SOURCE CODE FOR BIOLOGY AND MEDICINE

SOFTWARE REVIEW Open Access

Nephele: genotyping via complete composition vectors and MapReduce

Marc E Colosimo*, Matthew W Peterson, Scott Mardis and Lynette Hirschman

Abstract

Background: Current sequencing technology makes it practical to sequence many samples of a given organism, raising new challenges for the processing and interpretation of large genomics data sets with associated metadata. Traditional computational phylogenetic methods are ideal for studying the evolution of gene/protein families and using those to infer the evolution of an organism, but are less than ideal for the study of the whole organism mainly due to the presence of insertions/deletions/rearrangements. These methods provide the researcher with the ability to group a set of samples into distinct genotypic groups based on sequence similarity, which can then be associated with metadata, such as host information, pathogenicity, and time or location of occurrence. Genotyping is critical to understanding, at a genomic level, the origin and spread of infectious diseases. Increasingly, genotyping is coming into use for disease surveillance activities, as well as for microbial forensics. The classic genotyping approach has been based on phylogenetic analysis, starting with a multiple sequence alignment. Genotypes are then established by expert examination of phylogenetic trees. However, these traditional single-processor methods are suboptimal for rapidly growing sequence datasets being generated by next-generation DNA sequencing machines, because they increase in computational complexity quickly with the number of sequences.

Results: Nephele is a suite of tools that uses the complete composition vector algorithm to represent each sequence in the dataset as a vector derived from its constituent k-mers by passing the need for multiple sequence alignment, and affinity propagation clustering to group the sequences into genotypes based on a distance measure over the vectors. Our methods produce results that correlate well with expert-defined clades or genotypes, at a fraction of the computational cost of traditional phylogenetic methods run on traditional hardware. Nephele can use the open-source Hadoop implementation of MapReduce to parallelize execution using multiple compute nodes. We were able to generate a neighbour-joined tree of over 10,000 16S samples in less than 2 hours.

Conclusions: We conclude that using Nephele can substantially decrease the processing time required for generating genotype trees of tens to hundreds of organisms at genome scale sequence coverage.

Background

In the post-genomic era, as sequencing becomes ever cheaper and more routine, biological sequence analysis has provided many useful tools for the study and combat of infectious disease. These tools, which can include both experimental and computational methods, are important for molecular epidemiological studies [1-3], vaccine development [4-6], and microbial forensics [7-9]. One such method is genotyping, the grouping of samples based on their genetic sequence. This can be done experimentally [10-12] or computationally, either by identifying genetic signatures (nucleotide substrings which are only found in a single group of sequences) [13], or on the basis of genetic distance among the sequences [14-16]. These methods allow a researcher to split a group of sequences into distinct partitions for further analysis. In a forensics context, genotyping a sequence can yield clues on where the sequence comes from. In surveillance, genotyping can be used to examine the evolutionary footprint of a pathogen, for example, to identify areas where certain vaccines and other countermeasures should be used.

* Correspondence: mcolosimo@mitre.org
The MITRE Corporation, 202 Burlington Rd, Bedford MA 01730, USA

Sequence-based comparison involves three major steps. The first is to choose a set of sequences to study, based on some criteria, such as strain, time period or geographic region. Ideally, this set can be easily extracted from a well-populated reference database, containing not only the sequence data for the samples of interest, such as a particular serotype of *Influenza*, but also sufficient metadata. For infectious diseases, types of metadata include geospatial and temporal co-ordinates, host information, and pathogenicity. Once the appropriate dataset is chosen, the samples are compared and clustered in sequence space. From here, the metadata associated with the sequences is used to assess the evolutionary landscape of the organism or pathogen [17].

Sequence Comparison Methods

Traditionally, the first step in performing sequence comparisons is to generate a multiple sequence alignment (MSA) from the sequences of interest. This is most often done using heuristics found in utilities such as CLUSTAL W [18], MUSCLE [19], T-COFFEE [20], and ProbCons [21]. The dynamic programming solution, which can find the mathematically but not necessarily biologically correct solution, quickly becomes impractical with the sample sizes used in any meaningful analysis. A recent review [22] examined many of the issues in producing these alignments, most notably the trade-offs between alignment accuracy, time, and computational expense. Many of the most accurate algorithms cannot be used on a large number of sequences, or on very lengthy sequences, and were only recommended for sets of less than 100 sequences.

Because the alignment is dependent on each of the sequences from which it is calculated, the alignment must be recomputed whenever a new sequence is added. This becomes problematic for surveillance applications, where new sequences will be added constantly. While this problem has been mitigated to some extent using with algorithms such as Near-Alignment Space Termination (NAST) [23], this still adds a level of complexity if the dataset is continually growing, as is the case with Influenza and other infectious diseases. Another issue is that different heuristics will yield different alignments – they are only designed to find an acceptable answer, not the optimal alignment. While methods have been developed to find a "consensus alignment" [24] from a set of alignments, this requires a good deal of time and computing power.

The composition vector (CV) [25] method has been used to describe DNA/RNA and protein sequences as vectors, using the distance between these vectors as the genetic distance. This method involves using a sliding window to represent each sequence as a vector, where each element of the vector is calculated based on the actual and expected frequency of the k-mer (DNA/protein subsequence of length k) observed in that window. The vector representation allows the distance between two sequences to be calculated with any standard distance metric. The CV method was shown to produce trees which matched established taxonomies, as inferred from the 16S RNA segment by more conventional alignment-based methods [26]. The CV method was later expanded into the complete composition vector (CCV) method [27], which uses sliding windows over a range of lengths to describe the sequence. Since these methods do not require alignments to be calculated, distances calculated between sequences remain constant, rather than being dependent on the set of sequences being examined, making these methods ideal for the handling of rapidly growing datasets. No molecular models need be used to calculate distances – distances are calculated using any distance metric that can be used to calculate the distance between vectors.

The next step in sequence analysis is the clustering of the sequences. Traditionally, this is done by inferring a phylogenetic tree. Tools for this purpose include PHYLIP [28], PAUP* [29], or POY [30]. This work was initially performed using distance-based methods, such as the UPGMA or neighbour-joining algorithms [31], or cladistic methods such as Maximum Parsimony. As computational power increased, methods that inferred trees based on models of evolution were used. These include the Maximum Likelihood technique, as well as Bayesian Inference. While these methods produce phylogenetic trees, which provide a useful visualization, any further analysis and grouping must be performed manually. As the number of sequences to compare increases, this becomes more and more difficult. In fact, there has been much recent research into new methods to visualize phylogenetic trees with large numbers of leaves [32,33]. In addition, the phylogenetic tree view proves difficult to integrate with the metadata. For example, a recent paper discussing the spread of H5N1 Avian Influenza used Google Earth to draw a phylogenetic tree on top of the globe [34]. While this visualization works well for a small number of samples, it is ineffective for larger datasets, due to the "busyness" of the visualization.

Computational Genotyping

An alternative to the pure phylogenetic approach is computational genotyping. This involves partitioning the set of sequences into discrete groups, based on some criteria. This can be based on differences between known subtypes, such as tandem repeats or single nucleotide polymorphisms, or by genetic distance. In the case of genotyping based on distance, this becomes a clustering problem. In 2007, Frey and Dueck published a paper on a new clustering algorithm known as affinity propagation clustering [35]. In contrast to other clustering algorithms, such as k-means and Expectation Maximization (EM), the affinity propagation algorithm does not require the user to explicitly select a given number of exemplars at the start of clustering. Instead, affinity propagation simultaneously considers all points as potential exemplars, using an initial preference to determine the sensitivity, and therefore the number of clusters. This eliminates the need for large numbers of runs to determine the ideal number of clusters and any dependence on initial conditions seen in other partition clustering algorithms. Furthermore, this algorithm allows the user to set the preference for each data point. This is useful for a scenario where a partial set of representative samples are known, but there may be other exemplars along with these in a data set. The affinity propagation has been tested on geospatial, text, and gene expression data and showed improvements in both speed and accuracy over other clustering algorithms.

The main advantage of an automated computational genotyping method is that it gives the researcher the ability to combine a measure of sequence similarity (cluster membership) with the metadata. It is this metadata that yields the most information about a sample. A phylogenetic tree will tell what samples are close in sequence space, but any further inference is made using the metadata. By separating the sequences into discrete groups, the researcher is given much more flexibility to visualize the data and associated metadata.

MapReduce

MapReduce [36] is the software framework developed by Google™ to support parallel distributed execution of their data intensive applications. MapReduce is designed for fault-tolerant computations with extremely large datasets. MapReduce is divided into two major phases called map and reduce, separated by an internal shuffle phase of the intermediate results. Hadoop is an open-source version of MapReduce implemented in Java and sponsored by Amazon™, Yahoo™, and other major vendors. Recently, MapReduce has been used for sequence and phylogenetic applications. For example, CloudBurst uses Hadoop for parallel short read-mapping for use in a variety of biological analyses including SNP discovery, genotyping, and personal genomics [37]. The Genome Analysis Toolkit uses the MapReduce paradigm for shared memory platforms [38]. MrsRF (MapReduce Speeds up RF) is a multi-core, multi-machine algorithm that generates t × t Robinson-Foulds distance matrix between t trees [39] using Phoenix [40], a MapReduce implementation for shared memory multi-core platform, and OpenMPI [41]. These uses indicate that MapReduce is a promising tool to help solve the computational challenges with large datasets.

Nephele

In this paper, we describe a scalable complete genotyping system that brings together the complete composition vector and affinity propagation algorithms to produce genotypes from *Influenza* A sequences. The system has been tested on a variety of *Influenza* A and *Actinomycetes* genome data. In addition to providing discrete clusters representing genotypes, we use methods that produce trees that closely match the topologies of trees inferred using traditional phylogenetic methods, in order to provide scientists with a more familiar visualization.

Implementation

Datasets

The *Influenza* dataset used to develop our methods was that of Holmes *et al.* [42]. The clades and reassortment events found in these samples were discussed in detail, providing eight sets of sequences (one for each gene studied in the paper) for verification of our methods. This dataset consists of 155 samples, taken from New York State during the 1999-2000, 2001-2002, 2002-2003, and 2003-2004 flu seasons. The complete coding sequences are available, as well as the date and county of collection for these strains. We also used HA segments from H1N1 (1141) and H3N2 (2201) parsed from GenBank's viral division (gbvrl). For testing our implementation, we used 10,270 16S samples from GreenGenes (core_set_aligned.fasta retired on 07 February 2007; http://greengenes.lbl.gov/).

To test our methods, two additional datasets were identified. A set of 94 sequences representing WHO expert-defined genotypes (http://www.who.int/csr/disease/avian_influenza/guidelines/nomenclature/) was used to validate our methods, and another dataset representing an 2007 *Influenza* outbreak in Europe was chosen to demonstrate the utility of the computational genotyping approach for microbial forensic analysis (Additional File 1).

We also used 27 full length genomes of Actinomycetes bacteria from the Broad Institute along with their computed concatenated protein sequences, downloaded from the Tuberculosis Database (TBDB) [43].

Complete Composition Vector

The method used is based on that of Wu et al. [44]. Each sequence, S, of a given length L, can be broken into L − k + 1 overlapping substrings of length k. For each substring α, the probability of occurrence is calculated as

$$p(\alpha) = \frac{f(\alpha)}{L - k + 1},$$

where f(α) is the frequency of substring α in S. Next, the expected probability, q is calculated using a Markov model described by Brendel, Beckmann, and Trifonov, which takes into account the probabilities of length-(k-1) and length(k-2) strings [45].

$$q(\alpha) = \frac{p(\alpha_1\alpha_2 \ldots \alpha_{k-1})p(\alpha_2\alpha_3 \ldots \alpha_k)}{p(\alpha_2\alpha_3 \ldots \alpha_{k-1})}$$

This is designed to highlight the role of selective mutation, and it was found that phylogenetic trees produced without subtracting the background via the Markov model were not consistent with traditional approaches [25].

The composition value, π, for substring α is defined as:

$$\pi(\alpha) = \begin{cases} p(\alpha)/q(\alpha) - 1 & q \neq 0 \\ 0 & q = 0. \end{cases}$$

The k^{th} composition vector, $V_k(S)$, is comprised of the composition values for all possible substrings of length k. For amino acid sequences, V is of length 20 k, and for DNA/RNA, V is of length 4 k. This method has been shown to produce trees which match known taxonomies [26].

In 2004, Wu et al. extended the CV approach into the complete composition vector (CCV) [27]. This method combines the composition vector approach with the idea of the complete information set, in order to supplement any information loss from the background subtraction in the CV method [46]. The CCV is defined as the sequence of composition vectors from 3 to M, where M is a pre-determined constant.

For all experiments described in this paper, the complete composition vectors were calculated with M = 9. In addition, the revised relative entropy string selection string scoring scheme described by Wu *et al.* [44] was employed to reduce the dimensionality of the vectors. This is calculated as

$$RE(\alpha) = \sum\nolimits_i^n \left|\pi(a,i)\right| ln \left|\frac{\pi(\alpha,i)}{\Pi(\alpha)}\right|,$$

where Π represents the complete composition vector calculated from the concatenation of all n sequences in the dataset. In summary, this method evaluates the information content associated with each possible substring, and the most informative substrings are chosen for inclusion in the analysis. The number of n-mers used for distance calculations was chosen based on the dataset: if the absolute revised relative entropy was below 1.0, the substring was not used for any further calculations.

Once the final set of n-mers is chosen, the vectors are normalized by calculating the Z-score for each n-mer. From these normalized vectors, the distance matrix is then calculated. For each pair of samples, the distance between the normalized complete composition vectors Vi and Vj is calculated using cosine distance:

$$Dij = \frac{\frac{v_i v_j}{|v_i|\,|v_j|} + 1}{2},$$

We also experimented with using the Euclidian distance, calculated as

$$D_{ij} = \sqrt{\sum\nolimits_{k=1}^{n} \left(V_i(k) - V_j(k)\right)^2},$$

where n is the number of substrings kept after the substring selection (Figure 1). The CCV and distance calculation code was written in Java (1.5+), using custom classes to save space and memory. Experiments were run on Apple dual quad-core Intel Mac Pro with 8 GB running OS × 10.6. Additional testing was done under CentOS 5.5 and Ubuntu 9.4 Linux distributions.

Affinity Propagation Clustering

The input to the affinity propagation clustering algorithm is a similarity matrix. For Euclidian and Manhattan distances, the similarity is represented by the negative of the distance, while for cosine distances, the similarity is found by subtracting the distance matrix from 1. To determine the optimal preference, the mean silhouette value was used. This value is a measure of how similar a given sample is to others in the same clusters, versus samples found in other clusters. It ranges from 1 (sample is well-clustered) to -1 (the sample is found in an incorrect cluster) and is calculated as:

$$s(i) = \frac{b_i - a_i}{\max(a_i, b_i)},$$

Where a_i is the sample's average distance to the other samples in its cluster and b_i is the minimum average distance between the sample and the samples in each of the other clusters. The developers of the algorithm recommend using the minimum similarity between samples for a low number of clusters, and the median similarity for a moderate number of clusters. The preference resulting in the optimal partitioning, using the average silhouette value as a measurement was chosen from a set of four preferences spanning the minimum and median similarity. Affinity propagation was performed using the MATLAB function available at the authors' website (http://www.psi.toronto.edu/affinitypropagation/), with the default parameters or with our re-implementation written in Java as part of Nephele.

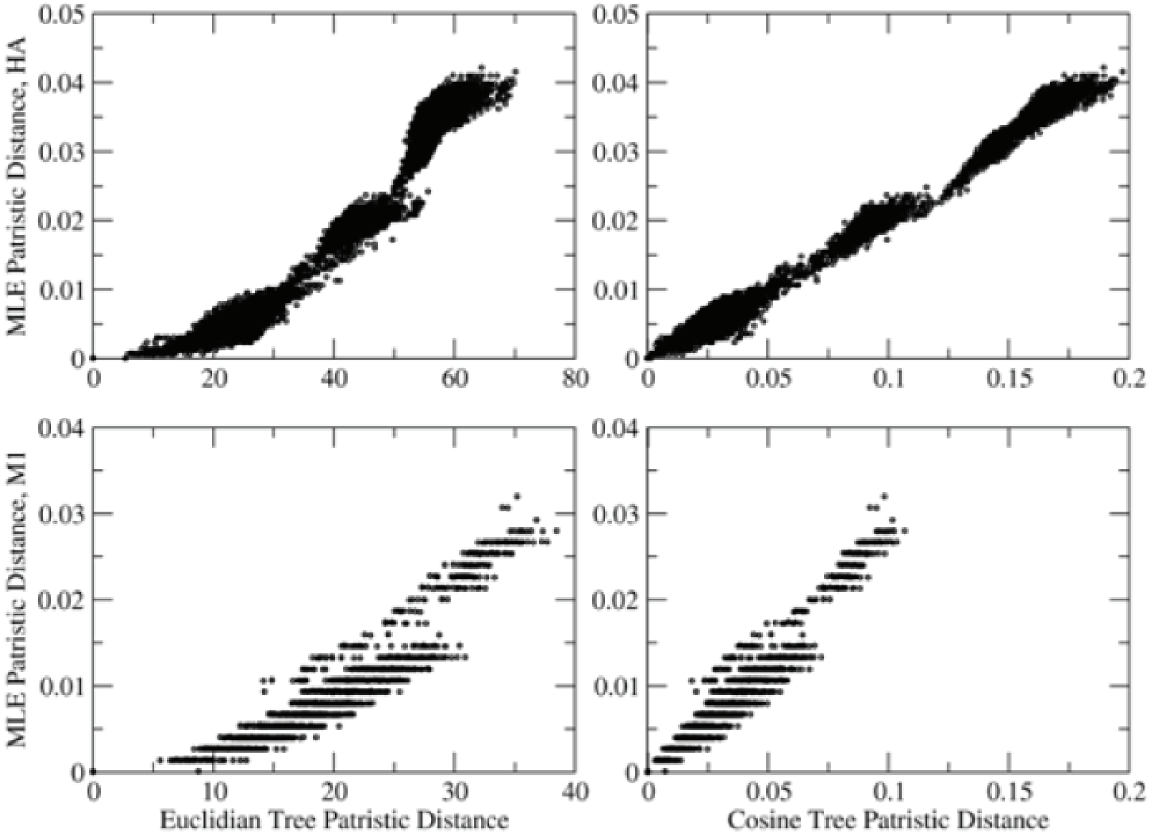

Figure 1 Comparison of CCV and Maximum Likelihood Trees. Patristic Distance plots of trees produced by Maximum Likelihood (Y axes) vs. those produced by CCV and cosine/Euclidian distance measures. (X axes).

Parallelization with Hadoop

We also implemented most of our CCV code for execution as a series of nine MapReduce jobs using the open-source MapReduce implementation Hadoop (http://hadoop.apache.org/). MapReduce is not ideal for the generation of neighbour-joined trees or affinity propagation. The MapReduce paradigm depends on the maps not dependent on any other data than what they are given. Both the neighbour-joined trees and affinity propagation algorithms depend on shared states, which breaks the MapReduce paradigm. However, Nephele provides a Message Passing Interface (MPI) version of Panjo, a neighbour-joining algorithm, that is able to handle very large trees [47]. Our version accepts row packed matrices instead of column packed, because it was easier to generate them as opposed to column packed matrices using MapReduce. All experiments were run on a Rocks (http://www.rocksclusters.org/) cluster running CentOS 5.4 on Intel Core 2 Quad and Core 2 Duo processors with 8 and 4 GB of memory, respectively, using Java 1.5 and Hadoop 0.20.1.

Results and Discussion

Genotyping the New York Dataset: Results, Computation Time and Choice of Distance Metric

The dataset used by Holmes and colleagues to study reassortment events throughout New York State provided a dataset to use to build and refine the genotyping methods. This set included 155 full genomes of H3N2 found in New York state between 1999 and 2004, collected as part of the *Influenza* Genome Sequencing Initiative [48], a worldwide sequencing initiative (this project has also sampled from the southern hemisphere, in Australia and New Zealand). Since the complete genome for each of the samples in this set was sequenced, this provided genes, with differing rates of evolution to test the pipeline.

Clustering was performed on the eight genes studied in detail in the paper (HA, M1, NA, NP, NS1, PA, PB1, PB2) as described in the Implementation section. Cluster counts ranged from 5 (NP, PB2) to 15 (PB1), with clusters ranging in size from 1 (representing an outlier in the dataset) to 36. The results from the affinity propagation clustering matched the clade structure of the trees. The trees for all eight genes, colored by cluster membership, can be seen in Additional File 2. All phylogenetic trees in this work were produced using TreeViewJ [49]. The segmented nature of the *Influenza* genome adds a level of complexity to the genotyping problem. For each sample, the set of clusters for the eight genes can be used to define a cluster "profile," which represents the genotype defined by the complete genome for that sample. This profile represents the composite genotype for that sample. For the 155 samples in the dataset, 32 genotypes were identified. From these, the groups of samples identified by Holmes and colleagues to be involved in reassortment events were found as distinct genotypes.

One of the major advantages of the complete composition vector approach over traditional phylogenetic methods is the speed of analysis. For the individual gene segments in the test dataset (155 sequences), execution times ranged from 1.50 minutes (M1) to 2.25 minutes (NA) In contrast, alignment times using MUSCLE were on the order of 5-10 minutes, and inference of maximum likelihood trees took roughly 20 minutes per gene If the genes are concatenated together to create a full genome sequence, the gains are even more impressive – trees were produced in a few minutes with the CCV-based approach, rather than hours for traditional alignment-based methods. This is consistent with initial results from composition vector based approaches, which focused on inferring trees for complete prokaryotic genomes [26].

In order to determine the ideal distance metric to use for clustering, the patristic distances (distances between leaves along the branches) of the phylogenetic trees inferred using the neighbour-joining algorithm on distances calculated using cosine and Euclidian distances were compared (see Implementation section for details of computation). Patristic distances (the distance between two leaves along the branches of a tree) were calculated using the TreeDistanceMatrix methods from the Phylogenetic Analysis Java Library [50]. Patristic distances for each gene and distance measure were plotted against each other. Figure 1 shows the patristic distance (distance between leaves along a tree) plots for HA and M1, which represent rapidly mutating and slowly mutating genes, respectively. It is clear that trees produced using the neighbour-joining algorithm on cosine distance matrices produce trees that are the most similar to the Maximum Likelihood trees, while the Euclidian distance metric produces trees which have overly large distances near the leaves of the tree, as shown in Figure 2. The cosine distance is shown to produce trees whose patristic distance has a linear relationship with that of the tree produced by maximum likelihood, while the trees produced using Euclidean distance show a higher-order relationship. These indicate that while the Euclidian distance has been used as a distance metric for the majority of previously published work involving composition/complete composition vectors [15,44], it appears that cosine distance provides a better correlation with trees produced by traditional phylogenetic methods.

Clustering on H5N1 Standard Nomenclature Dataset for Validation

In 2001, the World Health Organization (WHO), along with the World Organization for Animal Health (OIE) and Food and Agriculture Organization of the United Nations (FAO) released, in poster form, a standard nomenclature system for the various lineages of *Influenza* H5N1 found in over 50 countries throughout the world This nomenclature is intended to replace the current nomenclature used in publications, where samples are often identified by the location of the earliest sample with the closest genetic similarity (for example, "Fujian-like" or "Quinghai lineage"). Alignments of 904 HA sequences were created, and clades were chosen from the tree based on a set of rules. These clades, developed to define a new standard nomenclature, provided an opportunity to blind test set our genotyping system.

In addition to the complete 904 sample dataset, the authors provided a smaller, 109 sample representative dataset. Of these, we were able to find 94 which were in Genbank, and thus were available, with metadata, in our database. We ran our genotyping pipeline on these sequences, and found 18 clusters, as opposed to the 19 clades found in the nomenclature study. The phylogenetic tree, colored by cluster membership, is shown in Figure 3. To compare the results of our genotyping pipeline with the expert-defined genotypes, we used the Adjusted Rand Index [48], which has an expected value of zero, and a maximum value of 1. The Adjusted Rand Index for this experiment was 0.833, indicating a strong agreement between our results and the clades defined by the WHO/FAO/OIE.

We also performed a detailed examination of the trees produced by the CCV method with those from the study. We found that members of Clade 2.3.1 were found in two distinct groups on our tree, one of which was quite distant from the rest of the samples in the tree. Upon looking at the sequences, we found that these were much shorter (~1000 bp) than the rest of the sequences (~1600 bp), indicating that these sequences were most likely HA1 sequences, rather than the full HA coding sequence, even though they were labelled full HA. This highlights a problem with the quality of data that currently exists in the databases. These inconsistencies in the data can significantly distort the results of the various sequence analysis methods.

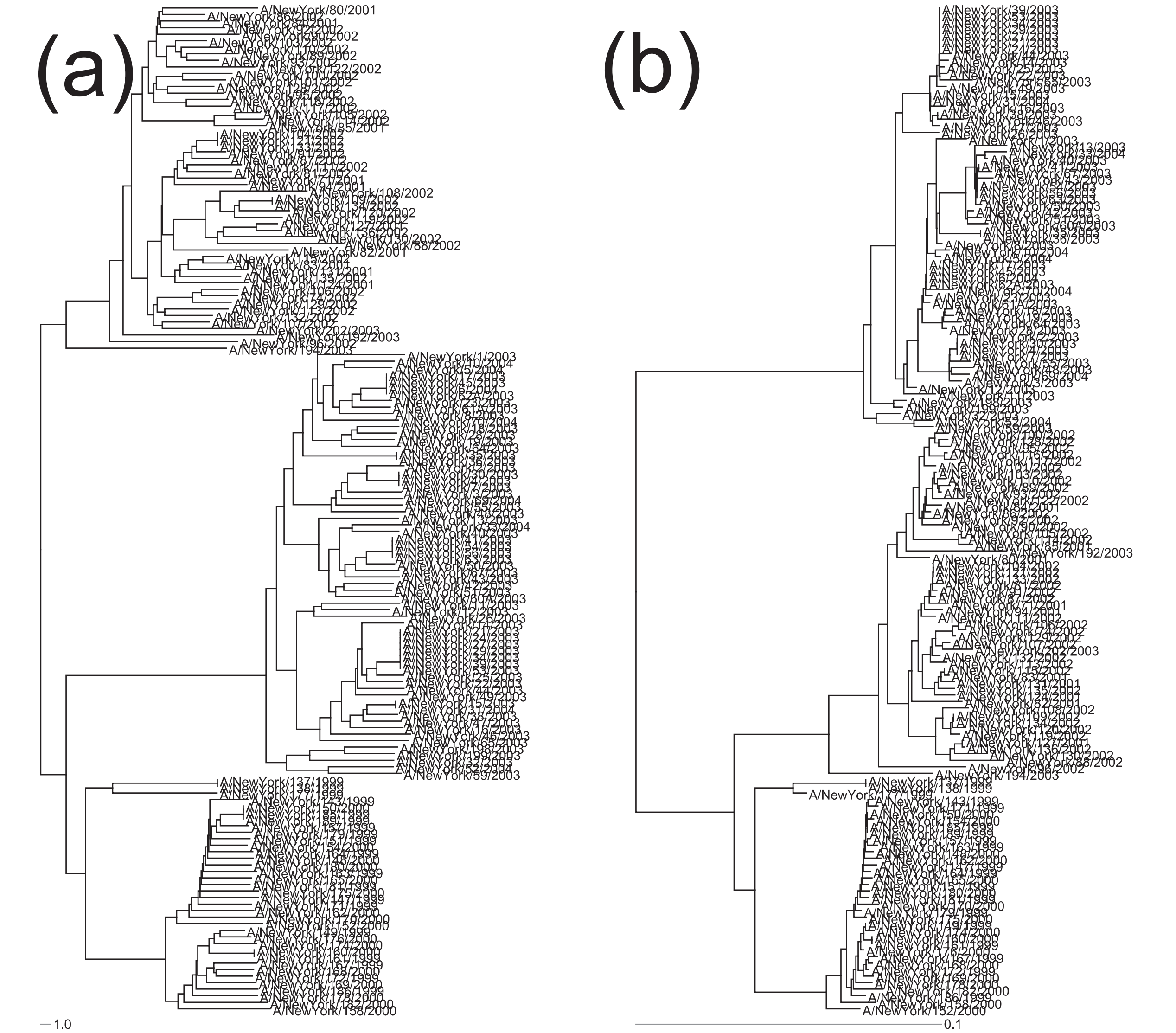

Figure 2 Comparison of Distance Metrics Used to Create Trees. Phylogenetic trees of the HA gene from the New York State dataset constructed with (a) Euclidian and (b) cosine distances. Note the long leaf-leaf distances on tree (a).

Bacterial Genomes

We investigated is our implementation can be used for larger and more complex genomes. We acquired 27 full length *Actinomycetes* genomes from the Tuberculosis Database at the Broad Institute and ran them through our pipeline. We were able to produce trees that had the same topology as those generated by the Broad in about 30 minutes compared to several hours for them using traditional tools (Brian Weiner, personal communication, from unpublished data). In addition, we compared the trees produced using the concatenation of the predicted proteins of the same set of genomes and we got similar results in both time to produce and the topology of the trees. However, it should be noted that the length of the branches are different.

MapReduce

During the development of the CCV code, we ran into memory bottle necks that required extensive coding to minimize. In addition, it was noted that several of the steps could be parallelized. We examined Hadoop to determine if we could utilize it for parallelizing our code across commodity hardware in a fault-tolerant way. We were able to code most of our algorithms using Hadoop. The few that we did not code were the neighbour-joining tree and the affinity propagation clustering algorithms. We provide a modified version of Panjo [47], a neighbour-joining algorithm, that uses the output of our Hadoop cosine distance matrix, which is in row major (packed) order. We also can output the matrix in the Phylip square format. We were able to generate a

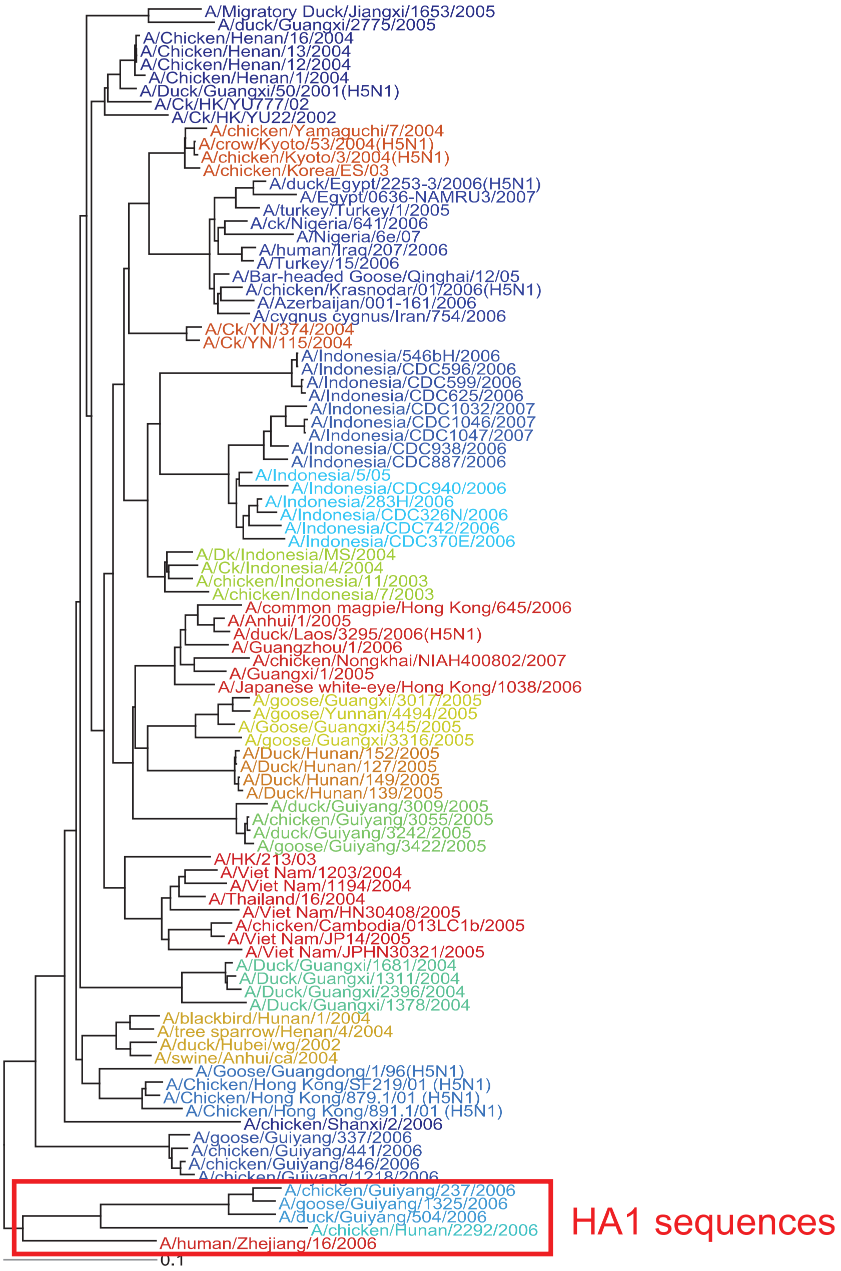

Figure 3 Clustering of Influenza Dataset. Phylogenetic tree of the WHO Dataset, colored by cluster membership. The shorter HA1 sequences are boxed in red.

neighbour-joined tree for 10,270 16S samples in 106 minutes using our Rocks Cluster of 30 machines, which we were not able to compute at all using our code on a single machine.

Conclusions

We have described a fast and accurate method for computational genotyping, using both human and avian *Influenza* as a model organism, full length *Actinomycetes* genomes, and 13 S samples. This method utilizes techniques that are faster than traditional methods for both sequence comparison and clustering. Our method produces genotypes that closely match those produced by expert analysis. In addition to providing discrete genotypes with minimal human intervention, the complete composition vector based method produces trees that correlate highly with those produced by sequence alignment and maximum likelihood methods, giving scientists a visualization of the data that they are familiar with in a fraction of the time. Possible uses of these

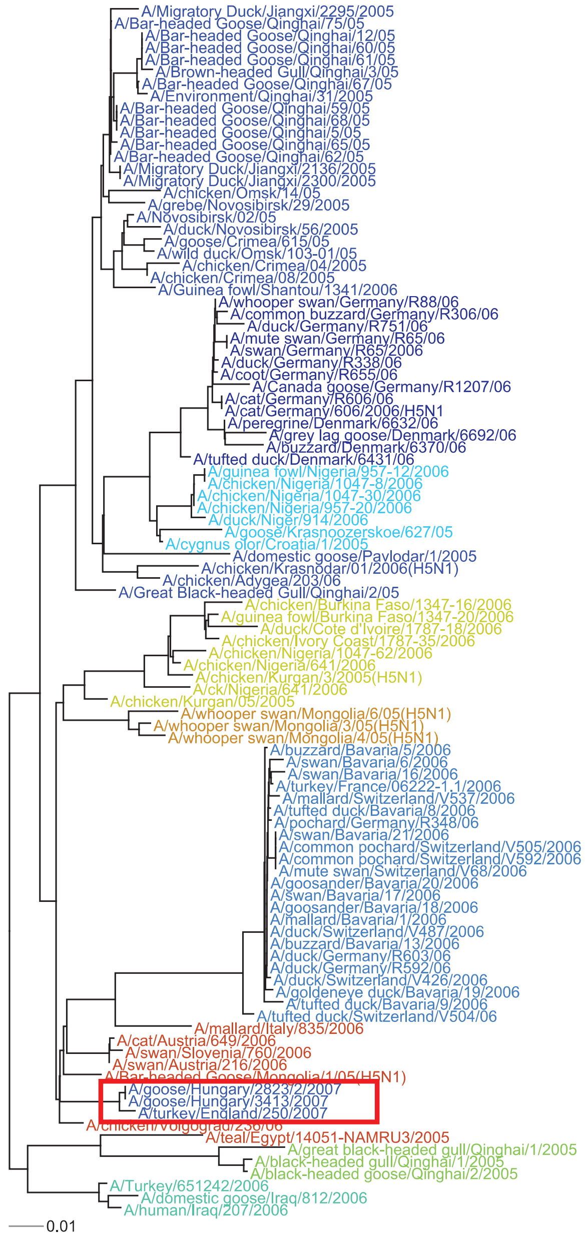
A/Migratory Duck/Jiangxi/2295/2005
A/Bar-headed Goose/Qinghai/75/05
A/Bar-headed Goose/Qinghai/12/05
A/Bar-headed Goose/Qinghai/60/05
A/Bar-headed Goose/Qinghai/61/05
A/Brown-headed Gull/Qinghai/3/05
A/Bar-headed Goose/Qinghai/67/05
A/Environment/Qinghai/31/2005
A/Bar-headed Goose/Qinghai/59/05
A/Bar-headed Goose/Qinghai/68/05
A/Bar-headed Goose/Qinghai/5/05
A/Bar-headed Goose/Qinghai/65/05
A/Bar-headed Goose/Qinghai/62/05
A/Migratory Duck/Jiangxi/2136/2005
A/Migratory Duck/Jiangxi/2300/2005
A/chicken/Omsk/14/05
A/grebe/Novosibirsk/29/2005
A/Novosibirsk/02/05
A/duck/Novosibirsk/56/2005
A/goose/Crimea/615/05
A/wild duck/Omsk/103-01/05
A/chicken/Crimea/04/2005
A/chicken/Crimea/08/2005
A/Guinea fowl/Shantou/1341/2006
A/whooper swan/Germany/R88/06
A/common buzzard/Germany/R306/06
A/duck/Germany/R751/06
A/mute swan/Germany/R65/06
A/swan/Germany/R65/2006
A/duck/Germany/R338/06
A/coot/Germany/R655/06
A/Canada goose/Germany/R1207/06
A/cat/Germany/R606/06
A/cat/Germany/606/2006/H5N1
A/peregrine/Denmark/6632/06
A/grey lag goose/Denmark/6692/06
A/buzzard/Denmark/6370/06
A/tufted duck/Denmark/6431/06
A/guinea fowl/Nigeria/957-12/2006
A/chicken/Nigeria/1047-8/2006
A/chicken/Nigeria/1047-30/2006
A/chicken/Nigeria/957-20/2006
A/duck/Niger/914/2006
A/goose/Krasnoozerskoe/627/05
A/cygnus olor/Croatia/1/2005
A/domestic goose/Pavlodar/1/2005
A/chicken/Krasnodar/01/2006(H5N1)
A/chicken/Adygea/203/06
A/Great Black-headed Gull/Qinghai/2/05
A/chicken/Burkina Faso/1347-16/2006
A/guinea fowl/Burkina Faso/1347-20/2006
A/duck/Cote d'Ivoire/1787-18/2006
A/chicken/Ivory Coast/1787-35/2006
A/chicken/Nigeria/1047-62/2006
A/chicken/Nigeria/641/2006
A/chicken/Kurgan/3/2005(H5N1)
A/ck/Nigeria/641/2006
A/chicken/Kurgan/05/2005
A/whooper swan/Mongolia/6/05(H5N1)
A/whooper swan/Mongolia/3/05(H5N1)
A/whooper swan/Mongolia/4/05(H5N1)
A/buzzard/Bavaria/5/2006
A/swan/Bavaria/6/2006
A/swan/Bavaria/16/2006
A/turkey/France/06222-1.1/2006
A/mallard/Switzerland/V537/2006
A/tufted duck/Bavaria/8/2006
A/pochard/Germany/R348/06
A/swan/Bavaria/21/2006
A/common pochard/Switzerland/V505/2006
A/common pochard/Switzerland/V592/2006
A/mute swan/Switzerland/V68/2006
A/goosander/Bavaria/20/2006
A/swan/Bavaria/17/2006
A/goosander/Bavaria/18/2006
A/mallard/Bavaria/1/2006
A/duck/Switzerland/V487/2006
A/buzzard/Bavaria/13/2006
A/duck/Germany/R603/06
A/duck/Germany/R592/06
A/duck/Switzerland/V426/2006
A/goldeneye duck/Bavaria/19/2006
A/tufted duck/Bavaria/9/2006
A/tufted duck/Switzerland/V504/06
A/mallard/Italy/835/2006
A/cat/Austria/649/2006
A/swan/Slovenia/760/2006
A/swan/Austria/216/2006
A/Bar-headed Goose/Mongolia/1/05(H5N1)
A/goose/Hungary/2823/2/2007
A/goose/Hungary/3413/2007
A/turkey/England/250/2007
A/chicken/Volgograd/236/06
A/teal/Egypt/14051-NAMRU3/2005
A/great black-headed gull/Qinghai/1/2005
A/black-headed gull/Qinghai/1/2005
A/black-headed goose/Qinghai/2/2005
A/Turkey/651242/2006
A/domestic goose/Iraq/812/2006
A/human/Iraq/207/2006
0.01

Figure 4 Integration of Metadata With Genotyping Results. Clustering of HA genes from the 2007 United Kingdom H5N1 outbreak with other samples isolated around the same time. This shows how researchers can combine genotyping with the metadata. Colors on the map represent the cluster membership. The red box indicates the cluster containing the UK Turkey Sample, along with the two Hungarian samples. Note the dates for the UK and Hungary samples (Light Blue) – these dates were only provided at the year level in Genbank, even though more accurate dates can be inferred from other sources.

tools include displaying the genotypes and associated metadata on a timeline or map, to show the geospatial and temporal distribution of the pathogen population (Figure 4). Finally, our MapReduce implementation should handle tens of thousands of bacterial size genomes and genomes of complex Eukaryote organisms (we have tested this with several *Fusarium sp.* and got similar trees, data not shown), such as those being produced from current and next generation sequencers, providing a method to analyze these large datasets.

Availability

Project name: Nephele
  Project home page: http://code.google.com/p/nephele/
  Operating system: Linux, Mac OS X, Unix
  Programming language: Java and C
  License: Apache License 2.0

Additional material

Additional file 1: Representative Standard Nomenclature Dataset of H5N1 Genotypes. A set of 94 sequences representing WHO expert-defined genotypes (http://www.who.int/csr/disease/avian_influenza/guidelines/nomenclature/).

Additional file 2: Clustering of Eight Genes from Influenza H3N2 Viruses (HA, M1, NA, NP, NS1, PA, PB1, PB2). This dataset consists of 155 samples, taken from New York State during the 1999-2000, 2001-2002, 2002-2003, and 2003-2004 flu seasons.

Acknowledgements

This work was funded through the MITRE Internal Research Program. Approved for Public Release: 10-9999. Distribution Unlimited.

Authors' contributions

MEC, MP, and SM wrote source code for CCV. MEC wrote source code for Hadoop CCV. MEC, LH, and MP conceived of the study, and participated in its design and coordination and helped to draft the manuscript. All authors have read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 5 April 2011 Accepted: 18 August 2011
Published: 18 August 2011

References

1. Li KS, Guan Y, Wang J, Smith GJ, Xu KM, Duan L, Rahardjo AP, Puthavathana P, Buranathai C, Nguyen TD, *et al*: **Genesis of a highly pathogenic and potentially pandemic H5N1 influenza virus in eastern Asia.** *Nature* 2004, **430**:209-213.
2. Campitelli L, Di Martino A, Spagnolo D, Smith GJ, Di Trani L, Facchini M, De Marco MA, Foni E, Chiapponi C, Martin AM, *et al*: **Molecular analysis of avian H7 influenza viruses circulating in Eurasia in 1999-2005: detection of multiple reassortant virus genotypes.** *J Gen Virol* 2008, **89**:48-59.
3. Rambaut A, Pybus OG, Nelson MI, Viboud C, Taubenberger JK, Holmes EC: **The genomic and epidemiological dynamics of human influenza A virus.** *Nature* 2008, **453**:615-619.
4. De Groot AS, Bosma A, Chinai N, Frost J, Jesdale BM, Gonzalez MA, Martin W, Saint-Aubin C: **From genome to vaccine: in silico predictions, ex vivo verification.** *Vaccine* 2001, **19**:4385-4395.
5. Yang HL, Zhu YZ, Qin JH, He P, Jiang XC, Zhao GP, Guo XK: **In silico and microarray-based genomic approaches to identifying potential vaccine candidates against Leptospira interrogans.** *BMC Genomics* 2006, **7**:293.
6. Macken C, Lu H, Goodman J, Boykin L: **The value of a database in surveillance and vaccine selection.** *International Congress Series* 2001, **1219**:103-106.

7. Cummings CA, Relman DA: **Genomics and microbiology. Microbial forensics–"cross-examining pathogens".** *Science* 2002, **296**:1976-1979.
8. Budowle B, Schutzer SE, Ascher MS, Atlas RM, Burans JP, Chakraborty R, Dunn JJ, Fraser CM, Franz DR, Leighton TJ, *et al*: **Toward a system of microbial forensics: from sample collection to interpretation of evidence.** *Appl Environ Microbiol* 2005, **71**:2209-2213.
9. McEwen SA, Wilson TM, Ashford DA, Heegaard ED, Kournikakis B: **Microbial forensics for natural and intentional incidents of infectious disease involving animals.** *Rev Sci Tech* 2006, **25**:329-339.
10. Wang D, Coscoy L, Zylberberg M, Avila PC, Boushey HA, Ganem D, DeRisi JL: **Microarray-based detection and genotyping of viral pathogens.** *Proc Natl Acad Sci USA* 2002, **99**:15687-15692.
11. Ghindilis AL, Smith MW, Schwarzkopf KR, Roth KM, Peyvan K, Munro SB, Lodes MJ, Stover AG, Bernards K, Dill K, McShea A: **CombiMatrix oligonucleotide arrays: genotyping and gene expression assays employing electrochemical detection.** *Biosens Bioelectron* 2007, **22**:1853-1860.
12. Lindh M, Andersson AS, Gusdal A: **Genotypes, nt 1858 variants, and geographic origin of hepatitis B virus–large-scale analysis using a new genotyping method.** *J Infect Dis* 1997, **175**:1285-1293.
13. Lin G, Cai Z, Wu J, Wan XF, Xu L, Goebel R: **Identifying a few foot-and-mouth disease virus signature nucleotide strings for computational genotyping.** *BMC Bioinformatics* 2008, **9**:279.
14. Lu G, Rowley T, Garten R, Donis RO: **FluGenome: a web tool for genotyping influenza A virus.** *Nucleic Acids Res* 2007, **35**:W275-279.
15. Wan XF, Chen G, Luo F, Emch M, Donis R: **A quantitative genotype algorithm reflecting H5N1 Avian influenza niches.** *Bioinformatics* 2007, **23**:2368-2375.
16. Stuyver L, De Gendt S, Van Geyt C, Zoulim F, Fried M, Schinazi RF, Rossau R: **A new genotype of hepatitis B virus: complete genome and phylogenetic relatedness.** *J Gen Virol* 2000, **81**:67-74.
17. Colosimo M, Hirschman L, Keybl M, Luciano J, Mardis S, Peterson M: **Genomics For Bioforensics: MITRE Sponsored Research Final Report.** Bedford, MA: The MITRE Corporation; 2008.
18. 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**:4673-4680.
19. Edgar RC: **MUSCLE: a multiple sequence alignment method with reduced time and space complexity.** *BMC Bioinformatics* 2004, **5**:113.
20. Notredame C, Higgins DG, Heringa J: **T-Coffee: A novel method for fast and accurate multiple sequence alignment.** *J Mol Biol* 2000, **302**:205-217.
21. Do CB, Mahabhashyam MS, Brudno M, Batzoglou S: **ProbCons: Probabilistic consistency-based multiple sequence alignment.** *Genome Res* 2005, **15**:330-340.
22. Edgar RC, Batzoglou S: **Multiple sequence alignment.** *Curr Opin Struct Biol* 2006, **16**:368-373.
23. DeSantis TZ Jr, Hugenholtz P, Keller K, Brodie EL, Larsen N, Piceno YM, Phan R, Andersen GL: **NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes.** *Nucleic Acids Res* 2006, **34**:W394-399.
24. Wallace IM, O'Sullivan O, Higgins DG, Notredame C: **M-Coffee: combining multiple sequence alignment methods with T-Coffee.** *Nucleic Acids Res* 2006, **34**:1692-1699.
25. Chu KH, Qi J, Yu ZG, Anh V: **Origin and phylogeny of chloroplasts revealed by a simple correlation analysis of complete genomes.** *Mol Biol Evol* 2004, **21**:200-206.
26. Gao L, Qi J, Sun J, Hao B: **Prokaryote phylogeny meets taxonomy: An exhaustive comparison of composition vector trees with systematic bacteriology.** *Sci China C Life Sci* 2007, **50**:587-599.
27. Wu X, Wan X-F, Wu G, Xu D, Lin G: **Whole Genome Phyogeny via Complete Composition Vectors.** *Technical Report TR05-06* Department of Computing Science, University of Alberta; 2005.
28. Retief JD: **Phylogenetic analysis using PHYLIP.** *Methods Mol Biol* 2000, **132**:243-258.
29. Wilgenbusch JC, Swofford D: **Inferring evolutionary trees with PAUP*.** *Curr Protoc Bioinformatics* 2003, **6, Unit 6 4.**
30. Giribet G: **Exploring the behavior of POY, a program for direct optimization of molecular data.** *Cladistics* 2001, **17**:S60-70.
31. Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Mol Biol Evol* 1987, **4**:406-425.
32. Rost U, Bornberg-Bauer E: **TreeWiz: interactive exploration of huge trees.** *Bioinformatics* 2002, **18**:109-114.
33. Hughes T, Hyun Y, Liberles DA: **Visualising very large phylogenetic trees in three dimensional hyperbolic space.** *BMC Bioinformatics* 2004, **5**:48.
34. Janies D, Hill AW, Guralnick R, Habib F, Waltari E, Wheeler WC: **Genomic analysis and geographic visualization of the spread of avian influenza (H5N1).** *Syst Biol* 2007, **56**:321-329.
35. Frey BJ, Dueck D: **Clustering by passing messages between data points.** *Science* 2007, **315**:972-976.
36. Dean J, Ghemawat S: **MapReduce: simplified data processing on large clusters.** *Proceedings of the 6th conference on Symposium on Opearting Systems Design\& Implementation - Volume 6* San Francisco, CA: USENIX Association; 2004.
37. Schatz MC: **CloudBurst: highly sensitive read mapping with MapReduce.** *Bioinformatics* 2009, **25**:1363-1369.
38. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, DePristo MA: **The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.** *Genome Res* 2010, **20**:1297-1303.
39. Matthews SJ, Williams TL: **MrsRF: an efficient MapReduce algorithm for analyzing large collections of evolutionary trees.** *BMC Bioinformatics* 2010, **11(Suppl 1)**:S15.
40. Ranger C, Raghuraman R, Penmetsa A, Bradski G, Kozyrakis C: **Evaluating MapReduce for Multi-core and Multiprocessor Systems.** *High Performance Computer Architecture, 2007 HPCA 2007 IEEE 13th International Symposium on* 2007, 13-24.
41. Gabriel E, Fagg GE, Bosilca G, Angskun T, Dongarra JJ, Squyres JM, Sahay V, Kambadur P, Barrett B, Lumsdaine A: **Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation.** *Proceedings, 11th European PVM/MPI Users' Group Meeting, Budapest, Hungary* 2004, 97-104.
42. Holmes EC, Ghedin E, Miller N, Taylor J, Bao Y, St George K, Grenfell BT, Salzberg SL, Fraser CM, Lipman DJ, Taubenberger JK: **Whole-genome analysis of human influenza A virus reveals multiple persistent lineages and reassortment among recent H3N2 viruses.** *PLoS Biol* 2005, **3**:e300.
43. Reddy TB, Riley R, Wymore F, Montgomery P, DeCaprio D, Engels R, Gellesch M, Hubble J, Jen D, Jin H, *et al*: **TB database: an integrated platform for tuberculosis research.** *Nucleic Acids Res* 2009, **37**:D499-508.
44. 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**:1744-1752.
45. Brendel V, Beckmann JS, Trifonov EN: **Linguistics of nucleotide sequences: morphology and comparison of vocabularies.** *J Biomol Struct Dyn* 1986, **4**:11-21.
46. Li M, Fang W, Ling L, Wang J, Xuan Z, Chen R: **Phylogeny based on whole genome as inferred from complete infomration set analysis.** *Journal of Biological Physics* 2002, 439-447.
47. Bullard J: **panjo: a parallel neighbor joining algorithm.** Berkeley; 2007.
48. Fauci AS: **Race against time.** *Nature* 2005, **435**:423-424.
49. Peterson MW, Colosimo ME: **TreeViewJ: an application for viewing and analyzing phylogenetic trees.** *Source Code Biol Med* 2007, **2**:7.
50. Drummond A, Strimmer K: **PAL: an object-oriented programming library for molecular evolution and phylogenetics.** *Bioinformatics* 2001, **17**:662-663.

doi:10.1186/1751-0473-6-13
Cite this article as: Colosimo *et al*.: Nephele: genotyping via complete composition vectors and MapReduce. *Source Code for Biology and Medicine* 2011 **6**:13.