Vector Quantized Spectral Clustering Applied to Whole Genome Sequences of Plants

Evolutionary Bioinformatics
Volume 15: 1–7

DOI: 10.1177/1176934319836997

Aditya A Shastri[1], Kapil Ahuja[1], Milind B Ratnaparkhe[2], Aditya Shah[1], Aishwary Gagrani[1] and Anant Lal[1]

[1]Computer Science and Engineering, Indian Institute of Technology Indore, Indore, India.
[2]ICAR-Indian Institute of Soybean Research, Indore, India.

ABSTRACT: We develop a Vector Quantized Spectral Clustering (VQSC) algorithm that is a combination of spectral clustering (SC) and vector quantization (VQ) sampling for grouping genome sequences of plants. The inspiration here is to use SC for its accuracy and VQ to make the algorithm computationally cheap (the complexity of SC is cubic in terms of the input size). Although the combination of SC and VQ is not new, the novelty of our work is in developing the crucial similarity matrix in SC as well as use of *k*-medoids in VQ, both adapted for the plant genome data. For Soybean, we compare our approach with commonly used techniques like Un-weighted Pair Graph Method with Arithmetic mean (UPGMA) and Neighbor Joining (NJ). Experimental results show that our VQSC outperforms both these techniques significantly in terms of cluster quality (average improvement of 21% over UPGMA and 24% over NJ) as well as time complexity (order of magnitude faster than both UPGMA and NJ).

KEYWORDS: spectral clustering, similarity matrix, alignment score, sampling, vector quantization, *k*-medoids, whole genome sequence (WGS), single nucleotide polymorphism (SNP).

RECEIVED: September 30, 2018. **ACCEPTED:** February 7, 2019.

TYPE: Rapid Communication

FUNDING: The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The first author would like to duly acknowledge the funding from Ministry of Electronics and Information Technology (MeitY), India under the Visvesvaraya PhD Scheme for Electronics & IT. The second author would like to acknowledge the funding from MATRICS Scheme of Department of Science and Technology (DST-SERB), India with project number MTR/2017/001023.

DECLARATION OF CONFLICTING INTERESTS: The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

CORRESPONDING AUTHOR: Kapil Ahuja, Computer Science and Engineering, Indian Institute of Technology Indore, Khandwa Road, Simrol, Indore 453552, Madhya Pradesh, India. Email: kapsahuja22@gmail.com

Introduction

Clustering is one of the most widely used techniques for data analysis having applications in almost every field like statistics, computer science, biology, social sciences, psychology, etc. People attempt to get a first impression of their data by trying to identify groups having similar behavior. Finding tight clusters, ie, well separated and compact, is very important. Commonly used clustering algorithms include *k*-means, Partition Around Medoids (PAM), Clustering LARge Applications (CLARA), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Density Based Spatial Clustering of Applications with Noise (DBSCAN), Wave-Cluster, and Expectation-Maximization (EM).[1] Compared with these traditional algorithms, a promising alternative is to use spectral methods for clustering.

Clustering algorithms that use spectral properties are widely used because of their accuracy (we get more tight clusters) and easy implementation (these algorithms can be solved efficiently by using standard linear algebra methods).[2] However, when the input data are very large, they become inefficient; computational complexity of $O(n^3)$, where n is the size of the input data. Hence, considerable research has been done to reduce this complexity without affecting the accuracy of the underlying algorithm.

One such method is sampling that can reduce the input size. Samples should be selected in a manner such that they represent the whole dataset uniformly. Many techniques exist for sampling like random sampling, stratified sampling, matrix factorization, vector quantization (VQ), pivotal sampling, the strip method, the mean method, the second derivative method, etc.[3,4] Among these, VQ[5] is commonly used and is easy to implement because it provides the reduced data in a single scan of elements.

Clustering of whole genome sequences (WGSs)—a sequence made from a combination of 4 nucleotides: A (Adenine), T (Thymine), G (Guanine), and C (Cytosine)[6]—is useful in developing better species of plants, eg, disease resistant and drought resistant. Here, the traditional methods for clustering, eg, Un-weighted Pair Graph Method with Arithmetic mean (UPGMA)[7] and Neighbor Joining (NJ),[7] which are currently used by plant biologists, do not provide the level of accuracy needed and are also not the most efficient methods because of their high computational complexity ($O(n^3)$).

In this article, we use the spectral clustering (SC) algorithm (for accuracy) along with VQ (for efficiency) for clustering single nucleotide polymorphism (SNP—the variation in the nucleotide that occurs at a specific position across sequences) data obtained from the WGSs of plants. Although this combination of SC and VQ is not new,[5] the novelty of our work is using the 2 for clustering SNP data.

Next, we present literature regarding usage of SC and VQ in the field of plant genome, and the novelty of our approach. Spectral clustering can be performed in 2 ways: recursive and non-recursive. Bouaziz et al[8] in 2012 used this method in a recursive way for genetic studies. However, we use a common non-recursive way,[2,9] because it is simpler and cheaper. It also gives tight and compact clusters.

The construction of the similarity matrix is the most important part of the SC algorithm. This can be done either by using basic techniques[10-13] like cosine similarity, pairwise distance, Jukes Cantor, and alignment score, or by using

advanced techniques[14] like identity-by-state, allele sharing distance, SNP edit distance, covariance, normalized covariance, and coancestry.

Li et al[15] in 2010 used SC for clustering gene sequences (which are a subset of WGSs) where they constructed the similarity matrix by cosine similarity. We use the earlier mentioned basic techniques besides cosine similarity because they capture the similarity between the SNP sequences in a better way. We do not use advanced techniques because they are more involved (and also not needed since basic work well).

Zhang et al[16] in 2011 used VQ to reduce the number of genome sequences of influenza A virus for better visualization of phylogenetic trees, which are an essential step in earlier mentioned clustering algorithms of UPGMA and NJ. They used the neural gas method as the basis of their sampling.

We use VQ as well, but in a different sense. We use *k*-medoids as the basis of our sampling instead of the neural gas method. This is because it is easy to find the medoids of the kind of data we have.

In the next section, we describe our Vector Quantized Spectral Clustering (VQSC) algorithm in detail. In the subsequent section ("Discussion" section), we test our algorithm on SNP sequences obtained from a standard plant database (Soybean). Here, we also compare our results with currently used methods of clustering SNP data (mentioned above). Experiments show that VQSC performs better than these 2 popular existing techniques in terms of cluster quality (average improvement of 21% over UPGMA and 24% over NJ) as well as time complexity (order of magnitude faster than both UPGMA and NJ) Further, we also discuss application of our technique to other plants, e.g., Wheat, Rice, Maize etc.

The VQSC Algorithm

The SC algorithm uses the concept of similarity graph to construct the similarity matrix (or the weighted adjacency matrix) that in turn is used to construct the Laplacian matrix (either normalized or non-normalized).[9] Then, the eigenvectors corresponding to the first *k* smallest eigenvalues (where *k* is the number of clusters to be formed) of the Laplacian matrix are used to cluster the data.

As mentioned earlier, construction of the similarity matrix is significant in this algorithm because better the quality of this matrix, better is the accuracy of the SC algorithm. The Laplacian matrix obtained from the above-mentioned similarity matrix is also important because the eigenvectors of this matrix are used for clustering. A detailed description of the similarity matrix, the Laplacian matrix, and the SC algorithm is given by Ulrike von Luxburg,[9] Binkiewicz et al,[17] and Arias-Castro et al.[18]

In this article, for constructing the similarity matrix, we compare every character in one SNP sequence with every character in other SNP sequences. This represents how much one sequence is different from another sequence. The dissimilarity $D(i, j)$ between any 2 SNP sequences X_i and X_j is defined as the number of positions at which X_i and X_j differ. The similarity value is calculated as

$$S(i, j) = l(seq) - D(i, j) \quad (1)$$

where $l(seq)$ is the length of the SNP sequence. This value is normalized and used as the similarity value for (i, j) index. We also use other similarity measures like pairwise distance,[11] Jukes Cantor,[12] and alignment score[13] to construct the similarity matrix. Results show that the quality of clusters is sensitive to the quality of the similarity matrix used.

As mentioned earlier, we use VQ to compress the original data into a small set of representative data entities. The goal now is to minimize the difference between the original and this representative set.

Although the standard VQ algorithm uses *k*-means, we achieve this minimized difference by using the *k*-medoids algorithm. This is because, as discussed earlier, data here are in the form of sequences of strings of A, T, G, and C characters and mean of these data does not exist. On the other hand, *k*-medoids provide us with representative sequences from the set of given sequences itself.

Following is the algorithm for our VQSC:

Input: n SNP sequences $\{x_i\}$ for $i = 1, \ldots, n$; k number of representative sequences to be selected; and m number of clusters to be formed.

Output: clustered SNP sequences.

1. Perform *k*-medoids as follows:
   (a) Compute medoids $y_1, \ldots, y_k$ as the k representative sequences.
   (b) Build a correspondence table to associate each x_i with the nearest medoid y_j.
2. Run the SC algorithm on $y_1, \ldots, y_k$ to obtain cluster indexes C_l; $l = 1, \ldots, m$ for each of y_j.
3. Recover the cluster membership for each x_i by looking up the correspondence table.

Discussion

We use SNP data of 31 Soybean sequences, which are taken from the database as follows:[19] http://chibba.pgml.uga.edu/snphylo/. These data contain 6 289 747 SNPs. As this is a raw data, we use SNPhylo software[19] to remove low-quality data. Specifically, false SNPs are removed and we get 31 SNP sequences each of length 4847. (This software also constructs a phylogenetic tree as used by other standard genome clustering algorithms.) Please refer to Figure 1 of Lee et al,[19] which shows the flowchart of SNPhylo pipeline, which is a commonly used standard procedure. Finally, these sequences are used to obtain the similarities among each other leading to the construction of the similarity matrix, which is an input to our VQSC algorithm.

Here, we first discuss the computational complexity of our and other standard algorithms (for SNP clustering). Next, we describe the criteria used to check the goodness of generated clusters, termed as validation metrics. Furthermore, we give our

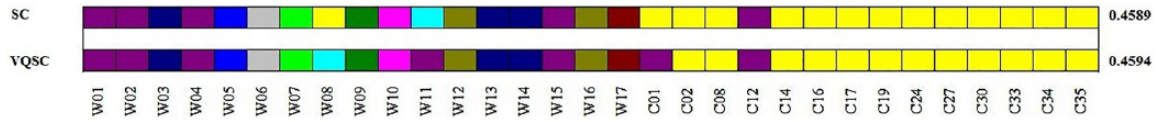

Figure 1. Cluster formation for SC and VQSC with alignment score and $m = 11$. SC indicates spectral clustering; VQSC, Vector Quantized Spectral Clustering.

results. Finally, we give concluding remarks and discuss future work.

Computational complexity

As mentioned in the "Introduction" section, complexities of the standard SC, UPGMA, and NJ algorithms are all $(O(n^3))$, where n is the size of the input data. This makes these algorithms computationally less efficient. However, the use of VQ sampling with SC reduces the complexity of VQSC to $O(k^3 + n^2 kt)$, where k is the number of representative samples chosen via k-medoids in VQ, and t is the number of iterations taken by VQ. Here, the first term (k^3) comes from SC, and the second term $(n^2 kt)$ comes from VQ. Application of VQ to UPGMA and NJ also leads to a comparable reduction in their complexity.

Validation metrics

There are various metrics available for validation of clustering algorithms. These include[1,20] Cluster Accuracy (CA), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), Compactness (CP), Separation (SP), Davis-Bouldin Index (DB), and Silhouette Value. For using the first 3 metrics, we should have prior knowledge of cluster labels. However, here we do not have ideal clustering results. Hence, we cannot use any of these validation metrics. Rest of the techniques do not have this requirement, and hence can be used for validation. We use Silhouette Value, which is usually used for validation of genome data.[21]

Silhouette Value is a measure of how similar an object is to its own cluster (intra-cluster similarity) compared with other clusters (inter-cluster similarity).[19] For any cluster $C_l (l = 1 \text{ to } m;\ \text{say } l = 1)$, let $a(i)$ be the average distance between the ith data point and all other points in cluster C_1, and let $b(i)$ be the average distance between the ith data point in cluster C_1 and all other points in clusters $C_l (l = 1 \text{ to } m \text{ and } l! = 1)$. Thus, Silhouette Value is given as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Here, $a(i)$ and $b(i)$ signify the intra-cluster and the inter-cluster similarities, respectively. Silhouette Value lies between −1 and 1, and average over all the data points is computed. A positive value indicates that the clusters are well separated from each other, and a negative value indicates that the clusters are overlapping.

Results

We first present the results of SC, UPGMA, and NJ without VQ. These data are given in Table 1. Column 1 gives the number of clusters chosen. As 2 to $n/2$ clusters, where n is the number of input data points, are commonly used in literature, we follow this. Hence, we provide results from 2 to 16 clusters (for us $n = 31$, and hence, $n/2 = 15.5 \approx 16$). Columns 2 to 5 refer to the Silhouette Values of the SC algorithm with 4 different similarity measures discussed earlier. Columns 6 and 7 give the Silhouette Values for UPGMA and NJ. As evident (highlighted in bold), SC with alignment score gives the best results for all the clusters.

The percentage improvement in SC (using alignment score as the similarity measure) in comparison with UPGMA and NJ is given in Table 2. We can observe from this table that the average improvement in SC over UPGMA is around 34% and over NJ is around 37%, which is considered to be a substantial improvement.

Next, we discuss the results for the same 3 clustering algorithms with VQ. Vector quantization can be performed in 2 ways: either we can reduce the length of each sequence or we can reduce the number of sequences. In this work, we reduce the number of sequences to reduce complexity. Results for these experiments are given in Table 3 (structure of which is similar to that of Table 2). From this table, we see a similar pattern, ie, our VQSC algorithm with alignment score is the best (highlighted in bold).

We also compare VQSC with Vector Quantized Un-weighted Pair Graph Method with Arithmetic mean (VQUPGMA) and Vector Quantized Neighbor Joining (VQNJ). The data for this is given in Table 4. Again, we observe substantial improvement by using VQSC. The average percentage improvement in VQSC over VQUPGMA is around 28% and over VQNJ it is around 347%.

Next, we calculate the loss of accuracy incurred because of sampling in the proposed SC algorithm (with alignment score as the similarity measure). For this, we compare the relevant SC and VQSC data from Tables 1 and 3, respectively. This loss for the different number of clusters chosen is listed in Table 5. We can observe from these data that the average of the loss of accuracy comes around 11%, which is considered acceptable because we are still better than the existing best algorithms (UPGMA and NJ; please see Table 7 and the accompanying discussion below).

We further validate the quality of these clusters using tools used by biologists at Indian Institute of Soybean Research. Here, we compare cluster formation for SC and VQSC for the

Table 1. Silhouette Values for different clustering algorithms without VQ.

NO. OF CLUSTERS	SC				UPGMA	NJ
	OUR SIMILARITY $S(I, J)$ FROM EQUATION (1)	PAIRWISE DISTANCE[11]	JUKES CANTOR[12]	ALIGNMENT SCORE[13]		
2	0.2012	0.2012	0.2590	**0.3169**	0.1831	0.2206
3	0.1987	0.1722	0.2440	**0.2845**	0.2002	0.2258
4	0.2053	0.2037	0.2621	**0.3241**	0.2546	0.2192
5	0.2488	0.2421	0.3017	**0.3528**	0.2791	0.2488
6	0.2771	0.2771	0.3214	**0.3886**	0.2389	0.2771
7	0.2990	0.3231	0.3414	**0.3882**	0.2612	0.2736
8	0.3451	0.3451	0.3811	**0.4007**	0.2906	0.2874
9	0.3490	0.3140	0.3785	**0.4130**	0.3112	0.3031
10	0.3522	0.3507	0.3771	**0.4464**	0.3430	0.2966
11	0.3687	0.3681	0.4045	**0.4589**	0.3831	0.3476
12	0.3799	0.4046	0.4258	**0.5031**	0.4089	0.3569
13	0.4329	0.3948	0.4611	**0.5375**	0.4153	0.3829
14	0.4470	0.4527	0.4646	**0.5415**	0.4610	0.4403
15	0.4481	0.4590	0.5093	**0.5701**	0.4881	0.4366
16	0.5014	0.5134	0.5301	**0.5917**	0.5139	0.4665

Abbreviations: NJ, Neighbor Joining; SC, spectral clustering; UPGMA, Un-weighted Pair Graph Method with Arithmetic mean; VQ, vector quantization.
Bold values indicate that SC with alignment score works best.

Table 2. Comparison of SC with UPGMA and NJ.

NO. OF CLUSTERS	PERCENTAGE IMPROVEMENT IN SC	
	OVER UPGMA	OVER NJ
2	73.07	43.65
3	42.11	26.00
4	27.30	47.86
5	26.41	41.80
6	62.66	40.24
7	48.62	41.89
8	37.89	39.42
9	32.71	36.26
10	30.15	50.51
11	19.79	32.02
12	23.04	40.96
13	29.42	40.38
14	17.46	22.98
15	16.80	30.58
16	15.14	26.84
Average	33.50	37.43

Abbreviations: NJ, Neighbor Joining; SC, spectral clustering; UPGMA, Un-weighted Pair Graph Method with Arithmetic mean.

different number of clusters. As above, we use the data corresponding to alignment score as the similarity measure because that gives the best results.

We do this comparison in 2 ways. For 2 cases ($m = 11$ and 12), we diagrammatically identify the sequences that are wrongly clustered by VQSC as compared with SC (in Figures 1 and 2). For all other values of m, we give the number of sequences wrongly clustered by VQSC as compared with SC (in Table 6). This 2-way strategy comprehensively depicts the goodness of VQSC without taking too much space.

In Figures 1 and 2, the x-axis lists the 31 sequences and the y-axis refers to the clustering algorithms used. The Silhouette Values from Tables 1 and 3 are given on the right. The different colors denote the different clusters, and the colored boxes signify which cluster each sequence belongs to. From Figure 1, we observe that our VQSC algorithm does not cluster sequences W08, W11, and C01 (ie, only 3 out of 31) in their respective clusters when compared with SC. Similar behavior can be observed from Figure 2. Sequences W05, C01, and C19 (again only 3 out of 31) are not correctly clustered by VQSC when compared with SC.

As evident from Table 6, on an average only 4 out of 31 (about 13%) sequences are wrongly clustered by VQSC as compared with SC. This is considered acceptable because, as earlier, we are still better than the existing best algorithms (please see Table 7 and the accompanying discussion below). (The outlier case of $m = 8$ needs further analysis and experimentation with more data.) To sum up, by using VQSC, we get

Table 3. Silhouette Values for different clustering algorithms with VQ.

NO. OF CLUSTERS	VQSC				VQUPGMA	VQNJ
	OUR SIMILARITY $S(I, J)$ FROM EQUATION (1)	PAIRWISE DISTANCE[11]	JUKES CANTOR[12]	ALIGNMENT SCORE[13]		
2	0.2012	0.2012	0.2590	**0.3169**	0.1835	0.0128
3	0.2002	0.2002	0.2474	**0.2876**	0.2002	0.0427
4	0.2159	0.2181	0.2610	**0.3052**	0.2192	0.0752
5	0.2211	0.2488	0.2887	**0.3232**	0.2488	0.0827
6	0.2639	0.2528	0.2922	**0.3046**	0.2532	0.0476
7	0.2446	0.2184	0.2867	**0.3259**	0.2604	0.0821
8	0.2727	0.2718	0.3189	**0.2935**	0.2752	0.1195
9	0.2861	0.3209	0.2890	**0.4004**	0.2886	0.1506
10	0.3361	0.2429	0.3561	**0.3726**	0.3264	0.1523
11	0.3035	0.2877	0.3672	**0.4594**	0.3456	0.2273
12	0.3299	0.3783	0.4078	**0.4743**	0.3650	0.2513
13	0.4268	0.4184	0.3811	**0.4843**	0.4216	0.3002
14	0.4128	0.4251	0.4450	**0.4966**	0.4111	0.3465
15	0.4560	0.4592	0.4796	**0.5334**	0.4592	0.3552
16	0.4552	0.4434	0.4587	**0.5004**	0.4434	0.4434

Abbreviations: VQ, vector quantization; VQNJ, Vector Quantized Neighbor Joining; VQSC, Vector Quantized Spectral Clustering; VQUPGMA, Vector Quantized Un-weighted Pair Graph Method with Arithmetic mean.
Bold values indicate that VQSC with alignment score works best.

Table 4. Comparison of VQSC with VQUPGMA and VQNJ.

NO. OF CLUSTERS	PERCENTAGE IMPROVEMENT IN VQSC	
	OVER VQUPGMA	OVER VQNJ
2	72.70	2375.78
3	43.66	573.54
4	39.23	305.85
5	29.90	290.81
6	20.30	539.92
7	25.15	296.95
8	6.65	145.61
9	38.74	165.87
10	14.15	144.65
11	32.93	102.11
12	29.95	88.74
13	14.87	61.33
14	20.80	43.32
15	16.16	50.17
16	12.86	12.86
Average	27.87	346.50

Abbreviation: VQNJ, Vector Quantized Neighbor Joining; VQSC, Vector Quantized Spectral Clustering; VQUPGMA, Vector Quantized Un-weighted Pair Graph Method with Arithmetic mean.

Table 5. Loss of accuracy because of sampling in SC.

NO. OF CLUSTERS	PERCENTAGE LOSS OF ACCURACY
2	0
3	+1.08
4	−6.19
5	−9.16
6	−27.58
7	−19.12
8	−36.52
9	−3.15
10	−19.81
11	−0.11
12	−6.07
13	−10.98
14	−9.04
15	−6.88
16	−18.25
Average	−11.45

Abbreviation: SC, spectral clustering.

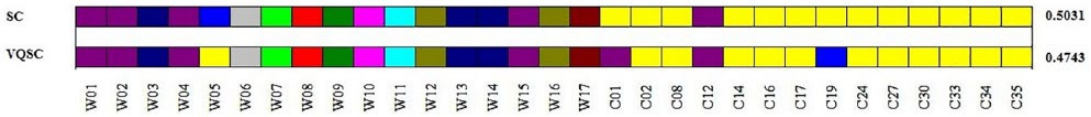

Figure 2. Cluster formation for SC and VQSC with alignment score and $m = 12$. SC indicates spectral clustering; VQSC, Vector Quantized Spectral Clustering.

Table 6. Wrongly clustered sequences by VQSC when compared with SC.

NO. OF CLUSTERS	NO. OF SEQUENCES WRONGLY CLUSTERED
2	0
3	1
4	4
5	4
6	4
7	4
8	12
9	2
10	5
11	3
12	3
13	5
14	6
15	4
16	4
Average	4.07

Abbreviations: SC, spectral clustering; VQSC, Vector Quantized Spectral Clustering.

Table 7. Comparison of VQSC with UPGMA and NJ.

NO. OF CLUSTERS	PERCENTAGE IMPROVEMENT IN VQSC	
	OVER UPGMA	OVER NJ
2	73.07	43.65
3	43.66	27.37
4	19.87	39.23
5	15.80	29.90
6	27.50	9.92
7	24.77	19.12
8	1.00	2.12
9	28.66	32.10
10	8.63	25.62
11	19.92	32.16
12	15.99	32.89
13	16.61	26.48
14	7.72	12.79
15	9.28	22.17
16	-2.63	7.27
Average	20.66	24.19

Abbreviations: NJ, Neighbor Joining; UPGMA, Un-weighted Pair Graph Method with Arithmetic mean; VQSC, Vector Quantized Spectral Clustering.

almost the same cluster formation as SC, but at a reduced computational cost.

Finally, we compare results of our efficient and accurate algorithm (VQSC using alignment score) with the existing best (UPGMA and NJ). Results for this are given in Table 7. As evident from this table, our VQSC is on an average 21% more accurate than UPGMA and on an average 24% more accurate than NJ. As earlier, we also have the added benefit of reduced computational complexity for VQSC as compared with both UPGMA and NJ.

Concluding Remarks

We present the VQSC algorithm that is a combination of SC and VQ sampling for clustering genome sequences of plants. We use SC for its accurate clustering and VQ for its accurate sample selection. Use of this combination makes our algorithm scalable for large data as well. As building the similarity matrix is critical to the SC algorithm, we exhaustively adapt 4 ways to build such a matrix for plant genome data. Adapting VQ for these data requires using k-medoids instead of traditional k-means for finding representative samples. For a sample plant data (Soybean), we compare the performance of our VQSC algorithm with other traditional and commonly used techniques of UPGMA and NJ. VQSC outperforms both of these in terms of cluster quality (average improvement of 21% over UPGMA and 24% over NJ) and computational complexity (order of magnitude faster than both UPGMA and NJ).

Future Work

In the future, we plan to extend this work to more number of sequences.[22] As earlier, here we reduce the number of sequences by sampling. However, we could also sample across the length of every sequence. As the quality of the similarity matrix has a big impact on the quality of clusters, we also intend to adapt other ways of constructing this matrix as part of our future work.[14] Also, we plan to test our algorithm for other genome

sequences. For example, genome sequences of wheat,[23] rice,[24] and maize[25] are also made from the combination of nucleotides A, T, G, and C. The only difference between Soybean sequences and these sequences is the length of sequences and the numbers of SNPs present in them, and both these things do not affect our algorithm. In SC, we can easily obtain similarities between these new sequences by using any of the measures mentioned earlier. Vector quantization can also be applied to these sequences without any change because the main aspect of VQ, the *k*-medoids algorithm, is independent of the above-mentioned changes in sequences.

Author Contributions

The main idea behind this paper was conceived and worked out by AAS and KA. The two of them also put the manuscript together including editing the various versions. MBR provided substantial help in the biology aspect of the project including understanding the data, applying the proposed techniques to Soybean plant, and analyzing the results obtained. AS, AG, AL worked on this as part of their undergraduate degree project, and performed most of the coding. They also helped in overcoming the challenges associated with applying sampling to plant genome data.

ORCID iD

Kapil Ahuja https://orcid.org/0000-0001-9640-4437

REFERENCES

1. Fahad A, Alshatri N, Tari Z, et al. A survey of clustering algorithms for big data: taxonomy and empirical analysis. *IEEE Trans Emerg Topics Comput*. 2014;2:267–279.
2. Ng AY, Jordan MI, Weiss Y. On spectral clustering: analysis and an algorithm. *NIPS*. 2001;14:849–856.
3. Tillé Y. *Sampling Algorithms*. 1st ed. New York: Springer-Verlag; 2006.
4. Friedrich A, Ripp R, Garnier N, et al. Blast sampling for structural and functional analyses. *BMC Bioinformatics*. 2007;8:62.
5. Wang X, Zheng X, Qin F, Zhao B. A fast spectral clustering method based on growing vector quantization for large data sets. In: Motoda H, Wu Z, Cao L, eds. *Proceedings of International Conference on Advanced Data Mining and Applications*. Berlin: Springer-Verlag; 2013:25–33.
6. Schmutz J, Cannon SB, Schlueter J, et al. Genome sequence of the palaeopolyploid soybean. *Nature*. 2010;463:178–183.
7. Backeljau T, Bruyn LD, Wolf HD, Jordaens K, Van Dongen S, Winnepenninckx B. Multiple UPGMA and neighbor-joining trees and the performance of some computer packages. *Mol Biol Evol*. 1996;13:309–313.
8. Bouaziz M, Paccard C, Guedj M, Ambroise C. SHIPS: spectral hierarchical clustering for the inference of population structure in genetic studies. *PLoS ONE*. 2012;7:e45685.
9. Luxburg UV. A tutorial on spectral clustering. *Stat Comput*. 2007;17:395–416.
10. Ye J. Cosine similarity measures for intuitionistic fuzzy sets and their applications. *Math Comput Model*. 2011;53:91–97.
11. Felsenstein J. An alternating least squares approach to inferring phylogenies from pairwise distances. *Syst Biol*. 1997;46:101–111.
12. Jukes TH, Cantor CR, Munro HN. Evolution of protein molecules. *Mamm Protein Metabol*. 1969; 3:132.
13. Betel D, Wilson M, Gabow A, Marks DS, Sander C. The microRNA.org resource: targets and expression. *Nucleic Acids Res*. 2008;36:D149–D153.
14. Lawson DJ, Falush D. Similarity matrices and clustering algorithms for population identification using genetic data. *Annu Rev Genomics Hum Genet*. 2012;13:337–361.
15. Li L, Shiga M, Ching WK, Mamitsuka H. Annotating gene functions with integrative spectral clustering on microarray expressions and sequences. *Genome Inform*. 2010;22:95–120.
16. Zhang J, Mamlouk AM, Martinetz T, Chang S, Wang J, Hilgenfeld R. PhyloMap: an algorithm for visualizing relationships of large sequence data sets and its application to the influenza A virus genome. *BMC Bioinformatics*. 2011;12:248.
17. Binkiewicz N, Vogelstein JT, Rohe K. Covariate-assisted spectral clustering. *Biometrika*. 2017;104:361–377.
18. Arias-Castro E, Lerman G, Zhang T. Spectral clustering based on local PCA. *J Mach Learn Res*. 2017;18:253–309.
19. Lee TH, Guo H, Wang X, Kim C, Paterson AH. SNPhylo: a pipeline to construct a phylogenetic tree from huge SNP data. *BMC Genomics*. 2014;15:162.
20. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math*. 1987;20:53–65.
21. Bolshakova N, Azuaje F. Cluster validation techniques for genome expression data. *Signal Process*. 2003;83:825–833.
22. Zhou Z, Jiang Y, Wang Z, et al. Resequencing 302 wild and cultivated accessions identifies genes related to domestication and improvement in soybean. *Nat Biotechnol*. 2015;33:408–414.
23. Brenchley R, Spannagl M, Pfeifer M, et al. Analysis of the bread wheat genome using whole-genome shotgun sequencing. *Nature*. 2012;491:705–710.
24. Goff SA, Ricke D, Lan TH, et al. A draft sequence of the Rice genome (Oryza sativa L. ssp. japonica). *Science*. 2002;296:92–100.
25. Schnable PS, Ware D, Fulton RS, et al. The B73 maize genome: complexity, diversity, and dynamics. *Science*. 2009;326:1112–1115.