Supplementary material

kmerPyramid: an interactive visualization tool for nucleobase and $k$-mer frequencies

Jochen Kruppa$^1$, Erhard van der Vries$^2$, Wendy K. Jo$^2$, Alexander Postel$^3$, Paul Becher$^3$, Albert Osterhaus$^2$, and Klaus Jung$^1$

$^1$Institute for Animal Breeding and Genetics, University of Veterinary Medicine Hannover, Germany
$^2$Research Center for Emerging Infections and Zoonoses (RIZ), University of Veterinary Medicine, Hannover, Germany
$^3$Institute of Virology, Department of Infectious Diseases, University of Veterinary Medicine Hannover, Germany

Contents

1 Introduction 2

2 Additional Method and Package Descriptions 3
   2.1 Visualization based on Nucleobase Frequencies 3
   2.2 Visualization based on $k$-mer Frequencies 4
   2.3 Dependent R packages 4

3 Additional Examples of Use for the kmerPyramid 5
   3.1 Visualization of Clustering Results of Genomic Regions 5
   3.2 Visualization of Genomic Islands 8
   3.3 Visualization of Low Complexity Regions 8
   3.4 Comparison of Sequences 10

4 Conclusion and Further Remarks 10

References 13
1 Introduction

As mentioned in the application note many methods in bioinformatics for the analysis of DNA and RNA sequences make use of the frequency distribution of the four nucleic bases (A, C, G, and T) or higher \( k \)-mers. Examples are the binning of sequence reads (Dodsworth et al., 2013; Imelfort et al., 2014), phylogenetic analysis (Podar et al., 2013), genome assembly (Simpson et al., 2009) and the annotation of large repetitive genomes (Kurtz et al., 2008). Several tools are available for visualizing the general composition of genomic sequences. The software Circos (Krzywinski et al., 2009) for example uses a circular information aesthetic to visualize different genome information such as histograms of a nucleobase contents. Another tool, called ‘KAT: a K-mer analysis toolkit’ (Mapleson et al., 2016), also allows the visualization of \( k \)-mer frequencies in the form of histograms. The kmerPyramid package provides functions for further visual representations of \( k \)-mer frequency distributions which enable the user to compare multiple sequences in the same 3-dimensional plot. In this supplementary document, we carry out further descriptions of the methods implemented in the kmerPyramid package (chapter 2), detail further examples of use (chapter 3) and give further discussion points.

The examples of use in chapter 3 specifically focus on the visualization of clustering results, of comparisons between genomic regions, of horizontal gene transfer as well as the display of low complexity regions. The visualization of clustering results based on \( k \)-mer frequency distributions are presented using the example of coding (CDS) and non-coding (non CDS) DNA sequences. Different machine learning algorithms based on the \( k \)-mer frequency distribution of sequences are for example described by Soueidan et al. (2016). The role of \( k \)-mer frequency distributions in the analysis of horizontal gene transfer was for example described by Langille et al. (2010). Differences in the nucleic base distribution can reveal genomic islands and therefore horizontal gene transfer. Especially lateral transfer can be observed in the genomes of bacteria. Therefore, a difference in the GC content between the transferred and the host genome can be observed. However, this difference in GC content seems to be diminished by mutations as time went by (Hayek, 2013). Furthermore, \( k \)-mer frequency distributions are relevant for detecting low complexity regions. Regions with a high amount of repeats or single nucleic bases can interfere with DNA search algorithms such as BLAST (Morgulis et al., 2006). The occurrence of sequences of low complexity can be visualized by the kmerPyramid. Thus, the kmerPyramid can be used as a tool for quality control. In addition, the occurrence of a low complexity regions can be an indication for coding or non-coding regions (Orlov and Potapov, 2004).
2 Additional Method and Package Descriptions

The functions implemented in the kmerPyramid-package enable the user to visualize nucleobase and \( k \)-mer frequency distributions in two ways. First, a sequence can be assigned to a specific coordinate within the kmerPyramid based on its individual nucleobase frequency distribution. When multiple sequences are studied, they form a point cloud within the pyramid. Sequences with similar nucleobase distribution appear close to each other. Second, for a user specified \( k \), the pyramid can be subdivided by a grid, where each grid point represents all possible permutations of a specific \( k \)-mer. A bubble is plotted at each grid point, where the size of this bubble mirrors the frequency of the related \( k \)-mer permutations in the studied sequence. Both variants of visualization are detailed in the following subsections.

2.1 Visualization based on Nucleobase Frequencies

Consider the user wants to explore and compare the nucleobase frequencies of \( n \) sequences. Since DNA and RNA sequences are usually composed of four nucleic bases, frequencies are stored in an \((n \times 4)\)-matrix \( X \). Applying principal component analysis (PCA) onto this matrix, the 4-dimensional space can be represented in two or three dimensions while preserving a certain proportion of information of the raw data. When choosing the 3-dimensional representation, the point cloud is always inscribed into a pyramid built by four triangles, i.e. a tetrahedron. The four vertices represent the extreme sequences consisting of only one nucleic base: A, C, G, or T and the six edges represent extreme sequences consisting only of two bases. The pyramid shape of the projected data becomes obvious when considering the most extreme matrix \( X = \text{diag}(1,1,1,1) \), i.e. with four sequences each of them composed of only one base. The representation of \( k \)-mer genome information in the form of a tetrahedral pyramid was also used by Ichinose et al. (2014), who employ a color code to visualize \( k \)-mer frequencies on the surface of the pyramid as well as a different method for transforming frequency values into the color code. Furthermore, the authors don’t provide a software tool but only an instruction for building paper crafts of the pyramid.

This variant of the pyramid can be created using the function \texttt{pyramid_3d()}\#. This function also allows to input classification results to be visualized in the pyramid, which can be achieved by providing a vector of colors or of text labels reflecting a classifiers assignments. Furthermore, two internal clustering methods are implemented within this function, the \( k \)-means clustering and a hierarchical clustering.
2.2 Visualization based on k-mer Frequencies

If the user wants to visualize the k-mer frequency distribution within a particular sequence, the grid-based version of the kmerPyramid can be employed. In this case, our implementation assesses the k-mer frequencies based on a sliding window approach. These frequencies can be stored in a \((n_{\text{window}} \times 4)\)-matrix \(X\), where \(n_{\text{window}}\) is the number of windows of size \(k\) in the studied sequence. After applying PCA on \(X\), the pyramid is subdivided by a grid. This grid forms smaller pyramids within the basic ACGT-pyramid. Each grid point represents all permutations of one k-mer, and the frequency of these permutations in the whole sequence of this k-mer is represented by the size of the bubble at this grid point. The grid variant of the pyramid can be created by the function \texttt{pyramid\_3d\_grid()}.

ACCTGCACATTGTGCACATGTACCCCTAAAACTTAAAGTATAATAA
TAATAAAATAAAAAATGCTACAGTATGACCCCACTCCTGG

Supplementary Table 1 shows the first four windows of size \(k = 2\) or \(k = 5\) of this sequence, and thus the first four rows of \(X\). The last two columns of both tables give the original sequence of each window and the ordered permutation of this sequence, respectively. The last column is then used to determine the frequency of the permutations of each k-mer. For the example of \(k = 5\), this becomes clear in the third and fourth row, where the ordered permutation \(ACCGT\) is counted twice, also this k-mer occurs as \(CTGCA\) and \(TGCAC\) on window 3 and 4, respectively. The counts of the ordered k-mers are then presented as bubble at the grid points of the kmerPyramid. Thus, for \(k = 3\), the \(AAC\) bubble includes the permutations \{AAC, ACA, AAC\}, and for an example of \(k = 4\), the bubble \(ACGT\) includes twenty-four permutations \{ACGT, ACTG, ..., CAGT, CATG\}.

This grid-variant of the pyramid can also be used to visualize differences in the k-mer frequency distribution of different sequences, say A and B. In this case, the size of a bubble reflects the magnitude of difference in the frequency. In our implementation, blue bubbles indicate a lower frequency of a specific k-mer in sequence B compared to sequence A. In opposite, a red bubble indicates a higher frequency in sequence B.

2.3 Dependent R packages

Several R-packages are used within the package kmerPyramid. We used for the generation of the kmerPyramid different R packages. The plot is rendered in 3D using the package \texttt{rgl} (Adler et al., 2017). The sliding windows are calculated using the function \texttt{letterFrequencyInSlidingView} of the R-package \texttt{biostrings} (Pages et al., 2016). The sequence data is stored and processed using functions of the R-packages \texttt{biostrings} and \texttt{ShortRead} (Morgan et al., 2009).
### 3 Additional Examples of Use for the kmerPyramid

In this section, we list additional examples of use that can be visualized by the kmerPyramid. Specifically, we demonstrate the usability of the kmerPyramid in visualization of clustering results of genomic regions, the visualization of genomic islands introduced by lateral gene transfer, the illustration of low complexity regions in a sequence, as well as the comparison of different sequences for a descriptive classification.

#### 3.1 Visualization of Clustering Results of Genomic Regions

A possible application of the kmerPyramid is the visualization of clustering results of genomic regions such as the coding (CDS) and non-coding (non-CDS) DNA sequences. Supplementary Figure 1 shows the colored visualization of the CDS (o) and non-CDS regions (x) of the Mavirus (NC_015230.1). Coding DNA sequences tend to have a higher GC contend than the intron regions of a sequence. Therefore, the CDS sequences should be nearer to the GC edge of the kmerPyramid than the intronic sequences. Further, we implemented the $k$-means and the hierarchical clustering using complete linkage for automatically clustering (Hartigan and Wong, 1979). 

Supplementary Figure 1 (top) shows the manually colored CDS and non-CDS regions of the Mavirus. Here, we see a clear separation of the CDS and intron sequences. The CDS sequences have a higher GC content and can be found nearer to the GC edge of the pyramid. Supplementary Figure 1 (bottom left) presents the $k$-means clustering with $k = 2$. Except for three samples the clustering results meet the original group labels very well. Nearly all CDS and non-CDS sequences are clustered correctly. The hierarchical clustering with an automatically cut into 2 groups performs not as well as the $k$-means clustering. Many CDS sequences are clustered as non-CDS sequences. This example also demonstrates the assumption that a $k$-means approach works better on 'spherical' data than a hierarchical clustering approach.
Supplementary Figure 1: Visualization of k-means and hierarchical clustering results of genomic regions of the Mavirus. Separation of the CDS (o) and non-CDS (x) regions after manual coloring (top). Separation of the CDS and non-CDS regions colored by the clustering assignment of a k-means clustering (bottom left) or hierarchical clustering (bottom right).
Supplementary Figure 2: Visualization of genomic islands in bacterial genomes. Left: Visualization of the GC content shift of gene *tetA* in ten selected species from Hayek (2013). Gene *tetA* (x) is potential lateral transferred into the genome of the host (o). A clear separation between the GC content of *tetA* and the GC content of the host genome can be seen. Right: Visualization of potential horizontal transferred genes in *Escherichia coli* plasmid pC15-1a (NC_005327.1). Gene *tetA* has been identified as lateral transferred by Fondi and Fani (2010). The genes *tetR*, *ydaA* and *tetA* are in close relationship to each other reflecting the tetracycline resistance in *Escherichia coli* (Møller et al., 2016). Gene *yigA* is part of a multidrug resistance plasmid to aminoglycosides (Sun et al., 2012).
3.2 Visualization of Genomic Islands

A further application of the kmerPyramid is the visualization of potential genomic islands. A genomic island is defined as a cluster of genes for which there is evidence of lateral or horizontal gene transfer. Horizontal gene transfer occurs mainly in bacterial genomes. Very often pathogenic genes or resistance plasmids are horizontally transferred and therefore lateral gene transfer is an important mechanism to adapt to changing environments (Juhas et al., 2009). Langille et al. (2010) give a broad overview of bioinformatic approaches for detecting genomic islands. Exemplarily, we concentrate on the visualization of the single gen $tetA$ (Gene ID 2716475) referenced by Hayek (2013). The gen $tetA$ has found to be laterally transferred to the *Escherichia coli* plasmid pC15-1a and other bacterial species (Fondi and Fani, 2010). First, we visualize all ten bacterial species that were regarded by Hayek (2013), and compare the genomic GC content with the GC content of $tetA$ itself. Supplementary Figure 2 (left side) clearly shows the differences in the GC content of the genomes and $tetA$. Hayek (2013) pointed out, that the GC content of laterally transferred genes and GC content of the host will converge over the time. Moreover, there are examples for higher GC content in transferred genes as well as for transferred genes with lower GC content such as the pathogenic islands of *Salmonella typhimurium* (Marcus et al., 2000). Supplementary Figure 2 (right side) visualizes the single genes in the *Escherichia coli* plasmid pC15-1a. As expected, $tetA$ is near to the GC edge of the kmerPyramid indicating a higher GC content. Three other genes are even closer to the GC edge since they have a higher GC content as $tetA$. We identified these genes as $yigA$ (Gene ID 13906989), $tetR$ (Gene ID 17035771) and $ydaA$ (Gene ID 13906937). The genes $tetA$, $tetR$, and $ydaA$ are all tetracycline resistance genes and have a strong relationship to each other (Møller et al., 2016; Sun et al., 2012). Furthermore, $yigA$ is a multidrug resistance plasmid with resistance to aminoglycosides (Zhang et al., 2013). All genes near the GC edge seem to be plasmids which might be laterally transferred to the host. A further investigation after this visualization step is needed to obtain more evidence for these findings.

3.3 Visualization of Low Complexity Regions

The next example of the kmerPyramid shows its application to low complexity regions. DNA sequences exhibit very often intervals with a high number of repeats or biased nucleotides also called low-complexity regions. In general, DNA database search engines align many sequences to low-complexity regions and produce therefore biologically irrelevant results. Similar problems can also occur in mapping and assembly algorithms. To demonstrate the usability to visualize low-complexity regions by the kmerPyramid, we take again the example sequence of 89 nucleotides from Morgulis et al. (2006). This sequence is located from position 198630 to 198718 of the Gen-
Supplementary Figure 3: Visualization of a low complexity regions using the grid version of the kmerPyramid. Comparison of a small sequence taken from Morgulis et al. (2006) and a random, equally distributed DNA sequence. Blue labeled bubbles indicate an increase of the 2-mers in the sequence, red labeled bubbles a decrease.

bank entry AC009229.5 (Homo sapiens BAC clone RP11-314C9):

ACCTGCACATTGTGCACATGTACCCTAAAACTTAAAGTATAATAA
TAATAAAAATTTTTTTTTATTGCTACGATGACCCCATCCTGG

In addition, we produced a random DNA sequence, also of length 89, with nucleotide probabilities of 25% of each base. Supplementary Figure 3 shows the comparison of the two sequences by the kmerPyramid using the grid version with a window size of \(k=2\). Compared to the random sequence the example sequence has a higher amount of AT. Thus, the kmerPyramid can be helpful to visualize sequences that have an abnormal nucleotide frequency distribution.

Orlov and Potapov (2004) present three low complexity regions of the bacterium Borrelia burgdorferi B31 (NC_001318.1). The three low complexity regions are connected mainly to the genes BB_0801, BB_0546, and BB_0210. Therefore, we used the CDS information of the Borrelia burgdorferi B31 sequence to visualize the 5-mer frequency distribution of the three genes. Supplementary Figure 4 shows the 5-mer frequency distribution of BB_0801 (top), BB_0546 (bottom left), and BB_0210 (bottom right) using the grid-based approach of the kmerPyramid. All three genes show a lower complexity expressed by an increased amount of 5-mers involving Adenine. Orlov and Potapov (2004) point out that BB_0210 (bottom right) has the region with the lowest complex-
ity. This can be seen in the kmerPyramids of the three genes, where BB_0210 has larger bubbles at grid points including adenine compared to the other two genes. BB_0801 (top) and BB_0546 (bottom left) have nearly the same complexity.

### 3.4 Comparison of Sequences

As last example, we demonstrate the visualization of differences between k-mer frequency distributions between different sequences. As was shown in the difference plot of the low complexity example (Supplementary Figure 3), the differences are determined using the sliding window approach with k=5. Here, we draw a random set of 120 sequencing reads with a median length of 100 from the genome of the Human herpesvirus 3 (NC_001348.1). Then, we paste all reads to one single sequence and calculated the 5-mer frequency distribution. This distribution is compared to the 5-mer frequency distribution of four Human herpesvirus sequences: Human herpesvirus 3 (NC_001348.1), Human herpesvirus 2 strain HG52 (NC_001798.2), Human herpesvirus 1 strain 17 (NC_001806.2) and Human herpesvirus 4 (NC_009334.1). Supplementary Figure 5 shows the visualization of the differences in the frequency distributions between the artificial sequence and the viral sequences. As expected it can be seen clearly that the difference between the artificial sequence and the Human herpesvirus 3 sequence is the smallest one. The difference between the artificial and the Human herpesvirus 4 is the second smallest. Here, a small decrease of the GC amount can be observed in comparison of both sequences.

### 4 Conclusion and Further Remarks

In this work, we present a new approach of visualizing k-mer frequency distributions based on a 3-dimensional interactive pyramid. The approach is useful in a wide range of applications for the visualization of genomic information. Here, we show the application of the kmerPyramid to whole DNA sequences as well as to specific genomic regions like CDS and non-CDS regions. The kmerPyramid allows to check for biased sequences, i.e. sequences with a low complexity and is therefore a useful component for quality control in other bioinformatic pipelines.

Many bacteria DNA motifs consist of repeats or sequences with a low nucleic complexity (Touzain et al., 2011). Therefore, the kmerPyramid is useful as a screening tool to detect these low complexity regions. It should be remarked that the pyramid does not picture the original k-mers of a sequence but the ordered ones. Hence, the grid based kmerPyramid can give hints for sequence motifs, which should however be further analyzed by the standard motif search tools (D’haeseleer, 2006).
Supplementary Figure 4: Visualization of low complexity regions of Borrelia burgdorferi B31. Visualization of the 5-mer frequency distribution of gene BB_0801 (top), gene BB_0546 (bottom left), and gene BB_0210 (bottom right). All three genes show a lower complexity and thus an increased amount of 5-mers including Adenine (Orlov and Potapov, 2004).
Supplementary Figure 5: Visualization of 5-mer frequency differences between sequences. Visualization of the frequency difference of 5-mers of four Human herpesviruses in comparison to randomly generated reads of length 100 from the Human herpesvirus 3: Difference to Human herpesvirus 1 (top left), difference to Human herpesvirus 2 (top right), difference to Human herpesvirus 3 (bottom left), and difference to Human herpesvirus 4 (bottom right).
References


