Advances and Applications in Bioinformatics and Chemistry

Dovepress
open access to scientific and medical research

Open Access Full Text Article

ORIGINAL RESEARCH

LifePrint: a novel *k*-tuple distance method for construction of phylogenetic trees

This article was published in the following Dove Press journal:
Advances and Applications in Bioinformatics and Chemistry
20 January 2011
Number of times this article has been viewed

Fabián Reyes-Prieto[1]
Adda J García-Chéquer[1]
Hueman Jaimes-Díaz[1]
Janet Casique-Almazán[1]
Juana M Espinosa-Lara[1]
Rosaura Palma-Orozco[2]
Alfonso Méndez-Tenorio[1]
Rogelio Maldonado-Rodríguez[1]
Kenneth L Beattie[3]

[1]Laboratory of Biotechnology and Genomic Bioinformatics, Department of Biochemistry, National School of Biological Sciences, [2]Superior School of Computer Sciences, National Polytechnic Institute, Mexico City, Mexico; [3]Amerigenics Inc, Crossville, Tennessee, USA

Purpose: Here we describe LifePrint, a sequence alignment-independent *k*-tuple distance method to estimate relatedness between complete genomes.

Methods: We designed a representative sample of all possible DNA tuples of length 9 (9-tuples). The final sample comprises 1878 tuples (called the LifePrint set of 9-tuples; LPS9) that are distinct from each other by at least two internal and noncontiguous nucleotide differences. For validation of our *k*-tuple distance method, we analyzed several real and simulated viroid genomes. Using different distance metrics, we scrutinized diverse viroid genomes to estimate the *k*-tuple distances between these genomic sequences. Then we used the estimated genomic *k*-tuple distances to construct phylogenetic trees using the neighbor-joining algorithm. A comparison of the accuracy of LPS9 and the previously reported 5-tuple method was made using symmetric differences between the trees estimated from each method and a simulated "true" phylogenetic tree.

Results: The identified optimal search scheme for LPS9 allows only up to two nucleotide differences between each 9-tuple and the scrutinized genome. Similarity search results of simulated viroid genomes indicate that, in most cases, LPS9 is able to detect single-base substitutions between genomes efficiently. Analysis of simulated genomic variants with a high proportion of base substitutions indicates that LPS9 is able to discern relationships between genomic variants with up to 40% of nucleotide substitution.

Conclusion: Our LPS9 method generates more accurate phylogenetic reconstructions than the previously proposed 5-tuples strategy. LPS9-reconstructed trees show higher bootstrap proportion values than distance trees derived from the 5-tuple method.

Keywords: phylogeny, sequence alignment, similarity search, tuple, viroid

Introduction

The most used and widespread representations of the evolutionary history of biologic entities are phylogenetic trees. Typically, molecular phylogenetic tree construction starts from a set of sequences (DNA or proteins), computation of a multiple sequence alignment, and then, based on the multiple sequence alignment, construction of a tree using one or several optimization criteria, such as distance, maximum parsimony, minimum evolution, maximum likelihood, and Bayesian inference. Among these criteria, a distance-based method using neighbor-joining (NJ)[1] is frequently used because it is considerably faster than character-based methods such as maximum parsimony, maximum likelihood, and Bayesian inference. However, the requirement of using multiple sequence alignment carries some disadvantages for typical tree construction methods. One of the major limitations of multiple alignments arises from the heuristic

Correspondence: Fabián Reyes-Prieto
Laboratory of Biotechnology and Genomic Bioinformatics, Department of Biochemistry, National School of Biological Sciences, National Polytechnic Institute, CP 11340, Mexico City, Mexico
Tel/Fax +52 55 5729 6000 Ext 62322
Email freyesp0900@ipn.mx

submit your manuscript | www.dovepress.com
Dovepress
DOI: 10.2147/AABC.S15021

methods used to calculate the multiple sequence alignment. These heuristic methods can present difficulties in handling long sequences, given that the underlying algorithms have a computational complexity of quadratic order (ie, discrete increases in length of the sequences involve major increases in the time needed to process multiple sequence alignment), which turns out to be impractical in some cases, eg, when analyzing relatedness between complete genomes. Additionally, because multiple sequence alignment often contains a number of homology ambiguities, phylogenetic inferences based on multiple sequence alignment analysis may produce equivocal trees.[2,3] Certain types of evolutionary events, like translocations and inversions, are hardly considered and included by multiple sequence alignment analysis. Another drawback of distance-based methods is that they consider only differences between sequences without considering their position. Some common computational programs for multiple sequence alignment construction are MUSCLE,[4] DIALIGN 2,[5] T-Coffee,[6] CLUSTAL W,[7] and Kalign.[8]

Classic phylogenetic surveys at the genomic level are computationally extremely demanding approaches and in some cases may be impractical. To overcome some of these practical limitations, alternative phylogenetic methods have been proposed that are independent of multiple sequence alignment. For example, gene content methods define an evolutionary distance between two genomes based on the percentage of shared homologous genes.[9–11] Recently, the use of signature genes corresponding to various taxonomic levels has been successfully tested.[12] The compression methods search for exact, approximate, direct, or inverted repeats and measure the similarity of whole genomes based on their relative "compression rate".[13–15] The composition vector method uses informative strings (ie, short nucleotide sequences) to construct phylogenies. An improved selection method extracts the strings with the best absolute relative entropy in a group of carefully sequence-curated strains, and then uses those strings to estimate evolutionary distances. This procedure was successfully applied to human immunodeficiency-1 subtyping using strings of 5–9 nucleotides in length.[16] This selection method has been improved statistically[17] and used to analyze large double-stranded DNA viruses.[18]

Another multiple sequence alignment-independent method for phylogenetic inference involves the estimation of *k*-tuple distance (also known as *k*-mer distance) between sequences. The *k*-tuple distance between two sequences refers to the sum of the differences in frequency, over all possible tuples of length *k*, between the sequences. Frequencies of 2-tuples in genomes enabled the creation of a biologically plausible phylogenetic tree for mitochondrial genomes.[19] This strategy has also been applied using amino acid strings.[20,21] Due to the amount of information to be processed, relatively large memory and central processing unit usage are required for this approach. Consequently, in practice, the *k* values used have been set to relatively small lengths, such as 5 and 6. Several multiple sequence alignment programs (eg, MUSCLE, CLUSTAL W, and Kalign) compute the *k*-tuple distance matrix for the sequences to be aligned, then these programs use algorithms such as NJ to construct a "guide tree" quickly that determines the order in which sequences are aligned. However, guide trees are rarely used as final phylogenetic trees, and other packages, such as PHYLIP[22] and PAUP,[23] are regularly used for this purpose. Recently, it has been shown that a 5-tuple distance method outperformed other distance estimators most of the time and could be at least twice as accurate as other distance estimators.[2]

Here we characterize and propose LifePrint, a *k*-tuple distance method that is independent of multiple sequence alignment and only uses a representative sample of all possible tuples of a given length *k*.

Methods

LifePrint set of 9-tuples

On the basis of previous analyses (Casique-Almazán et al, unpublished data) we observed that tuples of size 9 show optimal performance in the study of viroid genomes. To calculate the *k*-tuple distance between genomic sequences, we scrutinized real and simulated viroid genomes by similarity searches using a set of 1878 9-tuples. Each 9-tuple sequence of the set was distinct from the others by at least two internal and noncontiguous differences. The group of 1878 9-tuples, called LPS9, is a representative sample of all possible tuples of length 9, ie, 4^9 (262,144). Figure 1 illustrates the LPS9 distribution along the complete set of 262,144 9-tuples. LPS9 is available at the Universal Fingerprinting Chip Applications Server (UFCVH site).[24] We designed the LPS9 with the Universal program included in the Universal Fingerprinting Chip designer (UFC designer, unpublished manuscript) software. We used the following UFC designer criteria to define LPS9:

- The substitution criterion: this grouping criterion retrieved 9-tuples that had at least two nucleotide differences between them
- The block criterion: this criterion excluded 9-tuples with differences located in the ends; thus, the resulting group comprised sequences showing only internal differences
- Refining criterion: tuples sharing contiguous differences were excluded; after applying this criterion, the final

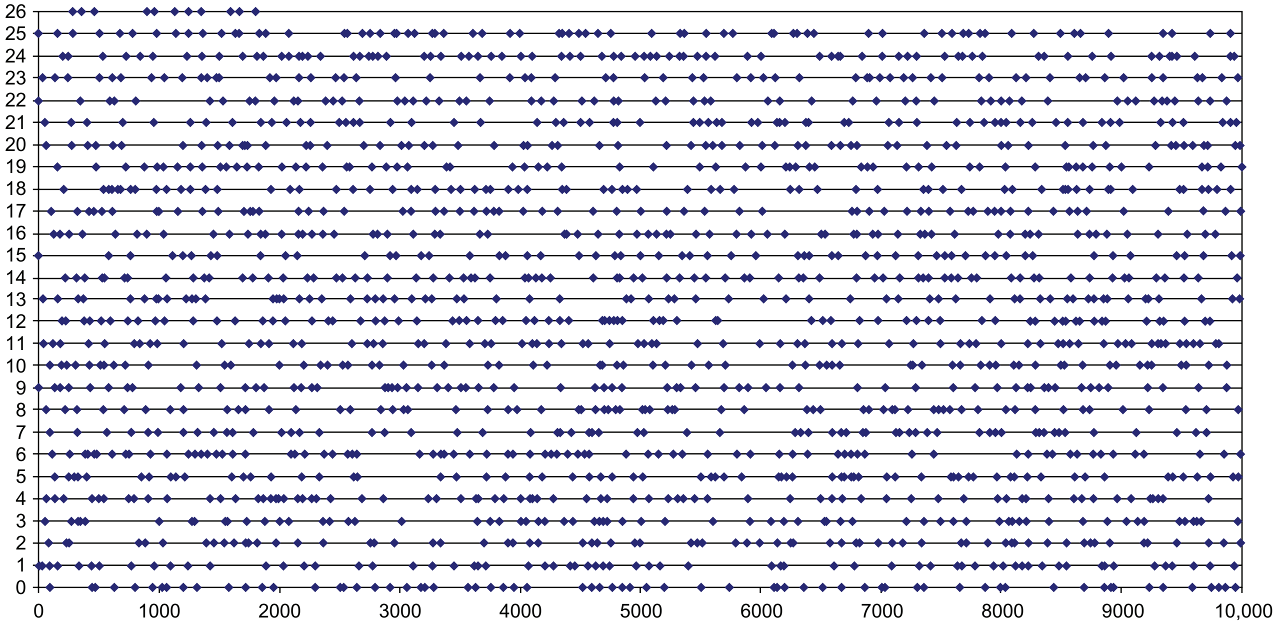


Figure 1 Distribution of the LifePrint set of 9-tuples (LPS9) inside the complete set of 9-tuples.[a]

Note: [a]In total, 1878 tuples of the LPS9 were represented graphically (blue points) in agreement with the positions inside the original list of all possible 9-tuples (262,144). Every line represents 10,000 tuples.

set contained sequences showing only noncontiguous differences.

Before applying each of three criteria, the Universal program "randomized" the query 9-tuple sets to diminish potential sampling bias during the grouping process. The sequences and number of the selected tuples depend on the randomization process and, as a consequence, the identified LPS9 set is not unique. However, any other LPS9 selected using this strategy will produce similar results.

Genomic sequences

We used real and simulated viroid genomes as models. The small size of the viroid genomes (approximately 300 nucleotides) facilitated the present analysis. See Appendix I for the NCBI access numbers of 36 real viroid genomes.

For accuracy of evaluation of differences between LifePrint and other tree construction methods, we used the EvolSeq program (available at the UFCVH site) to simulate the evolution of 32 viroid genomes (named from CVII31 to CVII62) derived from a common ancestor (Citrus viroid II). We used a five-generation evolutionary scheme considering a substitution model with a transition/transversion ratio of 2, as defined in the Kimura 2-parameter model. The 32 simulated viroid genomes were used to reconstruct phylogenetic trees with different methods. The simulated "true" phylogenetic tree (true tree) showing the real phylogenetic relationships between the 32 simulated viroid genomes is shown in Figure 2. The true tree was used as a reference to evaluate the accuracy of our phylogenetic reconstructions.

Similarity search

We used the Virtual Hybridization program to scrutinize the real and simulated viroid genomes with the 1878 9-tuples (LPS9). We compared four conditions allowing a distinct number of sequence differences between LPS9 and the 36 genomes, ie, no differences, no differences and allowing one difference, no differences and allowing up to two differences, and no differences and allowing up to three differences. The Virtual Hybridization program produces two different outputs. One is a detailed list of the positions at which each *k*-tuple is localized in the genomic sequence and a global table with the frequency of occurrences of each tuple in the genomic sequences (global frequency table). Alternatively, the global table can show just the presence (1) or absence (0) of the tuple in the sequences (global binary table). We used both frequency and binary tables to calculate three different kinds of *k*-tuple distances independently using the Characters program. The Virtual Hybridization and Characters programs are available at the UFCVH site.

Genomic coverage

Using LPS9 as the query, we carried out similarity searches to evaluate the capacity of LPS9 to cover viroid genomic sequences entirely. The LPS9 genomic coverage depicted in

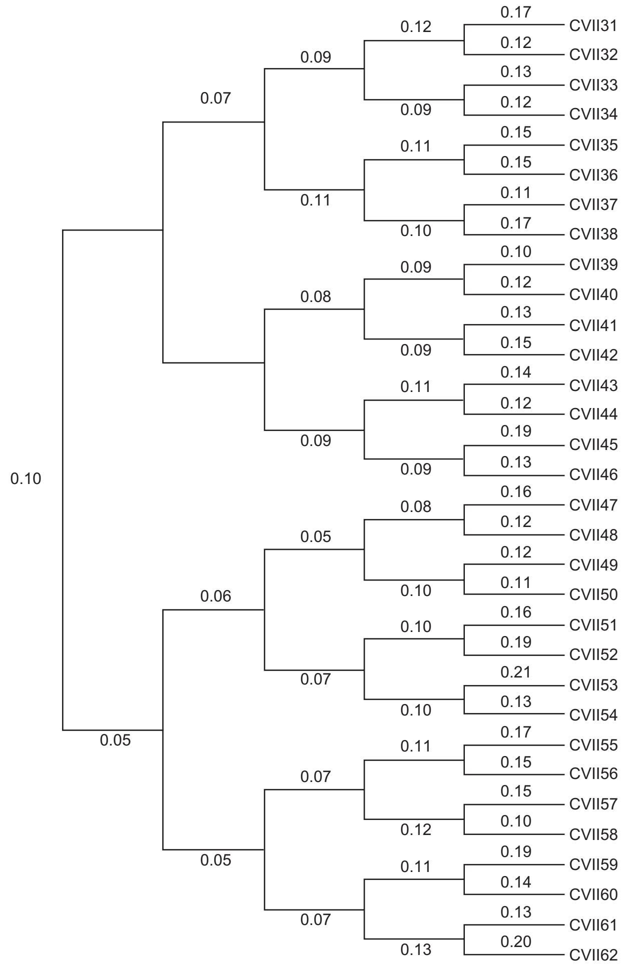


Figure 2 True tree.[a]

Note: [a]The true tree was manually constructed using as a reference the simulated evolution of 32 viroid genomes derived from the Citrus viroid II genome. Nucleotide substitutions were simulated following a 5-generation pattern and considering an evolutionary model with a transition/transversion ratio of 2 (Kimura 2-parameter).

Figure 3 shows the consensus sequence produced by tiling 9-tuples according to their matching position along the first 80 nucleotides located in the 5′ end of the Hop stunt viroid genome (302 nucleotides in length). Sequence identities and differences are shown in capital and lower case letters, respectively. For this representative analysis, we arbitrarily selected the Hop stunt viroid genome, but similar results were obtained with other genomes analyzed.

Detection of single base repeats

For assessment of the ability of LifePrint to detect single base repeats (homopolymers), we used the sequence AAAAAAAAATTTTTTTTTCCCCCCCCCG GGGGGGGG. Figure 4 shows tiling of 9-tuples according to their matching position along this model.

Calculating and bootstrapping *k*-tuple distances

We assumed that different distance metrics have different inherent accuracies for phylogenetic estimation. Here we used three different metrics to calculate *k*-tuple distances between viroid genomic sequences. First, we used the logarithmic *k*-tuple distance based on the Jaccard index (dLog), whereby distances based on the Jaccard index only consider tuples shared between genomic sequences, and distances are independent of the tuple frequency in the genome. Second, we used the *k*-tuple distance based on the Pearson's correlation coefficient (dPear), which takes into account the frequency of the signals. Third, we used the typical *k*-tuple distance (dk), which is based on the frequency of occurrence of the tuples in the genomic sequences and considers the length of these.[2]

The dLog, given two genomes, A and B, was calculated in two steps. First, the Jaccard index (also known as the Jaccard similarity coefficient) was calculated using the formula: $J = M_{11}/M_{01} + M_{10} + M_{11}$, where J is the Jaccard index; M_{11} is the total number of tuples that occur in both A and B genomes; M_{01} is the total number of distinctive tuples for B; and M_{10} is the total number of distinctive tuples for A. Second, the distance value was computed based on the formula: $S(A, B) = -1n\, J/k$, where S is k-tuple distance; J is the Jaccard index; and k is the tuple length.

The dPear given two genomes, A and B, was calculated using the formula: $dPear = 1 - r = 1 - \Sigma_i \overline{A_i - (A) B_i - (B)} \Big/ \left(\sqrt{\Sigma_i \overline{A_i - (A)^2}} \sqrt{\Sigma_i \overline{B_i - (B)^2}} \right)$, where r is the Pearson's correlation coefficient; A_i and B_i correspond to the tuple i's frequencies in sequences A and B, respectively; and $\overline{A}$ and $\overline{B}$ correspond to the average frequencies in sequences A and B, respectively.

The dk for any two sequences, A and B, is calculated using the formula: where A_i and B_i correspond to the tuple i's frequencies ($= counts/n - k + 1$) in sequences A and B, respectively; n is the sequence length of either sequence A or B; and k is the tuple length.

To estimate the accuracy of different tree construction methods we also used the Characters program to generate bootstrap replicates by random sampling of the tuple sets. The Characters program produces a matrix with the original k-tuple distances (original file) and a second output comprising the bootstrap replicates calculated from the original matrix (bootstrap file). We selected 1000 replicates in all cases. The general bootstrapping and tree construction strategies are illustrated in Figure 5.

Dynamic range

We evaluated the dynamic range of LPS9, which represents the capability of this particular set to estimate dLog values within a group of sequences with a wide interval of similarity, in such a way that we could distinguish between these sequences. For this survey, we used the Citrus viroid II genome as a reference. Additionally, we simulated sets of 100 genomic variants each. To evaluate the capability for distinguishing between highly related variants, we used two different approaches, called independent and successive, to introduce single base substitutions randomly in a reference genome. Substitutions were simulated using Perl scripts (Active Perl 5.8). Under the independent approach, single random substitutions were introduced in the reference genome, and the *k*-tuple distance between the reference genome and each variant was measured. In the second approach, successive and accumulative single random substitutions were introduced, and the *k*-tuple distance between each pair of new and previous variants was measured. For both approaches, we registered the minimum, maximum, and average *k*-tuple distances.

To better understand LifePrint cases in which single substitutions are located at the 5′ or 3′ ends, we calculated *k*-tuple distance for two different simulated sets of variants. The first set contained variants with single substitutions at each of the nine positions from 5′ or 3′ ends. The second set comprised variants with accumulative deletions at the 3′ end.

Dynamic range also refers to limit values of similarity between two sequences that can be interpreted to distinguish each other. To evaluate this property, we simulated 15 groups of 100 Citrus viroid II genomic variants with increasing proportion of substitutions. Each variant in a given group was made with the cumulative effect of successive and random single substitutions. The 15 groups were simulated by introduction of 1 (0.5%), 3 (1%), 6 (2%), 9 (3%), 12 (4%), 24 (8%), 36 (12%), 48 (16%), 60 (20%), 72 (24%), 84 (28%), 96 (32%), 120 (40%), 150 (50%), and 200 (66%) successive single base substitutions, respectively. Numbers

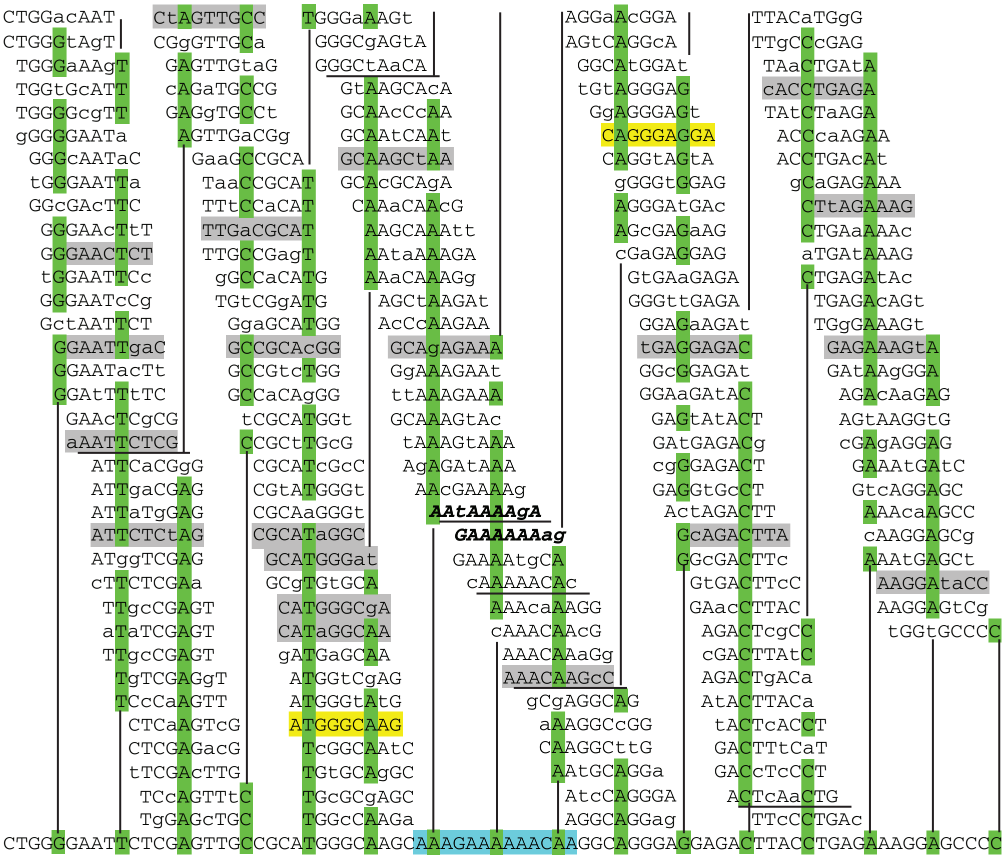
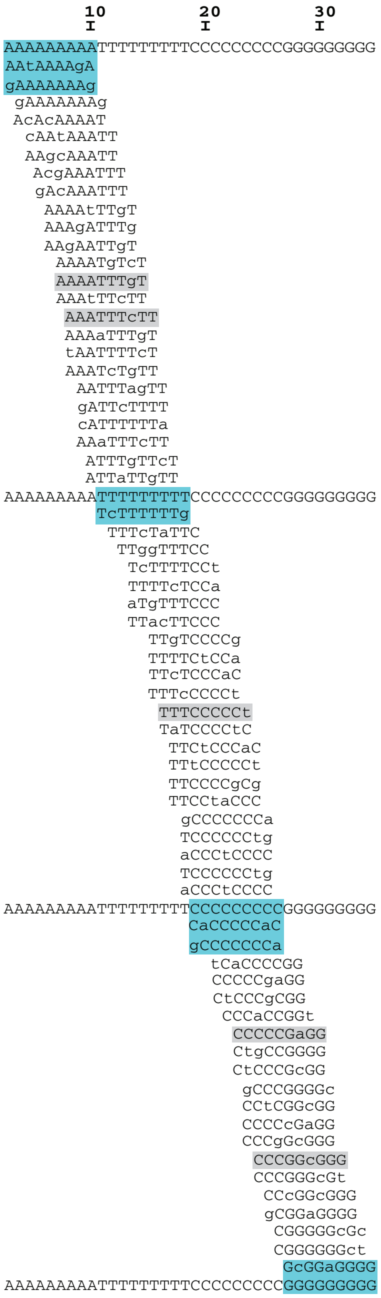


Figure 3 Genomic coverage.[a]

Note: [a]According to their corresponding matching position, we gathered all tuples of the LifePrint set of 9-tuples (LPS9) that detected identity and/or similarity in the first 80 nucleotides (5′ end) of the Hop stunt viroid genome. The coincidences between the most frequent nucleotides and their respective genomic positions are indicated in each column. Every five nucleotides, a green mark is placed as the nucleotide number reference. The identities and differences appear in capital and lower case letters, respectively. The tuples that found identities or sites with one difference are marked in yellow and gray, respectively. Six subsequences that were not detected directly by any tuple (beginning at nucleotide numbers 7, 27, 36, 39, 42, and 60) are underlined. Three of them (beginning at nucleotide numbers 36, 39, and 42) are located in a rich adenine region, which is marked in blue.

in parentheses are percentages of substitutions in relation to lengths of the genomic variants (300 nucleotides). The real number of accumulated substitutions in each variant also was determined using another Perl script able to call the water and needle programs from the EMBOSS software.[25] Later, the average number of accumulated substitutions was calculated for each group. Perl scripts are available at the UFCVH site.



```
        10         20         30
         I          I          I
AAAAAAAAATTTTTTTTTCCCCCCCCCGGGGGGGGG
AAtAAAAgA
gAAAAAAAg
 gAAAAAAAg
 AcAcAAAAT
  cAAtAAATT
  AAgcAAATT
   AcgAAATTT
   gAcAAATTT
    AAAAtTTgT
    AAAgATTTg
    AAgAATTgT
     AAAATgTcT
     AAAATTTgT
     AAAtTTcTT
      AAATTTcTT
      AAAaTTTgT
      tAATTTTcT
      AAATcTgTT
       AATTTagTT
       gATTcTTTT
       cATTTTTTa
       AAaTTTcTT
        ATTTgTTcT
        ATTaTTgTT
AAAAAAAAATTTTTTTTTCCCCCCCCCGGGGGGGGG
         TcTTTTTTg
          TTTcTaTTC
           TTggTTTCC
            TcTTTTCCt
            TTTTcTCCa
            aTgTTTCCC
            TTacTTCCC
              TTgTCCCCg
              TTTTCtCCa
              TTcTCCCaC
              TTTcCCCCt
               TTTCCCCCt
               TaTCCCCtC
                TTCtCCCaC
                TTtCCCCCt
                TTCCCCgCg
                TTCCtaCCC
                 gCCCCCCCa
                 TCCCCCCtg
                 aCCCtCCCC
                 TCCCCCCtg
                 aCCCtCCCC
AAAAAAAAATTTTTTTTTCCCCCCCCCGGGGGGGGG
                  CaCCCCCaC
                  gCCCCCCCa
                    tCaCCCCGG
                    CCCCCgaGG
                    CtCCCgCGG
                     CCCaCCGGt
                      CCCCCGaGG
                      CtgCCGGGG
                      CtCCCGcGG
                       gCCCGGGGc
                       CCtCGGcGG
                       CCCCcGaGG
                       CCCgGcGGG
                        CCCGGcGGG
                        CCCGGGcGt
                         CCcGGcGGG
                         gCGGaGGGG
                          CGGGGGcGc
                          CGGGGGGct
                           GcGGaGGGG
AAAAAAAAATTTTTTTTTCCCCCCCCCGGGGGGGGG
```

Figure 4 LifePrint detection of single base repeats.[a]

Note: [a]Using a model of 36 nucleotides comprising four different single nucleotide repeats, we carried out a similarity search. We tiled 9-tuples according to their matching position along the model. Sequence identities and differences appear in capital and lower case letters, respectively. Tuples able to identify the repeat and the respective genome repeat direct are highlighted in blue.

Accuracy of different tree construction methods

As described in the Genomic sequences section, we simulated the evolution of 32 viroid genomes from the common ancestor Citrus viroid II sequence. We then used the results derived from this simulation to prepare a Newick representation of the true tree manually for further analyses (see below). We visualized and edited the tree (Figure 2) with MEGA 4.0.[26] Later, we used the 32 simulated viroid genomes to reconstruct their phylogeny with our LPS9 method and all possible tuples of size 5. All phylogenetic trees in our study are considered unrooted.

As described in the Similarity search section, we used the Virtual Hybridization program to scrutinize the viroid sequences with our *k*-tuple sets as the query, allowing a defined number of sequence differences. The Virtual Hybridization program locates thermodynamically stable sites for the hybridization of *k*-tuples within genomic sequences.[27] Given that thermodynamic information is not relevant for this study, a free energy cutoff value of 0 was used to identify the matching positions for our *k*-tuples. For the LPS9 search we allowed up to two sequence mismatches, whereas for 5-tuples only identical matches were allowed. For both the LPS9 and 5-tuple searches, we used the Characters program to calculate the three different kinds of *k*-tuple distances (see Calculating and bootstrapping k-tuple distances section) from the global tables (binary- and frequency-based). Phylogenetic trees were estimated for each distance table using the NJ method implemented in the Neighbor program of the PHYLIP 3.69 package. Topologies of the resulting 12 NJ trees were contrasted with the true tree to evaluate the accuracy (see Evaluation of accuracy section) of the *k*-tuple methods.

We used global frequency tables from 36 real viroid genomes and generated bootstrap files for LPS9 and 5-tuples using dPear. We constructed NJ trees using the Neighbor program included in PHYLIP 3.69 package. We then estimated consensus trees (Figures 6 and 7) and subsequently performed visualization and editing with MEGA 4.0.

Evaluation of accuracy

We used the symmetric difference between trees to evaluate the accuracy of the different tree construction methods analyzed here. The symmetric difference or topologic distance is a widely used metric to evaluate differences between phylogenetic trees. It has to be mentioned that symmetric difference estimation is independent of the tree branch lengths. For two unrooted bifurcating trees, the symmetric difference is twice the number of interior branches at which the sequence partitions is different between the compared trees.[28] Another

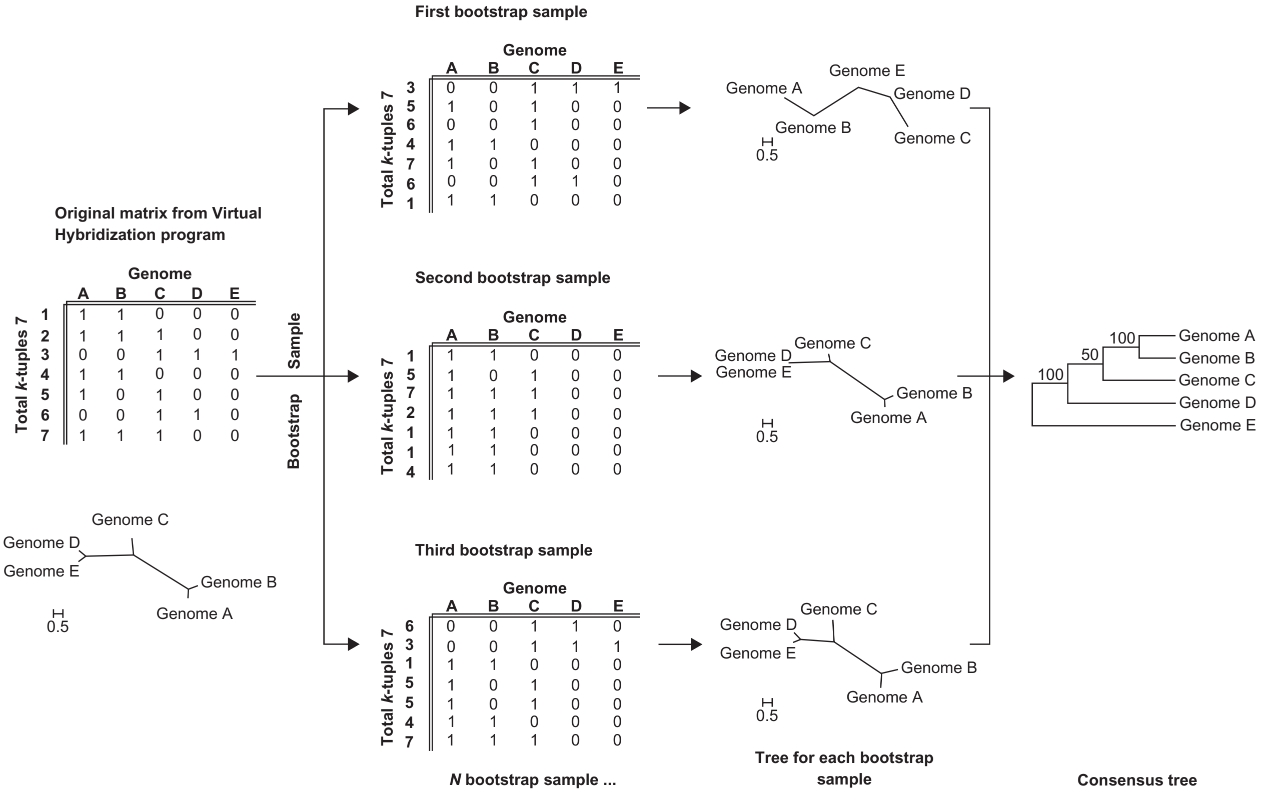


Figure 5 General bootstrapping scheme for *k*-tuple distance and tree construction in LifePrint.[a]
Note: [a]The Virtual Hybridization program generates a matrix for the identity/similarity for each tuple (rows 1 to 7) against each genome sequence (columns A to E). Then, entire rows from the original matrix are randomly sampled with replacement in order to produce a new bootstrap matrix with the same number of rows as the original matrix. A distance table for each bootstrap sample matrix is calculated and used to estimate a phylogenetic tree. Finally, a consensus tree is calculated from all the bootstrap trees. The numbers in the consensus tree show the percentage of abundance of the groups in the bootstrap samples.

equivalent definition of symmetric difference is the distance between a pair of trees based on the number of branches that differ between the trees. Thus, symmetric difference is simply a count of how many partitions there are between the two trees compared.[29] Briefly, we conducted a paired comparison of each NJ tree against the true tree. We estimated the symmetric difference between the phylogenetic trees using the Treedist program included in the PHYLIP 3.69 package.

Using the Phylocomparison program,[30] we compared the topologies for LPS9 and 5-tuples NJ trees constructed with dPear (from the original Characters file) and the true tree. See Appendix II for graphic representations of these comparisons.

Finally, we compared bootstrap support values between the consensus NJ trees obtained for the 36 real viroid genomes and the viroid phylogeny proposed in the International Committee on Taxonomy of Viruses.[31]

Results

LPS9

Our grouping criteria (see Methods) generated the following subsets. After the substitution criterion, the number of 9-tuples diminished from 262,144 to 29,868 tuples (a circa eight-fold reduction). The following block criterion produced 4206 tuples and finally the refining criterion identified the final set of 1878 9-tuples defining LPS9. In Figure 1, each line represents 10,000 9-tuples. The number of tuples of the LPS9 in every line changes from 64 to 84. On average, a tuple of the LPS9 is selected by every 144 of the complete set of 262,144 9-tuples. The homogeneous distribution of the LPS9 inside the complete set of 9-tuples allowed us to consider it to be a representative sample.

Similarity search

Table 1 compiles the average number of tuples sharing identity and/or similarity under four different conditions (see Methods). The optimal scheme is reached at condition C (allowing up to two differences). We consider condition C to be our optimal scheme given that the proportion of 9-tuples with identity and/or similarity with the genomes is between 20% and 80% of the LPS9. Under this scheme, we avoid both underutilization and saturation of LPS9. All subsequent similarity searches (with the exception of the Accuracy of different tree construction

Table 1 Number of LifePrint sets of identical and/or similar 9-tuples (LPS9) under four different similarity search schemes[a]

Conditions	Allowed differences between sequences	Average number of identical and/or similar LPS9 tuples in sequences	Percentage in relation to the number of tuples of the LPS9
A	0	3	0.2
B	0 and 1	64	3.4
C	0, 1, and 2	605	32.2
D	0, 1, 2, and 3	1705	90.8

Note: [a]We carried out a similarity search between LPS9 and 36 viroid genomes, allowing a different number of differences between the sequences. We calculated the average number of 9-tuples that are identical and/or similar found in four different conditions (A, B, C, and D).

methods section) were carried out under condition C. In conditions A and B the LPS9 is underutilized, whereas in condition D it reaches saturation.

The results of the similarity search included the sequences of the tuples that were sharing identity (no differences) or similarity (one or two differences) with subsequences in the genomes.

Each one of 1878 tuples of the LPS9 scrutinized 352 different sequences (one identical one, 27 allowing one difference, and 324 allowing two differences), which means that 661,056 (1878 × 352) sequences of 9-mer are searched. Given that all possible 9-mer sequences are 262,144, it is

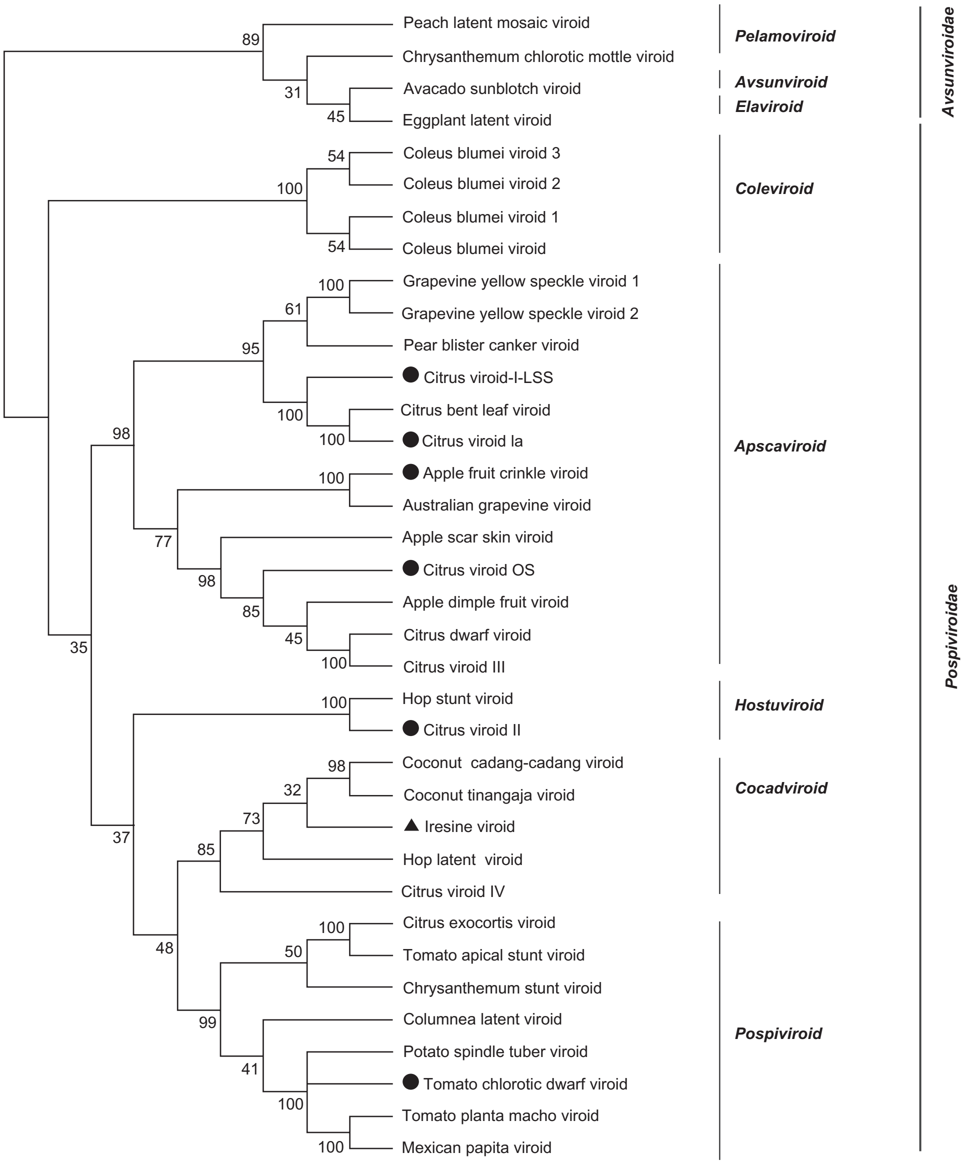


Figure 6 LifePrint set of 9-tuples (LPS9) bootstrap consensus tree from 36 real viroid genomes (*k*-tuple distance based on Pearson's correlation coefficient, 1000 replicates).[a]
Note: [a]Families were assigned according to the International Committee on Taxonomy of Viruses classification. Numbers represent bootstrap confidence values for the sequence groups. The black circles correspond to unclassified viroid genomes. The black triangle corresponds to a viroid that should properly be grouped in the subfamily Pospiviroid.

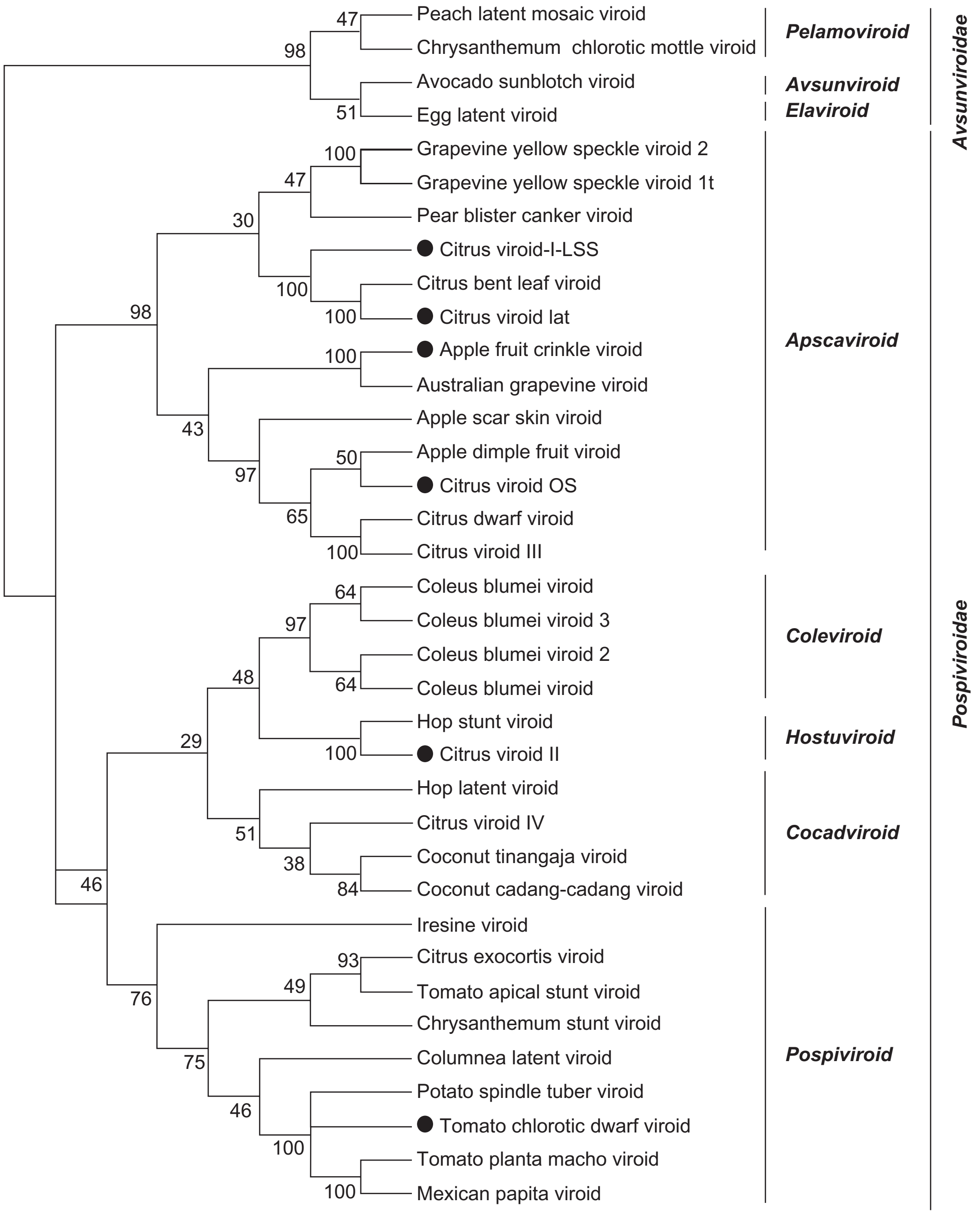


Figure 7 The 5-tuple method bootstrap consensus tree from 36 real viroid genomes (*k*-tuple distance based on Pearson's correlation coefficient, 1000 replicates).[a]
Note: [a]Families were assigned according the International Committee on Taxonomy of Viruses classification. Numbers represent bootstrap confidence values for the sequence groups. The black circles correspond to unclassified viroid genomes.

expected that every sequence would be searched, on average, 2.5 times. In fact, a genome 300 nucleotides in length, ie, containing 292 subsequences 9-mer, is covered by 605 tuples of the LPS9, which corresponds to an average of 2.05 tuples per sequence.

Genomic coverage

LifePrint has an advantage in comparison with the original *k*-tuple distance methods, given that, under the optimal similarity search scheme, the tuple length of nine nucleotides provides complete coverage of the analyzed sequences.

Figure 3 shows that in the first 80 nucleotides of the Hop stunt viroid, six 9-mer subsequences were not detected directly by LPS9. The region with minor coverage (from nucleotide numbers 34–46, marked in blue) has a high adenine (A) composition. In addition, this region has three of six subsequences that were not detected directly by LPS9 (beginning at nucleotide numbers 32, 39, and 42). Only two 9-tuples (AATAAAAGA and GAAAAAAAG) shared similarity with this region. Even though LPS9 did not detect all possible 9-mer subsequences directly, our 9-tuple set was capable of fully covering the genomic sequences in this study. Every position within the genomes was recognized by several tuples, which increases the sensibility to detect simple changes. This property is particularly relevant in the case of single nucleotide substitutions.

Detection of single base repeats

Given that the design of LifePrint implies the selection of sequences with a minimum of entropy, LPS9 is not able to identify regions of low complexity (ie, sequence repeats) directly. The results obtained with repeat model sequencing (see Methods) show that two 9-tuples comprising seven with A, one with seven T, two with seven C, and one with seven G (Figure 4, all in bold type) were capable of detecting subsequences of nine consecutive and identical nucleotides.

How important are low complexity regions to an accurate phylogeny reconstruction? What happen if these regions are not included in the analysis? To answer these questions, it is important to distinguish whether or not the region of interest encodes protein products. Coding or noncoding regions tend to evolve by different mechanisms. When genomic regions encode proteins, even those comprising repeated regions, changes are, in principle, constrained to synonymous substitutions or mutations producing conserved amino acids. In these cases, it is important to consider such regions in the estimation of evolutionary distances, although their impact is also related to the proportion that they represent in the genome, and if they are present in the other genome sequences. However, noncoding regions are not exposed to the same evolutionary pressures as are coding regions. Most mutations are neutral in the noncoding zones, but some substitutions may follow complex evolutionary mechanisms (eg, covariation), such as the case of noncoding sequencing important for other functions (eg, regulation of gene expression).

Dynamic range

Here we selected dLog to identify shared tuples between genomes independently of the frequency that tuples are present in the viroid genomes, and the length of the particular genome. Table 2 shows the results obtained from independent and successive approaches (see Methods section). Under the independent approach we obtained a value of 0 in a variant with a substitution in the end 3′.

Analysis of variants with substitutions or eliminations located in the 5′ or 3′ ends revealed that only in a few cases were sequence ends presenting punctual changes not detected directly by the LPS9. This result indicates that the ability to discern between variants was not affected. Table 3 shows *k*-tuple distance values between variants mentioned above and the Citrus II viroid genome.

In Table 2, the *k*-tuple average distance for a single substitution presents values from 0.00378 to 0.00390. We examined the list of tuples involved in detecting a simple substitution that implies a *k*-tuple distance within the mentioned interval. We selected the simulated variant 144 A→G, which presents a *k*-tuple distance of 0.00390 in relation to the Citrus II viroid genome. In Figure 8, we list the tuples that detected the substitution A for G in the position 144. It has to be noted that 20 tuples are distinctive in this position, 15 for A and five for G. Figure 8 explains graphically how LPS9 detects efficiently simple substitutions.

In order to estimate the limits on the degree of relatedness between two sequences, which putatively will allow us to distinguish between two closely related sequences, the results depicted in Table 4 indicate that LifePrint reaches saturation at approximately 40% of substitutions. It is expected that when a critical number of variants is included in the phylogenetic study, a given variant considerably distant to

Table 2 *k*-tuple distance values on single substitutions variants[a]

Value	Independent	Successive
Minimum	0.000000	0.00040
Maximum	0.005894	0.00580
Average	0.003780	0.00390

Note: [a]We calculated the minimum, maximum, and average values of the *k*-tuple distances for variants of the Citrus II viroid genome using independent and successive approaches described in the Methods section.

Table 3 *k*-tuple distance values for variants with single substitutions or eliminations located in the ends of sequences[a]

Position from end	*k*-tuple distance			Deleted nucleotides (n)
1	0.00000	0.00069	0.000000	0
2	0.00069	0.00117	0.000380	1
3	0.00082	0.00179	0.000580	2
4	0.00137	0.00248	0.000580	3
5	0.00145	0.00331	0.000960	4
6	0.00255	0.00324	0.000962	5
7	0.00234	0.00441	0.001349	6
8	0.00381	0.00531	0.001543	7
9	0.00426	0.00552	0.002128	8
	5′ end	3′ end		

Note: [a]We calculated the *k*-tuple distance between every variant and the Citrus II viroid genome. In the second and third columns we present the results for three possible single substitutions in the 5′ or 3′ ends. In the fourth column the results represent the combined effect of successive eliminations in the 3′ end and the resulting substitutions in the new ends.

another sequence will be closer (eg, more similar) to some other variant. Therefore, the *k*-distance saturation should not be a limitation for the construction of trees when many strains are included in the analysis.

Evaluation of accuracy

In Table 5, we summarize the results of symmetric difference comparisons between 12 different NJ trees and the true tree (see Construction of trees in Methods section). These results indicate that both methods based on 9-tuples and 5-tuples, respectively, fail to recover the true tree. These findings illustrate that the metric used for the *k*-tuple distances results in different accuracies of tree reconstruction. See Appendix II for a graphic representation of topologic comparisons between the true tree and the LPS9 and 5-tuple NJ trees constructed with dPear.

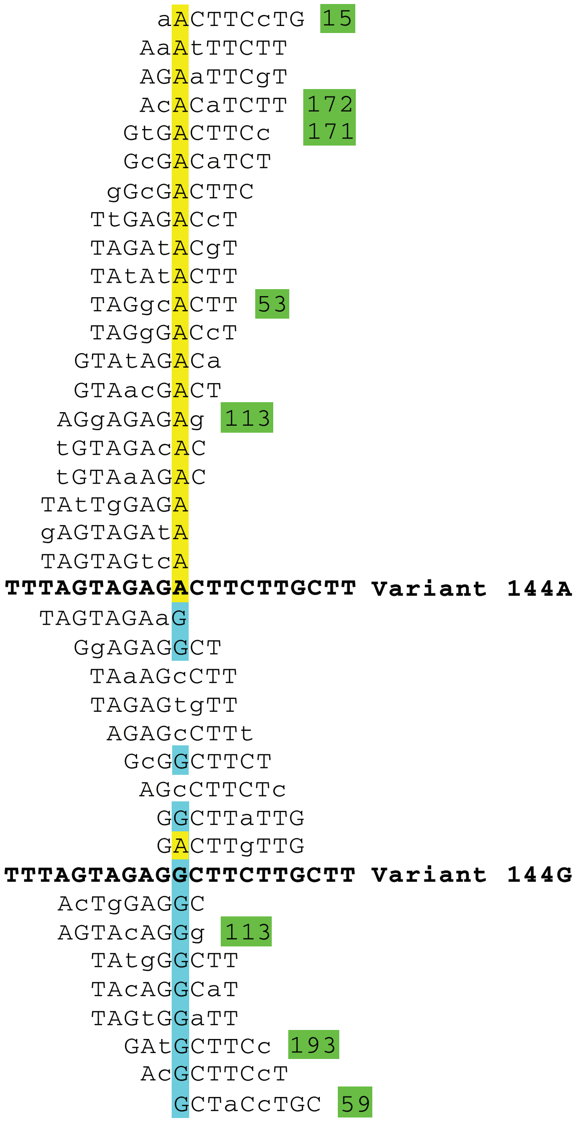
Visual inspection of the NJ trees from the 36 real viroid genomes indicates that the bootstrap support values are higher for trees reconstructed with the LPS9 method than for trees derived from the 5-tuple method. For both 5-tuple trees and LPS9-based trees, low bootstrap proportion values (less than 30%) were observed in those clades inconsistent with the true tree. Therefore, it seems that the bootstrap test can be used confidently as a proxy to evaluate the accuracy of the reconstruction.

Figures 6 and 7 indicate that although both reconstructions are consistent with each other, bootstrap proportion values are higher for the tree based on 9-tuples. The major viroid families were identified by our 9-tuple methods, although some clusters are organized differently, such as the case of the Avsunviroidae members. However, such conflicts are associated with relatively low bootstrap confidence values.

In our view, branch length comparisons between trees are critical when topologies have been estimated using the same optimization criteria. However, in this particular case, we are evaluating only different *k*-tuple (LifePrint versus 5-tuple) distance methods including trees from character-based methods, such as maximum likelihood or maximum parsimony, which in our assessment would produce uncertain comparisons, given that each optimization criterion reflects different change measurements in the branch lengths. For the purposes of this work, we consider it adequate to constrain the topologic comparison only between trees obtained with *k*-tuple distance methods.

Conclusion

Previous studies suggest that phylogenetic reconstructions based on 5-tuples work better than those based on 9-tuples. However, the LifePrint approach is different from the 5-tuple) distance methods including trees from character-based methods, such as maximum likelihood or maximum parsimony, which in our assessment would produce uncertain comparisons, given that each optimization criterion reflects different change measurements in the branch lengths. For the purposes of this work, we consider it adequate to constrain the topologic comparison only between trees obtained with *k*-tuple distance methods.



```
         aACTTCcTG 15
        AaAtTTCTT
        AGAaTTCgT
        AcACaTCTT 172
       GtGACTTCc  171
       GcGACaTCT
      gGcGACTTC
     TtGAGACcT
     TAGAtACgT
     TAtAtACTT
     TAGgcACTT 53
     TAGgGACcT
    GTAtAGACa
    GTAacGACT
   AGgAGAGAg 113
   tGTAGAcAC
   tGTAaAGAC
  TAtTgGAGA
  gAGTAGAtA
  TAGTAGtcA
TTTAGTAGAGACTTCTTGCTT Variant 144A
  TAGTAGAaG
    GgAGAGGCT
     TAaAGcCTT
     TAGAGtgTT
      AGAGcCTTt
       GcGGCTTCT
        AGcCTTCTc
         GGCTTaTTG
         GACTTgTTG
TTTAGTAGAGGCTTCTTGCTT Variant 144G
   AcTgGAGGC
   AGTAcAGGg 113
     TAtgGGCTT
     TAcAGGCaT
     TAGtGGaTT
       GAtGCTTCc 193
        AcGCTTCcT
          GCTaCcTGC 59
```

Figure 8 Differential detection of variants with a single substitution that implies an average *k*-tuple distance.[a]

Note: [a]We calculated the *k*-tuple distance between simulated variant 144 A→G and the Citrus II viroid genome. Tuples of the LifePrint set of 9-tuples (LPS9) that found identity or similarity in both sequences in this region of interest are between both sequences (in bold type), and the distinctive tuples are placed above or below the respective sequence. Substitution is marked in yellow when there is identity with A and in blue when the identity is with G. We highlight with green other positions where the same tuples find identity or similarity. These tuples were not considered to be distinctive.

Table 4 Ability of LifePrint to distinguish between sequences with different degree of relatedness[a]

Single substitutions			*k*-tuple distance			σ
Real	**Observed**		**Minimum**	**Maximum**	**Average**	
	Number	**Percentage**				
1	1.00	0.334	0.00134	0.00589	0.00382	0.00098828
3	2.97	0.993	0.00423	0.01575	0.01150	0.00187987
6	5.96	1.993	0.01437	0.02847	0.02208	0.00287570
9	8.89	2.973	0.02312	0.04112	0.03246	0.00361394
12	11.82	3.953	0.03117	0.04995	0.04205	0.00401709
24	22.82	7.632	0.06276	0.09471	0.07808	0.00704353
36	32.91	11.006	0.09334	0.13755	0.11449	0.01101853
48	42.99	14.378	0.11396	0.21124	0.16224	0.01946713
60	51.61	17.261	0.14709	0.29580	0.21528	0.03098441
72	60.87	20.358	0.17289	0.42443	0.28682	0.04982284
84	68.55	22.927	0.22703	0.76171	0.36894	0.08921467
96	75.29	25.181	0.26546	0.76253	0.45189	0.12901015
120	87.86	29.385	0.30701	0.76664	0.61653	0.14854929
150	99.86	33.398	0.35974	0.76852	0.72943	0.07186317
200	116.55	38.980	0.39897	0.77006	0.76087	0.01373281

Note: [a]Fifteen groups of 100 Citrus viroid II virtual variants containing an average of 1–116 substitutions were created. The minimum, maximum, average, and standard deviation of *k*-tuple distance between each variant and the original viroid were determined. Column 3 (percent) is computed by column 2 (number) divided by 299.

previously described *k*-tuple method, given that the present method allows a defined number of sequence differences, whereas dk methods search for perfect coincidences. Therefore, the performance is, in principle, different.

Our analyses indicate that distances based on LPS9 enable more accurate reconstructions than distances estimated by the 5-tuples method. Bootstrap support values were also higher for trees reconstructed from LPS9 than for those trees derived from 5-tuples. Moreover, dLog and dPear work better for binary and frequency tables, respectively.

Our results are consistent with previous findings that suggest that *k*-tuple distance methods are more accurate than phylogenetic methods based on multiple sequence alignment.[2] It should be noted that LifePrint uses a representative sample of all possible 9-tuples to estimate distance between genomic sequences. This characteristic will allow us to implement approaches that would save a considerable amount of time for phylogenetic reconstructions using the entire sequence information available in genome-scale data.

However, our results do not provide definitive evidence to distinguish the more accurate *k*-tuple method tested in this study. As mentioned, the fact that LifePrint allows a certain number of sequence differences, in contrast with the identity of the 5-tuples, may explain discrepancies between these methods. Our results indicate that the accuracy of particular *k*-tuple reconstruction methods depends on both the length of the target sequences and the similarity between the sequences. In further studies, it will be important to evaluate the relationship between these two factors comprehensively to determine the length and sequence characteristics of the "ideal" *k*-tuple method, and in parallel to explore its intrinsic accuracy.

Other areas to be explored using our 9-tuple method include the incorporation of position-sensible strategies to identify regions of sequence identity or similarity, and the use of parsimony optimization criteria to estimate trees from our similarity search scheme. Additionally, the use of *k*-tuple methods using amino acid-based data would be

Table 5 Symmetric difference values between true tree and neighbor-joining trees constructed from *k*-tuple distance based on three different distances metrics[a]

Distance metrics	LPS9 (from global binary table)	LPS9 (from global frequency table)	5-tuple (from global binary table)	5-tuple (from global frequency table)
dLog	10	10	18	18
dPear	14	6	18	26
dk	14	8	18	26

Note: [a]We measured the accuracy of the LifePrint set of 9-tuples (LPS9) and complete set of 5-tuple methods by comparing each neighbor-joining tree with the true tree using a symmetric difference.

Abbreviations: dLog, *k*-tuple distance based on the Jaccard index; dPear, *k*-tuple distance based on the Pearson's correlation coefficient; dk, typical *k*-tuple distance.

important to analyze protein encoding regions and/or highly divergent related genomes.

Acknowledgment

Support from the National Polytechnic Institute and CONACYT-Mexico for this research is gratefully acknowledged.

Disclosure

The authors report no conflicts of interest in this work.

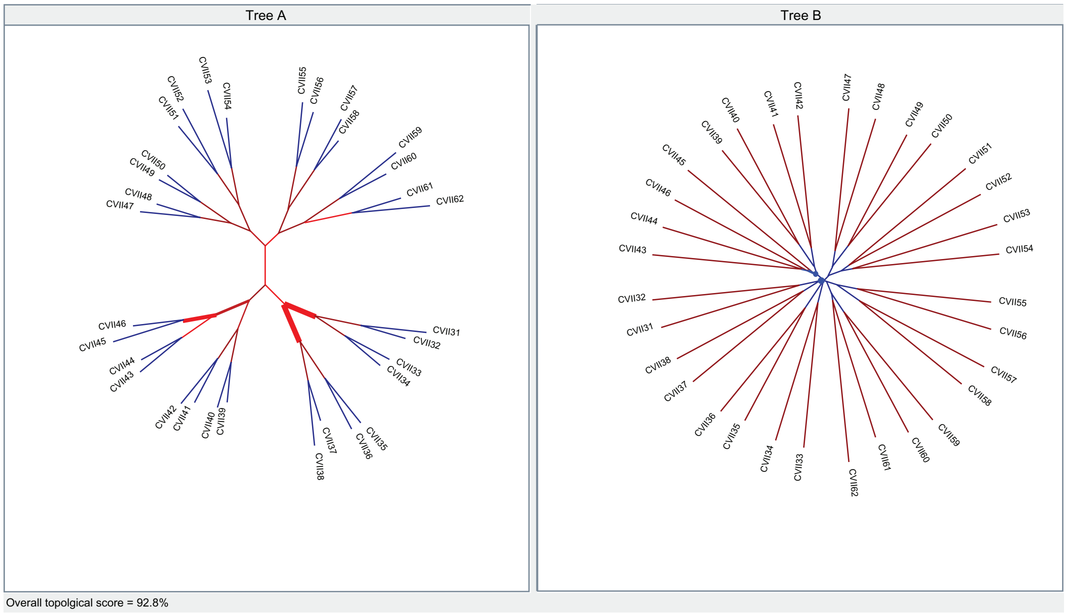
References

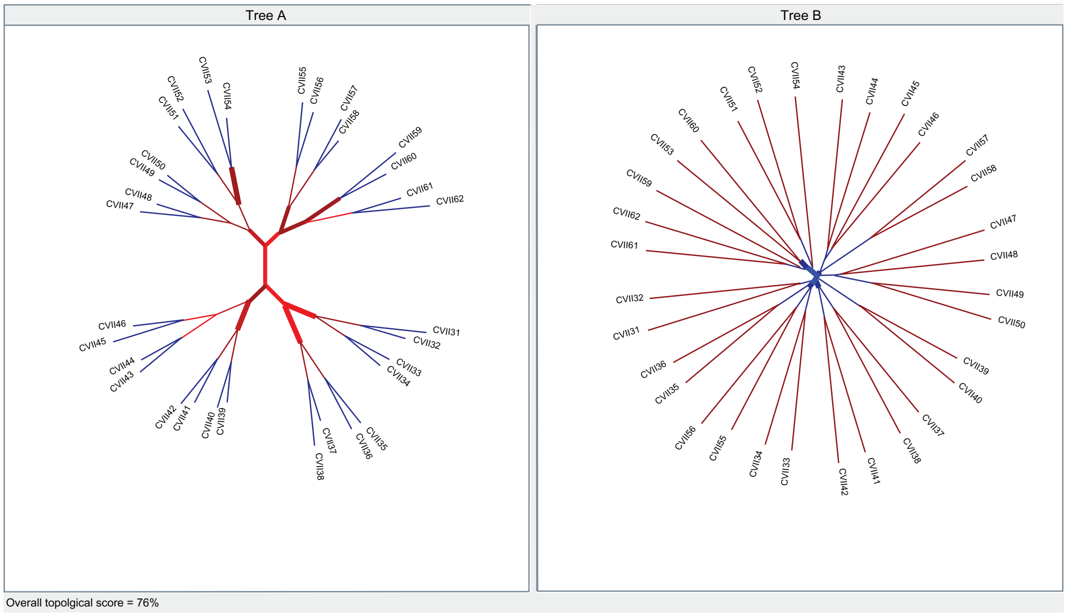
1. Saitou N, Nei M. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol Biol Evol*. 1987;4(4):406–425.
2. Yang K, Zhang L. Performance comparison between *k*-tuple distance and four model-based distances in phylogenetic tree reconstruction. *Nucleic Acids Res*. 2008;36(5):e33.
3. Whelan S. Inferring trees. In: Keith JM, editor. *Bioinformatics, Volume 1: Data, Sequence Analysis and Evolution*. Totowa, NJ: Humana Press; 2008.
4. Edgar RC. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*. 2004;32(5):1792–1797.
5. Morgenstern B. DIALIGN 2: Improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*. 1999;15(3): 211–218.
6. Notredame C, Higgins DG, Heringa J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*. 2000; 302(1):205–217.
7. Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*. 1994;22(22):4673–4680.
8. Lassmann T, Sonnhammer EL. Kalign – an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*. 2005;6:298.
9. Snel B, Bork P, Huynen M. Genome evolution. Gene fusion versus gene fission. *Trends Genet*. 2000;16(1):9–11.
10. Herniou EA, Luque T, Chen X, et al. Use of whole genome sequence data to infer baculovirus phylogeny. *J Virol*. 2001;75(17):8117–8126.
11. House CH, Fitz-Gibbon ST. Using homolog groups to create a whole-genomic tree of free-living organisms: An update. *J Mol Evol*. 2002; 54(4):539–547.
12. Dutilh BE, Snel B, Ettema TJ, Huynen MA. Signature genes as a phylogenomic tool. *Mol Biol Evol*. 2008;25(8):1659–1667.
13. Milosavljević A. Discovering sequence similarity by the algorithmic significance method. *Proc Int Conf Intell Syst Mol Biol*. 1993;1: 284–291.
14. Chen X, Kwong S, Li M. A compression algorithm for DNA sequences and its applications in genome comparison. *Genome Inform Ser Workshop Genome Inform*. 1999;10:51–61.
15. Chen X, Li M, Ma B, Tromp J. DNA Compress: Fast and effective DNA sequence compression. *Bioinformatics*. 2002;18(12):1696–1698.
16. Wu X, Cai Z, Wan XF, Hoang T, Goebel R, Lin G. Nucleotide composition string selection in HIV-1 subtyping using whole genomes. *Bioinformatics*. 2007;23(14):1744–1752.
17. Lu G, Zhang S, Fang X. An improved string composition method for sequence comparison. *BMC Bioinformatics*. 2008;9 Suppl 6:S15.
18. Gao L, Qi J. Whole genome molecular phylogeny of large dsDNA viruses using composition vector method. *BMC Evol Biol*. 2007;7:41.
19. Karlin S, Burge C. Dinucleotide relative abundance extremes: A genomic signature. *Trends Genet*. 1995;11(7):283–290.
20. Stuart GW, Moffett K, Baker S. Integrated gene and species phylogenies from unaligned whole genome protein sequences. *Bioinformatics*. 2002; 18(1):100–108.
21. Qi J, Wang B, Hao BI. Whole proteome prokaryote phylogeny without sequence alignment: A K-string composition approach. *J Mol Evol*. 2004;58(1):1–11.
22. Felsenstein J. PHYLIP – Phylogeny Inference Package (Version 3.2). *Cladistics*. 1989;5:164–166.
23. Swofford DL. *PAUP*.Phylogenetic Analysis Using Parsimony (*and Other Methods).Version 4*. Sunderland, MA: Sinauer Associates; 2003.
24. The Universal Fingerprinting Chip Applications Server. Available at: http://biomedbiotec.encb.ipn.mx/UFCVH/. Accessed Dec 22, 2010.
25. Rice P, Longden I, Bleasby A. EMBOSS: The European Molecular Biology Open Software Suite. *Trends Genet*. 2000;16(6):276–277.
26. Tamura K, Dudley J, Nei M, Kumar S. MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) Software Version 4.0. *Mol Biol Evol*. 2007;24(8):1596–1599.
27. Reyes-Lopez MA, Méndez-Tenorio A, Maldonado-Rodríguez R, Doktycz MJ, Fleming JT, Beattie KL. Fingerprinting of prokaryotic 16S rRNA genes using oligodeoxyribonucleotide microarrays and virtual hybridization. *Nucleic Acids Res*. 2003;31(2):779–789.
28. Nei M, Kumar S. *Molecular Evolution and Phylogenetics*. New York, NY: Oxford University Press; 2000.
29. Felsenstein J. *Inferring Phylogenies*. Sunderland, MA: Sinauer Associates; 2003.
30. Nye TM, Liò P, Gilks WR. A novel algorithm and web-based tool for comparing two alternative phylogenetic trees. *Bioinformatics*. 2006; 22(1):117–119.
31. Flores R, Randles JW, Owens RA, Bar-Joseph M, Diener TO. Subviral agents: Viroids. In: Fauquet CM, Mayo MA, Maniloff J, Desselberger U, Ball LA, editors. *Virus Taxonomy. VIIIth Report of the International Committee on Taxonomy of Viruses*. New York, NY: Elsevier Academic Press; 2005.

Appendices

Apple dimple fruit viroid, complete genome	NC_003463
Apple fruit crinkle viroid, complete genome	NC_003777
Apple scar skin viroid, complete genome	NC_001340
Australian grapevine viroid, complete genome	NC_003553
Avocado sunblotch viroid, complete genome	NC_001410
Chrysanthemum chlorotic mottle viroid, complete genome	NC_003540
Chrysanthemum stunt viroid, complete genome	NC_002015
Citrus bent leaf viroid, complete genome	NC_001651
Citrus dwarf viroid, complete genome	NC_005821
Citrus exocortis viroid, complete genome	NC_001464
Citrus viroid Ia, complete genome	NC_001907
Citrus viroid II, complete genome	NC_003881
Citrus viroid III, complete genome	NC_003264
Citrus viroid IV, complete genome	NC_003539
Citrus viroid OS, complete genome	NC_004359
Citrus viroid-I-LSS, complete genome	NC_004358
Coconut cadang-cadang viroid, complete genome	NC_001462
Coconut tinangaja viroid, complete genome	NC_001471
Coleus blumei viroid 1, complete genome	NC_003681
Coleus blumei viroid 2, complete genome	NC_003682
Coleus blumei viroid 3, complete genome	NC_003683
Coleus blumei viroid, complete genome	NC_003882
Columnea latent viroid, complete genome	NC_003538
Eggplant latent viroid, complete genome	NC_004728
Grapevine yellow speckle viroid 1, complete genome	NC_001920
Grapevine yellow speckle viroid 2, complete genome	NC_003612
Hop latent viroid, complete genome	NC_003611
Hop stunt viroid, complete genome	NC_001351
Iresine viroid, complete genome	NC_003613
Mexican papita viroid, complete genome	NC_003637
Peach latent mosaic viroid, complete genome	NC_003636
Pear blister canker viroid PBCVd, complete genome	NC_001830
Potato spindle tuber viroid, complete genome	NC_002030
Tomato apical stunt viroid, complete genome	NC_001553
Tomato chlorotic dwarf viroid, complete genome	NC_000885
Tomato planta macho viroid, complete genome	NC_001558

Appendix 1 National Center for Biotechnology Information access numbers of 36 real viroid genomes.





Appendix II For LPS9 and 5-tuple neighbor-joining trees constructed with dPear (Characters original file) and true tree we compared the topologies using the Phylocomparison program. The first figure compared the topologies of the true tree (Tree A) and the LPS9 NJ-dPear tree (Tree B). The second figure compared topologies of true tree (Tree A) and 5-tuple neighbor-joining dPear tree (Tree B). In both figures, thicker lines show a poor match. Topologic score is proportional to line thickness, ie, for major thickness the difference in this clade is bigger. Also, they appear in the low part of the image as overall topologic scores. Observing the figures and overall topologic scores, it can be established that topologic differences are evidently minor between the true tree and the LPS9 NJ-dPear tree.