Bioinformatics, 31(21), 2015, 3522–3528
doi: 10.1093/bioinformatics/btv407
Advance Access Publication Date: 7 July 2015
Original Paper

OXFORD

Genetics and population analysis

LayerCake: a tool for the visual comparison of viral deep sequencing data

Michael Correll[1,*], Adam L. Bailey[2], Alper Sarikaya[1], David H. O'Connor[2] and Michael Gleicher[1]

[1]Department of Computer Sciences and [2]Department of Pathology and Laboratory Medicine, University of Wisconsin, Madison, Madison, WI 53706, USA

*To whom correspondence should be addressed.
Associate Editor: Inanc Birol

Received on February 13, 2015; revised on May 28, 2015; accepted on July 1, 2015

Abstract

Motivation: The advent of next-generation sequencing (NGS) has created unprecedented opportunities to examine viral populations within individual hosts, among infected individuals and over time. Comparing sequence variability across viral genomes allows for the construction of complex population structures, the analysis of which can yield powerful biological insights. However, the simultaneous display of sequence variation, coverage depth and quality scores across thousands of bases presents a unique visualization challenge that has not been fully met by current NGS analysis tools.

Results: Here, we present LayerCake, a self-contained visualization tool that allows for the rapid analysis of variation in viral NGS data. LayerCake enables the user to simultaneously visualize variations in multiple viral populations across entire genomes within a highly customizable framework, drawing attention to pertinent and interesting patterns of variation. We have successfully deployed LayerCake to assist with a variety of different genomics datasets.

Availability and implementation: Program downloads and detailed instructions are available at http://graphics.cs.wisc.edu/WP/layercake under a modified MIT license. LayerCake is a cross-platform tool written in the Processing framework for Java.

Contact: mcorrell@cs.wisc.edu

1 Introduction

Comparative sequence analysis can reveal evolutionary relationships that could otherwise not be discerned. Sequence comparisons can also identify signatures of natural selection and, when analyzed in conjunction with appropriate phenotypic data, can be used to infer the 'pressures' driving selection processes.

Prior to the arrival of next-generation sequencing (NGS), comparative sequence analysis was largely restricted to the comparison of consensus sequences; i.e. sequences represented by the most abundant nucleotide at a given position in a particular sample. The limitations of consensus-level sequence analyses are particularly apparent when examining RNA viruses, as samples often contain a highly heterogeneous 'swarm of mutants'—the diversity of which cannot be represented by a consensus sequence. NGS yields thousands of short 'reads' that together represent the full diversity of virus sequences in a sample. The assembly of these sequencing reads using either a pre-determined reference or a reference assembled *de novo* from the reads themselves allows for the reconstruction of coding-complete viral genomes with the detection of nucleotide variants that exist in as little as 1% of a viral population. With this paradigm, it is now possible to overlay useful information such as nucleotide polymorphisms, polymorphism frequencies and sequencing coverage depth onto every position of a whole-genome consensus sequence. Conveying the read depth at each position in conjunction with the above information creates a large multi-dimensional matrix, which can be difficult to display visually in a manner that

facilitates the discovery of motifs by the investigator. This problem is further compounded when NGS data from multiple samples ('isolates') is compared, especially when the virus in question has a high degree of intra-sample sequence variability. However, it is in precisely these contexts that a visualization tool can be most useful for evaluating variation across genomes.

Relevant to the discussion of genomic sequence variability is the notion of a sample. Rather than a *sequence* of nucleotides, an individual sample contains the *population* of nucleotides observed at different locations along a genome, derived from NGS data. These populations can be compared to a reference sequence (or 'reference population', see Section 2.2.2) to define a certain proportion of variability at each location. By visualizing different samples simultaneously, we can observe change in variability over time (if we take multiple samples from the same infected organisms but at different time points) or observe subgroups within a particular virus (if we take samples from multiple organisms and compare them). In both cases, the analyst compares multiple samples at once.

We have therefore developed the LayerCake visualization tool to address the problem of visualizing sequence variability in viral populations. In LayerCake, samples are visualized as a colored row or layer in a single view, with variability and confidence information encoded as color. LayerCake automatically aggregates regions of the genome into discrete bins, the size of which can be controlled by the user. This design allows viewers to immediately receive an overview of the entire dataset and quickly locate regions of interest within or among samples. Zooming and side displays allow the user to retrieve detailed, nucleotide-level statistics with a single click. Interaction allows the user to adjust the aggregation, update the metrics used to define variation or update metrics related to data quality or importance. In this article, we describe the LayerCake system in detail, contextualizing its design with respect to other visual analytics tools for genomics and presenting case studies of how LayerCake has been used in multiple genomic analysis settings. We expand upon the initial LayerCake prototype detailed in Correll *et al.* (2011), supporting a more robust model of sequencing data, the capacity to deal with multiple settings of 'references' and 'pseudo-references' and adaptation to more general datasets.

1.1 Related works

There are a number of general purpose genome browsers which employ principles from visualization [see Nielsen *et al.* (2010) for a survey and discussion of the difficulties in building such systems], including some which are track based (in which different samples or data types are placed in their own distinct rows and visualized simultaneously). Many of these systems are visually similar to LayerCake in design, relying on comparison across rows or tracks and the heavy use of color to encode value (e.g. see Robinson *et al.* 2011; Zhou *et al.* 2011; Zhu *et al.* 2009). The LayerCake system differs in two key ways from these systems: first, it supports flexible aggregation and zooming, allowing the analyst to compare across an entire genome and examine small regions of interest simultaneously. Second, LayerCake is tailored for NGS data models and can adapt to the specificities of examining this sort of sequencing data (as opposed to treating each of the variables involved in NGS sequencing and alignment as orthogonal tracks).

Tools for the visualization of NGS data specifically must display the heterogeneity of reads at particular locations. Most of these NGS tools have relied on the 'scaffold view' in which sequencing reads are assembled against a reference sequence and stacked atop one another. Nucleotides that vary from the reference are highlighted within their respective read, and the frequency of these variants is represented by proportional sequence logos at the bottom of the stack (see Carver *et al.* 2012; Hou *et al.* 2010; Milne *et al.* 2010; Schatz *et al.* 2007 for a partial list of NGS visualization tools employing the scaffold view). These sequence logos are notoriously difficult to interpret (see Maguire *et al.* 2014; Ray *et al.* 2014), making it difficult for analysts to compare variation at individual locations, let alone large regions of a genome. Even if other aggregation strategies are used, the scaffold view is most useful when examining a single sequence of reads, since each scaffold is large and visually complex (requiring the display of potentially thousands of reads, hundreds of base pairs long). Even tools which do not use the scaffold metaphor are still limited to the exploration of variants within a single NGS sample (such as Ferstay *et al.* 2013). A survey of tools for NGS variant analysis (Pabinger *et al.* 2014) confirmed that most tools for this task afford the viewing of only a few separate tracks of reads at a time (one or two per window), although some tools allow the analyst to dynamically combine samples (Bigelow *et al.* 2012).

One exception to the tools which present only one (or a few) samples at a time is the Sequence Surveyor tool (Albers *et al.* 2011). Originally designed for the analysis of linkage and conservation across large numbers of genomes, Sequence Surveyor encodes each genome as a row in a large display and aggregates sections of the genome into discrete colored blocks, allowing hundreds of sequences of millions of base pairs in length to be summarized on a single screen. Swihart *et al.* (2010) recommend a similar layered design for observing trends in longitudinal data. The initial design of LayerCake adapts Sequence Surveyor techniques to the variant analysis task while maintaining a scalable design based on the arrangement of colored rows of blocks. A key difference between the two methods is that Sequence Surveyor is for viewing static sequences of genes or nucleobases. LayerCake deals with the simultaneous comparison of multiple sample *populations* of sequentially organized reads. Instead of one bit of information per location (for instance 'what is the nucleobase at this location?'), LayerCake must contend with at least four (how many of each type of nucleobase are at this location?). This problem becomes even more challenging when we compare populations to each other. Section 2.2.2 expands on this formalistic difference.

2 System and methods

LayerCake, as a tool for the quick, visual comparison of large amounts of genomic variability data, has three primary design components:

1. Techniques for visually *aggregating* large amounts of genomic variability data from multiple samples and populations.
2. Techniques for calculating and displaying various conceptions of *variation and reference*
3. Techniques for calculating and displaying various conceptions of data quality and *confidence*.

Central to LayerCake is the notion of a layer—each separate sample of viral sequence data is visually represented as a row of colored glyphs. Figure 2 shows an example of a LayerCake layer; red regions of the layer correspond to locations along the genome for which this particular population has high variance compared with the current reference. Figure 1 shows the entire LayerCake system: dozens of discrete layers organized and displayed simultaneously, with annotations and tools for viewer interaction.

2.1 Aggregation

Although viral genomes are smaller in length than mammalian genomes (tens of thousands of nucleobases rather than billions), it is still

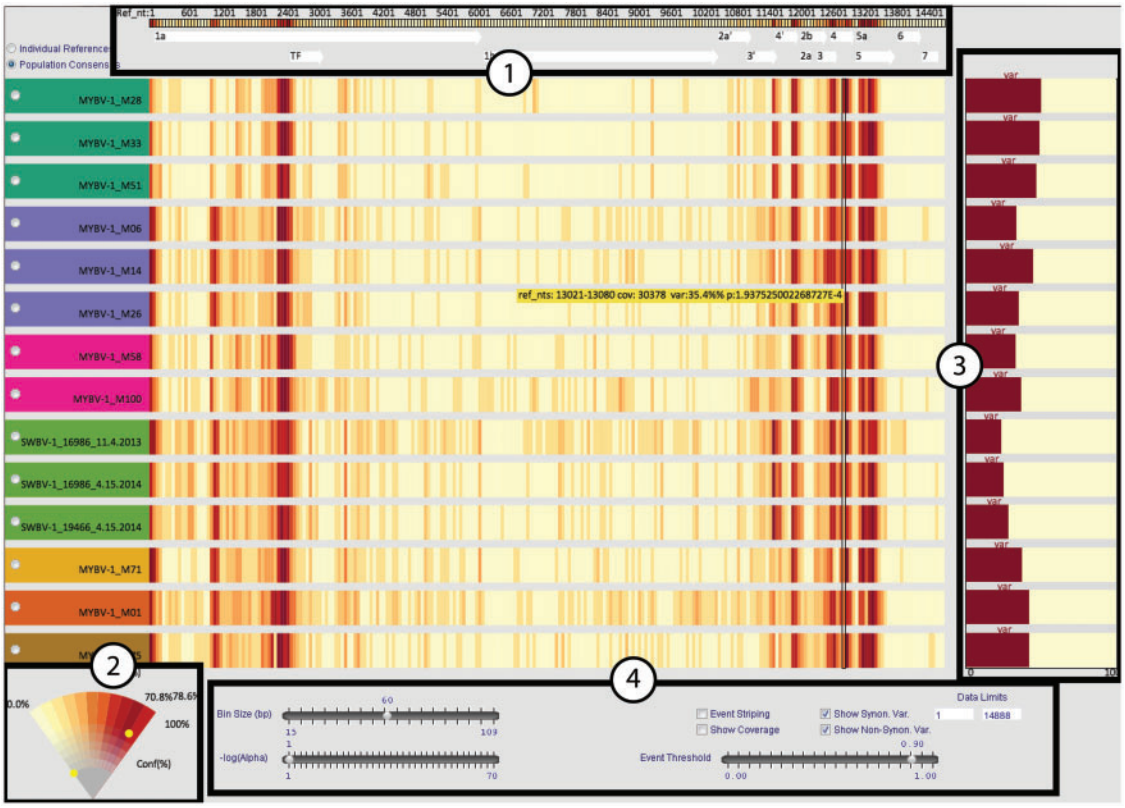

Fig. 1. An overview of LayerCake, on the simian arterivirus (SAV) dataset (see Section 3.1). Central to the display are a series of *layers*, each representing a sample of a viral population. Dark red sections of the layers correspond to areas with high deviation from a reference. The radio buttons on the left allow the viewer to choose between different conceptions of a reference (see Section 2.2.4). **(1)** An overview of variation across all samples, as a colored *histogram*. Dark red regions correspond to sections of the genome with high variation. ORFs are depicted as directional arrows. **(2)** The *color wedge*, which is both legend and interactive filtering tool. Viewers can move the yellow dots to define their own standards of important amounts of variation and acceptable levels of uncertainty (see Section 2.3). **(3)** If the viewer mouses over a particular region of interest, the *detail view* shows histograms of variations for each population. If the viewer is zoomed into a particular bin, this will show variation information at the level of individual nucleobases. **(4)** *Interaction tools* for manipulating the range of data, the size of bins and minimum standards of uncertainty, among other options

not feasible to visually present all the information from dozens of samples simultaneously. To present a meaningful overview in limited space, LayerCake compresses the sequence and chooses a visual representation of each sample that is compact enough to afford the simultaneous presentation of many samples in a single screen. Sequence compression must be considered not just in pixels, but also in visual complexity. By definition, this compression inherently aggregates some information, but LayerCake gives the viewer the ability to recover these details on demand.

The primary form of aggregation LayerCake supports is binning: contiguous locations in the genome are aggregated together into discrete blocks. The resulting color of the block represents the average variation of all sites within the block. We used color to encode data rather than, for instance, vertical position (as in a line graph or scatterplot) as prior work has shown that viewers are better at estimating and comparing average color values from sequences as opposed to average positional values (Albers *et al.* 2014; Correll *et al.* 2012). A typical viral genome consisting of tens of thousands of nucleobases can then be reduced to a few hundred blocks, which can easily fit within the dimensions of a standard computer monitor. The viewer can interactively choose how many base pairs are contained within a single bin, which alters the aspect ratio of each block as the entire layer is stretched to fit the available space. To guarantee the visibility of each block, the number of nucleobases within a block cannot be reduced to a number so low that a block would be less than a pixel wide. Conversely, the number of nucleobases within a block cannot be a number so high that the display of a bin's contents will not fit in the available space. In practical use cases, viewers tend to make bins dozens of base pairs large, to reduce the visual complexity of the display while still permitting the investigation of small-scale features in the data.

2.1.1 Recovering Detail

LayerCake averages together multiple locations into a single bin; this aggregation can create ambiguity (is this location somewhat red because many of the locations within it are somewhat variant or is it because there is one highly variant location surrounded by locations with little or no variation?) and erase details (since a region of interest in the overview could ambiguously refer to any location within a region dozens or hundreds of nucleobases long). We therefore include two techniques to recover detail: focus + context lenses and 'event striping'.

When the viewer right-clicks on a particular bin, LayerCake expands the contents of the bin to a detail view and shrinks the rest of the layer to maintain total length. Since this zooming occurs discontinuously, this is a 'table' or 'Manhattan' lens [see Carpendale and Montagnese (2001) for an overview of this and other lens types for information displays]. This detail view explicitly shows the variation at each location within a bin. Figure 2 shows an example.

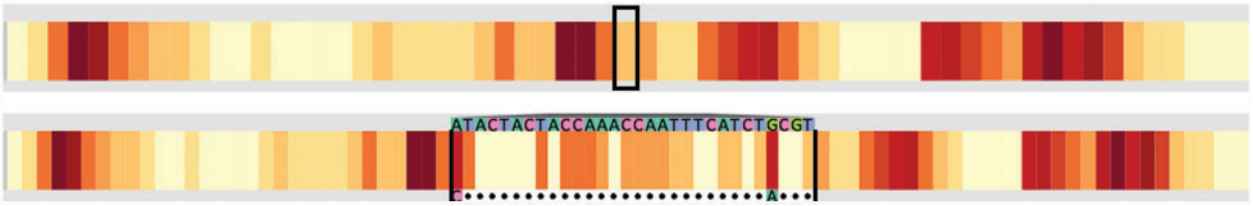

Fig. 2. A LayerCake layer. Variation at multiple sequential locations on the genome is averaged together into bins, presenting an overview of the entire genome at once (above). By right clicking on a bin (below), the viewer can recover specific information about a section of the genome while keeping the overview in context

The overview merely shows the average value of each bin. A single point of high variation can be lost in this averaging process. If the viewer wishes to see small scale (but important) features, we support a technique called 'event striping' (see Fig. 3). When enabled, the viewer selects a threshold of interest, and then LayerCake will draw thin red stripes on bins which contain locations where variation meets or exceeds this threshold. For instance, a viewer might use event striping to highlight locations on the genome where more than 50% of reads are variant. An individual bin in the main display might, on average, have significantly less than 50% variation but still have a number of visible red stripes which suggest that the viewer might wish to investigate this bin with zooming. Event striping increases the visual complexity of the display (since the number of events is only limited by the number of locations in the dataset) but allows viewers to find locations that would otherwise be lost in the averaging. Prior work has shown the utility of event striping for identifying outliers in sequence data (Albers *et al.* 2014).

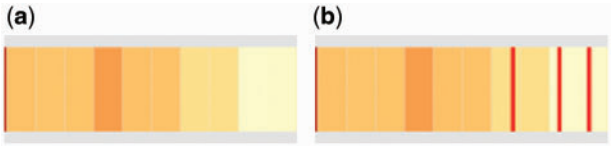

Fig. 3. An example of event striping—since each bin contains the average of information from many locations, it is possible that specific locations with high variation will be drowned out by their low variation neighbors [as in (**a**), where there appears to be very little variability in the last few bins]. Event striping draws a dark red bar on high variation outliers, adding to the visual complexity of the display but showing outliers that could be missed when data are aggregated [as in (**b**), where three specific points of high variation are now visible]

2.2 Defining Variation and Reference

Variation presupposes a non-variant sequence or population from which deviation can be measured—a reference. Typically, this is a *reference sequence*; however, in LayerCake we expand on the definition of reference to include more complex situations—for instance we may be concerned in how a viral population has changed compared to a particular time point, as opposed to some initial pre-infection reference. Different datasets will have different *references* (sequences, pseudo-sequences or populations against which we define variation), but they also might have different *definitions* of what constitutes a valid reference. These definitions might even change dynamically over the course of a session.

2.2.1 Variation from a Static Reference Sequence

Let $\overrightarrow{\text{Reads}}_n$ be a four dimensional convex vector whose components sum to 1.0, denoting the population of all reads at a location n. $\overrightarrow{\text{Reads}}_{n,A}$ would then be the proportion of reads at location n that were identified as adenine. Let Ref_n denote the reference at n. If Ref_n is a static, single base pair, then the *variation* from the reference at n is straightforward to compute. Namely, it is the percentage of reads which do not match the reference base pair:

$$1.0 - \overrightarrow{\text{Reads}}_{n,\text{Ref}_n} \tag{1}$$

2.2.2 Variation from a Reference Population

In real tasks, the assumption of a static reference is frequently violated. For instance, we might want to compare against a population at a particular timepoint, or an individual might have been infected by a diverse population of viruses rather than a single homogeneous population. In this case we would represent not just the sample but also the *reference* as another four dimensional vector $\overrightarrow{\text{Ref}}_n$. Variation should then be represented as some sort of *distance* from one vector to another. Many possible distance metrics exist; however, for this task, the distance metric ought to be easily comparable to Equation (1) above: it should preserve the semantic meaning of 'more' or 'less' variation and have a range in the interval [0,1].

We chose a distance metric based on the central metaphor of swapping. That is to make two locations identical, one would change individual reads until the distributions matched. For instance, if the population was entirely adenine at a location, but the reference was one entirely cytosine, one would 'swap' out 100% of the adenine and replace it with 100% cytosine: 100% of the reads would be swapped, so the total variation would be 100%. Likewise, if the reference was 50% A and 50% C, only half as many swaps would need to be performed, so variation would be 50%. This behavior of examining distance at each dimension (or nucleotide) individually and then summing up is captured by the ℓ^1-norm or Manhattan distance. To avoid double-counting swaps (adding more adenine by necessity means subtracting quantities for another nucleotide), we divide the ℓ^1-norm by 2.0 to derive the final metric for variation between two populations:

$$\frac{\|\overrightarrow{\text{Reads}}_n - \overrightarrow{\text{Ref}}_n\|_1}{2.0} \tag{2}$$

2.2.3 Synonymous and non-synonymous variation

The analysis of viral NGS data within a sample or from multiple samples can be used to identify signatures of natural selection—an exercise that can yield powerful biological insights, especially when supported by phenotypic data. At the core of this analysis is the identification of 'non-synonymous' mutations: those which change the amino acid sequence of the encoded protein. Since mutations are generated randomly, a high density of non-synonymous mutations in a particular region is indicative of natural selection favoring

diversification of the respective protein sequence: a phenomenon referred as 'positive selection'. The opposite is also true: a paucity of non-synonymous mutations indicates selection against protein sequence changes (i.e., 'purifying selection'). To enable the visualization of non-synonymous variation across the genome, LayerCake can display either non-synonymous mutations, synonymous mutations or both when open reading frame (ORF) annotations are included in the input reference sequence. A mutation is considered non-synonymous if it would result in a changed amino acid for even one of the relevant ORFs. The metrics presented above extend to this case by filtering out the relevant types of variation before calculating total variation.

2.2.4 Defining References in LayerCake

LayerCake considers three different reference scenarios:

1. **Individual references**: In this scenario, each discrete population considers variation *separately*—for each population, the user either provides a reference sequence (for instance from a FASTA file) or LayerCake will generate a consensus sequence for each sequence. This scenario highlights regions which have systematically high variation *within* a sample.
2. **Population consensus**: In this scenario, variation is defined with reference to a single reference sequence. This sequence is either provided from a source file (for instance a GFF file) or LayerCake will generate a single consensus sequence by voting. That is if there are 10 populations in the dataset and 6 of them have an adenine at a given position, then the population consensus will also be adenine, regardless of the read depth of any individual sequence. This scenario affords the quick apprehension of particular regions of particular samples that have high variations.
3. **Per sample comparison**: Individual samples, through the method described in Equation (2), can be used as a pseudo-reference for the rest of the dataset. This scenario readily shows variation *between* samples and also the identification of sub-groups of samples. See Figure 6 for an example.

Users may dynamically choose between different reference scenarios, even in the course of a single session. For instance, if one is interested in general regions where variation occurs, they might begin with individual references. Once those locations are identified, they might choose a particular population as a reference, to see if there are groups of populations that have different sorts of variation in these hotspots.

2.3 Confidence Visualization

Uncertainty about variation at a particular location on the genome can occur for a number of reasons. There can be error in assembling reads, aligning reads, identifying base pairs and sampling error that could arise from insufficient read coverage at a location.

Uncertainty data, no matter the source, must be visualized along with the variation information, especially for tasks where the viewer must decide which locations of the genome require more detailed analysis—highly variant but uncertain information might warrant less attention than a location with less variation but little uncertainty.

In LayerCake, color is used in each layer to display information. Color has been shown in Albers *et al.* (2014) to be a useful visual variable for helping analysts to quickly find outliers and estimate average value in regions. Since we have two types of information to display (frequency of variation and average uncertainty), this means that we must use a bivariate color map to represent the data. To avoid many theoretical obstacles to creating these color scales (see Trumbo 1981), we presume that highly uncertain values are unimportant, regardless of the variation at this location. Thus, rather than our color map resembling a square (two equal orthogonal axes), our color map resembles a wedge (with the uncertainty axis converging to a point). This makes the choice of colors significantly easier, while maintaining the desired visual behavior (important regions are highly visible, unimportant regions recede into the background). While we interpolate in multiple color attributes (both hue and saturation) to make discriminability easier, as confidence decreases it is intentionally more difficult to distinguish colors; in effect we have fewer distinct color values as we descend the wedge, replicating the intended effect of making value less important as confidence decreases. Figure 4 shows the color wedge in detail. While a bivariate encoding (such as color and size or color and orientation) would allow us to faithfully present value and confidence simultaneously, we wished to make it easier for analysts to filter out uncertain (and likely irrelevant) portions of the dataset without having to integrate multiple channels of visual information.

3 Discussion

The LayerCake system has been widely deployed across a number of viral datasets. In this section, we highlight three case studies that highlight the benefits of the LayerCake system: the presentation of a genome-scale overview of data, the ability to interactively alter notions of reference and variation and the alignment and aggregation of many samples in a single, all-encompassing display.

In addition to finding regions of high variation (as described in Correll *et al.* 2011), LayerCake affords longitudinal comparison of variation (as in Fig. 5) and allows for the identification of subgroups with similar variation signatures (as in Section 3.1).

3.1 Simian arterivirus

LayerCake also allows for the description of nucelotide variation and deep population analysis of novel viruses for which little or no prior data on sequence evolution exists. In Bailey *et al.* (2014a, b), we used LayerCake to examine nucleotide variation in novel, highly divergent simian arteriviruses that we discovered in wild red

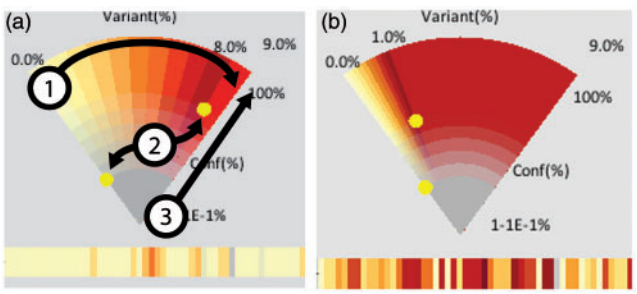

Fig. 4. The LayerCake color wedge, showing the mapping from the two axes of variation (1) and uncertainty (3) to color. Highly uncertain data are all mapped to the same grey color, giving the visual impression of data receding into a "fog" of unimportance. The two yellow dots (2) can be moved by the viewer to redefine standards of interest and importance. On the right (4b) the viewer has moved the topmost yellow dot counterclockwise, making all locations with more than 1% variation dark red, which is interactively reflected in the layers

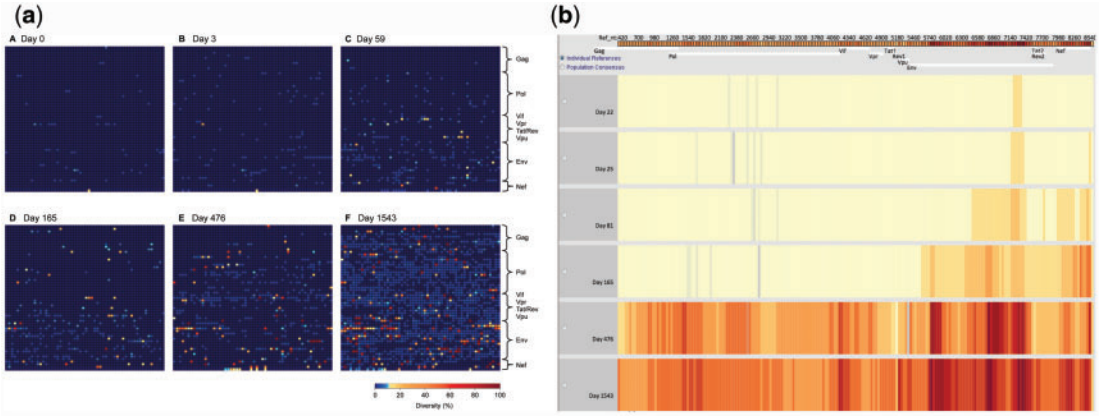

Fig. 5. An example of the utility of LayerCake for viewing systematic patterns of variation, illustrated by examining the evolution of HIV-1 in an infected individual over time. While the standard heatmap display (**a**) makes the overall trend visible (variability increases over the course of the infection), it is difficult to compare specific locations over time. In LayerCake (**b**), each row represents the viral population at a different timepoint in the infection. Change over time at a particular location can be estimated by visually scanning a particular column. Annotations (across the top of the LayerCake display) also adds context to the pattern of variation accumulated over time

Fig. 6. An example of how changing the conception of the reference in LayerCake can identify intrasequence patterns of variability, here on a dataset of simian arterivirus (SAV). Dataset from Bailey *et al.* (2014a). By defining variation from a particular population rather than a reference sequence, we can easily identify subgroups. The first row is selected as the reference population. Here, the first three rows are very similar to each other but not to the other sample, indicating a meaningful subgroup

colobus monkeys and yellow baboons living in Uganda and Tanzania, respectively. With a population-wide consensus selected (the Population Consensus option described in Section 2.2.4), LayerCake revealed several genomic regions with high levels of non-synonymous diversity. Follow-up analysis showed that the region with the most intense signal was within the ORF encoding the major envelope glycoprotein. When compared with functional data from more extensively characterized arteriviruses, this region aligned with the primary neutralizing antibody epitope of these viruses (i.e., the region of the viral protein targeted by adaptive humoral immune responses)—again providing mechanistic insight into the selective pressures driving the accumulation of non-synonymous mutations. Selecting individual references in LayerCake (the Per Sample Comparison option described in Section 2.2.4) quickly revealed varying degrees of viral sequence homology between animals, reflecting the pattern of transmission among individual monkeys (see Fig. 6). In the red colobus, this exercise identified one animal that was super-infected with two unique virus strains.

4 Conclusion

LayerCake is a full-featured visualization tool for exploring patterns of variability in viral genomes. We have deployed LayerCake to experts in the field and incorporated their feedback into further refinements. The tool, and more broadly the analytics and visual metaphor of the per-sample layer, has been applied to a large number of datasets, with positive scholastic results. The LayerCake tool is freely available and extensible to datasets beyond those we present.

Acknowledgement

We thank Todd Allen from the MGH/MIT/Harvard Ragon Institute for permission to reproduce figures from Henn *et al.* (2012).

Funding

This work was supported in part by NSF award IIS-1162037 and NIH award R01 AI077376.

Conflict of Interest: none declared.

References

Albers,D. *et al.* (2011) Sequence surveyor: leveraging overview for scalable genomic alignment visualization. *IEEE Trans. Visualization Comput. Graph.*, **17**, 2392–2401.

Albers,D. *et al.* (2014) Task-driven evaluation of aggregation in time series visualization. In: *Proceedings of the 2014 ACM Annual Conference on Human Factors in Computing Systems*. ACM, New York, NY, pp. 551–560.

Bailey,A. *et al.* (2014a) High genetic diversity and adaptive potential of two simian hemorrhagic fever viruses in a wild primate population. *PLoS One*, **9**, e90714.

Bailey,A.L. *et al.* (2014b) Two novel simian arteriviruses in captive and wild baboons (*Papio* spp.). *J. Virol.*, **88**, 13231–13239.

Bigelow,A. *et al.* (2012) CompreheNGSive: a tool for exploring next-gen sequencing variants. In: *Poster Proceedings of the 2nd IEEE Symposium on Biological Data Visualization (BioVis 2012)*.

Carpendale,M.S.T. and Montagnese,C. (2001) A framework for unifying presentation space. In: *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, pp. 61–70.

Carver,T. *et al.* (2012) Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data. *Bioinformatics*, **28**, 464–469.

Correll,M. *et al.* (2011) Visualizing virus population variability from next generation sequencing data. In: *2011 IEEE Symposium on Biological Data Visualization (BioVis)*. IEEE, New York, NY, pp. 135–142.

Correll,M. *et al.* (2012) Comparing averages in time series data. In: *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems*. ACM, New York, NY, pp. 1095–1104.

Ferstay,J.A. *et al.* (2013) Variant view: Visualizing sequence variants in their gene context. *IEEE Trans. Visualization Comput. Graph.*, **19**, 2546–2555.

Hou,H. *et al.* (2010) Magicviewer: integrated solution for next-generation sequencing data visualization and genetic variation detection and annotation. *Nucleic Acids Res.*, **38**(Suppl. 2), W732–W736.

Maguire,E. *et al.* (2014) Redesigning the sequence logo with glyph-based approaches to aid interpretation. In: *Proceedings of EuroVis 2014 Short Paper, IEEE Visualization and Graphics Technical Committee (IEEE VGTC)*.

Milne,I. *et al.* (2010) Tablet—next generation sequence assembly visualization. *Bioinformatics*, **26**, 401–402.

Nielsen,C.B. *et al.* (2010) Visualizing genomes: techniques and challenges. *Nat. Methods*, **7**, S5–S15.

O'Connor,S. *et al.* (2012) Conditional CD8 + t cell escape during acute simian immunodeficiency virus infection. *J. Virol.*, **86**, 605–609.

Pabinger,S. *et al.* (2014) A survey of tools for variant analysis of next-generation genome sequencing data. *Brief. Bioinform.*, **15**, 256–278.

Ray,W.C. *et al.* (2014) Understanding the sequence requirements of protein families: insights from the biovis 2013 contests. In: *BMC Proceedings*, Vol. 8. BioMed Central Ltd., IEEE, New York, NY, pp. S1.

Robinson,J.T. *et al.* (2011) Integrative genomics viewer. *Nat. Biotechnol.*, **29**, 24–26.

Schatz,M.C. *et al.* (2007) Hawkeye: an interactive visual analytics tool for genome assemblies. *Genome Biol.*, **8**, R34.

Swihart,B.J. *et al.* (2010) Lasagna plots: a saucy alternative to spaghetti plots. *Epidemiology*, **21**, 621–625.

Trumbo,B. (1981) A theory for coloring bivariate statistical maps. *Am. Stat.*, **35**, 220–226.

Zhou,X. *et al.* (2011) The human epigenome browser at Washington university. *Nat. Methods*, **8**, 989–990.

Zhu,J. *et al.* (2009) The UCSC cancer genomics browser. *Nat. Methods*, **6**, 239–240.