Reconstruction of Viral Variants via Monte Carlo Clustering

Akshay Juyal,[1†] Roya Hosseini,[1†] Daniel Novikov,[1]

Mark Grinshpon,[2*] Alex Zelikovsky[1*]

[1]Department of Computer Science, Georgia State University,

Atlanta, GA 30303, USA

[2]Department of Mathematics and Statistics, Georgia State University,

Atlanta, GA 30303, USA

[†]Joint first authors

[*]To whom correspondence should be addressed;

E-mail: mgrinshpon@gsu.edu, alexz@gsu.edu.

September 20, 2023

Keywords: Clustering, Entropy, Hamming distance, Monte Carlo optimization, Viral genomic sequences

Abstract: Identifying viral variants via clustering is essential for understanding the composition and structure of viral populations within and between hosts, which play a crucial role in disease progression and epidemic spread. This paper proposes and validates novel Monte Carlo methods for clustering aligned viral sequences by minimizing either entropy or Hamming distance from consensuses.

We validate these methods on four benchmarks: two SARS-CoV-2 interhost datasets and two HIV intrahost datasets. A parallelized version of our tool is scalable to very large datasets. We show that both entropy and Hamming distance based Monte Carlo clusterings discern the meaningful information from sequencing data. The proposed clustering methods consistently converge to similar clusterings across different runs. Finally, we show that Monte Carlo clustering improves reconstruction of intrahost viral population from sequencing data.

1 Introduction

Clustering viral sequences is crucial in characterizing the composition and structure of intrahost and interhost viral populations, which are significant factors in disease progression and epidemic spread. In the case of intrahost populations, clustering enables us to identify the various viral variants present in a patient, including minor low-frequency variants that can cause immune evasion, drug resistance, and an increase in virulence and infectivity (Beerenwinkel et al., 2005; Douek et al., 2006; Gaschen et al., 2002; Holland et al., 1992; Rhee et al., 2007; Campo et al., 2014; Skums et al., 2015). These minor variants are often responsible for transmissions and the establishment of infection in new hosts. Therefore, clustering viral sequences not only provides a better understanding of the virus's behavior but also aids in the development of effective control strategies (Campo et al., 2016; Glebova et al., 2017; Skums et al., 2017).

On the other hand, clustering viral populations across different hosts provides valuable insights into major strains of closely related viral samples. This information is particularly helpful in tracking transmissions and informing public health strategies (Bousali et al., 2021). By clustering viral sequences, we

can identify the source of an outbreak and determine whether it is present in the sampled population. Clustering can also indicate whether two viral samples belong to the same outbreak and whether one infected the other (Melnyk et al., 2020). As a result, accurate characterization of viral mutation profiles from infected individuals through clustering is essential for viral research, therapeutics, and epidemiological investigations.

Viral sequences can be thought of as vectors of categorical data, as they are composed of strings from a fixed nucleotide alphabet. In the optimal clustering scenario, the sequences within a cluster are as homogeneous as possible at each site. Traditionally, this is accomplished by minimizing the Hamming distances between sequences within a cluster or by minimizing the distances to the cluster's consensus (de la Vega et al., 2003). However, Hamming distance assumes that all mutations at all sites are of equal cost, which may not be true in reality. Furthermore, Hamming distance does not consider the distribution of values within a given category, treating all mismatches as equal. To address these shortcomings of Hamming distance, in this paper we propose, as an alternative, the use of entropy for clustering viral sequences. Unlike Hamming distance, entropy takes into account the distribution of nucleotides at each site, enabling us to capture different types of mismatches.

This paper proposes a Monte Carlo (MC) optimization method for clustering viral sequences by minimizing either the total Hamming distance or the total entropy. The MC method repeatedly improves an initial clustering by attempting to randomly move a sequence between clusters and accepting such move if it reduces Hamming distance or entropy.

To improve runtime and ensure that the method is scalable to very large datasets, we incorporate a tag selection preprocessing step, which chooses a smaller predefined number of sites with the highest site entropies. We show

that this enhancement results in a significant reduction in runtime.

We have validated the effectiveness of our MC clustering method on real viral sequencing datasets. We estimate the amount of meaningful information extracted by the clustering. Furthermore, we demonstrate stability of the MC method by showing that it converges to similar clusterings across multiple runs.

We also show that the MC clustering method enhances the CliqueSNV tool (Knyazev et al., 2021). We apply MC clustering to haplotypes produced by CliqueSNV for HIV intrahost sequencing benchmarks and obtain better characterizations of the intrahost populations.

2 Methods

In this section, we begin by reviewing the definitions of entropy and Hamming distance measures for clusterings of set of aligned viral sequences. We then present a Monte Carlo clustering algorithm that is a modification of a previously proposed algorithm from Li et al. (2004). This algorithm can use either entropy or Hamming distance as the measure to minimize. Finally, we describe a preprocessing step of tag selection, which significantly reduces the runtime of the algorithm by selecting the highest entropy sites to represent the sequences.

2.1 Entropy Based Clustering of Viral Sequences

We consider aligned viral sequences as vectors of categorical data, where the categories are the sites along the sequences, and the values are the letters of the nucleotide alphabet $\{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$, not counting the gap symbol (`-`).

The entropy of a set of sequences at a site quantifies homogeneity of the values at this site: it is lower when a single value is highly frequent, and it is higher when all values are equally frequent. Summing over all sites, we measure homogeneity of the set of viral sequences. Given a clustering of a set of aligned

viral sequences, we can do this for every cluster to obtain a measure of similarity within each cluster.

In Juyal et al. (2022), the entropy of a cluster is defined as the sum of site entropies within this cluster, and then the entropy of the clustering is the sum of cluster entropies weighted by clusters' relative sizes.

In Li et al. (2004), the authors prove that clustering entropy defined in this way is a convex function, allowing any optimization procedure to reach a global minimum. Therefore, minimizing clustering entropy is a valid objective for applying minimization techniques to clustering aligned viral sequences.

2.2 Hamming Distance Based Clustering of Viral Sequences

The Hamming distance of a clustering was also defined in Juyal et al. (2022) in a similar fashion. The Hamming distance at a site is total Hamming distance from the consensus letter at his site; effectively, it is the total frequency of all letters different from the consensus one at this site. Then the Hamming distance of a cluster is defined as the sum of site Hamming distances, and the Hamming distance of the clustering is the sum of cluster Hamming distances.

2.3 Monte Carlo Optimization Algorithm

In Juyal et al. (2022), the authors describe an algorithm implementing a Monte Carlo optimization method, with certain modifications that are intended to reduce its runtime and improve the quality of the clusterings it produces.

Given an initial clustering of a set of viral sequences, Monte Carlo optimization attempts to apply random changes, accepting a change only if it improves the objective. In this case, the objective is to minimize either the clustering entropy or the clustering Hamming distance.

The algorithm in Juyal et al. (2022) starts with an initialization phase, where

it computes nucleotide counts for each column in each cluster. These counts are then used to compute the values of the entropy or the Hamming distance for each cluster, as well as the overall clustering entropy or Hamming distance.

Then the algorithm enters an optimization loop by repeatedly attempting a trial step, which will be accepted or rejected depending on whether it sufficiently improves the objective. Each trial step amounts to picking a random sequence s from a randomly selected cluster A and moving it to another randomly selected cluster B. If the relative entropy reduction from this move is positive and higher than a predefined threshold value, the move is accepted; otherwise, the move is rejected and the clustering is reverted to its previous state (i.e., the sequence s is returned to cluster A).

The algorithm terminates if no moves are accepted for a sufficiently long time, i.e., if there are no changes for a certain predefined number of consecutive iterations of the trial loop.

3 Clustering of Inter/Intrahost Viral Populations

We validate the Monte Carlo clustering optimization method by estimating improvement over an existing clustering technique. To that end, we apply this algorithm to clusterings obtained from the CliqueSNV tool (Knyazev et al., 2021).

3.1 Datasets

We validate our entropy based and Hamming distance based Monte Carlo clustering methods on four datasets: two interhost benchmarks of the SARS-CoV-2 virus and two intrahost benchmarks of the HIV virus, see Table 1.

3.1.1 Interhost viral sequencing benchmarks

We use two datasets of SARS-CoV-2 sequences introduced in Melnyk et al. (2021). For both datasets, we applied CliqueSNV to obtain an initial clustering.

Interhost D1: This dataset encompasses all the sequences that were submitted to the global GISAID viral database (Khare et al., 2021) between December 2019 and the beginning of March 2020. It comprises a total of 3688 aligned SARS-CoV-2 sequences, each 29891 nucleotides long. The initial clustering of this dataset consists of 28 clusters produced by CliqueSNV.

Interhost D2: This dataset includes all sequences submitted to the UK-based EMBL-EBI (2023) database from the end of January 2020 to the end of December 2020. It consists of 148000 aligned SARS-CoV-2 sequences, each 29903 nucleotides long. CliqueSNV produced an initial clustering of this dataset consisting of 15 clusters.

3.1.2 Intrahost viral sequencing benchmarks

We use two datasets HIV2 (Intrahost D3) and HIV5 (Intrahost D4) of HIV sequences with 2 and 5 true haplotypes, respectively, introduced in Knyazev et al. (2021). For both datasets, we applied CliqueSNV to obtain an initial clustering.

Intrahost D3: This dataset contains 49988 Illumina MiSeq 2×300 bp paired reads from the 1074 bp long region of HIV known protease and reverses transcriptase genes. The initiasl clustering of this dataset consists of 45 haplotypes produced by CliqueSNV for the threshold frequency of TF = 0.001 and 27 haplotypes for TF = 0.002.

Intrahost D4: This dataset contains 711228 Illumina MiSeq 2×250 bp paired reads from the 9275 bp long region of HIV known protease and reverses

transcriptase genes.

These updated datasets provide a significant number of aligned sequences for each HIV type, allowing for more in-depth analysis of genetic variations and clustering using the CliqueSNV method.

3.2 Distinguishing Signal from Noise

We estimate the amount of meaningful information extracted by the clusterings obtained by our Monte Carlo optimization method. To distinguish between sample-specific noise and meaningfully extracted information, we run our method on a perturbed version of the input with the same starting entropy. For this experiment, we use the D1 dataset with 3688 sequences, alongside the initial clustering from CliqueSNV of 28 clusters.

The permutation procedure is as follows. Within each cluster, every site is shuffled into a random permutation. Importantly, by respecting clusters during permutation, the initial nucleotide frequencies within each site in each cluster stay the same. Thus, the permuted input has the same starting entropy as the original input; what changes is the haplotypes being clustered.

We run the program on both of these inputs for exactly 100000 Monte Carlo trials each, accepting all moves that reduce entropy. We compare the resulting entropy reductions between the two runs. Any entropy reduction present in the permuted data is sample-specific noise extracted by our method, while the difference in resulting entropies between the original and permuted inputs corresponds to the amount of meaningful information extracted by our method.

3.3 Stability of Monte Carlo Clustering

Now we evaluate the robustness of our method against slight permutations of the input data as well as changes in random seed. Rather than completely shuffling

each site, we only shuffle a small percentage p of nucleotides at each site. We still respect clusters when permuting the data, to ensure that nucleotide frequencies in each site in each cluster remain unchanged.

We chose two values of p to create slightly permuted data sets for validation, $p = 1\%$ and $p = 5\%$, to be compared with the original data with 0% permutation. For each of the three datasets (two permuted and one original), we run our method three times, on two different objectives: first minimizing entropy, and second minimizing Hamming distance between sequences and their cluster consensus. As a result, for each degree of permutation and for each Monte Carlo objective, we obtain three clusterings.

The Rand index, which quantifies the degree of agreement between two clusterings, is measured between the initial clustering and each of the resulting clusterings, to get a sense of how far away the resulting clusterings have moved from the initial one under varying degrees of permutation. Further, we also measure the Rand index between the resulting clusterings, to determine whether the proposed method converges to similar clusterings across multiple runs.

3.4 Enhancing CliqueSNV with Read Clustering

In this section we show how we enhance CliqueSNV to more accurately recover intrahost haplotypes from Illumina sequencing data. Since the MC clustering requires full-length genomes, we fill all uncovered regions with dashes, see Figure (1). Here each dash represents a single undetermined nucleotide. If the paired reads overlap, then we also replace the mismatched sites within the overlap with dashes, Figure (1b).

CliqueSNV outputs a set of haplotypes with their frequencies. The *CliqueSNV clustering* is obtained by assigning each read to the closest haplotype with the highest frequency, i.e., if a read is within the same minimal Hamming distance

to multiple haplotypes, it is assigned to the most frequent one. The updated haplotypes are consensuses of the resulting clusters.

We initialize entropy and Hamming distance clusterings with the CliqueSNV clustering and update them using the Monte Carlo algorithm.

3.5 Validation Metrics for Viral Population Inference

3.5.1 Distance to the closest predicted haplotype (DCPH).

DCPH measures the quality of true haplotype prediction. For each true haplotype, we report the Hamming distance, or the number of mismatches, to the closest predicted haplotype. For dataset D3 we reported two numbers (Table 4), and for dataset D4 we report five numbers (Table 5).

3.5.2 Earth mover's distance (EMD) between populations.

In order to simultaneously match haplotype sequences and their frequencies for true population T and predicted population P, we allowed for a fractional matching when portions of a single haplotype p of population P are matched to portions of possibly several haplotypes of T and vice versa. Thus, we partition the frequency of p, f_p, as $f_p = \sum_{t \in T} f_{pt}$, $f_{pt} \geq 0$, where each f_{pt} denotes the portion of p matched to t. Symmetrically, the frequency of each t, f_t, is also partitioned as $f_t = \sum_{p \in P} f_{pt}$. The matching error between haplotypes p and t is equal to the Hamming distance between them and is denoted d_{pt}. Finally, we choose the values of all f_{pt} as to minimize the total error of matching T to P. This is known as Wasserstein metric or the Earth Mover's Distance between T and P (Levina and Bickel, 2001; Mallows, 1972):

$$EMD(T,P) = \min_{f_{pt} \geq 0} \left[\sum_{t \in T} \sum_{p \in P} f_{pt} d_{pt} \right]$$

such that

$$\sum_{t \in T} f_{pt} = f_p \quad \text{and} \quad \sum_{p \in P} f_{pt} = f_t.$$

EMD is efficiently computed as an instance of the transportation problem using network flows.

Values of EMD can vary significantly over different benchmarks since they may have different complexities, which depend on the number of true variants, the frequency distribution, the similarity between haplotypes, sequencing depth, sequencing error rate, and many other parameters. Hence, we measured the complexity of a benchmark as the EMD between the true population and a population consisting of a single consensus haplotype (Yang et al., 2012).

4 Results

We ran an implementation of the proposed MC clustering method on the cluster hardware consisting of 128 cores Intel® Xeon® CPU E7-4850 v4 CPU @ 2.10GHz, with 3 TB of RAM, running Ubuntu 16.04.7 LTS.

4.1 Distinguishing Signal in Clustering

By minimizing entropy on the permuted data, we find that the method reduces entropy to 29.62, while on the original, unshuffled data the method reaches a much lower entropy of 24.77 (Table 2). This difference in the entropies of resulting clusterings, $29.62 - 24.77 = 4.85$, accounts for the amount of meaningful information, which is not noise, that our method was able to extract from the real data.

For the entropy and Hamming distance, we report the average, the minimum and the standard deviation σ achieved over 20 runs for the benchmark D1 and for the benchmark D3 (TF = 0.001, TF = 0.002).

Notice that the standard deviation σ for the original unpermuted data is significantly smaller than for the permuted data.

4.2 Stability of Monte Carlo Output

Table 3 shows the results of stability validation, in which we compare clustering similarity for various degrees of permutation of the input data.

The second column in Table 3 compares the resulting clusterings to the initial clustering. Without any permutations, the resultant clustering moves significantly further away from the initial one, giving a Rand index of 0.93. As the permutation degree increases, we observed that the clusterings produced by the Monte Carlo algorithm do not move as far away from the initial clustering; in other words, even after Monte Carlo was applied, the resulting clusterings had a high degree of agreement with the initial clustering.

The third column in Table 3 gives the average Rand index between multiple runs of Monte Carlo for a given permutation. We see that for all degrees of permutation, the method stably converges towards similar clusterings, with Rand index scores of 0.97–0.98. The same trends can be observed when using Hamming distance to cluster toward the consensus as the objective, as shown in the fifth column.

4.3 Monte Carlo Clustering Enhancement of CliqueSNV

Tables 4 and 5 compare performance of CliqueSNV with three clustering methods: CliqueSNV clustering, entropy MC clustering, and Hamming distance MC clustering. Both tables show the numbers of haplotypes or clusters, the values of the earth mover's distance (EMD) between the true and predicted haplotypes, and the distances to the closest predicted haplotype (DCPH) for each true haplotype.

For the benchmark D3 (see Table 4). for the both TF = 0.001 and TF = 0.002, the original CliqueSNV matches the second haplotype exactly and predicts the first haplotype with a single mismatch. While clustering methods lose to the original CliqueSNV for TF = 0.001, they match the quality of CliqueSNV for TF = 0.002.

For the benchmark D4, CliqueSNV produced 13 haplotypes for the threshold frequency of TF = 0.05 (Table 5). Here, Monte Carlo clusterings improve over both CliqueSNV and CliqueSNV clustering in matching true haplotypes (DCPH).

EMD to the true solution is smaller for CliqueSNV than for the clustering methods because CliqueSNV more accurately estimates hyplotype frequencies using the expectation-maximization method.

Figures (2), (3), and (4) show how the clustering entropy decreases with the number of Monte Carlo moves. Note that for larger datasets the entropy reduction is relatively smaller.

5 Conclusions

We have developed Monte Carlo methods for clustering sets of aligned viral genomic sequences. The methods are scalable to millions of sequences and is made even faster without significant loss of accuracy by picking a subset of tags with maximum entropy to represent the sequences. We have shown that both minimum entropy and minimum Hamming distance Monte Carlo clustering methods discern the meaningful information from sequencing data and that both clustering methods consistently converge to similar clusterings across different runs. Finally, we have shown that Monte Carlo clusterings achieve more accurate reconstruction of intrahost viral populations.

6 Acknowledgements

None.

7 Author Contributions

Akshay Juyal: Methodology (lead), software (lead), formal analysis (lead), writing - original draft (supporting), writing - review and editing (supporting). **Roya Hosseini:** Methodology (lead), software (supporting), formal analysis (lead), writing - original draft (lead), writing - review and editing (supporting). **Daniel Novikov:** Methodology (supporting), writing - original draft (supporting). **Mark Grinshpon:** Formal analysis (supporting), writing - original draft (supporting), writing - review and editing (lead). **Alex Zelikovsky:** Conceptualization (lead), methodology (supporting), writing - original draft (supporting), writing - review and editing (lead), supervision (lead).

8 Author Disclosure Statement

The authors declare they have no conflicting financial interests.

9 Funding Statement

The work at Georgia State University (GSU) was partially supported by NIH grant 1R21CA241044-01A1, NSF grant IIS-2212508, by the GSU Molecular Basis of Disease Fellowship, and by the GSU Brain and Behavior Fellowship.

10 References

References

Beerenwinkel, N., Sing, T., Lengauer, T., et al. Computational methods for the design of effective therapies against drug resistant HIV strains. *Bioinformatics*, 21(21):3943–3950, September 2005. doi: 10.1093/bioinformatics/bti654.

Bousali, M., Dimadi, A., Kostaki, E.-G., et al. SARS-CoV-2 molecular transmission clusters and containment measures in ten european regions during the first pandemic wave. *Life*, 11(3), 2021. doi: 10.3390/life11030219.

Campo, D. S., Skums, P., Dimitrova, Z., et al. Drug resistance of a viral population and its individual intrahost variants during the first 48 hours of therapy. *Clinical Pharmacology and Therapeutics*, 95(6):627–635, June 2014. doi: 10.1038/clpt.2014.20.

Campo, D. S., Xia, G.-L., Dimitrova, Z., et al. Accurate genetic detection of hepatitis C virus transmissions in outbreak settings. *The Journal of Infectious Diseases*, 213(6):957–965, March 2016. doi: 10.1093/infdis/jiv542.

de la Vega, W. F., Karpinski, M., Kenyon, C., and Rabani, Y. Approximation schemes for clustering problems. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 50–58. Association for Computing Machinery, 2003. doi: 10.1145/780542.780550.

Douek, D. C., Kwong, P. D., and Nabel, G. J. The rational design of an AIDS vaccine. *Cell*, 124(4):677–681, 2006. doi: 10.1016/j.cell.2006.02.005.

EMBL-EBI. EMBL's European Bioinformatics Institute, 2023. Available from: https://www.ebi.ac.uk/. [Last accessed: May 17, 2023].

Gaschen, B., Taylor, J., Yusim, K., et al. Diversity considerations in HIV-1 vaccine selection. *Science*, 296(5577):2354–2360, 2002. doi: 10.1126/science.1070441.

Glebova, O., Knyazev, S., Melnyk, A., et al. Inference of genetic relatedness between viral quasispecies from sequencing data. *BMC Genomics*, 18(Suppl 10), 2017. doi: 10.1186/s12864-017-4274-5.

Holland, J. J., Torre, J. C. D. L., and Steinhauer, D. A. RNA virus populations as quasispecies. *Current Topics in Microbiology and Immunology*, 176:1–20, 1992. doi: 10.1007/978-3-642-77011-1_1.

Juyal, A., Hosseini, R., Novikov, D., Grinshpon, M., and Zelikovsky, A. Entropy based clustering of viral sequences. In Bansal, M. S., Cai, Z., and Mangul, S., editors, *Bioinformatics Research and Applications. ISBRA 2022. Lecture Notes in Computer Science*, volume 13760, pages 369–380. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-23198-8_33.

Khare, S., Gurry, C., Freitas, L., et al. GISAID's role in pandemic response. *China CDC weekly*, 3(49):1049—1051, 2021. doi: 10.46234/ccdcw2021.255.

Knyazev, S., Tsyvina, V., Shankar, A., et al. Accurate assembly of minority viral haplotypes from next-generation sequencing through efficient noise reduction. *Nucleic Acids Research*, 49(17):e102–e102, July 2021. doi: 10.1093/nar/gkab576.

Levina, E. and Bickel, P. The earth mover's distance is the Mallows distance: Some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 251–256. IEEE, 2001. doi: 10.1109/ICCV.2001.937632.

Li, T., Ma, S., and Ogihara, M. Entropy-based criterion in categorical clustering. In *Proceedingsof the Twenty-First International Conference on Machine Learning, ICML 2004*, volume 3, pages 536–543. Association for Computing Machinery, 2004. doi: 10.1145/1015330.1015404.

Mallows, C. L. A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, 43(2):508–515, 1972. doi: 10.1214/aoms/1177692631.

Melnyk, A., Knyazev, S., Vannberg, F., et al. Using earth mover's distance for viral outbreak investigations. *BMC Genomics*, 21(582), 2020. doi: 10.1186/s12864-020-06982-4.

Melnyk, A., Mohebbi, F., Knyazev, S., et al. From Alpha to Zeta: Identifying variants and subtypes of SARS-CoV-2 via clustering. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, 28(11):1113–1129, 2021. doi: 10.1089/cmb.2021.0302.

Rhee, S.-Y., Liu, T. F., Holmes, S. P., et al. HIV-1 subtype B protease and reverse transcriptase amino acid covariation. *PLOS Computational Biology*, 3(5):1–8, May 2007. doi: 10.1371/journal.pcbi.0030087.

Skums, P., Bunimovich, L., and Khudyakov, Y. Antigenic cooperation among intrahost HCV variants organized into a complex network of cross-immunoreactivity. *Proceedings of the National Academy of Sciences*, 112(21):6653–6658, 2015. doi: 10.1073/pnas.1422942112.

Skums, P., Zelikovsky, A., Singh, P., et al. QUENTIN: Reconstruction of disease transmissions from viral quasispecies genomic data. *Bioinformatics*, 34(1):163–170, June 2017. doi: 10.1093/bioinformatics/btx402.

Yang, X., Charlebois, P., Gnerre, S., et al. De novo assembly of highly di-

verse viral populations. *BMC genomics*, 13:1–13, 2012. doi: 10.1186/1471-2164-13-475.

11 Tables

Table 1: Four sequencing datasets of SARS-CoV-2 and HIV.

Dataset	Data type	Virus	Number of input sequences	Type of input sequences	Number of true sequences or clusters
D1	Interhost	SARS-CoV-2	3688	Full genomes	28
D2	Interhost	SARS-CoV-2	148000	Full genomes	15
D3	Intrahost	HIV	49988	Illumina reads	2
D4	Intrahost	HIV	711228	Illumina reads	5

Table 2: Results after running Monte Carlo clustering optimizing entropy and Hamming distance over three datasets.

Benchmarks			**D1**	**D3, TF=0.001**	**D3, TF=0.002**
Entropy	Initial		31.524	846.19	875.22
	Unpermuted	Avg	24.77	616.56	708.72
		Min	24.7	599.16	708.72
		σ	0.049	12.28	8.28
	Permuted	Avg	29.65	779.3	860.54
		Min	28.6	701.23	839.07
		σ	0.7	55.2	15.18
Entropy reduction over permuted data			3.95	102.05	130.35
Hamming distance	Initial		1008.41	59165.26	176293.27
	Unpermuted	Avg	373.14	34935.99	96535.57
		Min	369.14	34815.03	93919.13
		σ	2.82	85.53	1850.1
	Permuted	Avg	770.39	44719.8	135549.43
		Min	689.97	43675.66	124817.3
		σ	56.56	738.31	7588.76
Distance reduction over permuted data			320.83	8860.63	30898.17

Table 3: Clustering similarity (Rand index) across three choices of the degree of permutation. The method was run three times for each permuted instance, each run consisting of 100000 Monte Carlo trials. Reported are average Rand index similarity of the resulting clusterings to the initial clustering, as well as between resulting clusterings.

% permutation	Cluster similarity (Rand index)			
	Entropy		Hamming distance	
	With original	With runs	With original	With runs
0	0.936476	0.970195	0.936114	0.970135
1	0.978898	0.979435	0.936126	0.970374
5	0.980458	0.980688	0.936180	0.970616

Table 4: Results for CliqueSNV with minimum frequencies 0.1% and 0.2% run on the benchmark D3, and for enhancements by CliqueSNV clustering, entropy MC clustering, and Hamming distance MC clustering. We report the number of haplotypes/clusters, earth mover's distance (EMD) to the true sequences, and distance from the closest predicted haplotype (DCPH).

Benchmark D3, TF = 0.001			
Method	**# haplotypes/clusters**	**EMD**	**DCPH**
CliqueSNV	45	3	$[1, 0]$
CliqueSNV clustering	45	19	$[5, 0]$
Entropy MC clustering	45	49	$[5, 21]$
Hamming distance MC clustering	45	7	$[5, 11]$
Benchmark D3, TF = 0.002			
Method	**# haplotypes/clusters**	**EMD**	**DCPH**
CliqueSNV	27	2	$[1, 0]$
CliqueSNV clustering	27	27	$[0, 3]$
Entropy MC clustering	27	61	$[0, 1]$
Hamming distance MC clustering	27	7	$[0, 1]$

Table 5: Results for CliqueSNV with minimum frequency 5% run on the benchmark D4, and for enhancements by CliqueSNV clustering, entropy MC clustering, and Hamming distance MC clustering. We report the number of haplotypes/clusters, earth mover's distance (EMD) to the true sequences, and distance from the closest predicted haplotype (DCPH).

Benchmark D4, TF = 0.05			
Method	**# haplotypes/clusters**	**EMD**	**DCPH**
CliqueSNV	13	243	$[3, 1, 9, 7, 1]$
CliqueSNV clustering	8	294	$[1, 3, 2, 4, 3]$
Entropy MC clustering	8	311	$[2, 0, 6, 3, 2]$
Hamming distance MC clustering	8	311	$[2, 0, 6, 3, 2]$

12 Figure Legends

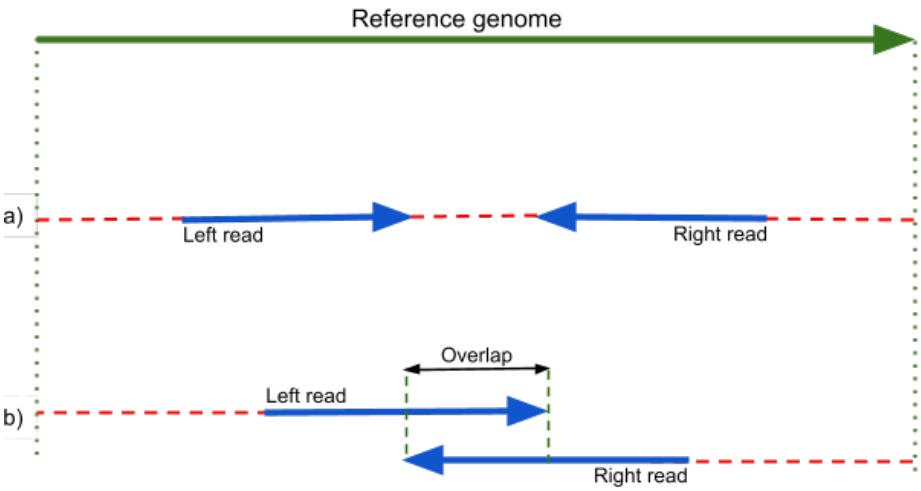

Figure 1: Illumina paired reads aligned to the full length reference genome. All uncovered position are filled with dashes (a). If the paired reads overlap, then any mismatched positions in the overlap are replaced with a dash (b).

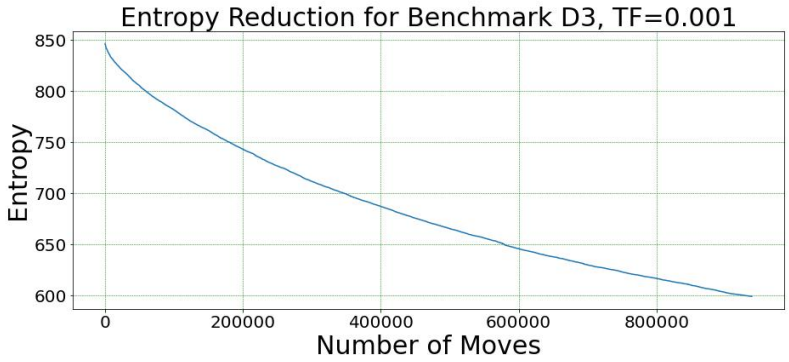

Figure 2: Entropy reduction for benchmark D3, with the threshold frequency of TF = 0.001.

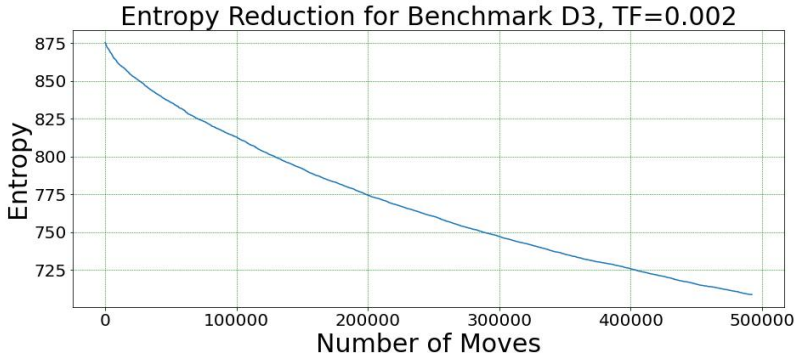

Figure 3: Entropy reduction for benchmark D3, with the threshold frequency of TF = 0.002.

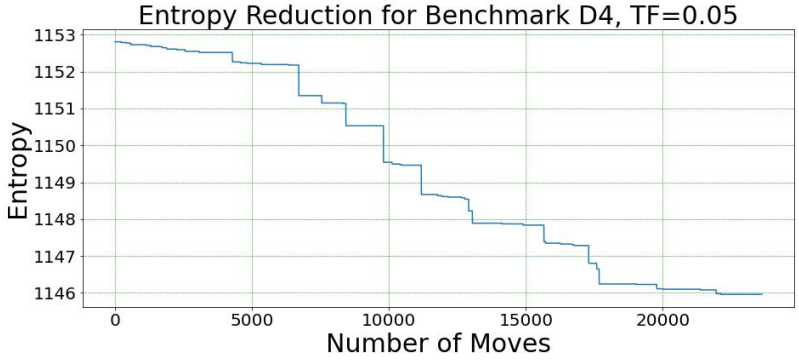

Figure 4: Entropy reduction for benchmark D4, with the threshold frequency of TF = 0.05. Entropy decreased by approximately 0.6%.