jpHMM - Documentation

http://jphmm.gobics.de

Anne-Kathrin Schultz
Department of Bioinformatics
Institute of Microbiology and Genetics
Georg-August-Universität Göttingen, Germany

October 26, 2011

Contents

1 Short Introduction **3**

2 jpHMM **3**
2.1 Method 3
2.2 Reliability of the recombination prediction 4
2.2.1 Uncertainty regions 4
2.2.2 Breakpoint intervals 4
2.3 Reduction of the runtime (optional) 4
2.3.1 Pre-alignment of a query sequence to the multiple sequence alignment with BLAT 5
2.3.2 Accuracy 7
2.3.3 Usage and download of BLAT 8
2.4 Applications 8
2.4.1 Viruses with linear genomes such as HIV-1 9
2.4.2 Viruses with circular genomes such as HBV 9

3 Installation **10**

4 Copyright **13**

5 Running jpHMM **13**
5.1 Main jpHMM program 13
5.1.1 Parameters 14
5.2 Definition of uncertainty regions and breakpoint intervals 19
5.2.1 Graphical output of the posterior probabilities (for HIV) 21

5.3 Graphical visualization of predicted recombination for circular genomes . 22
5.4 Example 23

6 Web server **24**

7 Contact **26**

1 Short Introduction

jpHMM (jumping profile Hidden Markov Model) is a probabilistic approach to compare a query sequence to a multiple alignment of a sequence family [8, 11, 7]. It was applied to detect recombinations in genomic sequences of HIV-1, the hepatitis C virus (HCV) and the hepatitis B virus (HBV). It is available online at `http://jphmm.gobics.de/` as a webserver for HIV-1 and HBV and for download. The download version can also be used for detecting recombinations in other viruses.
The application of jpHMM to detect recombinations in HIV-1 and HCV genomes was developed in close collaboration with the HIV Sequence Database Group of the Los Alamos National Laboratory, USA. The application of jpHMM to HBV was developed in collaboration with scientists from the Laboratoire INSERM U871 in Lyon, France, and the Laboratoire associé au Centre National de Référence des hépatites B, C et delta, UFR Santé Médecine Biologie Humaine, at the Université Paris 13, France.

jpHMM aligns a query sequence to a pre-calculated multiple sequence alignment of pure-subtype sequences, predicts recombination breakpoints and assigns to each segment of the sequence one of the given subtypes/genotypes. Based on the posterior probabilities of the subtypes calculated by jpHMM, information about uncertainty regions in the recombination prediction and interval estimates of breakpoints, as opposed to point estimates, are determined in a post-processing step [7].

2 jpHMM

2.1 Method

jpHMM is a probabilistic model that predicts phylogenetic recombination breakpoints in a viral genomic sequence and assigns to each segment of the sequence one subtype/genotype. The idea for the model is based on the 'jumping alignment' algorithm (JALI) proposed by Spang *et al.* as a strategy to database searching [10]. In contrast to standard methods, JALI does not align and compare a database sequence to a multiple sequence alignment as a whole, but aligns local segments of the sequence to those segments of individual sequences from the alignment that are most similar to them. Within this alignment, the sequence can jump between the different sequences of the alignment at arbitrary positions.
The recombination prediction of jpHMM is based on a pre-calculated multiple sequence alignment subdivided into different subclasses, so-called *subtypes*. Each subtype in the alignment is modeled as a profile HMM. In addition to the usual state transitions within these profile HMMs, transitions, called jumps, between the different profile HMMs are allowed. Thus the model can jump between states corresponding to the different subtypes, depending on which subtype is locally most similar to the database sequence. The recombination prediction for a query sequence is then defined by the most probable path through the jpHMM, the so-called Viterbi path, that generates the query sequence. Since

each state of the jpHMM only belongs to one profile HMM and each sequence position is generated by one state of the model, each position of the query sequence is assigned to exactly one parental subtype. Positions of jumps between different subtypes define recombination breakpoints.

2.2 Reliability of the recombination prediction

To get a hint about the reliability of the jpHMM prediction, i.e. the accuracy of the predicted breakpoint positions and parental subtypes, we extended the output of the jpHMM to include the information on regions where the model is 'uncertain' about the parental subtype and provide an interval estimate of the breakpoint, as opposed to a point estimate [7]. For each sequence position, the so-called posterior probability for each subtype is calculated. This is the probability that the respective sequence position belongs to the considered subtype under the assumption that the whole sequence is generated by the model. The posterior probabilities are used to define *uncertainty regions* in the recombination prediction and interval estimates of breakpoints, called *breakpoint intervals* here.

2.2.1 Uncertainty regions

If at a certain position of the sequence the posterior probability of the predicted subtype is lower than a certain threshold t_{UR}, this position is marked as uncertain. By examining the graph of the posterior probabilities the user can also see which parental subtypes are most closely related in these regions.

2.2.2 Breakpoint intervals

A breakpoint interval is defined by an interval around a predicted breakpoint position where the posterior probabilities of the two successive predicted subtypes are lower than a certain threshold t_{BPI} but higher than the posterior probabilities of all other subtypes.

The length of a breakpoint interval depends on how precisely the breakpoint can be located. A large interval is the consequence of the uncertainty of the model to locate the exact breakpoint position between two subtypes. Thus, the user can see, which breakpoints can be located relative precisely or which breakpoints are approximative. For uncertainty regions, no parental strain can confidently be determined. However, by examining the graph of the posterior probabilities the user can see which subtypes are closest related in these regions. At positions outside uncertainty regions and breakpoint intervals, the user can now be more confident in the predicted parental subtype, as our results show [7].

2.3 Reduction of the runtime (optional)

In general, each query sequence position can be aligned to each column in the given multiple sequence alignment, i.e. it can be emitted by each (non-mute) state in the model.

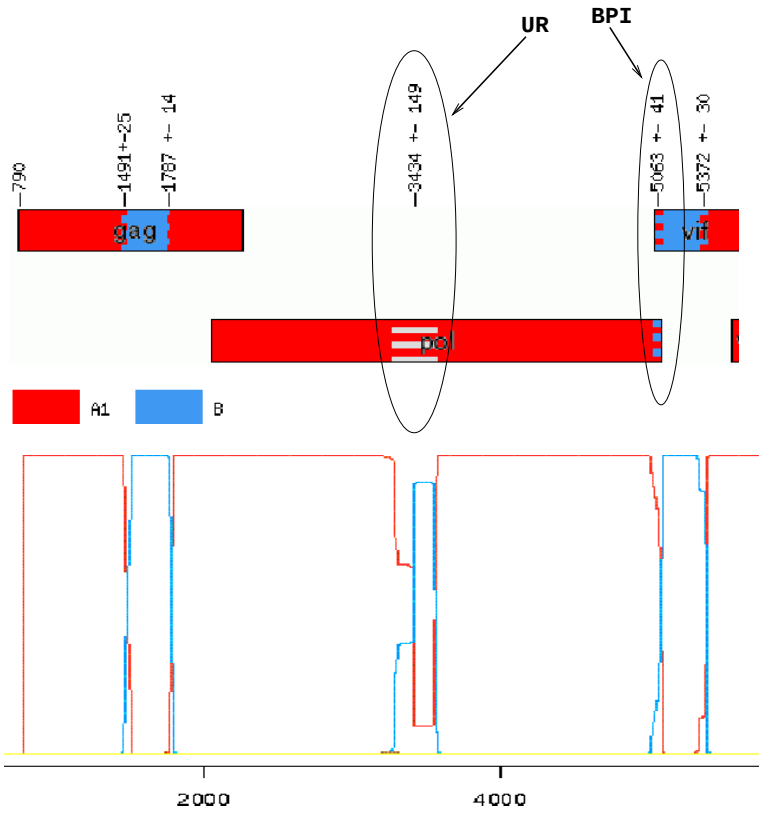

Figure 1: Part of the jpHMM web server output for a full-length semi-artificial HIV-1 recombinant of subtype A1 containing segments of length 300nt from subtype B. Above the genome map of the predicted recombination is shown (drawn by HIV Sequence Locator Tool, http://hiv.lanl.gov), below the posterior probabilities of the subtypes. Breakpoint intervals (BPI) are shown by an interfingering of the colors of the two predicted subtypes, uncertainty regions (UR) by an interfingering of grey and the color of the predicted subtype. For the UR at position 3434 ± 149, the posterior probabilities give a hint to the correct subtype B.

This leads to a very high complexity of the Viterbi algorithm and thus to a large runtime of the program. For example, for full-length HIV-1 sequences, the average runtime of the program is about 7 to 8 minutes.

2.3.1 Pre-alignment of a query sequence to the multiple sequence alignment with BLAT

For sequences that share a certain degree of similarity with the sequences included in the multiple sequence alignment, the complexity of the Viterbi algorithm can be reduced considerably. By a pre-alignment of the query sequence to the multiple alignment, it is possible to map each query sequence position to a certain column (or a certain region) in the alignment. This column (or region) defines the part of the alignment to which the

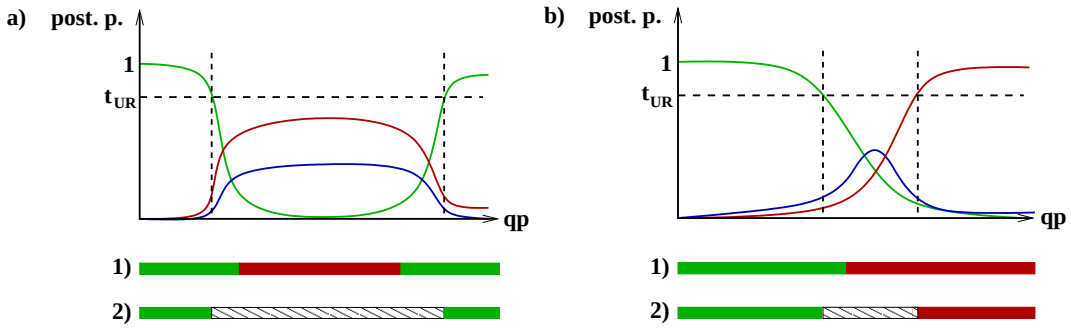

Figure 2: Two examples for uncertainty regions in predicted recombinations. In both figures, for each query sequence position (qp), the posterior probabilities (post. p.) of three subtypes are plotted. t_{UR} marks the posterior probability threshold for the definition of uncertainty regions. Vertical dashed lines define the extent of uncertainty regions. The first bar (1) below the plot of the posterior probabilities shows the original recombination prediction with precise breakpoint positions. The second bar (2) shows the predicted recombination including uncertainty regions (hatched regions). In a) an uncertainty region is defined because the posterior probability of the predicted subtype (red) is below t_{UR}. In b) the region around the predicted breakpoint is not defined as a breakpoint interval since the posterior probability of a third subtype (blue) is higher than the posterior probabilities of the subtypes predicted to the left (green) and to the right (red) of the breakpoint.

respective query sequence position is allowed to be aligned with jpHMM. Thus, for each query sequence position, the search space of the Viterbi algorithm is restricted to states corresponding to the respective assigned column (or region).

For the pre-alignment of the query sequence to the given alignment, a set of sequences ($n \approx 100$), equally distributed on all subtypes, is selected from the multiple alignment. The query sequence is aligned to each of these sequences pairwisely so that each query sequence position is mapped to a certain set of alignment columns. These columns define the region in the alignment to which the respective sequence position is allowed to be aligned with jpHMM. Since most of the columns in the alignments we study are reasonably conserved, the size of these regions is very short, usually.

As pairwise alignment tool, the BLAST-like alignment tool (BLAT) [2] is chosen. It is a very fast and accurate tool for mRNA/DNA and cross-species protein alignments with an easy-to-use output. At DNA level, BLAT works well for sequences with a similarity greater than or equal to $90\,\%$. More divergent alignments might be missed but BLAT is able to align sequences that include large inserts. We chose BLAT since most parts of the HIV-1 as well as the HBV genome are very conserved and show a genetic divergence lower than $10\,\%$. Only few regions with a higher genetic variability are included. For example, the highest genetic divergence between different HIV-1 subtypes can be observed in the env and the gag region with a maximum divergence of $\sim 35\,\%$ and $14\,\%$ respectively. Genotypes of HBV are distinguished on the basis of a genetic divergence of at

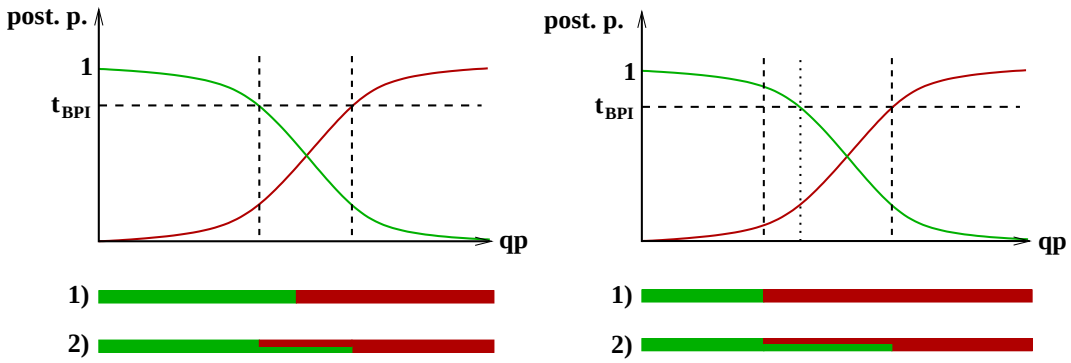

Figure 3: In both figures, for each query sequence position (qp), the posterior probabilities (post. p.) of two subtypes are plotted. t_{BPI} marks the posterior probability threshold for the definition of breakpoint intervals. Vertical dashed lines define the extent of breakpoint intervals. The first bar (1) below the plot of the posterior probabilities shows the original recombination prediction with precise breakpoint positions. The second bar (2) shows the predicted recombination including breakpoint intervals (two-color region). In a) a breakpoint interval around the predicted breakpoint is defined by the region where the posterior probability of the predicted subtypes (green and red) is below t_{BPI}. In b) the original left end (dotted line) of the breakpoint interval defined by the posterior probabilities of the predicted subtypes is moved to the left (dashed line) to include the predicted breakpoint position.

least $8\,\%$. Thus, in some regions in the HBV genome, the genetic divergence may also exceed $10\,\%$. In these regions, poor or even no alignments will be produced with BLAT. But, since most of the genomic regions show a much lower genetic divergence and as we are only interested in reliable alignments, only alignments in conserved regions are taken into account anyway. In variable regions, the whole region is assigned to the respective query sequence position. A detailed description of the algorithm can be found in [5].

2.3.2 Accuracy

To assess the accuracy of the pre-alignment with BLAT, each sequence in the multiple alignment was chosen as query sequence. It was aligned pairwisely to the chosen set of sequences so that each sequence position is mapped to a certain set of alignment columns. For each sequence position, it was checked whether the assigned region in the alignment contains the original alignment column of the sequence position. For the given HBV and HIV-1 alignment, 100% and 99.95% of the sequence positions were correctly aligned, respectively. The 0.05% of the HIV-1 sequence positions that were not able to be correctly aligned are located either in insert regions or are part of repeats of which one corresponds to an insertion in the alignment. For these positions, an alignment is not possible.

The jpHMM recombination prediction on the basis of a pre-alignment of the query

sequence to the given multiple sequence alignment was compared to that of the original jpHMM using semi-artificial recombinant sequences [7]. For all tested sequences, the results of both methods were identical. Additionally, the jpHMM runtime could be reduced by half.

2.3.3 Usage and download of BLAT

The latest jpHMM version contains an implementation for the pre-alignment of the query sequence to the given multiple alignment with BLAT. If you want to use such a pre-alignment for a speed-up of the program, please use option `-Q blat` (see section 5, Additional optional parameters). Additionally, it is necessary to download the BLAT program and adapt the respective path to the executable in the jpHMM source code (`src/const_def.cpp`). For a detailed description please see section 3.

The BLAT source and executables are freely available for academic, nonprofit and personal use (`http://genome.ucsc.edu/FAQ/FAQblat.html`). Commercial licensing information is available on the Kent Informatics website (`http://www.kentinformatics.com/`). BLAT source may be downloaded from `http://www.soe.ucsc.edu/ kent`. For BLAT executables, go to `http://hgdownload.cse.ucsc.edu/admin/exe/`; and choose your machine type.

2.4 Applications

Originally, jpHMM was applied to detect recombinations in genomic sequences of HIV-1, but it can also be used to detect recombinations in other viruses. We implemented a jpHMM version for viruses with circular genomes and applied it to HBV. For HIV-1 and HBV, jpHMM is available as a webserver (`http://jphmm.gobics.de`).

To run jpHMM for other viruses, the user only needs a multiple sequence alignment subdivided into different subtypes/ subfamilies, and a file containing the prior parameters/pseudocounts for the jpHMM parameters. Additionally, the jump probability (option `-j`, see section 5.1) can be adjusted. For viruses with a larger genome and a larger number of subtypes than the HIV-1 genome, the beam-width parameter `-B` (see section 5.1) should be adjusted. This parameter influences the size of the search space for the Viterbi algorithm (this is explained in detail in [8]). Larger values (e.g. `-B 1e-10` instead of `-B 1e-20`) lead to a stronger restriction of the search space.

For recombination detection in sequences from the hepatitis B (HBV) and hepatitis C virus (HCV), the jpHMM parameters are provided. For HIV-1, we use a jump probability of `-j 1e-09`, and for HBV a jump probability of `-j 1e-07`. The HIV-1 alignment is divided into 14 (sub)subtypes, the HBV alignment into 8 genotypes.

For other viruses, we did not estimate any model parameters yet. You can either use the given parameters or set, for example, all pseudocounts to 1 (see description below) for a first try. We are looking forward for suggestions for applications.

2.4.1 Viruses with linear genomes such as HIV-1

As HIV-1, HCV is a virus with a genome in a linear form. Thus, the application of jpHMM to recombination detection in HCV genomes is identical to the application to HIV-1 genomes as it will be described below. It is only necessary to use appropriate parameters. This is also described below.

2.4.2 Viruses with circular genomes such as HBV

We also developed a jpHMM version for viruses with genomes in a circular form such as the hepatitis B virus (HBV) [6].

Recombination analysis in circular genomes is usually performed on the linearized version of the genome using linear models. When local dependencies exist in a circular genome, these imply dependencies between the 5' and 3' end in the linearized version of the genome. Since linear models are usually unable to model such dependencies, this can result in inaccurate predictions of recombination breakpoints and thus in incorrect classification of circular viral genomes.

The jpHMM version for circular genomes takes into account the circularity of the genome and is not biased against recombination breakpoints close to the 5' or 3' end of the linearized version of the genome: Each full-length (linearized) input sequence is extended at both sequence ends such that the prefix (5') and the suffix (3') of the sequence are copied and concatenated to the original 3' and 5' end, respectively. By this extension, dependencies between nucleotides at the 5' and 3' end of the linearized sequence can be taken into account in the recombination prediction. Additionally, the given multiple sequence alignment is extended by duplicating it and copying and concatenating a prefix to the end of the alignment. This allows an alignment of extended, full-length as well as of fragmental sequences to the multiple alignment, regardless of the chosen origin for the sequence coordinates. On the basis of this extended alignment, the model is built.

To use the circular jpHMM, choose option `-C` (see section 5).

Since each query sequence can be nearly completely (full-length sequences) or completely (fragmental sequences) aligned to two different regions in the extended multiple sequence alignment, it is useful to define the location of the query sequence in the multiple sequence alignment before jpHMM is applied. For this purpose, each query sequence is initially aligned to the extended multiple sequence alignment with BLAT (as described in the previous section 2.3) to reduce the runtime and memory of the program. To use this pre-alignment, it is necessary to download the BLAT program and adapt the respective file names and variables as described above. If BLAT is not available, the recombination prediction is performed without any pre-alignment.

Graphical output and reference genome for HBV The output of the method is visualized in a circular form (Fig. 4) using the software package Circos [3]. All sequence position numbers are given relative to the HBV reference genome AM282986. Circos is called with the Perl script `jphmm_out_to_circos.pl` (see section 3). The output of

this script also includes the predicted recombination with position numbers given relative to the reference genome in text format. To use this script and to plot the output in circular form, it is necessary to download the program from the Circos homepage `http://mkweb.bcgsc.ca/circos/`.

For extended, full-length sequences, the output only comprises the prediction for the original input sequence. Different subtypes at the 5' and 3' end of the sequence imply a breakpoint at this position. The circular jpHMM is currently applied to detect recombinations in genomic sequences of HBV. The estimated model parameters are available for download. If you do not use option `-C` in the command line, the original jpHMM, i.e. the linear jpHMM version, is applied and the HBV genomes are treated as linear genomes.

The circular jpHMM approach can also be applied to other viruses with circular genomes. Please adapt the respective variable names in `src/const_def.cpp`. For plotting the predicted recombination for other viruses with circular genomes the Circos input files (see section 3) must be adapted. For detailed information about the Circos input format please also visit the Circos homepage.

3 Installation

jpHMM is delivered as a command line program called `jpHMM`. It is written in the programming language C++.
Currently, the program is implemented for Linux. To make it executable for Windows, the line `FILE *p = popen(buffer, "r")` in `jpHMM.cpp` (see below), around l. 211, must be changed (maybe to `FILE *p = _popen(buffer, "r")`).

1. If you want to use the pre-alignment option, please download BLAT: The BLAT source may be downloaded from `http://www.soe.ucsc.edu/ kent`. For BLAT executables, go to `http://hgdownload.cse.ucsc.edu/admin/exe/`; and choose your machine type.

2. If you want to to apply the method to circular genomes and plot the output in a circular form, please download the software package Circos from `http://mkweb.bcgsc.ca/circos/`.

3. Unpack the source code: `tar -xzf jpHMM.tar.gz`
   The tar-archive contains one directory `jpHMM/` with the six subdirectories `doc/`, `src/`, `input/`, `priors/`, `example/` and `obj/`.
   The subdirectory `doc/` contains a documentation for jpHMM.

   The subdirectory `src/` contains

   - the jpHMM source code:
     - `Makefile`,
     - `alignment.h, alphabet.h, const_def.h,`

`emissionPriors.h, emissionProfile.h, emTransProfile.h, error.h, extEmTrProbs.h, hmm.h, html_cell.h, html_tab_def.h, html_tab.h, io.h, jpHMM.h, profileHMM.h, query.h, std_header.h, std_source.h, transitionPriors.h,transitionProfile.h, -alignment.cpp, alphabet.cpp, const_def.cpp, emissionPriors.cpp, emissionProfile.cpp, emTransProfile.cpp, error.cpp, extEmTrProbs.cpp, hmm_algorithms.cpp, hmm.cpp, hmm_output.cpp, html_cell.cpp, html_tab.cpp, jpHMM.cpp, main.cpp, profileHMM.cpp, query.cpp, test.cpp, transitionPriors.cpp, transitionProfile.cpp`

- a Perl script for evaluating the pre-alignment (optional) of the query sequence(s) to the given alignment with BLAT (section 2.3.1):
  `eval_blat_output.pl`
  If you use the pre-alignment option, please adapt the following file names in the jpHMM source code file `const_def.cpp` and indicate the correct paths, e.g.
  `m_blat_script_name = "~/jpHMM/src/eval_blat_output.pl"`
  and
  `m_blat_program_name = "~/Blat/blat"`

- a Perl script for extending full-length (linearized) circular genomes and the given multiple sequence alignment (section 2.4.2):
  `extend_circ_gen.pl`
  If you use the circular option (-C, see below), please adapt the following file name in the jpHMM source code file `const_def.cpp` and indicate the correct path, e.g.
  `m_ext_script_name = "~/jpHMM/src/extend_circ_gen.pl"`.
- the Boost C++ libraries (www.boost.org):
  `./libs/boost/`
- the Perl program for the determination of uncertainty regions and breakpoint intervals in the predicted recombination:
  `determine_UR_and_BPI.pl`
- a R script for plotting the posterior probabilities for HIV sequences:
  `R_posterior_probabilities_plot.R`
- a Perl script for plotting the predicted recombination and the posterior probabilities for (circular) HBV sequences in a circular form using the software package Circos:
  `jphmm_out_to_circos.pl`.
  If you want to plot the predicted recombination in circular form, please down-

load the software package Circos and adapt the directory of the Circos program in the script, e.g.
`my $circosDir = "~/Circos/circos-0.52/bin/circos"`

- the input files for Circos for HBV genomes:
  `./circos_input_files/`
  for plotting the predicted recombination for other circular genomes the input files must be adapted. For detailed information about the Circos input format please also visit the Circos homepage.
- a Perl script running jpHMM and the post-processing files in a pipeline:
  `call_jphmm.pl`

The subdirectory `input/` contains the input alignment files:

- `*_alignment.fas, * = HIV, HCV, HBV`

and a file mapping all columns in the multiple HBV sequence alignment to a position in the HBV reference genome (AM282986, included in the multilple alignment). This file is necessary for the visualization of the predicted recombination in circular form with Circos;

- `hbv_msa_to_refseq.txt`

The subdirectory `priors/` contains the emission and transition prior (pseudo counts for the emission and transition probabilities) input files:

- `emissionPriors_*.txt, * = HIV, HCV, HBV`

The subdirectory `example/` contains the following subdirectories and files for a toy example:

- `data/`:
  - `sample_seq_HIV.fas`: sample HIV-1 sequence
  - `sample_seq_HBV.fas`: sample HBV sequence
- `results/`:
  in the subdirectory `results/`, the output files for the HIV-1 and the HBV sample sequence are given respectively:
  `recombination.txt`, `recombination_incl_UR_and_BPR.txt`, `uncertainty_and_BP_regions.txt`, `recombination_incl_gaps.txt`, `recombination_without_positions.txt`, `viterbi_path.gff`, `posterior_probabilities_for_seq_1.txt`, `posterior_recombination_path.txt`, `query_to_ref_alignments.txt`, `variable_regions.txt`, `alignment_to_msa.txt`,

posterior_prob_plot_for_seq_1.png for HIV-1 and recombination_prediction_AB076679_A_AB073846_B.png for HBV (the recombination of this sequence is based on the multiple alignments given in the /input/ directory.

- LDA/:
  in the subdirectory LDA/, we provide example files illustrating the use of some optional parameters required for the usage of jpHMM demonstrated in [1].

The subdirectory obj/ is needed for the C++ object data.

4. In const_def.cpp, various parameters and file names are defined. If necessary, please adapt the path to the given input and priors directory (*m_input_dir_name*, *m_priors_dir_name*) (and, if necessary, to the output directory *m_output_dir_name*). If you use the pre-alignment option, please adapt the following file names and indicate the correct paths:
   m_blat_script_name = "~/jpHMM/src/eval_blat_output.pl"
   and
   m_blat_program_name = "~/Blat/blat"
   If you use the circular option (-C, see below), please adapt the following file name and indicate the correct path, e.g.
   m_ext_script_name = "~/jpHMM/src/extend_circ_gen.pl".

5. Move to the folder jpHMM/src/ and compile the program with the command 'make' (for Linux). Now the jpHMM program should be executable (./jpHMM).

6. Make the Perl and the R scripts executable.

4 Copyright

Please read the following files:
COPYRIGHT – copyright notice, and information on the free software license
LICENSE – Full text of the GNU Public License, version 2 (see COPYING)

5 Running jpHMM

5.1 Main jpHMM program

jpHMM is started in the command line by:
./jpHMM -s query_sequence_file -v virus_type ....
If you start the program in the src/ folder,

- for HIV, the usual program call is:
  `./jpHMM -s ../example/data/sample_seq_HIV.fas -v HIV`,
- for HBV:
  `./jpHMM -s ../jpHMM/example/data/sample_seq_HBV.fas -v HBV -C`.

Both automatically use the respective input alignment.

5.1.1 Parameters

Obligatory parameters

-s `query_sequence_file`:
`query_sequence_file` is the name of the file containing the query sequence(s) for which the recombination should be predicted. The number of query sequences is not limited. The format of the file is:

```
>name_of_sequence_1 ref_begin_seq1 ref_end_seq1
aagggtaca
>name_of_sequence_2 ref_begin_seq2 ref_end_seq2
ccgatacctaa
```

`ref_begin_seq*` and `ref_end_seq*` are optional parameters, they define the start and end position of the query sequence relative to the reference genome (HXB2 for HIV, AM282986 for HBV). Especially for short sequences, these two parameters can decrease the runtime of jpHMM enormously, since the search space is reduced considerably. They can be determined with the 'Los Alamos HIV Sequence Locator Tool' at `http://hiv.lanl.gov/`.
! All parameters are separated by ONE white space !

For the circular jpHMM, no start and end position are needed.

-v considered type of virus:
Only 'HIV', 'HCV' and 'HBV' are allowed yet. For these three viruses, the pseudo counts to estimate the paramters of jpHMM are delivered.
The program can, of course, be used for other viruses. Please see the description for `em_parameter_file` (below) for the correct format of the file containing the respective priors for the emission probabilities. In `const_def.cpp`, the program can be modified to use it for other viruses. (For HBV: if you do not choose option `-C` in addition, the genomic sequences are considered as linear sequences and the original jpHMM version is applied.)

Additional optional parameters

-a `em_parameter_file`:
`em_parameter_file` is the name (including the path to the corresponding directory) of the file containing the priors (pseudo counts) for the emission probabilities in the model. The default parameter files are
`../priors/emissionPriors_*.txt, * = HIV, HBV, HCV`.

-b `trans_parameter_file`:
`trans_parameter_file` is the name (including the path to the corresponding directory) of the file containing the priors (pseudo counts) for the transition probabilities in the model. The default parameter file is
`../priors/transition_priors.txt`. (For all types of viruses, the same transition priors are used.)

-i `multiple_alignment_file`:
`multiple_alignment_file` is the name (including the path to the corresponding directory) of the file containing a multiple sequence alignment subdivided into several subtypes in the following format (the number of subtypes and sequences is not limited, but note that the runtime and memory increases with the number of subtypes):

```
>>name_of_subtype_1
>name_of_sequence_1
aagtaatg
>name_of_sequence_2
aaataatc
>>name_of_subtype_2
>name_of_sequence_3
agggtcca
>name_of_sequence_4
ggggtcaa
>name_of_sequence_5
aaggtcca
```

The default alignment files are `../input/*_alignment.fas, * = HIV, HBV, HCV`.

-j Jump probability (jp). This is the probability for a jump from one subtype to any other subtype in the alignment. For HIV-1, the default parameter is `-j 1e-09`, for HBV it is `-j 1e-07`. Thus, the probability of a jump from, for example, HIV-1 subtype A to subtype B is $\frac{1e-09}{\text{number of subtypes}}$.

-B Beam width (bw). This parameter determines the size of the search space for the jpHMM. Default parameter is `-B 1e-20`. Larger values reduce the search

space and as consequence the runtime and the memory, but also the accuracy of jpHMM. Smaller values increase the accuracy of jpHMM, but also the runtime and the memory. For full-length HIV-1 genomes, the runtime for the default parameter `-B 1e-20` is about 1.5 GB.

-P `prior_dir`:
`prior_dir` is the name of the prior directory, where the parameter files (emission and transition) can be found. This parameter is only useful, if `-a` is not chosen. If `-a` is not chosen, this parameter must be chosen, if the program is not started in the directory `./jpHMM`. The default is `../priors/`.

-I `input_dir`:
`input_dir` is the name of the input directory, where the alignment files can be found This parameter is only useful, if `-i` is not chosen. If `-i` is not chosen, this parameter must be chosen, if the program is not started in the directory `./jpHMM`. The default is `../input`.

-o `output_dir`:
`output_dir` is the name of the output directory, where ALL jpHMM output files are stored. The default is `./output`.

-Q `speed-up_algo`:
`speed-up_algo` is the name of the algorithm that is used to speed up the program runtime. This can be either `beam_search` or `blat`. `blat` means that each query sequence is pre-aligned to the given multiple alignment using the program BLAT. A description for this algorithm and its usage is given in section 2.3.1. The default option is `beam_search` (The beam-search algorithm is always used, also when the option `blat` is chosen [5]). If you use the BLAT pre-alignment option, please adapt the following file names in the source code file `const_def.cpp` and indicate the correct paths: `m_blat_script_name` and `m_blat_program_name`. Then compile the program again (see above).

-C if this parameter is chosen, the input sequences are treated as *circular genomes*. Please find a description of the usage of jpHMM for circular genomes above in section 2.4.2. Per default the sequences are treated as non-circular sequences. If this option is chosen, automatically the `blat` option is chosen to pre-align the (extended) sequences to the extended multiple sequence alignment. If the BLAT program cannot be found, the circular jpHMM is applied without a pre-alignment of the sequences. This will extend the jpHMM runtime and memory.

-e File containing the emission probabilities. The emission probabilities have to be provided in list form, with each row being composed of the alignment position, the subtype and the emission probabilities. See `EP.txt` in `example/LDA` for an example.

-t File containing the transition probabilities. The transition probabilities have to be provided in list form, with each row being composed of the alignment position, the subtype and the transition probabilities (M→M, M→I, M→D, I→M, I→I, D→M, D→D). See `TP.txt` in `example/LDA` for an example.

-c File containing the consensus columns. The consensus columns have to be provided in list form, with each row containing one position in the alignment which is a consensus column. See `cons.txt` in `example/LDA` for an example.

Input files

- `emissionPriors_*.txt, * = HIV, HBV, HCV`:
  files containing the priors for the estimation of the emission probabilities, i.e. the pseudo-counts, for the respective virus.
  For HIV and HCV, these pseudo-counts are estimated for a given alignment using a Dirichlet distribution [9, 8]. For HBV, the pseudo-counts are estimated on the basis of training sequences optimizing the recombination prediction in terms of the accuracy of predicted breakpoint intervals and subtypes (described in Schultz *et al.*, in preparation). If you use other parameters, please use the same format for the parameter file.

Output files Except for the posterior probabilities, the results are saved within one single file for all query sequences.

- `recombination.txt`:
  predicted recombination for the query sequence(s) with breakpoints based on raw sequence positions (without gaps) of the tested sequence(s).

  ```
  >sequence_1 (bw=1e-20)
  1 152 A1
  153 280 B
  281 300 G
  301 350 B

  >sequence_2 (bw=1e-20)
  ...
  ```

  For each sequence, the beam-width (bw) that was used for the restriction of the Viterbi search space is given. For circular sequences, it is additionally given whether the sequence is a full-length or a fragmental sequence (i.e. if it was extended or not), and if BLAT was used as pre-alignment method.

- `recombination_without_positions.txt`:
  predicted subtypes for the query sequence(s) (without breakpoint positions; including chosen parameters)

```
>sequence_1 bw=1e-20, jp=1e-09 A1BG
>sequence_2 bw=1e-20, jp=1e-09 ...
```

- `posterior_probabilities_for_seq_*.txt`:
  for each query sequence, a file containing the posterior probabilities of the subtypes at each sequence position is given

```
 #sequence_1 A1 A2 B C ... G
1 0 0 0
0.9999 0.0001 0 0
0.9999 0.0001 0 0
1 0 0 0
...
```

- `alignment_to_msa.txt`:
  the alignment of each query sequence to the multiple sequence alignment: for each sequence position, the aligned column in the multiple alignment is given. Special cases: '0': insert at the beginning of the sequence, 'num alignment columns + 1': insert at the end of the sequence;

- `query_to_ref_alignments.txt`:
  for each query sequence, the (local) alignment of the sequence to the reference sequence (HXB2 for HIV, AM282986 for HBV) determined by jpHMM. Start and end position of the reference sequence in this pairwise alignment are indicated by the two numbers following the reference sequence name (e.g. 5 is the start and 400 the end position of the HXB2 sequence in the following example)
  Please note that UPPER-CASE letters are considered to be aligned, lower-case letters are NOT aligned. '.' denote identic aligned residues, '-' denote gaps.

```
 >HXB2_sequence 5 400
acA.........G......A......aaaa....A
>sequence_1
ccG.........A......G......aaga....G
```

- `recombination_incl_gaps.txt`:
  predicted recombination for the query sequence(s) with breakpoints based on sequence positions including gaps in the tested sequence(s) (only if the query sequence(s) contains gaps)

- `viterbi_path.gff`:
  predicted recombination for the query sequence(s) with breakpoints based on raw sequence positions in gff-format

- `posterior_recombination_path.txt`:
  the 'path' of the subtypes with the highest posterior probability at each position

- `variable_regions.txt`:
  only important for HIV sequences (for other viruses, this file is empty): the variable regions in the query sequence based on raw sequence positions, determined by the alignment of the query to the reference sequence (given in `query_to_ref_alignments.txt`; only if the reference sequence (HXB2 sequence) is given in the multiple alignment that is used to build the model).
  As variable regions in the HIV genome (HXB2 sequence), we use:
  $6615-6691, 6696-6811, 7110-7216, 7377-7477, 7602-7636$.

- If option `-Q blat` is chosen (speed-up by a pre-alignment of the query sequence to the multiple sequence alignment with BLAT; this option is automatically chosen for circular genomes):

  - `active_alignment_columns.txt`:
    For each query sequence position, the assigned region ([start_pos, end_pos]) in the multiple sequence alignment is given, e.g.

    ```
     #sequence_1
    1 3 1 4 2 4 ...
    ```

  - `raw_sequences.fas` and `blat_output.psl`:
    Temporary (BLAT) output files for the alignment of the query sequence to the given multiple sequence alignment. Not needed for the jpHMM method.

For each sequence, the output is only given if at least to one query sequence position a subtype could be assigned to.

Please notice that, in contrast to the web server output, the output of the recombination breakpoints is only based on raw query sequence positions, not relative to the reference genome (i.e. not in HXB2 numbering for HIV sequences).

5.2 Definition of uncertainty regions and breakpoint intervals

Run the script `determine_UR_and_BPI.pl` to determine uncertainty regions in the recombination prediction and breakpoint intervals:
`./determine_UR_and_BPI.pl perc_var_UR perc_non_var_UR perc_var_BPI perc_non_var_BPI`
It defines in one step the uncertainty regions and breakpoint intervals for all considered query sequences.

Parameters

- `perc_var_UR`: threshold for an UR in a variable region of the genome
- `perc_non_var_UR`: threshold for an UR in a non-variable region of the genome

- `perc_var_BPI`: threshold for a BPI in a variable region of the genome
- `perc_non_var_BPI`: threshold for a BPI in a non-variable region of the genome
- `recomb_out_file`: output file containing the predicted recombination incl. uncertainty regions and breakpoint intervals
- `UR_out_file`: output file containing the uncertainty regions and breakpoint intervals in each query sequence

All parameters are optional parameters. `recombination_incl_UR_and_BPR.txt` and `uncertainty_and_BP_regions.txt` are the default output files for `recomb_out_file` and `UR_out_file`.
If you don't choose any parameters `perc_...`, the default of 99% is chosen as default for each of the first four parameters.

Input files The script needs the following jpHMM output files as input:

- `recombination.txt`
- `variable_regions.txt` (optional)
- `posterior_probabilities_for_seq_*.txt`

Ouput files The following files are produced:

- `recombination_incl_UR_and_BPR.txt`
  (or the file specified by `recomb_out_file`:
  Recombination predicted by jpHMM (based on raw sequence positions) including uncertainty regions in the prediction and breakpoint intervals instead of exact breakpoint positions. Uncertainty regions are labeled with '?', but the subtype predicted by jpHMM is still included. For breakpoint intervals, both subtypes are given.
  Example:

```
 >sequence_1
1 150 A1
151 155 A1/B
156 280 B
281 300 ?/G
301 350 B
```

  Positions 151 - 155 are defined as a breakpoint interval between subtypes A1 and B, i.e. in this region, a breakpoint between subtype A1 and B is predicted, but the exact breakpoint position is not clear. For positions 281 to 300, subtype G was predicted by jpHMM, but since the posterior probability of subtype G is lower than the given threshold, this prediction is marked as uncertain.

- uncertainty_and_BP_regions.txt
  (or the file specified by UR_out_file:
  File containing the uncertainty regions and breakpoint intervals for each query sequence based on raw sequence positions.

```
 >sequence_1
#uncertainty regions
281 300 ?/G
#breakpoint intervals
151 155 A1/B
```

5.2.1 Graphical output of the posterior probabilities (for HIV)

For uncertainty regions, no parental strain can confidently be determined. By examining the graph of the posterior probabilities you may see which subtypes are most closely related in these regions.
For HIV, the posterior probabilities of each subtype at each position in the query sequence can be plotted with the following program:

R_posterior_probabilities_plot.R

This script is written in the programming language R that must be installed (www.r-project.org, open access). Additionally the R library 'seqinr' must be installed.
To run the script you can either start R in the jpHMM folder and run the script by the command

```
source("R_posterior_probabilities_plot.R -args al_type")
```

or start the script by a command line call, e.g.:

```
R -no-save < R_posterior_probabilities_plot.R
-args al_type >/dev/null
```

Please adapt the script to the subtypes used in your multiple HIV sequence alignment (multiple_alignment_file) (line 53 - 67)!
-args al_type is optional. al_type specifies the chosen alignment of the query sequences to the reference (HXB2) sequence. As default, the posterior probabilities for HIV sequences are plotted based on HXB2 numbering given by the alignment of the query to the HXB2 sequence predicted by jpHMM (query_to_ref_alignments.txt), if the HXB2 sequence is given in the alignment. You can of course use any other alignment. For the web server we use HXB2 numbering predicted by the Los Alamos HIV sequence locator tool (SLT).

For HBV, the posterior probabilities are plotted together with the predicted recombination in a circular form using the software package Circos [3]. The respective script will be available soon.

Input files The R script needs the following jpHMM output files as input:

- `recombination_withoutPositions.txt`
- `query_to_ref_alignments.txt` (or any other alignment you specify in the code)
- `posterior_probabilities_for_seq_*.txt`

Output files For each sequence the following file is produced:
`posterior_prob_plot_for_seq_*.png`

In Figure 1, at the bottom, the posterior probabilities are plotted for an artifical recombinant.

5.3 Graphical visualization of predicted recombination for circular genomes

For circular genomes, the predicted recombination and the posterior probabilities of the genotypes are visualized in a circular form (Fig. 4) using the software package Circos [3]. All sequence position numbers are given relative to the HBV reference genome AM282986 [4].s

The output is produced with the Perl script `jphmm_out_to_circos.pl`. The output of this script also includes the predicted recombination with position numbers given relative to the reference genome in text format. To use this script and to plot the output in circular form, it is necessary to download the program from the Circos homepage `http://mkweb.bcgsc.ca/circos/`.

Run the script by:
`./jphmm_out_to_circos.pl dir_for_circos_input_files recomb_pred_file_name recomb_incl_UR_and_BPI_pred_file_name query_alignment_to_msa_file_name msa2refSeq_file_name output_directory`.

Parameters/Input files

- `dir_for_circos_input_files`: directory containing the input files for Circos: `src/circos_input_files`
- `recomb_pred_file_name`: file with predicted recombinations for query sequences: `recombination.txt`

- `recomb_incl_UR_and_BPI_pred_file_name`: file with predicted recombinations including uncertainty regions and breakpoint intervals: `recombination_incl_UR_and_BPR.txt`
- `query_alignment_to_msa_file_name`: file with alignments of query to given alignment determined with jpHMM: `alignment_to_msa.txt`
- `msa2refSeq_file_name`: file with mapping of alignment columns to reference sequence (AM282986) positions: `input/hbv_msa_to_refseq.txt`
- `output_directory`: directory for the output: `./` if the script is called directly in the output directory

All parameters/input files are obligatory. In the jpHMM output directory, the script can be called by:

```
/jpHMM/src/jphmm_out_to_circos.pl  /jpHMM/src/circos_input_files/
recombination.txt recombination_incl_UR_and_BPR.txt
alignment_to_msa.txt  /jpHMM/input/hbv_msa_to_refseq.txt ./
```

If you call the script from another directory, some paths must eventually be adapted in the script.

Output files For each sequence, a subdirectory is produced in the given output directory. Besides format files for creating the plot, it contains the following files:

- `recombination_prediction_seqname.png` plot of the predicted recombination including posterior probabilities of the genotypes in png format. For an example, see Figure 4.
- `recombination_seqname.mapped.txt`
  predicted recombination mapped to the reference genome (i.e. sequence positions given relative to the reference genome).
- `recombination_incl_UR_and_BPR_seqname.mapped.txt`:
  predicted recombination including uncertainty regions and breakpoint intervals mapped to the reference genome.
- `uncertainty_and_BP_regions_seqname.mapped.txt`
  uncertainty regions and breakpoint intervals mapped to the reference genome.

5.4 Example

Try a test run in the directory `src/`:

1. `./jpHMM -s ../example/data/sample_seq_HIV.fas -v HIV`
   This should reproduce the jpHMM output files given in the subdirectory `example/results/HIV/`. The results of this test run are saved in the subdirectory `src/output/`, if no other output directory (`-o`) is given in the list of arguments.

2. `./determine_UR_and_BPI.pl`
   This should reproduce the output files `recombination_incl_UR_and_BPR.txt` and `uncertainty_and_BP_regions.txt` given in the subdirectory `example/results/HIV/`.

3. `R -no-save < R_posterior_probabilities_plot.R -args jpHMM >/dev/null` This should reproduce the output file `posterior_prob_plot_for_seq_1.png` given in the subdirectory `example/results/HIV/`.

6 Web server

jpHMM is also available as a web interface [11] for HIV-1 as well as HBV sequences at `http://jphmm.gobics.de/`. Figure 1 shows an excerpt of its output for an artificial HIV-1 recombinant. The predicted recombination including uncertainty regions and breakpoint intervals as well as the posterior probabilities of all subtypes are drawn. For HBV, the results are presented in a circular form (Figure 4). $t_{\text{BPI}} = t_{\text{UR}} = 0.99$ are used as default thresholds for the posterior probabilities. The original recombination prediction with precise breakpoints and without uncertainty regions can still be downloaded from the results page. Additionally, the alignment of the query sequence to the reference genome (HXB2 for HIV, AM282986 for HBV), determined with jpHMM, and a file containing the posterior probabilities of the subtypes in text format can be downloaded.

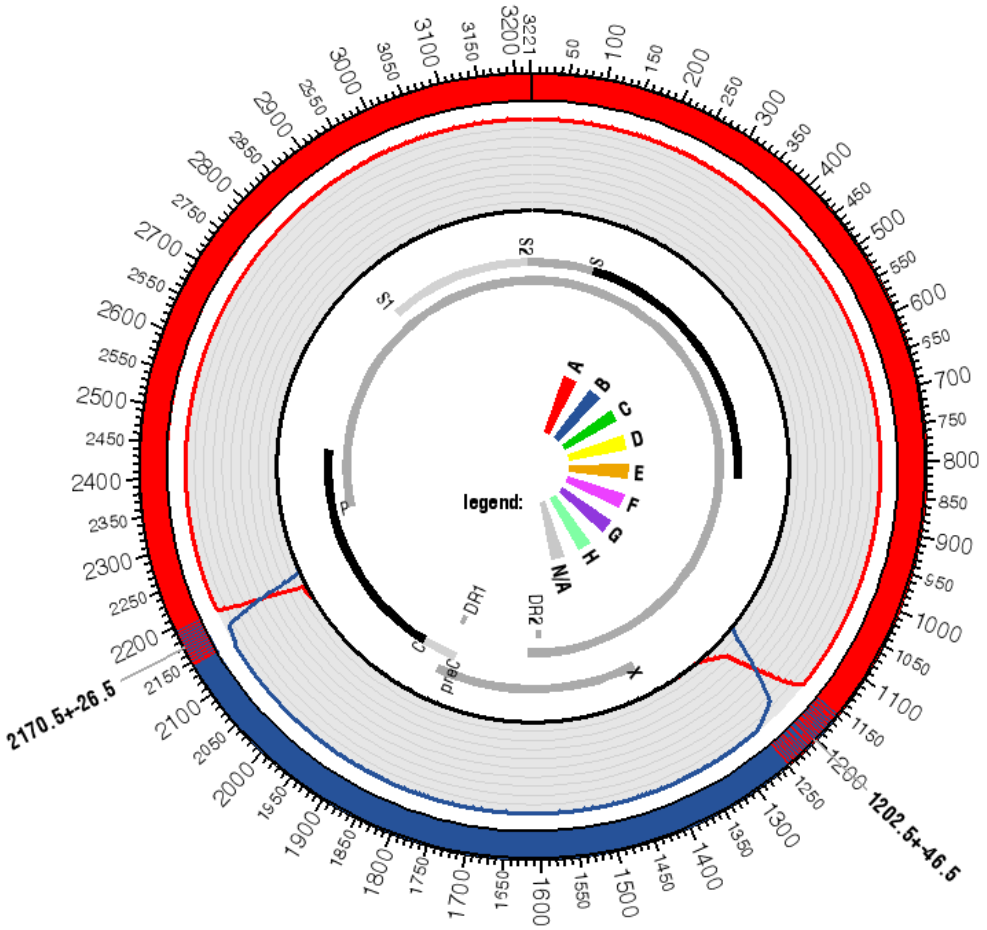

Figure 4: Predicted recombination for an artificial HBV recombinant plotted in circular form (outer ring). Regions with a shading of two colors mark breakpoint intervals, e.g. region 1202.5 ± 46.5. The posterior probabilities for each genotype are plotted in the second inner ring. In the inner circle, positions of genes in the HBV reference genome (AM282986) are marked with grey and black bars. In the middle of the figure, a color legend for the genotypes is given. N/A denotes for "not assigned". The plot is generated with Circos.

7 Contact

Dr. Anne-Kathrin Schultz
University of Göttingen
Institute of Microbiology and Genetics
Department of Bioinformatics
Germany
phone: +49-551-3913884
email: anne@gobics.de

References

[1] I. Bulla, A.-K. Schultz, and P. Meinicke. Improving Hidden Markov Models for Classification of Human Immunodeficiency Virus-1 subtypes through Linear Classifier Learning. *submitted*.

[2] W. J. Kent. BLAT - The BLAST-Like Alignment Tool. *Genome Research*, 12(4):656–664, 2002.

[3] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, 2009.

[4] N. Panjaworayan, S. Roessner, A. Firth, and C. Brown. HBVRegDB: Annotation, comparison, detection and visualization of regulatory elements in hepatitis B virus sequences. *Virology Journal*, 4(1):136, 2007.

[5] A.-K. Schultz. *Improvement of the jpHMM approach to recombination detection in viral genomes and its application to HIV and HBV*. PhD thesis, Georg-August-Universität Göttingen, 2011.

[6] A.-K. Schultz, I. Bulla, M. Abdou-Chekaraou, E. Gordien, B. Morgenstern, F. Zoulim, P. Dény, and M. Stanke. jpHMM: Recombination analysis in viruses with circular genomes such as the hepatitis B virus. *submitted*.

[7] A.-K. Schultz, M. Zhang, I. Bulla, T. Leitner, B. Korber, B. Morgenstern, and M. Stanke. jpHMM: Improving the reliability of recombination prediction in HIV-1. *Nucleic Acids Research*, 37(suppl 2):W647–W651, 2009.

[8] A.-K. Schultz, M. Zhang, T. Leitner, C. Kuiken, B. Korber, B. Morgenstern, and M. Stanke. A jumping profile Hidden Markov Model and applications to recombination sites in HIV and HCV genomes. *BMC Bioinformatics*, 7(1):265, 2006.

[9] K. Sjölander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet Mixtures: A Method for Improved Detection of Weak but Significant Protein Sequence Homology. *Comput. Applic. Biosci.*, 12:327–345, 1996.

[10] R. Spang, M. Rehmsmeier, and J. Stoye. A Novel Approach to Remote Homology Detection: Jumping Alignments. *J. Comp. Biol.*, 9(5):747–760, 2002.

[11] M. Zhang, A.-K. Schultz, C. Calef, C. Kuiken, T. Leitner, B. Korber, B. Morgenstern, and M. Stanke. jpHMM at GOBICS: a web server to detect genomic recombinations in HIV-1. *Nucleic Acids Research*, 34(suppl 2):W463–W465, 2006.

Last update by Anne-Kathrin Schultz, Goettingen, October 26, 2011