100(16.4)**	99.6(17)	98(30.3)	0(0)	95.8(11.4)
	$CPR_{89.6}$	**100(27.1)**	98.7(17)	**100(17.3)**	**100(17.3)**	98.6(18.1)	**100(19.7)**	100(22.2)	100(20.6)	**100(16.3)**	92(10.4)	96.5(20.2)	98.9(23.7)	95.5(16.4)
	CPR_{JR-SCF}	**100(31.3)**	99.6(24.6)	**100(25.8)**	**100(29.9)**	99(21.5)	**100(22.1)**	100(20.8)	100(32.7)	**100(27)**	98.8(26.7)	97.7(21.4)	99.1(29.7)	98.2(21.1)
	CPR_{NL4-3}	**100(12.9)**	100(21.6)	**100(25.6)**	**100(20.1)**	98.9(17.7)	**100(30)**	100(39.5)	99.8(28.5)	**100(23.2)**	100(41.3)	96.3(28)	98.8(36.6)	100(31.8)
	CPR_{YU2}	**100(12.4)**	99.7(15.8)	**100(9.2)**	**100(20.3)**	99.2(18.5)	**100(12.2)**	99.5(7.9)	99.7(9)	**100(17.1)**	100(4.6)	0(0)	98.6(10.1)	99.2(14)

Predicted Proportion (PredProp) and Correct Phasing Rate (CPR (%)) for GAEseq, PredictHaplo, TenSQR and aBayesQR applied to reconstruction of HIV-1HXB2, HIV-189.6, HIV-1JR-CSF, HIV-1NL4-3 and HIV-1YU2 for all 13 genes of the HIV-1 dataset. Frequencies are reported in parenthesis.

References

Diederik P. Kingma and Jimmy Ba 2015. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs.LG]*.

Xavier Glorot and Yoshua Bengio 2010. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249-256.

Ahn, S.; Ke, Z.; and Vikalo, H. (2018). Viral quasispecies reconstruction via tensor factorization with successive read removal. *Bioinformatics (Oxford, England)*, 34(13), i23–i31.

Ahn, S.; and Vikalo, H. 2017. aBayesQR: A bayesian method for reconstruction of viral populations characterized by low diversity. In *International Conference on Research in Computational Molecular Biology*, pages 353–369. Springer.

Motazedi, E.; Finkers, R.; Maliepaard, C.; and de Ridder, D. 2018. Exploiting next-generation sequencing to solve the haplotyping puzzle in polyploids: a simulation study. *Briefings in bioinformatics*, 19(3), 387–403.

Potato Genome Sequencing Consortium 2011. Genome sequence and analysis of the tuber crop potato. *Nature*, 475, 189–195.

Huang, W.; and Li, L.; Myers, J. R. and Marth, G. T. 2012. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4), 593–594.

Li, H.; and Durbin, R. 2009. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25(14), 1754–1760.

Edge, P.; Bafna, V.; and Bansal, V. 2017. Hapcut2: robust and accurate haplotype assembly for diverse sequencing technologies. *Genome Res.*, 2017; 27(5):801–12.