JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 23, Number 10, 2016

Pp. 810–820
DOI: 10.1089/cmb.2013.0132

Descriptive Statistics of the Genome: Phylogenetic Classification of Viruses

TROY HERNANDEZ[1] and JIE YANG

ABSTRACT

The typical process for classifying and submitting a newly sequenced virus to the NCBI database involves two steps. First, a BLAST search is performed to determine likely family candidates. That is followed by checking the candidate families with the pairwise sequence alignment tool for similar species. The submitter's judgment is then used to determine the most likely species classification. The aim of this article is to show that this process can be automated into a fast, accurate, one-step process using the proposed alignment-free method and properly implemented machine learning techniques.

We present a new family of alignment-free vectorizations of the genome, the generalized vector, that maintains the speed of existing alignment-free methods while outperforming all available methods. 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 five different characterizations of genome similarity using *k*-nearest neighbor classification and evaluate these on two collections of viruses totaling over 10,000 viruses. We show that our proposed method performs better than, or as well as, other methods at every level of the phylogenetic hierarchy. The data and R code is available upon request.

Key words: alignment-free, classification, machine learning, phylogenetics, virology.

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 (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.

[1]Mathematical Sciences Center, Tsinghua University, Beijing, China.
[2]Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, Illinois.

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 aligment (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 precomputed. After the pairwise identities of a new virus are calculated a histogram of the identity scores are displayed—color-coded according to their subfamily, genus, and species. From there, experts use their best judgment 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; for example, for $k=3$, ''AGC,'' ''CTA,'' and ''TAG.'' For a given k, a vector of k-mer frequencies can be used in 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 $A,C,G,$ and T in addition to positional information. That is, the mean position of the nucleotides and their central moments; that is, the 2nd, 3rd, 4th, etc., central moments.

In this article 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 five 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; that is, they use statistics collected from a genome as components in a vector. Those vectors are then used in 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 preprocessing used to implement the classifications are described. This will affect the measures of similarity differently for the different representations.

2.1.1. Sequence alignment. A review of sequence alignment is beyond the scope of this article, but one can be found in Waterman (1995). What is important, with regard to this article, is the computational complexity of MSA. Given a collection of m sequences of length n the complexity is $O(n^m)$. Newer implementations have brought speed-ups beyond the naive implementation, but large-scale comparisons can still be prohibitive. PASC gets around this by aligning every pair of sequences and uses those pairwise scores for a similarity matrix.

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 in which each component represents a word. This conversion results in the loss of grammar and word order information.

Within bioinformatics, the bag-of-words model has been adapted to work on genomes. The ''words'' in this case are nucleotides in the genome. Substrings 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 1st position to the $(n - k+1)$st position. For example, in the string $S=$ GATTACA there are six nonzero 2-mers:

$$n_{AC}=1, n_{AT}=1, n_{CA}=1, n_{GA}=1, n_{TA}=1, n_{TT}=1$$

This results in a vector of counts:

$$\begin{aligned} n_2 = (&n_{AA}, n_{AC}, n_{AG}, n_{AT}, n_{CA}, n_{CC}, n_{CG}, n_{CT}, \\ &n_{GA}, n_{GC}, n_{GG}, n_{GT}, n_{TA}, n_{TC}, n_{TG}, n_{TT}) \quad (1) \\ = (&0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1) \quad (2) \end{aligned}$$

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

$$f_2 = (0, \tfrac{1}{6}, 0, \tfrac{1}{6}, \tfrac{1}{6}, 0, 0, 0, \tfrac{1}{6}, 0, 0, 0, \tfrac{1}{6}, 0, 0, \tfrac{1}{6}) \quad (3)$$

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 (Hao et al., 2003). 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 Chan et al. (2010).

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 Chan et al. (2010), we estimate its expected frequency:

$$\begin{aligned} \mathcal{P}(LwR) &= \mathcal{P}(Lw)\mathcal{P}(R|Lw) \quad (4) \\ &\approx \mathcal{P}(Lw)\mathcal{P}(R|w) \quad (5) \\ &= \frac{\mathcal{P}(Lw)\mathcal{P}(wR)}{\mathcal{P}(w)} \quad (6) \end{aligned}$$

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

$$c_u = \frac{f_u - \mathcal{P}_u}{\sqrt{\mathcal{P}_u}} \quad (7)$$

For a given k this results in the composition vector:

$$c_k = (c_{u_1}, \ldots, c_{u_4k}). \quad (8)$$

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 (Wu et al., 2004). This produces the CCV:

$$v_k = (c_1, \ldots, c_k) \quad (9)$$

For the CV and a fixed k, using the values without additional transformations is sufficient. When using the CCV with distance matrices another transformation is necessary for the following reason: Concatenating the CVs of a genome from $k=1\ldots5$, the vector will have four 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 use 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 (Deng et al., 2011; Yu et al., 2013) 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; that is, $s_i \in \{A,C,G,T\}$ 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

$$s_u[1] < \cdots < s_u[n_u] \tag{10}$$

and

$$S[s_u[i]] = u, \text{ for } i=1, \ldots, n_u. \tag{11}$$

(4) Let the mean position of letter u be

$$\mu_u = \sum_{i=1}^{n_u} s_u[i]/n_u \tag{12}$$

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

$$d_u^j = \sum_{i=1}^{n_u} \frac{(s_u[i]-\mu_u)^j}{n_u^{j-1} n^{j-1}}. \tag{13}$$

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:

$$(n_A, \mu_A, d_A^2, n_C, \mu_C, d_C^2, n_G, \mu_G, d_G^2, n_T, \mu_T, d_T^2) \tag{14}$$

2.2. Proposed vectorization

Given the k-mer, composition vector, complete composition vector, and natural vector representations of the genome, we introduce the generalized vector (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 is being ignored. In addition to extending the natural vector definition to k-mers with values of k greater than 1, we also make some adjustments.

2.2.1. Coordinates of natural vector. Suppose n is large enough. Let s_u be a randomly chosen position for the nucleotide u. Assume that s_i follows an *iid* discrete distribution with four 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,

$$(s_u-\mu_u)/n \dot{\sim} \text{Unif}(-1/2, 1/2) \tag{15}$$

$$\mu_u \dot{\sim} \frac{n}{2} \tag{16}$$

and

$$d_u^j \dot{\sim} \begin{cases} \frac{n}{2^j(j+1)n_u^{j-2}} & \text{if } j=2d \\ 0 & \text{if } j=2d-1 \end{cases} \tag{17}$$

because

$$\frac{1}{n_u}\sum_{i=1}^{n_u} \frac{(s_u[i]-\mu_u)^j}{n^j} \dot{\sim} \int_{-1/2}^{1/2} x^j dx \tag{18}$$

$$= \begin{cases} \frac{1}{2^j(j+1)} & \text{if } j=2d \\ 0 & \text{if } j=2d-1 \end{cases} \tag{19}$$

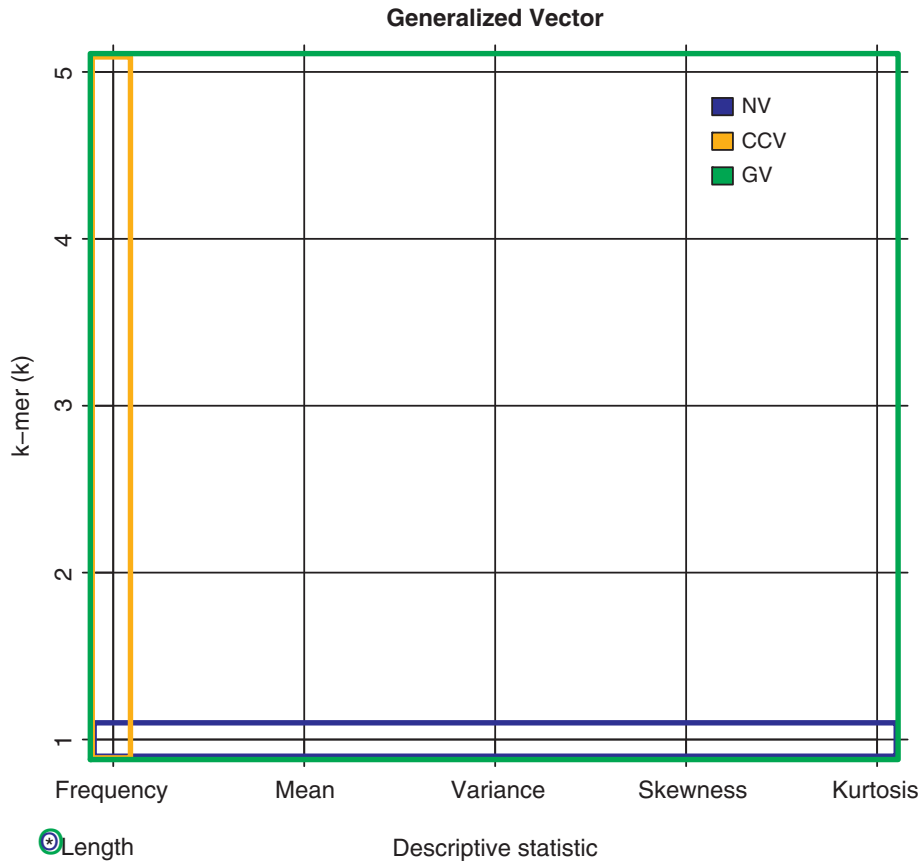

FIG. 1. The descriptive space of genome vectorizations.

Due to the term "n_u^{j-2}" in (17), which is roughly $(np_u)^{j-2}$, d_u^j will be much smaller than n_u and μ_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. 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.

Third, we use the standardized moments, $\frac{\mu^j}{\sigma^j}$, where μ^j represents the j-th moment about the mean and σ represents the standard deviation,

$$\mu_u^j = \frac{1}{n_u}\sum_{i=1}^{n_u}(s_u[i]-\mu_u)^j \tag{20}$$

$$\sigma_u = \sqrt{\frac{1}{n_u}\sum_{i=1}^{n_u}(s_u[i]-\mu_u)^2} \tag{21}$$

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 reason for this is that the scaling of the central moment by $\frac{1}{n^{j-1}}$ 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; for example, $k=1\ldots5$. The *generalized vector*, g_k^j, of a DNA sequence S is defined by

$$(n, v_k, \mu_1, \ldots, \mu_k, \sigma_1^2, \ldots, \sigma_k^2, \frac{\mu_1^3}{\sigma_1^3}, \ldots, \frac{\mu_k^3}{\sigma_k^3}, \ldots, \frac{\mu_1^j}{\sigma_1^j}, \ldots, \frac{\mu_k^j}{\sigma_k^j}) \tag{22}$$

where

$$\mu_k^j = (\mu_{u_1}^j, \ldots, \mu_{u_{4^k}}^j) \tag{23}$$

$$\sigma_k^j = (\sigma_{u_1}^j, \ldots, \sigma_{u_{4^k}}^j) \tag{24}$$

and

$$\frac{\mu_k^j}{\sigma_k^j} = \left(\frac{\mu_{u_1}^j}{\sigma_{u_1}^j}, \ldots, \frac{\mu_{u_{4^k}}^j}{\sigma_{u_{4^k}}^j} \right). \tag{25}$$

Figure 1 shows the approximate descriptive space occupied by the various vectorizations. The complete composition vector uses 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-mers with $k = n$. That is, for a genome of length n and a k-mer vector with $k = n$, there is exactly one k-mer in the 4^k length vector that is nonzero. The generalized vector maintains the one-to-one correspondence given that one may fix $k \geq 1$ and let $j = max\{n_{u_1}^j, \ldots, nu_{4k}{}^j\}$, which guarantees one-to-one correspondence. In practice, we use $k \leq 5$ and $j \leq 4$.

3. ALGORITHM

3.1. Phylogenetic classes

Viruses are classified phylogenetically using two complementary systems. The first system is known as *Baltimore classification* (Baltimore, 1971). 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). This results in seven mutually exclusive viral classes.

The International Committee on Taxonomy of Viruses (ICTV) provides the second method of classification (King et al., 2011). The classifications are made by a subcommittee 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*, *subfamily*, *genus*, *and species*. Additionally, each family belongs to only one Baltimore class. There are additional levels of the hierarchy, for example, *subgenus*, but for the data used here only the Baltimore class, family, genus, and species are analyzed.

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.

Since we perform cross-validation to determine optimal parameters, and because some of the labels are small in number, it is required that a class label have at least three samples: one sample for testing and two for training. Classes with fewer than three samples are removed. In practice, the viruses in these classes can be added back into the training set for the final model. The procedure for determining if a virus belongs to a new class is discussed below. We also require proportional distribution of the classes among the training and testing sets in addition to proportional distribution among the cross-validation splits. We use 10-fold cross validation where possible, and smaller where it is not.

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 after removing viruses with multiple segments or without Baltimore classes, only 1881 viruses remain.

The PASC data consist of 51 families with 8862 viruses in total. These data are used to predict species since that is the primary objective of the web tool.

3.4. PASC

The PASC web tool uses a BLAST-based alignment method. 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 resulted in the k-nearest neighbor algorithm being the most straightforward to implement. 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; for example, there are only four 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 among 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.

3.7. Partitions

We perform the above analysis on each dataset five times using five randomly chosen testing and training set partitions to ensure the reliability of the results. From the single-segment 2044 RefSeq viruses, 1881 viruses are used for training (1413) and testing (468) in total. For each partition of the PASC data there are 5559 training samples and 1758 testing samples.

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 a reasonable parameter space (Hastie et al., 2001). In Bergstra and Bengio (2012) a randomized search is shown to be a more efficient method and is used here.

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 labels given in the NCBI or PaSC datasets 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 reliable.

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 five 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 seven Baltimore classes and five partitions compared to 8.9%, 13.3%, and 14.7% misclassification rates for CCV, CV, and NV respectively.

Table 1. Baltimore Errors and Samples Averaged Over Five Partitions

	I	*II*	*III*	*IV*	*V*	*VI*	*VII*	*Totals*
No. train	582	246	34	423	51	44	33	1413
No. test	194	82	11	140	16	14	11	468
No. removed	0	0	0	0	0	0	0	0
No. total	776	328	45	563	67	58	44	1881
NV errors	4.8	7.0	4.4	10.4	3.2	6.6	1.8	38.2
CV errors	4.8	20.4	7.8	17.2	1.4	1.8	1.8	55.2
CCV errors	2.6	13.4	4.8	8.2	1.0	1.2	0.2	31.4
GV errors	1.6	5.8	2.4	1.8	0.6	0.8	0.4	13.4

Table 2. Family Errors and Samples Given Baltimore Class Averaged Over Five Partitions

	I	*II*	*III*	*IV*	*V*	*VI*	*VII*	*Totals*
No. train	558	238	28	400	48	0	33	1305
No. test	178	76	8	124	15	0	11	412
No. removed	40	14	9	39	4	58	0	164
No. total	776	328	45	563	67	58	44	1881
NV errors	42.8	3.0	1.0	13.0	0.6	0.0	0.0	60.4
CV errors	27.4	3.2	1.4	21.6	1.2	0.0	0.0	54.8
CCV errors	23.6	2.6	1.0	8.8	0.6	0.0	0.0	36.6
GV errors	17.0	0.6	0.4	4.4	0.2	0.0	0.0	22.6

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 web tool is not portable in the sense that it relies on NCBI resources and cannot be implemented on a PC. The other four methods can be utilized on a PC easily.

One case where GV noticeably underperforms compared to PASC and CV is in the family *Picornaviridae*, with nine 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

Table 3. Genus Errors and Samples Given Family Class Averaged Over Five Partitions

	I	*II*	*III*	*IV*	*V*	*VI*	*VII*	*Totals*
No. train	252	221	16	330	32	43	23	917
No. test	77	67	4	101	10	10	7	276
No. removed	447	40	25	132	25	5	14	688
No. total	776	328	45	563	67	58	44	1881
NV errors	10.6	2.4	1.4	10.4	2.4	2.4	1.2	30.8
CV errors	2.8	1.8	2.2	10.8	2.0	0.4	1.0	21.0
CCV errors	2.0	2.4	1.2	7.0	0.6	0.0	1.2	14.4
GV errors	3.0	1.6	1.6	6.0	1.2	0.2	0.8	14.4

Table 4. Errors and Samples by Family Averaged Over Five Partitions

	No. train	*No. test*	*No. removed*	*No. total*	*CV*	*CCV*	*NV*	*PASC*	*GV*
Adenoviridae	70	22	31	123	0.0	0.0	0.2	0.0	0.0
Alloherpesviridae	0	0	5	5	0.0	0.0	0.0	0.0	0.0
Alphaflexiviridae	66	20	39	125	0.0	0.0	0.0	0.0	0.0
Anelloviridae	151	49	38	238	0.0	0.0	0.4	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	0.2	0.0	0.2
Avsunviroidae	292	95	0	387	0.0	0.2	0.0	0.0	0.2
Baculoviridae	8	3	52	63	0.0	0.0	0.6	0.0	0.0
Betaflexiviridae	73	22	46	141	0.0	0.4	0.2	0.0	0.0
Caliciviridae	227	73	10	310	0.8	1.4	1.2	1.2	0.8
Caulimoviridae	33	10	52	95	0.0	0.0	0.2	0.0	0.0
Circoviridae	272	88	18	378	0.0	0.0	0.4	0.0	0.0
Coronaviridae	108	34	28	170	0.8	0.0	0.4	0.0	0.2
Dicistroviridae	21	7	13	41	0.0	0.0	0.0	0.0	0.0
Endornaviridae	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	562	183	41	786	2.8	0.6	4.8	0.4	0.6
Geminiviridae	505	154	220	879	3.0	3.8	14.0	4.0	3.4
Hepadnaviridae	50	15	8	73	0.6	0.4	1.0	1.0	0.0
Herpesviridae	8	2	55	65	0.0	0.0	0.0	0.0	0.0
Hypoviridae	0	0	9	9	0.0	0.0	0.0	0.0	0.0
Iflavirus	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	699	230	10	939	4.4	1.2	6.0	0.8	1.6
Leviviridae	23	6	3	32	0.4	0.0	0.2	0.0	0.0
Lipothrixviridae	0	0	8	8	0.0	0.0	0.0	0.0	0.0
Luteoviridae	73	22	19	114	0.0	0.4	0.0	0.0	0.4
Microviridae	44	13	15	72	0.0	0.0	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
Narnaviridae	0	0	13	13	0.0	0.0	0.0	0.0	0.0
Papillomaviridae	157	49	86	292	4.6	0.4	5.0	0.0	0.4
Paramyxoviridae	168	51	17	236	2.4	1.8	1.6	2.0	1.2
Parvoviridae	84	24	62	170	0.6	0.2	2.4	0.2	0.2
Picornaviridae	491	155	39	685	4.8	0.2	4.8	0.0	1.8
Podoviridae	7	2	113	122	0.0	0.0	0.0	0.0	0.2
Polyomaviridae	109	34	28	171	0.0	0.0	0.4	0.0	0.4
Pospiviroidae	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	87	28	27	142	0.2	0.0	0.0	0.0	0.0
SecoviridaeRNA1	21	7	34	62	0.8	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	0.4	0.2	0.0	0.4
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

maximum in the *Paramyxoviridae* and *Togaviridae* families respectively. PASC's error rate within families reaches its maximum in the *Hepadnaviridae* family with 6.67%.

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 are simple and intuitive, but other methodologies may be more effective; for example, principle component analysis (Jolliffe, 2005), neighborhood component analysis (Goldberger et al., 2004), or large-margin nearest neighbors (Blitzer et al., 2005). PASC includes 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.

Future work could include the GV being extended to maximal length using the suffix-tree methods that have already been shown to be effective with CCA methods in phylogenetic classification (Apostolico et al., 2010). Additionally, the method described above should be considered a proof-of-concept. The determination of new virus classes (and incorrect labels) can be handled in practice using techniques developed in the deep $\boldsymbol{k}$-nearest neighbor literature (Denceux, 1995), one-class SVMs (Chen et al., 2001), and cluster analysis (Tibshirani et al., 2001).

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.

ACKNOWLEDGMENTS

Yiming Bao and Chenglong Yu provided helpful suggestions. This work was supported by National Science Foundation [DMS-1120824], University of Illinois at Chicago, and Tsinghua University.

AUTHOR DISCLOSURE STATEMENT

The corresponding author has previously contacted the editor, Michael Waterman, concerning future collaboration. No competing financial interests exist.

REFERENCES

Apostolico, A., Denas, O., and Dress, A. 2010. Efficient tools for comparative substring analysis. *J. Biotechnol.* 149, 120–126.

Baltimore, D. 1971. Expression of animal virus genomes. *Bacteriol. Rev.* 35, 235.

Bao, Y. 2012. National Center Biotechnology Information, Pairwise Sequence Comparison Web Tool. www.ncbi.nlm.nih.gov/sutils/pasc/viridty.cgi (accessed Dec. 21, 2012).

Bao, Y., Kapustin, Y., and Tatusova, T. 2008. Virus classification by pairwise sequence comparison (PASC), 342–348. In *Encyclopedia of Virology*, vol. 5. Elsevier, Oxford, UK.

Bergstra, J., and Bengio, Y. 2012. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 13, 281–305.

Blitzer, J., Weinberger, K.Q., and Saul, L.K. 2005. Distance metric learning for large margin nearest neighbor classification, 1473–1480. In *Advances in Neural Information Processing Systems*. MIT Press, New York.

Chan, R., Wang, R., and Yeung, H. 2010. Composition vector method for phylogenetics—a review, 13. In *Proc. 9th Int. Symp. Operations Research and Its Applications*.

Chen, Y., Zhou, X.S., and Huang, T.S. 2001. One-class SVM for learning in image retrieval, 34–37. In *Proceedings of 2001 International Conference on Image Processing*, vol. 1. IEEE, New York.

Denceux, T. 1995. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. Syst. Man Cybernet.* 25, 804–813.

Deng, M., Yu, C., Liang, Q., et al. 2011. A novel method of characterizing genetic sequences: Genome space with biological distance and applications. *PLoS One* 6, e17,293.

Domingos, P. 2012. A few useful things to know about machine learning. *Commun. ACM* 55, 78–87.

Goldberger, J., Roweis, S., Hinton, G., et al. 2004. Neighbourhood components analysis. *Adv. Neural Inform. Process. Syst.* 17, E513–E520.

Hao, B., Qi, J., and Wang, B. 2003. Prokaryotic phylogeny based on complete genomes without sequence alignment. *Modern Phys. Lett. B* 17, 91–94.

Hastie, T., Tibshirani, R., and Friedman, J.J.H. 2001. *The Elements of Statistical Learning*, vol. 1. Springer, New York.

Jolliffe, I. 2005. *Principal Component Analysis*. Wiley Online Library.

King, A.M., Adams, M.J., Lefkowitz, E.J., et al. 2011. *Virus Taxonomy: IXth Report of the International Committee on Taxonomy of Viruses*, vol. 9. Elsevier, New York.

Lewis, D. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. Machine Learning: ECML-98, 4–15. Volume 1398 of the series Lecture Notes in Computer Science.

Rosen, G., Garbarine, E., Caseiro, D., et al. 2012. Naive Bayesian Classification Tool. www.nbc.ece.drexel.edu/creators.php (accessed Dec. 21, 2012).

Shental, N., Hertz, T., Weinshall, D., et al. 2006. Adjustment learning and relevant component analysis, 776–790. In *Computer VisionECCV 2002*. Springer, New York.

Tibshirani, R., Walther, G., and Hastie, T. 2001. Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 63, 411–423.

Vinga, S., and Almeida, J. 2003. Alignment-free sequence comparisona review. *Bioinformatics* 19, 513–523.

Waterman, M. 1995. *Introduction to Computational Biology: Maps, Sequences and Genomes.* Chapman & Hall/CRC, New York.

Wu, X., Wan, X., Wu, G., et al. 2004. Whole genome phylogeny via complete composition vectors. *Int. J. Bioinform. Res. Appl*. 2, 219–248.

Yu, C., Hernandez, T., Zheng, H., et al. 2012. The natural vector classification tool. www.r720.math.tsinghua.edu.cn/Virus/index.php (accessed Dec. 21, 2012).

Yu, C., Hernandez, T., Zheng, H., et al. 2013. Real time classification of viruses in 12 dimensions. *PLoS One* 8, e64,328.

Address correspondence to:
Dr. Jie Yang
Department of Mathematics, Statistics, and Computer Science
University of Illinois at Chicago
513 Science and Engineering Offices
851 S. Morgan Street
Chicago, IL 60607-7045

E-mail: troy.hernandez.phd@gmail.com