INTRODUCTION

RAMBO-K is a reference-based tool for rapid and sensitive extraction of one organisms reads from a mixed dataset. It is based on a Markov chain implementation, which uses genomic characteristics of each reference to assign reads to the associated set.

SYSTEM REQUIREMENTS

RAMBO-K is implemented in python and Java. Thus, it requires Java SE 7 as well as python 2.7 or higher (including modules numpy and matplotlib).

LICENSE

RAMBO-K - Read Assignment Based On K-mers

RUNNING RAMBO-K

References and names

To use RAMBO-K you need to specify two reference sets (using parameters $-R$ and $-r$). References must be provided as fasta-files and may contain one or more sequences corresponding to the desired species. If you lack an exact reference, a multiple FASTA file containing sequences from a set of species related as closely as possible will also work. You can enter two names for the reference sets (using $-N$ corresponding to $-R$ and $-n$ corresponding to $-r$), which will be used for naming of the result files and labelling the plots.

Unassigned reads

The unassigned reads from the original dataset can be provided as fastq or fasta files (if fasta, filetype option must be set to $-t$ fasta) either as single- or paired end sets using parameter *-1* (and *-2* for paired reads). If paired end information is available, using it is strongly recommended.

File structure

RAMBO-K will create a folder for every new set of species (named $name1_$name2). In that folder, the assigned reads will be saved as $name*.fastq (or .fasta if input format is specified as fasta). All files are named according to the entered options.
There will also be a folder $\$name1\_\$name2/temp/$ containing temporary files. This can be used to rerun RAMBO-K without repeating all calculations. The temp folder can be deleted after a run using the $-d$

parameter, but as temporary data are rather small and save a lot of time when re-running RAMBO-K, this is not recommended.
In the $\$name1\_\$name2/graphics/$ folder you will find some plots which are helpful for determining your final parameters (see Cutoffs). If you want to write your results to a folder other than your working directory, use the parameter $-o$.

K-mer sizes

The size of k has a crucial influence on assignment quality. It is recommended to run RAMBO-K over a range of k values first to find a suitable value. Generally, a low k (<5) works best if the reference is rather distant, while higher values (>8) can be used if a close reference is known. Best results are usually achieved with k between 4 and 12. You can enter k as a range (e.g. $-k$ 4:8), a list (e.g. $-k$ 4,8,12) or a single integer (e.g. $-k$ 8). For a detailed discussion on the choice of k in different cases see figures 1 and 2 in the appendix.

Cutoffs

After the first run of RAMBO-K, you will find a series of plots provided in the graphics folder. All plots are named according to the entered options. The score_histogram_*.png plot will show the theoretical distribution of scores of each species. Ideally, the curves for the two species, which are to be separated, should overlap as little as possible. Theoretical specificity and sensitivity is shown in ROCplot_*.png. To check the correlation of the theoretical distributions to your real dataset, fitted_histograms_*.png will show the distribution of scores in your original data. Choose k such that both peaks are as far apart as possible, while the correlation of your data with the theoretical distributions is as high as possible. Your cutoff should be set such that most desired reads are assigned while assigning as few background reads as possible. In most cases, this should be a value close to 0.
Cutoffs are provided either using parameter $-c$ to assign all reads with scores lower than the given number, or parameter $-C$ to assign all reads scoring higher than the given number.
For negative cutoffs, use m instead of - (e.g. m1 = -1).

Number of reads used for simulation

The $-a$ parameter allows you to vary the number of reads used to simulate data and draw the plots. Especially for very large reference genomes, a higher number is recommended to yield more precise results. Also, the plots will look smoother using more data. However, using higher numbers of reads for this step may slow down the precalculation phase.

Threading

RAMBO-K is implemented to use multiple threads with near-linear speedup until hard drive access speed becomes the limiting factor. The number of threads is specified by parameter $-t$. Per default, multithreading is disabled.

Hints

In case of low quality data, quality trimming will significantly improve the results of RAMBO-K.
For a short help and explanation of all parameters type ./RAMBOK -h .

EXAMPLE

Divide a set of viral and human reads from a paired end data set in fastq format:
1) Run RAMBO-K to evaluate optimal parameters (optional):

```
./RAMBOK -r human_reference_sequences.fasta -n H.sapiens -R viral_reference_sequences.fasta
-N virus -1 unassigned_reads.1.fastq -2 unassigned_reads.2.fastq -t 4 -k 4:8
```

2) Now take a look at the plots to choose ideal values of *k* and *c*. Run RAMBO-K again specifying *k* and *c*. Do not change any of the other options except for the number of threads. In this example, the best discrimination would be obtained using $k = 6$ and setting the cutoff to 0. To assign viral reads while dismissing the human background, run:

```
./RAMBOK -r human_reference_sequences.fasta -n H.sapiens -R viral_reference_sequences.fasta
-N virus -1 unassigned_reads.1.fastq -2 unassigned_reads.2.fastq -t 4 -k 6 -c 0
```

Results will be saved in virus_cutoff_0_k_6_1.fastq and virus_cutoff_0_k_6_2.fastq in your working directory or, if specified, the output folder.
Or, to assign the human reads while dismissing viral reads, run:

```
./RAMBOK -r human_reference_sequences.fasta -n H.sapiens -R viral_reference_sequences.fasta
-N virus -1 unassigned_reads.1.fastq -2 unassigned_reads.2.fastq -t 4 -k 6 -C 0
```

Results will be saved in H.sapiens_cutoff_0_k_6_1.fastq and H.sapiens_cutoff_0_k_6_2.fastq in your working directory or, if specified, the output folder.

Or, to divide a set of viral and human reads from a single end data set in fasta format:
1) Run RAMBO-K to evaluate optimal parameters (optional):

```
./RAMBOK -r human_reference_sequences.fasta -n H.sapiens -R viral_reference_sequences.fasta
-N virus -1 unassigned_reads.1.fasta -f fasta -t 4 -k 4:8
```

2) Now take a look at the plots to choose ideal values of *k* and c. Run RAMBO-K again specifying *k* and *c*. Do not change any of the other options except for the number of threads. In this example, the best discrimination would be obtained using $k = 10$ and setting the cutoff to -10. To assign viral reads while dismissing the human background, run:

```
./RAMBOK -r human_reference_sequences.fasta -n H.sapiens -R viral_reference_sequences.fasta
-N virus -1 unassigned_reads.1.fasta -f fasta -t 4 -k 10 -c m10
```

Results will be saved in virus_cutoff_m10_k_10.fasta in your working directory or, if specified, the output folder. Or, to assign the human reads while dismissing viral reads, run:

```
./RAMBOK -r human_reference_sequences.fasta -n H.sapiens -R viral_reference_sequences.fasta
-N virus -1 unassigned_reads.1.fasta -f fasta -t 4 -k 10 -C m10
```

Results will be saved in H.sapiens_cutoff_m10_k_10.fasta in your working directory or, if specified, the output folder.

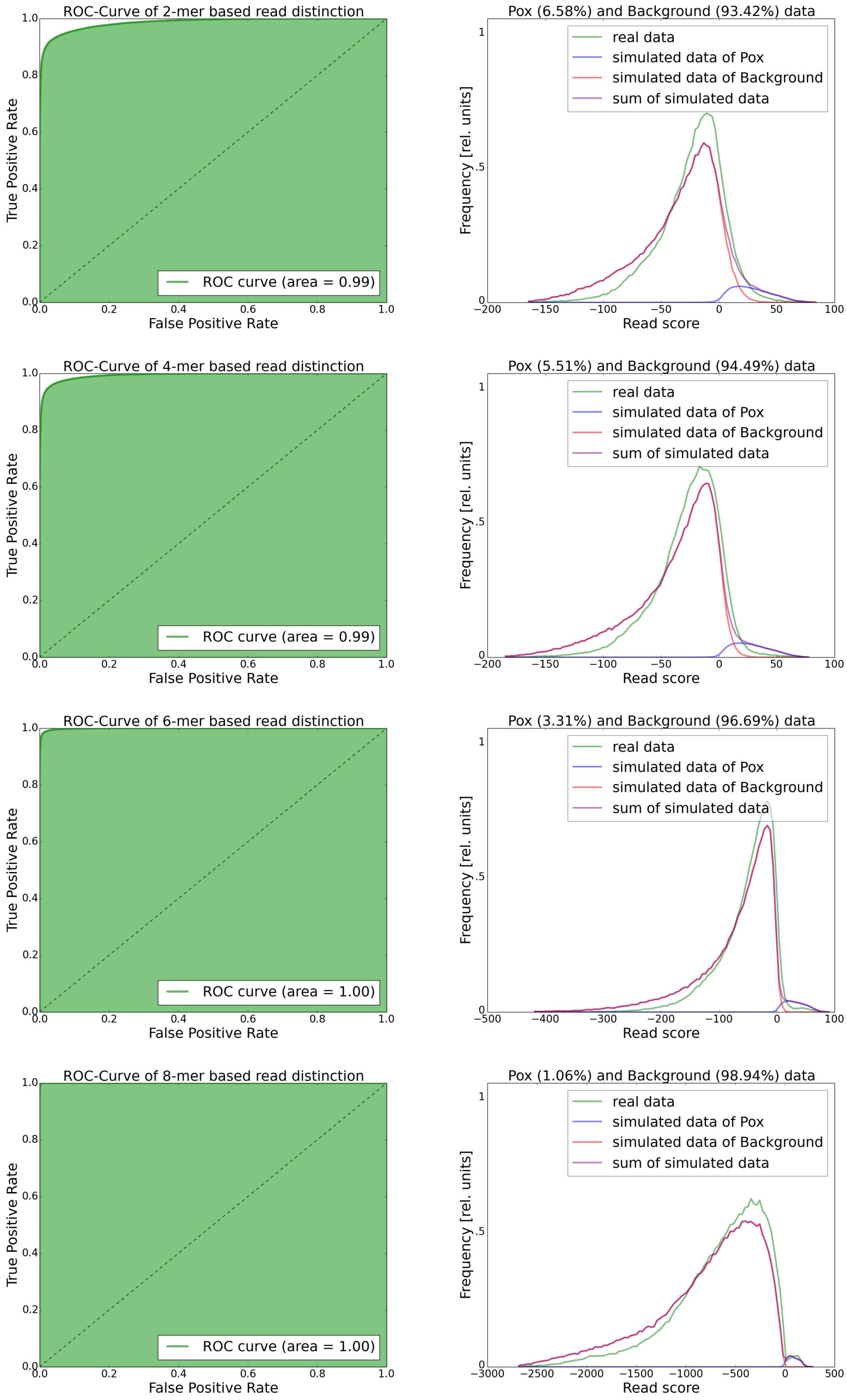

Figure 1: ROC-plots (right) and fitted histograms (left) of mixed reads of pox and human background. Theoretical read separation (as represented by the ROC-plots) and fitting of the score distributions (represented by the fitted histograms) improve with increasing k. An optimal choice of values would here be $k = 8$ and $c = 0$.

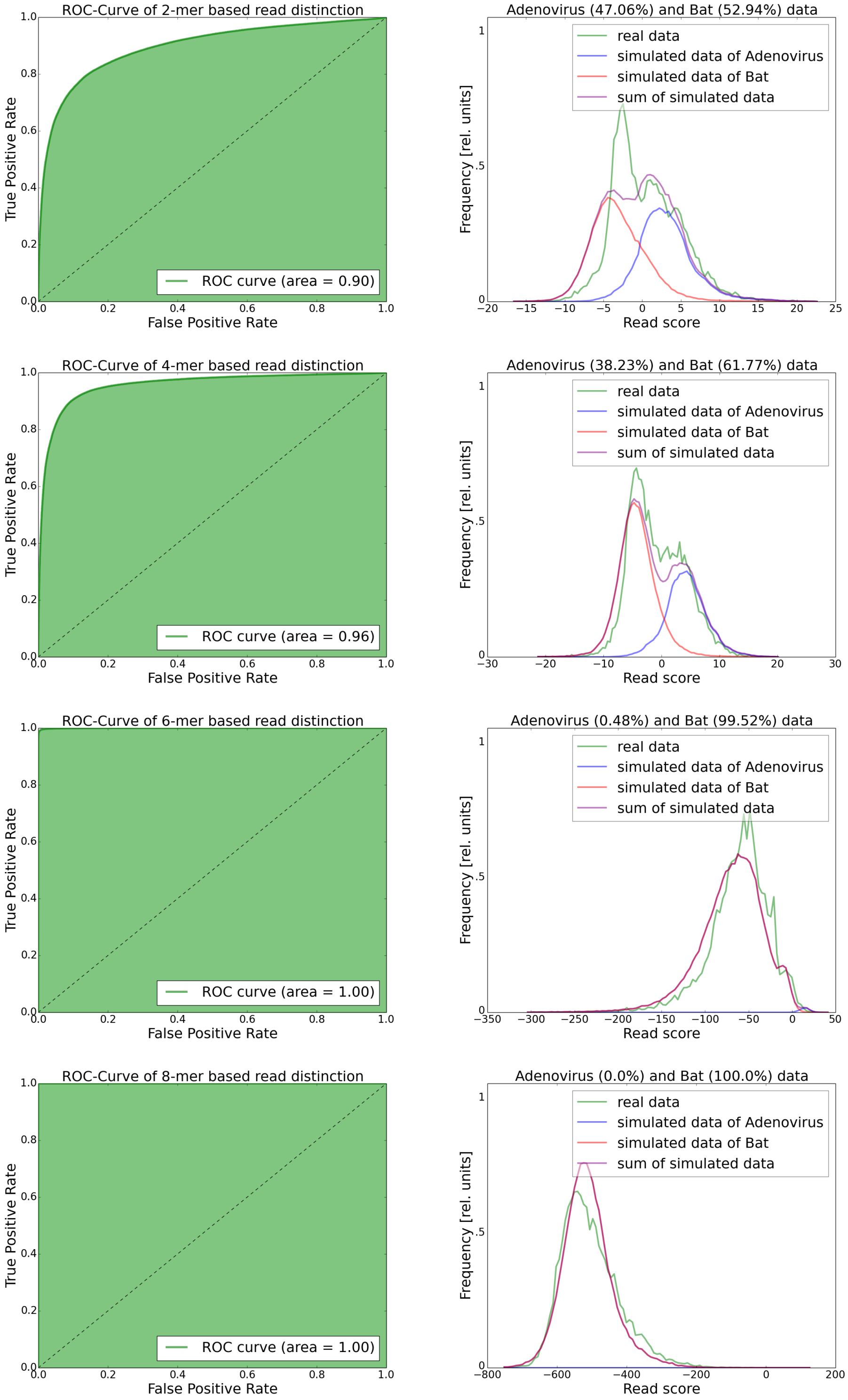

Figure 2: ROC-plots (right) and fitted histograms (left) of mixed reads of bat adenovirus and bat background. Training was executed using bat sequences and more distant canine adenovirus sequences. Theoretical read separation (as represented by the ROC-plots) improves with increasing k, but fitting of the score distributions (represented by the fitted histograms) diminishes. This is due to the fact that no k-mers of high length are identical in the canine adenovirus reference and the real dataset. In this case, a compromise must be found. Here, best results are achieved using $k = 4$ and $c = 0$.