Descriptive Statistics of the Genome: 
Phylogenetic Classification of Viruses

Troy Hernandez∗† troy@math.tsinghua.edu.cn
Jie Yang† jyang06@uic.edu

Abstract
In recent years there have been many proposed methods for phylogenetic classification of viruses. Performing these classifications in a timely manner is of interest to researchers and public health authorities. While multiple sequence alignment (MSA) remains the tool of choice for practitioners, alignment-free methods have gained popularity due to their substantial increases in speed and decreases in computational resources. Pairwise sequence alignment (PASC) is a middle-ground between the preceding two.

We present a new family of alignment-free vectorizations of the genome, the generalized vector (GV), that maintains the speed of existing alignment-free methods and incorporates some of the interpretability of sequence alignment. This new alignment-free vectorization uses the frequency of genomic words (k-mers), as is done in the composition vector, and incorporates descriptive statistics of those k-mers’ positional information, as inspired by the natural vector.

We analyze 5 different characterizations of genome similarity using k-nearest neighbor classification, and evaluate these on two collections of viruses totalling over 10,000 viruses. We show that our proposed method performs better than, or at least as well as, other methods at every level of the phylogenetic hierarchy.

The data and R code is available upon request.

1 Introduction

At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.

−Paul Domingos (Domingos, 2012)

The proliferation of low-cost, high-speed genomic sequencing technology has and will continue to give the scientific community ever-increasing amounts of genomic data. Experts will no longer have the ability to manually classify this torrent of biological data. Automated virus classification systems have begun appearing in the past few years to assist experts and practitioners (Bao, 2012; Rosen et al., 2012; Yu et al., 2012). These classification systems rely broadly on two different measures of similarity between the genome; sequence alignment identity and alignment-free vectorizations.

∗Mathematical Sciences Center, Tsinghua University, 131 Jin Chan Yuan West Building, Haidian, Beijing, CN 100084.
†Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 322 Science and Engineering Offices, 851 S. Morgan Street, Chicago, IL 60607-7045.
Virus classification by pairwise sequence comparison (Bao et al., 2008) relies on the sequence alignment identity between every pair of viruses. For reasons of computational complexity, all pairs of viruses are aligned instead of all viruses being aligned at once as in multiple sequence alignment (MSA). MSA (i.e., aligning entire groups of genomic sequences at once) has a computational complexity of $O(n^m)$, where $n$ is the length of a viral sequence and $m$ is the number of viruses being compared. For this reason, all of the pairwise identities of viruses in a given family are pre-computed. After the pairwise identities of a new virus are calculated a histogram of the identity scores are displayed, color-coded according to their sub-family, genus, and species. From there, an expert uses their best judgement to determine the proper subfamily, genus, and species classifications.

In the alignment-free/vectorization approach, statistics of each genome are compiled, stored, and new viruses are then classified according to various learning algorithms. The bulk of the literature in alignment-free methods relies on the bag-of-words model, also known as k-mers within the bioinformatics community. K-mers are genomic words from the alphabet $\{A, C, G, T\}$ of length $k$; e.g., for $k=3$, “AGC”, “CTA”, and “TAG”. For a given $k$, a vector of k-mer frequencies can be used within learning algorithms for clustering or classification (Vinga and Almeida, 2003).

Another alignment-free approach is the natural vector (Deng et al., 2011). The natural vector characterizes the distribution of a genome’s nucleotides. That characterization consists of the counts of $A, C, G, T$ in addition to positional information. That is, the mean position of the nucleotides and their central moments; i.e., the 2nd, 3rd, 4th, etc. central moments.

In this paper we extend the idea of incorporating information about the distribution of k-mer positions to a genomic vectorization. The primary contributions are as follows:

- Characterizing k-mer positional distribution information in a vector via the proposed generalized vector (GV).
- Analysis of 5 different characterizations of genome similarity; the composition vector (CV), the complete composition vector (CCV), the natural vector (NV), pairwise sequence alignment (PASC), and GV.
- Comparative evaluation of the two collections of viruses families genera mentioned above totaling over 10,000 unique viruses.

In section 3 we describe the source, curation, and details of the data in addition to the algorithm, the implementation details, and the expectations for performance on each method. We evaluate the different methods in section 4 and conclude in section 5.

2 Methods

2.1 Related Work

In this section we describe various methods used in the literature to quantify similarity in genomes. Three of the methods are alignment-free; i.e., they use statistics collected from a genome as components in a vector. Those vectors are then used within learning algorithms for clustering or classification. Alternatively, MSA and PASC aligns genomes and measures similarity directly from those alignments. In section 3 the algorithms and pre-processing used to implement the classifications are described. This will affect the measures of similarity differently for the different representations.
2.1.1 Sequence Alignment

Sequence alignment is a method of aligning the nucleotides within genomes so that functional or structural regions of similarity can be observed and measured. In locations where exact nucleotide matches are not possible, substitutions are made or gaps are inserted. Different alignment procedures are used for different purposes. Global alignment tries to align two entire genomes, whereas local alignment tries to align only regions suspected of having conserved regions. Different scores for nucleotide identity and different penalties for gaps and substitutions will also lead to different measures of similarity between genomes. Additionally, aligning multiple sequences at once can lead to different similarity measures between two genomes if different collections of sequences are used. A more thorough explanation of these issues can be found in [Waterman, 1995].

What is most important, with regards to alignment, is the computational complexity of MSA. Given a collection of \(m\) sequences of length \(n\) the complexity is \(O(n^m)\). For the applications used by today’s public health professionals, this can be prohibitive. PASC gets around this by aligning every pair of sequences and uses those pairwise scores for a similarity matrix. From the PASC website: “PASC is a web tool for analysis of pairwise identity distribution within viral families. The identities are pre-computed for every pair within the families and with distribution plotted in a form of histogram where each bar corresponds to an interval of identities.” [Bao, 2012]

2.1.2 K-mers

The bag-of-words model is ubiquitous in natural language processing [Lewis, 1998]. In this model a text document is converted into a vector where each component represents a word. This conversion results in the loss of grammar and word order information. Despite this loss of information, it is effective.

Within bioinformatics, the bag-of-words model has been adapted to work on genomes instead of text documents. The ‘words’ in this case are sub-strings of nucleotides in the genome. Sub-strings of length \(k\), known as \(k\)-mers, can be of length 1 to \(n\) for a given sequence of length \(n\). These \(k\)-mers are extracted from the sequence by sliding a window of length \(k\) over the genome from the \(1^{st}\) position to the \((n-k+1)^{st}\) position. For example, in the string \(S=\text{GATTACA}\) there are 6 non-zero \(k\)-mers (\(k=2\)):

\[
n_{AC} = 1, n_{AT} = 1, n_{CA} = 1, n_{GA} = 1, n_{TA} = 1, n_{TT} = 1
\]

This results in a vector of counts:

\[
n_2 = (n_{AA}, n_{AC}, n_{AG}, n_{AT}, n_{CA}, n_{CG}, n_{CT}, n_{GA}, n_{GC}, n_{GG}, n_{GT}, n_{TA}, n_{TC}, n_{TG}, n_{TT})
\]

\[
= (0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1)
\]

Typically, by dividing by \(l-k+1\) these \(k\)-mer counts are converted to frequency vectors, \(f_k\). Due to the 4 letter nucleotide alphabet, for a given \(k\), there are \(4^k\) components in the \(k\)-mer frequency vector. For example:

\[
f_2 = (0, \frac{1}{6}, 0, \frac{1}{6}, \frac{1}{6}, 0, 0, 0, \frac{1}{6}, 0, 0, \frac{1}{6}, 0, 0, \frac{1}{6})
\]
2.1.3 Composition Vector

It has been shown that classification using k-mers can be improved by using some informed scale and location shifts of the frequency vector \cite{Bailin2003}. This is known as the composition vector (CV). There are many different proposed parameters for the scale and location shifts. Here we focus on a Markov Model as described in \cite{Chan2010}.

For a k-mer \( u \), we estimate its expected frequency using its two component \( k-1 \) length words. As an example, let \( u = LwR = GATTACA \). Where \( L = G \), \( w = ATTAC \), and \( R = A \). Following \cite{Chan2010}, we estimate its expected frequency:

\[
\begin{align*}
\mathcal{P}(LwR) &= \mathcal{P}(Lw)\mathcal{P}(R | Lw) \\
&\approx \mathcal{P}(Lw)\mathcal{P}(R | w) \\
&= \frac{\mathcal{P}(Lw)\mathcal{P}(wR)}{\mathcal{P}(w)}
\end{align*}
\]

To get the composition vector component for k-mer \( u \), \( c_u \), we use the frequency of \( u \), \( f_u \), and its expected frequency, \( \mathcal{P}_u \):

\[c_u = \frac{f_u - \mathcal{P}_u}{\sqrt{\mathcal{P}_u}}\]

For a given \( k \) this results in the composition vector:

\[c_k = (c_{u1}, \ldots, c_{u_{k}})\]

2.1.4 Complete Composition Vector

The complete composition vector (CCV) takes the composition vector for various values of \( k \), \( c_k \), and concatenates them \cite{Wu2004}. This produces the CCV:

\[v_k = (c_1, \ldots, c_k)\]

For the CV and a fixed \( k \), using the values without any transformations is sufficient. When using the CCV another transformation seems necessary for the following reason: When concatenating the CVs of a genome from \( k = 1 \ldots 5 \), assuming the CVs for the various \( k \)s have the same distribution, the vector will have 4 components from \( c_1 \) and \( 4^5 = 1024 \) components from \( c_5 \). This makes the contribution of \( c_1 \) negligible to the distances computed. For this reason, as shown in section 3.6 we suggest a transformation informed by the data.

2.1.5 Natural Vector

K-mers and the composition vector throw out all location information for the nucleotides, the natural vector does not. The natural vector characterization of genomes \cite{Deng2011, Yu2013} consists of the counts, mean position, and central moments of the nucleotides A, C, G, and T.

For \( u = A, C, G, T \),

1. Let \( S = (s_1, s_2, \ldots, s_n) \) be a nucleotide sequence of length \( n \); i.e. \( s_i \in \{A, C, G, T\} \) for \( i = 1, 2, \ldots, n \).
(2) Let $n_u$ denote the number of letter $u$ in $S$ and $n$ denote the length of $S$, such that $\sum_u n_u = n$

(3) Let $s_u[i]$ denote the position of the $i^{th}$ letter $u$, that is $s_u[1] < \cdots < s_u[n_u]$ and $S[s_u[i]] = u$, for $i = 1, \ldots, n_u$.

(4) Let the mean position of letter $u$ be

$$\mu_u = \frac{\sum_{i=1}^{n_u} s_u[i]}{n_u}$$

(5) For $j = 2, \ldots, n_u$, let

$$d_{u}^j = \sum_{i=1}^{n_u} \frac{(s_u[i] - \mu_u)^j}{n_u^{j-1} n^{j-1}}.$$

In theory, any number of central moments can be used. In practice, only the second central moment (i.e. $j = 2$) is used resulting in a 12-dimensional vector (Yu et al., 2013). This results in a vector:

$$(n_A, \mu_A, d_A^2, n_C, \mu_C, d_C^2, n_G, \mu_G, d_G^2, n_T, \mu_T, d_T^2)$$

2.2 Proposed Vectorization

Given the k-mer, composition vector, complete composition vector, and natural vector representations of the genome, we introduce the GV. Observing that the composition vector throws out the positional information of the genome and the natural vector retains this information, but only for k-mers of length 1, it becomes clear that a large space of descriptive statistics of the genome are being ignored. Rather than simply extending the natural vector definition to k-mers with values of $k$ greater than 1, we also make some adjustments. These adjustments are explained by the analysis below and improve results (not shown).

2.2.1 Coordinates of Natural Vector

Suppose $n$ is large enough. Let $s_u$ be a randomly chosen position for the letter $u$. Assume that $s_i$ follows an i.i.d. discrete distribution with 4 outcomes for $i = \{1, \ldots, n\}$ with proportions $(p_A, p_C, p_G, p_T)$, where $0 < p_u < 1$, $u = A, C, G, T$, and $\sum_u p_u = 1$. Then approximately,

$$s_u - \mu_u \sim \text{Unif}(-1/2, 1/2)$$

and

$$\mu_u \sim \frac{n}{2}$$

and

$$d_{u}^j \sim \begin{cases} \frac{n}{2^{j+1} n_u} & \text{if } j = 2d \\ 0 & \text{if } j = 2d - 1 \end{cases}$$
because

\[
\frac{1}{n_u} \sum_{i=1}^{n_u} \frac{(s_u[i] - \mu_u)^j}{n_j} \sim \int_{-1/2}^{1/2} x^j dx
\]

\[
= \begin{cases} 
\frac{1}{2^j(j+1)} & \text{if } j = 2d \\
0 & \text{if } j = 2d - 1
\end{cases}
\]

Due to the term “\(n_j^{-2}\)” in (8), which is roughly \(n_j^{-2}d_u^{-2}\) for some constant \(c > 0\), \(d_u\) will be much less than \(n_u\) and \(\mu_u\) for large \(n\) and \(j > 2\). Therefore, the coordinates after the first 12 of the natural vector will be negligible when calculating the distances used to measure similarity.

### 2.2.2 Generalized Vector

In extending the natural vector to values of \(k\) greater than 1, we first replace counts of k-mers, \(n_u\), with their respective CVs, \(c_u\). The insight of the CV, which is especially important for the CCV, is that the frequencies of k-mers and (k-1)-mers are generally highly correlated (Wu et al., 2004). Additionally, we concatenate the collection of CVs, \(c_k\), resulting in \(v_k\) as defined in section 2.1.4.

Secondly, we add in the length \(n\). Experiments suggest that when trying to distinguish between different families of viruses, instead of just distinguishing between different species, the length of a genome is one of the most important factors. Additionally, adding genome length to the vector maintains the one-to-one nature of the representation.

Third, we use the standardized moments, \(\mu_j/\sigma_j\), where \(\mu_j\) represents the \(j^{th}\) moment about the mean and \(\sigma\) represents the standard deviation,

\[
\mu_j = \sum_{i=1}^{n_u} (s_u[i] - \mu_u)^j
\]

\[
\sigma = \sqrt{\sum_{i=1}^{n_u} (s_u[i] - \mu_u)^2}
\]

This is used instead of the scaled central moments that are used in the natural vector. In particular, \(j = 3\) is skewness and \(j = 4\) is kurtosis. The reasons for this being that (1) the central moments are highly correlated for odd and even moments, respectively, for the natural vector and (2) the scaling of the central moment by \(1/n_j\) makes it so that the higher order moments converge very quickly to 0. Lastly, similarly to CCV, we concatenate the vectors described above for various values of \(k\); e.g. \(k = 1 \ldots 5\). The generalized vector, \(g_k\), of a DNA sequence \(S\) is defined by

\[
(n, v_k, \mu_1, \ldots, \mu_k, \sigma_1^2, \ldots, \sigma_k^2, \mu_1^2/\sigma_1^2, \ldots, \mu_k^2/\sigma_k^2, \mu_1^3/\sigma_1^3, \ldots, \mu_k^3/\sigma_k^3)
\]

where \(\mu_j\) is the vector consisting of the \(j^{th}\) moment about the mean positions of all k-mers of length \(k\) and \(\sigma_k\) is the vector consisting of the standard deviations of the positions of all k-mers of length \(k\). That is,
\[ \mu_k^j = (\mu_{u_1}^j, \ldots, \mu_{u_k}^j) \]

\[ \sigma_k^j = (\sigma_{u_1}^j, \ldots, \sigma_{u_k}^j) \]

and

\[ \frac{\mu_k^j}{\sigma_k^j} = \left( \frac{\mu_{u_1}^j}{\sigma_{u_1}^j}, \ldots, \frac{\mu_{u_k}^j}{\sigma_{u_k}^j} \right) \]

Figure 1: The descriptive space of genome vectorizations.

Figure 1 shows the approximate descriptive space occupied by the various vectorizations. The complete composition vector looks at the frequencies but ignores all additional position information and throws out length. The natural vector uses counts and so length is described, in addition to mean, variance, and higher-order descriptive statistics that can be transformed to describe skewness and kurtosis. The generalized vector uses the length in addition to the frequency, mean, variance, etc. of all k-mers.

2.2.3 One to One

In [Deng et al., 2011] the authors show that there is a one-to-one correspondence between a genome and its natural vector. The same is true for k-mer counts with values of \( k = l \). That is, for
a genome of length \( l \), there is exactly one k-mer with \( k = l \) in the collection of \( 4^k \) different k-mers that has a non-zero count. The generalized vector maintains the one to one correspondence given a large enough number of centralized moments. Alternatively, one may fix \( k \geq 1 \) but let \( j = \max\{n_A, n_C, n_G, n_T\} \) which guarantees one-to-one correspondence too. In practice, we suggest \( k \leq 5 \) and \( j \leq 4 \).

3 Algorithm

3.1 Virus Phylogeny

Viruses are classified phylogenetically using two complementary systems. The first system is known as Baltimore classification \(^{[1]}\). Baltimore classifications are defined by the genomic material of the virus (RNA/DNA), strandedness (single/double), the method of replication (reverse-transcribing), and sense (positive/negative). These classes break the viruses up into 7 classes.

The International Committee for the Taxonomy of Viruses (ICTV) provides the second method of classification \(^{[2]}\). The classifications are made by a sub-committee of the ICTV based on features of the virus (e.g. capsid shape, host, genome sequence, etc.) These classifications are hierarchical. The levels of the hierarchy, ordered from the broadest to the most specific, are order, family, sub-family, genus, species, strain. Additionally, each family belongs to only one Baltimore class. There are additional levels of the hierarchy, e.g. sub-genus, but for the analysis of the data used here we test the Baltimore class, family, genus, and species.

3.2 Training and Testing

Each dataset is split up randomly into a training set of 75% of the data and a testing set of the remaining 25%. The same cross-validation folds (training) and testing sets are used for all of the vectorizations. For the CCV and GV vectorizations, we perform transformations (Box-Cox and then standardization) on the data. Those transformations are calculated using only the training data. The same transformations are then performed on the testing data. To avoid having the transformations induce outliers on the testing set, values of the testing data are truncated at the maxima and minima of each component within the training set. The transformations are not performed for CV and NV, but cross-validation is used to select the best \( k \) for the k-mer that makes up the CV. These are used as baseline performances.

Since we perform cross-validation to determine optimal parameters, and because some of the labels are small in number, we require that a class label have at least 3 samples: 1 sample for testing and 2 for training. Classes with fewer than 3 samples are removed. We also require proportional distribution of the classes amongst the training and testing sets in addition to proportional distribution amongst the cross-validation splits. We use 10-fold cross validation where possible, and smaller where it is not possible.

3.3 Data

The two data sets used are the Reference Sequence data (RefSeq) published by the National Center for Biotechnology Information (NCBI) and the PASC data. The RefSeq data consists of over 2000 viruses, but removing viruses with multiple segments or without a baltimore class label leaves 1881 viruses. An additional 164 viruses are removed when testing family classifications due to having
no label or labels with fewer than three samples. Similarly, 688 viruses are removed when testing genus classifications.

The PASC data consists of 51 families with 8862 viruses in total. This data is used to predict species since that is the primary objective of the web tool. Of the 8862 viruses, 1545 are removed due to sparse labeling. Of the 51 families, 8 of them are removed entirely due to sparse labeling.

### 3.4 PASC

The PASC web tool uses a BLAST-based alignment method. From the (Bao, 2012), “BLAST-based alignment method calculates pairwise identity as number of identical bases in local hits divided by the longest sequence length. Local hits are found by blastn (nucleotide to nucleotide) and tblastn (translated sequences in all 6 frames from one genome to nucleotide sequence of another genome and vice versa)” (Bao, 2012). While a review of BLAST is beyond the scope of this work, it should be noted that BLAST-alignment methods are able to account for large and small indels, duplications, and inversions. This makes it one of the more robust alignment methods. The precomputed similarity scores were downloaded, and are accessible, from the PASC website (Bao, 2012). PASC matrices are not calculated for the RefSeq data and the method is ignored for that evaluation.

### 3.5 k-Nearest Neighbors

The restriction of PASC to similarity matrices restricts the usable classification algorithms. For this reason, we use $k$-nearest neighbor classification. The value of $k$ within the $k$-nearest neighbor algorithm is chosen by cross-validation.

### 3.6 Relevant Component Analysis

With regards to GV and CCV, the exponential growth of the vector size for larger values of $k$ within k-mers ensures that the smaller values of $k$ will be overwhelmed by the larger values of $k$; e.g. there are only 4 1-mers while there are 1024 5-mers. For this reason we perform a version of relevant component analysis (RCA) to (1) improve classification accuracy and (2) because the transformations may provide valuable information for practitioners.

Where the standard RCA (Shental et al., 2006) takes the average of the absolute value of a component’s correlation amongst all labels, we instead use cross-validation to:

1. take the absolute value of the correlation to some power between 0 and 10 before taking the average and
2. we enforce some sparsity by reducing to 0 some percentage of the smallest coefficients. Again, the power and the percentage of coefficients reduced to zero are determined by cross-validation.

### 3.7 Partitions

We perform the above analysis on each dataset 5 times using 5 randomly chosen testing and training set partitions to ensure the reliability of the results. From the single-segment 2044 RefSeq viruses, 163 were removed to make Baltimore classifications due to missing labels or because they are satellites which are known as sub-viral agents and not technically viruses. This leaves 1881 viruses for training (1413) and testing (468) in total. For family level classifications there are 1305 training samples and 412 testing samples. For genus level classifications there are 834 training samples and 251 testing samples.
For each partition of the PASC data there are 5559 training samples and 1758 testing samples. Of the 8790 samples, there were 5883 unique samples representing 80.4% of the testable data. Those 5883 samples were tested, on average, 1.6 times each.

3.8 Cross-validation

Cross-validation is used to tune the parameters of a model. Typically, this is done by performing a grid search over reasonable parameter values [Hastie et al., 2001]. In [Bergstra and Bengio, 2012] a randomized search is shown to be a more efficient method and is used here. In the event that two sets of parameters produce the same classification error in training, one is chosen randomly. If more than two sets of parameters produce the same classification error, then the parameter set that is closest (using the Mahalanobis metric) to the mean of the sets of parameters is chosen.

3.9 Predictions and Errors

Within the PASC data evaluations, predicted class labels are recorded. Viruses where the predicted class labels do not match the given labels are assumed to be errors. While this is not always true due to the inherently messy nature of the data, the low error rates described below indicate that the overwhelming majority of the species labels are correct. Given the quantity of real data here, an amount of uncertainty and/or errors in the labels is to be expected. For example, [Muhire et al., 2013] shows that a number of viruses in just one genus (Mastrevirus) could be recategorized in a more satisfactory manner. Therefore, the results should only be interpreted as an approximation of the quality of the vectorizations.

4 Implementation

4.1 Reference Sequence Results

For Baltimore classifications, with results shown in Table 1, GV performs the best and has an average misclassification rate of 2.9% over the 5 partitions. CCV, NV, and CV have average misclassification rates of 6.8%, 8.2%, and 11.8% respectively.

Results for family classifications given the Baltimore class are shown in Table 2. GV again performs the best and has an average misclassification rate of 5.5% over the 7 Baltimore classes and 5 partitions compared to 8.9%, 13.3%, and 14.7% misclassification rates for CCV, CV, and NV respectively.

Results for genus classifications given family labels are shown in Table 3. GV again performs the best, but this time it ties with CCV with an average misclassification rate over the 72 families and 5 partitions of 5.7% compared to 8.4% and 12.3% misclassification rates for CV and NV respectively.

4.2 PASC results

The totals on the bottom of Table 4 show that CCV and GV are both very competitive with PASC on this data hand-picked for PASC with error rates of 0.7% and 0.8%, respectively, compared to PASC’s 0.6%. CV and NV on the other hand, struggle in many cases. Additionally, the PASC webtool is not a portable in the sense that it relies on NCBI resources and is not available on a PC, while the other four methods can be utilized on a PC easily.
The one case where GV noticeably underperforms compared to PASC and CV is in the family Picornaviridae, with 9 errors total. While this bears more investigation the error rate within that family remains below 1.2%. For CCV and GV, the error rates never exceed 4% on any virus family, reaching their maximum in the Paramyxoviridae and Togaviridae families respectively. PASC’s error rate within families reaches its maximum in the Hepadnaviridae family with 6.67%.

While the overwhelming majority of the species labels are believed to be correct, there is a certain amount of uncertainty and/or errors in the labels. For example, GI #133919132 (Wheat dwarf virus - barley[Germany:Baden:2005]) is predicted to be barley dwarf virus. While it is believed that this virus is indeed a wheat dwarf virus, the prediction stems from the fact that many viruses listed as wheat dwarf are in fact barley dwarf viruses. Therefore, the results described in Table 4 should again only be interpreted as an approximation of the quality of the different methods.

## 5 Discussion

We have generalized the class of genome statistics for sequences that comprise the vectorizations used for phylogenetic classification, thereby avoiding the troubles that accompany sequence alignment. The performance of the GV is superior to the other vectorizations on Baltimore and family classifications. On genus-level and species-level classifications GV performs as well as, or almost as well as, CCV and PASC.

The coefficients generated by the RCA methodology can point practitioners and public health
professionals to the relevant features of a genome more effectively than the CCV vectorization. PASC is unable to do this and requires a two-step process that requires first identifying the appropriate virus family. Additionally, PASC requires the use of high-performance computing that may not be available in low-resource environments. The GV method described here requires less than a second to classify new viruses using existing models and less than a minute to generate entirely new models on a consumer laptop.

Taking classification performance and computational performance features into account, the GV method provides a useful alternative to PASC for phylogenetic classification. Given the many and varied applications of k-mers, this new class of genome statistics may prove to be additionally useful outside the field of phylogenetics.

6 Funding

This work was supported by National Science Foundation [DMS-1120824].

7 Acknowledgement

Yiming Bao and Chenglong Yu provided helpful suggestions.
References

Bailin, H., et al. (2003). Prokaryotic phylogeny based on complete genomes without sequence alignment. *Modern Physics Letters B*, **17**(03), 91–94.

Baltimore, D. (1971). Expression of animal virus genomes. *Bacteriological Reviews*, **35**(3), 235.

Bao, Y. (2012). National center biotechnology information, pairwise sequence comparison web tool. [http://www.ncbi.nlm.nih.gov/sutils/pasc/viridty.cgi](http://www.ncbi.nlm.nih.gov/sutils/pasc/viridty.cgi).

Bao, Y., et al. (2008). Virus classification by pairwise sequence comparison (pasc).

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, **13**, 281–305.

Chan, R., et al. (2010). Composition vector method for phylogenetics-a review. In *Proc. 9th Int. Symp. Operations Research and Its Applications*, page 13.

Deng, M., et al. (2011). A novel method of characterizing genetic sequences: genome space with biological distance and applications. *PloS one*, **6**(3), e17293.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, **55**(10), 78–87.

Hastie, T., et al. (2001). *The elements of statistical learning*, volume 1. Springer New York.

King, A. M., et al. (2011). *Virus taxonomy: IXth report of the International Committee on Taxonomy of Viruses*, volume 9. Access Online via Elsevier.

Lewis, D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *Machine Learning: ECML-98*, pages 4–15.

Muhire, B., et al. (2013). A genome-wide pairwise-identity-based proposal for the classification of viruses in the genus mastrevirus (family geminiviridae). *Archives of virology*, pages 1–14.

Rosen, G., et al. (2012). Naive bayesian classification tool. [http://www.nbc.ece.drexel.edu/creators.php](http://www.nbc.ece.drexel.edu/creators.php).

Shental, N., et al. (2006). Adjustment learning and relevant component analysis. In *Computer Vision ECCV 2002*, pages 776–790. Springer.

Vinga, S. and Almeida, J. (2003). Alignment-free sequence comparison-a review. *Bioinformatics*, **19**(4), 513–523.

Waterman, M. (1995). *Introduction to computational biology: maps, sequences and genomes*. Chapman & Hall/CRC.

Wu, X., et al. (2004). Whole genome phylogeny via complete composition vectors.

Yu, C., et al. (2012). The natural vector classification tool. [http://www.r720.math.tsinghua.edu.cn/Virus/index.php](http://www.r720.math.tsinghua.edu.cn/Virus/index.php).

Yu, C., et al. (2013). Real time classification of viruses in 12 dimensions. *PloS one*, **8**(5), e64328.
| Family                      | # Train | # Test | # Remv’d | # Total | CV  | CCV | NV  | PASC | GV  |
|-----------------------------|---------|--------|----------|---------|-----|-----|-----|------|-----|
| Adenoviridae                | 70      | 22     | 31       | 123     | 0.0 | 0.0 | 0.2 | 0.0  | 0.0 |
| Alphaherpesviridae          | 0       | 0      | 5        | 5       | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Alphaflexiviridae           | 66      | 20     | 39       | 125     | 0.0 | 0.0 | 0.4 | 0.0  | 0.0 |
| Anelloviridae               | 151     | 49     | 38       | 238     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Arteriviridae               | 120     | 39     | 3        | 162     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Astroviridae                | 26      | 8      | 15       | 49      | 0.0 | 0.2 | 0.0 | 0.2  | 0.2 |
| Avsuviroidae                | 292     | 95     | 0        | 387     | 0.0 | 0.2 | 0.0 | 0.0  | 0.0 |
| Baculoviridae               | 8       | 3      | 52       | 63      | 0.0 | 0.0 | 0.6 | 0.0  | 0.0 |
| Betatetralviridae           | 73      | 22     | 46       | 141     | 0.0 | 0.4 | 0.2 | 0.0  | 0.0 |
| Caliciviridae               | 227     | 73     | 10       | 310     | 0.0 | 0.4 | 0.2 | 0.0  | 0.2 |
| Caulimoviridae              | 0       | 0      | 11       | 11      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Coronaviridae               | 66      | 20     | 39       | 125     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Dicistroviridae             | 21      | 7      | 13       | 41      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Endornavirida               | 0       | 0      | 11       | 11      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Filoviridae                 | 20      | 6      | 3        | 29      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Flaviviridae                | 151     | 49     | 38       | 238     | 0.0 | 0.0 | 0.2 | 0.0  | 0.0 |
| Geminiviridae               | 292     | 95     | 0        | 387     | 0.0 | 0.2 | 0.0 | 0.0  | 0.0 |
| Hepdnaviridae               | 8       | 2      | 55       | 65      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Herpesvirida                | 66      | 20     | 39       | 125     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Hypoviridae                 | 20      | 6      | 3        | 29      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Ilioviridae                 | 13      | 3      | 7        | 23      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Inoviridae                  | 0       | 0      | 38       | 38      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Iridoviridae                | 6       | 2      | 10       | 18      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Lentivirus                  | 73      | 22     | 46       | 141     | 0.0 | 0.4 | 0.2 | 0.0  | 0.0 |
| Leviviridae                 | 23      | 6      | 3        | 32      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Lipothrixvirida             | 66      | 20     | 39       | 125     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Luteoviridae                | 151     | 49     | 38       | 238     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Microviridae                | 292     | 95     | 0        | 387     | 0.0 | 0.2 | 0.0 | 0.0  | 0.0 |
| Nanoviridae_CP              | 25      | 8      | 6        | 39      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Nanoviridae_Rep             | 0       | 0      | 48       | 48      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Narnavirida                 | 0       | 0      | 13       | 13      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Papillomaviridae            | 151     | 49     | 38       | 238     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Paramyxoviridae             | 292     | 95     | 0        | 387     | 0.0 | 0.2 | 0.0 | 0.0  | 0.0 |
| Parvoviridae                | 8       | 2      | 10       | 18      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Picornaviridae              | 151     | 49     | 38       | 238     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Podoviridae                 | 6       | 2      | 10       | 18      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Polyomaviridae              | 109     | 34     | 28       | 171     | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Pssovirusoid                | 491     | 155    | 8        | 654     | 1.4 | 1.2 | 3.0 | 0.6  | 1.2 |
| Potyviridae                 | 209     | 66     | 59       | 334     | 1.0 | 0.0 | 0.4 | 0.0  | 0.0 |
| Poxviridae                  | 8       | 3      | 30       | 41      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Rhabdoviridae               | 25      | 8      | 6        | 39      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Secoviridae RNA1            | 21      | 7      | 34       | 62      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Sobemovirus                 | 28      | 9      | 13       | 50      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Tectiviridae                | 0       | 0      | 8        | 8       | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Tobamovirus                 | 78      | 23     | 22       | 123     | 0.0 | 0.0 | 0.4 | 0.0  | 0.0 |
| Togaviridae                 | 92      | 26     | 13       | 131     | 1.4 | 0.2 | 2.0 | 0.4  | 1.0 |
| Tombusviridae               | 18      | 6      | 48       | 72      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Totiviridae                 | 4       | 2      | 32       | 38      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Tymoviridae                 | 5       | 2      | 29       | 36      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |
| Umbravirus                  | 7       | 2      | 3        | 12      | 0.0 | 0.0 | 0.0 | 0.0  | 0.0 |

| Totals                      | 5559    | 1758   | 1545     | 8862    | 30.0 | 13.0 | 49.6 | 10.6  | 14.2 |

Table 4: PASC Errors and samples by family averaged over 5 partitions