TABAJARA

A tool for rational design of profile HMMs

A Quick Guide and Tutorial

TABAJARA - A Quick Guide and Tutorial

1 Introduction

TABAJARA is a tool for rational design of profile HMMs. Starting from a multiple sequence alignment (MSA), TABAJARA is able to find blocks that are either (1) conserved across all sequences or (2) discriminative for two specific groups of sequences (Figure 1). For the identification of regions conserved across all protein sequences of an MSA, we implemented a previously described algorithm (Capra & Singh, 2007), based on Jensen–Shannon divergence method (Lin, 1991). This is the method of choice to determine the level of character conservation across all sequences of an MSA. In the case of nucleotide sequences, TABAJARA uses Shannon entropy (Shannon, 1948; Shannon & Weaver, 1949) to calculate position-specific scores. To find group-discriminative blocks, the program can use either Mutual Information (Adami, 2004; Cover & Thomas, 2006) or Sequence Harmony (Feenstra *et al*., 2007; Pirovano *et al*., 2006) for both, DNA or protein sequences.

What does the name TABAJARA mean? The classical definition is that TABAJARA is the name of one of Tupi tribes of indigenous people, the Tabajaras, that used to live along the Atlantic coast region of northeast Brazil. An alternative definition is that TABAJARA is the name of a fictional company (TABAJARA Organization) created by a Brazilian humoristic TV show in the 1990's, associated with products of doubtful utility and sensationalist advertising. In our context, TABAJARA is an acronym for **T**ool for **A**lignment **B**lock **A**nalysis **J**oining **A**ppropriate **R**ational **A**pproaches. We let the users to choose their favorite definition.

2 Main features

- Fully configurable
- Multiple sequence alignments of either protein or nucleotide (still experimental) sequences can be used
- Identification of the most conserved regions from a set of functionally and taxonomically related proteins using Jensen-Shannon divergence for protein sequences and Shannon entropy for nucleotide sequences

- Identification of group-specific regions from a set of functionally and taxonomically related proteins using Mutual Information (MI), Sequence Harmony (SH) and combined results
- Automatic elimination of redundant sequences and gap-only columns from MSAs
- Automatic choice of the best candidate alignment blocks
- Automatic construction of profile HMMs from selected alignment blocks
- Generates different types of text reports

3 The algorithm

`TABAJARA` uses as input MSA files in both FASTA and Clustal (`aln`) formats. According to the user's choice, the program can detect identical sequences and leave only one representative of each identical group. This step avoids biasing the alignment block selection and profile HMM construction. By default, `TABAJARA` identifies and removes gap-only columns. Depending on the dataset (nucleotide or protein sequences) and the chosen task (detecting sequence-conserved or group-specific regions), `TABAJARA` uses one of the aforementioned methods (see Introduction) to calculate position-specific scores across the entire alignment. Once position-specific scores have been determined, the program uses a sliding-window to screen the whole alignment and delimit top-scoring regions. A second sliding-window step can be performed using a smaller window to identify short high-scoring regions. This is particularly interesting if regions highly specific to a particular subset of sequences are sought. The program automatically extracts the selected alignment blocks, discards identical sequences, eliminates gap-only columns and builds the corresponding profile HMMs, which can then be used for downstream applications.

3 Self-validation of the models

After building de profile HMMs, `TABAJARA` executes a series of validation steps to discard models that do not fulfill a series of quality criteria. First, `TABAJARA` converts the training set MSA file into a plain-sequence FASTA file without gaps. Either in Conservation or in Discrimination mode, all models are submitted to similarity searches against all sequences of the training set using `hmmsearch` program. After saving the output in both tabular and complete formats, `TABAJARA` inspects the results and checks if the models have fulfilled

quality control criteria defined by the user. Approved models are stored, and the remaining are discarded.

3.1 Validating models for conservation

Running TABAJARA in Conservation mode will result in a set of models that should ideally be able to detect all sequences used in the training set. To validate the profile HMMs, TABAJARA invokes hmmsearch to run similarity searches of the models against the sequences of the training set, using default parameters. TABAJARA calculates for each model if the percentage of detected sequences is equal to or greater than the value defined in parameter -pd (see below). If the user declares a list of category prefixes (e.g. DENV and ZIKV) in parameter -c (see below), then a valid model will be required to detect a percentage of all sequences, plus the same percentage of sequences of each individual category. For example, being a dataset composed of 50 sequences of dengue virus, 40 of Zika virus and 310 sequences of other flaviviruses, if the user chooses parameters -c DENV,ZIKV and -pd 80, TABAJARA will only accept models that detect at least 80% of dengue (≥ 40) and Zika (≥ 32) sequences, in addition to detecting at least 80% of the total number of sequences of the training set (80% of 400 - ≥ 320 sequences). Finally, if parameter -cs yes is used, TABAJARA inserts a CUTOFF SCORE tag in the profile HMM, using a value corresponding to 80% of the score of the last similarity hit of the training set.

3.2 Validating models for discrimination

When running in Discrimination mode, TABAJARA also validates the models following a series of criteria. Given a dataset composed of sequences of the chosen category (group A) and the remaining sequences (group B), TABAJARA runs hmmsearch with default parameters using each model as a query against the training set, including sequences of groups A and B. The first requirement for a model to be accepted is that the top hit sequence must be a member of group A, otherwise the model is discarded. If all positive sequences belong to the chosen category (group A) and parameter -cs yes is used, TABAJARA inserts a CUTOFF SCORE tag in the profile HMM, corresponding to a percentage (defined by parameter -pt) of the lowest score.

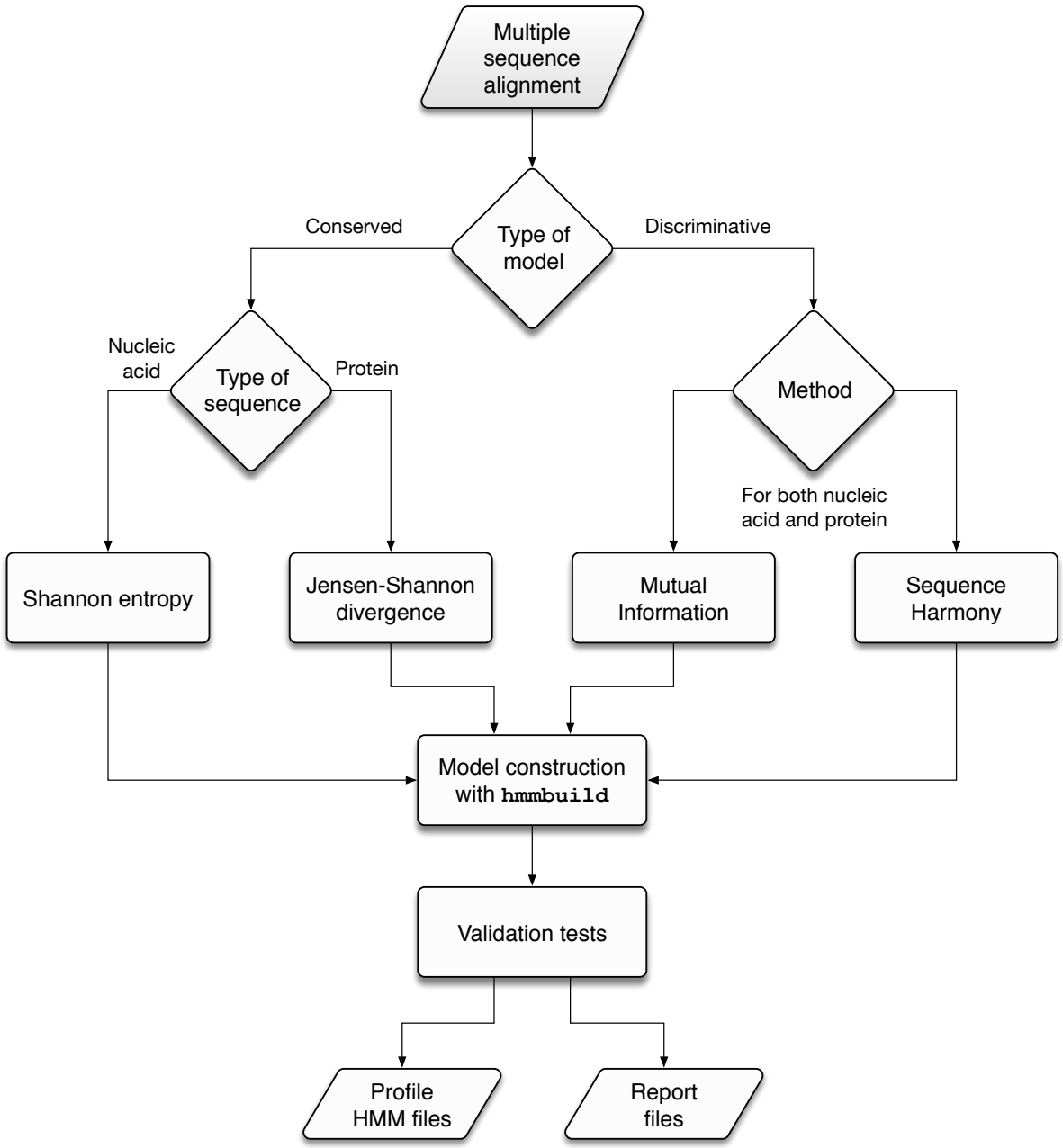


Figure 1 – Workflow of profile HMM construction. TABAJARA uses a multiple sequence alignment as an input training set. In conservation mode, alignment blocks can be selected for regions conserved across all sequences, using Shannon entropy for nucleic acid and Jensen-Shannon divergence for protein sequences. To select discriminative blocks, TABAJARA can use either Mutual Information of Sequence Harmony, or a combination of both methods. Selected alignment blocks are used to build profile HMMs with hmmbuild program. All HMMs are then submitted to validation tests and the selected models are stored, together with some report files of the pipeline.

There are many instances where hits corresponding to sequences of group B can also be found. TABAJARA screens the list of hits in decreasing order of score and calculates the ratio of the score of the first positive sequence of group B to the score of the immediately preceding sequence of group A. This value must be equal to or lower than the value defined by parameter -pt, otherwise the model is discarded. For example, given the list of hmmsearch results below and assuming that category Herbe is the chosen group, the ratio of 33.8 to 153.7 would be 21.99. Assuming a -pt value of 80, this model would be accepted.

Sequence	**Score**
Herbe_AGX32058.1	158
Herbe_AGX32061.1	155.4
Herbe_AFR34023.1	154.8
Herbe_AGX32060.1	154.6
Herbe_AGX32059.1	154.6
Herbe_AGX32057.1	153.7
Ortho_NC_022595_Murrumbidgee	33.8
Ortho_KJ481929_Buffalo_Creek	33.7
Ortho_KJ867181_Pongola	28.2
Ortho_KJ481923_Mapputta	27.6
Ortho_KF981633_Khurdun	26.1
Ortho_KM272183_Calchaqui	25.1
Ortho_KT950264_Alajuela_GAM_122	24.4

If the criterion above is fulfilled, and parameter -cs yes is used, TABAJARA inserts a CUTOFF SCORE tag in the profile HMM. The program now calculates the absolute difference between the scores of the first positive sequence of group B (33.8) and the immediately preceding sequence of group A (153.7). A value corresponding to 80% of this difference (in this case 0.8 x 119.9 – 95.92) is added to the score of the top performer sequence of group B (33.8). Thus, is this example the cutoff score would be 128.8. The rationale behind such heuristics is to guarantee that sequences of the chosen category that present a slightly lower score (80%) would be still detected by the model, whereas avoiding

the detection of sequences not belonging to this category. Finally, using the ascribed cutoff values of each model, TABAJARA verifies if the percentage of detected sequences of group A is equal to or higher than the value defined by parameter -pd. Only models that meet this condition are accepted and stored.

4 Before using TABAJARA

4.1 System requirements

TABAJARA was developed in Perl language and can be used in any POSIX compliant operating system such as UNIX and Linux distributions with an installed Perl interpreter (http://www.perl.org).

4.2 Third-party programs

TABAJARA requires the program hmmbuild (HMMER3 package - (http://hmmer.org/) to build profile HMMs. The program must be located in a directory listed in the PATH of the operating system.

5 Understanding TABAJARA parameters

5.1 Mandatory parameters:

TABAJARA has six mandatory parameters that must be specified by the user at the command line (or in the configuration file):

- -i <file name>: Input file (multiple sequence alignment - FASTA and Clustal formats are accepted)
- -t <decimal>: Score threshold of the alignment position for block extraction (valid values: 0 to 1)

`-w` `<integer>`: Window size for block extraction

`-p` `<integer>`: Percentage of positions in sliding window with score ≥`t`

`-b` `<integer>`: Minimum block size (value must be ≥ `w`)

`-m` `<c|mi|sh|b>`: Method to generate position-specific scores

TABAJARA uses several different methods to calculate position-specific scores, according to the global level of character conservation along all positions of the MSA. Alternatively, the program can evaluate each position for its ability of discriminating two groups of sequences. In this case, TABAJARA can use two distinct methods, Mutual Information or Sequence Harmony, or a combination of both. The following options are valid for parameter `-m`:

`c` – Conservation. TABAJARA automatically detects the nature of the sequences comprising the MSA. For protein sequences, TABAJARA uses a previously described algorithm (Capra & Singh, 2007), based on Jensen–Shannon divergence method (Lin, 1991). In the case of nucleotide sequences, TABAJARA calculates Shannon entropy (Shannon, 1948; Shannon & Weaver, 1949) and converts the results to normalized Shannon entropy values (Kumar *et al*., 1986). In both cases, position-specific scores are calculated to estimate the level of character conservation overall.

`mi` - Mutual Information. TABAJARA calculates position-specific scores using mutual information, as described by Adami (2004) and Cover & Thomas (2006).

`sh` - Sequence Harmony. TABAJARA calculates position-specific scores using complements of sequence harmony (Pirovano *et al*., 2006; Feenstra *et al*., 2007) values.

`b` - Both methods: Sequence Harmony and Mutual Information. TABAJARA calculates position-specific scores using both Mutual Information and Sequence Harmony. Using the sliding window approach, the best regions are selected for each method. The coordinates of both regions are compared to each other. Regions selected by both methods are stored. In case the program finds an overlap between regions of different methods,

these regions are merged and TABAJARA calculates an average score value for each position. Finally, an additional sliding window step is used to find the top scoring window within each merged region.

-c <string>: Name of the category(ies) to be analyzed. Ex. Alpavirinae. This parameter is only mandatory if a discriminative scoring method is chosen (Mutual Information, Sequence Harmony or both). Sequences belonging to a category are identified in the FASTA header by a prefix composed of an initial string of characters delimited at the end by an underline character. For example, the MSA of *Microviridae* sequences, provided as a dataset for this tutorial, presents the following naming format:

```
>Alpavirinae_Human_feces_C_016_Mi
>Gokushovirinae_Human_gut_34_012_Microvir
>Pichovirinae_Pavin_279_Microviri
```

For example, using Alpavirinae as the string of parameter -c will imply that all sequences whose headers start with Alpavirinae will be treated as members of a single group, while all remaining sequences will compose a second group. Multiple categories can be specified, separated by commas:

```
-c Alpavirinae,Gokushovirinae,Pichovirinae
```

In this example, after processing and finding Alpavirinae-specific blocks, TABAJARA will repeat the process for Gokushovirinae and Pichovirinae, treating the respective subsets of sequences as groups to be compared against all remaining sequences.

Name prefixes are also used to guide the model validation step (see section "Self-validation of the models"). When using TABAJARA in Conservation mode, declaring sequence categories is optional. However, if these prefixes are used, the program will report the detection rate of the models, discriminated according to each group of sequences. If prefixes are not declared, then

TABAJARA will only report the detection rate in regard to the whole set of sequences of the training set.

5.2 Optional parameters:

-clean <yes|no> (default = yes) - Delete some processing files.

-conf <configuration file> - use a configuration file that lists all parameters for execution, overriding any parameter of the command line. An example of a configuration file follows below:

```
i=peribunyaviridae.aln
o=output_dir
m=b
t=0.5
p=50
c=Herbevirus,Orthobunyavirus
w=15
sa=1
b=20
md=3
gs=20
wg=yes
di=yes
cs=yes
pd=80
gc=30
wg=yes
pt=80
sv=0.8
mb=60
sa=1
sr=yes
```

-cs <yes|no>: Insert cutoff scores in the profile HMMs (default = no)

-di <yes|no>: Discard identical sequences (default = no).

-gc <integer>: Gap cutoff (default=30). Percentage of allowed gaps in each position of the MSA. If most of the sequences in a given position are represented by gap characters, this position may introduce noise to the model (the profile HMM). To avoid this occurrence, positions presenting higher percentages than the gc value will receive a zero score. If parameter -wg yes is used, any window containing zero-score positions will be discarded. Therefore, this former parameter allows to discard blocks containing any number of gaps. In Conservation mode, the percentage of gaps in a column is

calculated using all sequences of the alignment. Conversely, in Discrimination mode, calculation only takes into account the sequences of the chosen category. This latter feature allows to select group-specific blocks that are absent in the remaining sequences (they correspond to an inserted region).

`-gs <integer>`: Percentage of allowed gaps in each sequence of the selected alignment block. Any sequence presenting higher percentages of gap characters than `gs` is discarded from the MSA and not used in the construction of the profile HMM.

`-h:` Print a help screen.

`-hb <string>`: Any set of `hmmbuild`'s valid parameters can be entered under double quotes. Example: `-hb "--wblosum -wid 0.8"`. Default: no parameters.

`-mb <integer>`: Maximum block size. Given an alignment block with a minimum length defined by parameter `-b`, the user can also define the maximum length of this block. `TABAJARA` scans the region with another sliding window with `-mb` length and then selects the sub-region which presents the highest score sum.

`-md <integer>`: Maximum accepted distance for two alignment blocks to be joined as a single block. If `TABAJARA` finds two contiguous alignment blocks that are separated by a distance equal or shorter than `md`, the program will join them into a single block, otherwise the blocks will be stored separately.

`-o <file name>`: Output directory (default = `output_dir_#`).

`-pd <integer>`: Minimum detection rate (in percentage) of training set (MSA) sequences by the constructed profile HMM. This is equivalent to the minimum accepted sensitivity of the model (default: `80`).

`-pt <integer>`: This parameter applies for the validation of HMMs designed for group discrimination. Given two groups, A (selected category) and B (remaining sequences), an `hmmsearch` is performed with the constructed model against the training set (MSA) sequences. The highest score obtained by group B sequences must be lower than n % (defined by parameter `-pt`) of the lowest score obtained by group A sequences, otherwise the model is not accepted (default = `80`).

`-sa <integer>`: Minimum allowed number of sequences to build a profile HMM (default = `2`). This value applies to the original MSA and to the post-scoring selected alignment blocks. In both cases, the redundant sequences are identified and discarded. `TABAJARA` only builds a profile HMM from an alignment if the total number of sequences remaining after redundancy removal is equal or greater than the number defined by parameter `-sa`. In general, no model should be built with less than two sequences. In fact, `hmmbuild`, the program that constructs the models, does not accept a single sequence as input. However, if a value of 1 is specified, `TABAJARA` creates multiple copies of the single sequence and this pseudo set is used to build the profile HMM.

`-sr <yes|no>`: If a maximum block size (parameter `-mb`) is defined, in addition to the selection of the best alignment block, the program also screens the remaining stretches of sequence to find additional blocks that fulfill the specified scoring selection criteria. This option maximizes the number of models that can be built from a high-scoring region (default = `no`).

`-sv <integer>`: Minimum accepted score per alignment position to be ascribed in the `CUTOFF SCORE` tag of the profile HMM (default = `1`). For instance, if a `-sv 1` is used for a 20-position block, then the ascribed `CUTOFF SCORE` will be 20. This parameter supersedes the standard `CUTOFF SCORE` calculated in the model validation step (see parameter `-pt`), thus avoiding too low cutoff values.

`-v`: Print the program version.

`-wg <yes|no>`: Discard sliding windows presenting gaps (default = `no`).

6 Running TABAJARA

To install the data for the tutorials, copy the file `tabajara.tar.gz` to a directory of your choice. Decompress the file using the following command:

```
tar xzvf tabajara.tar.gz
```

This command will create the `tutorial` directory. This directory contains the following items:

- Subdirectory `data` - contains two multiple sequence alignment files:
  - `microviridae.fasta` – contains 83 protein sequences of VP1 (major capsid protein) of *Microviridae* phages (Roux *et al.,* 2012).
  - `flavivirus.fasta` – contains 100 polyprotein sequences from viruses of the *Flavivirus* genus (Prof. Paolo M. A. Zanotto, unpublished).
- Subdirectory `results` - contains multiple subdirectories with pre-run results for *Microviridae* e *Flavivirus* sequences, as expected for the commands covered in this tutorial.
- `tabajara.pl` – the executable file.
- `tabajara.xls` – an MS Excel file containing spreadsheets and graphs with results of the commands covered in this tutorial.
- `Manual.pdf` – a PDF file of this tutorial.

6.1 Example 1 – Finding regions conserved across all sequences of a multiple sequence alignment

6.1.1 Executing TABAJARA

The commands below executes `TABAJARA` using MSA files `microviridae.fasta` and `flavivirus.fasta` as input, respectively.

```
./tabajara.pl -i data/microviridae.fasta -t 0.5 -w 15 -p 50 -b
15 -m c -md 3 -gs 20 -mb 20 -o microviridae_dir
```

```
./tabajara.pl -i data/flavivirus.fasta -t 0.5 -w 15 -p 50 -b
15 -m c -md 3 -gs 20 -mb 20 -o flavivirus_dir
```

6.1.2 Understanding the parameters

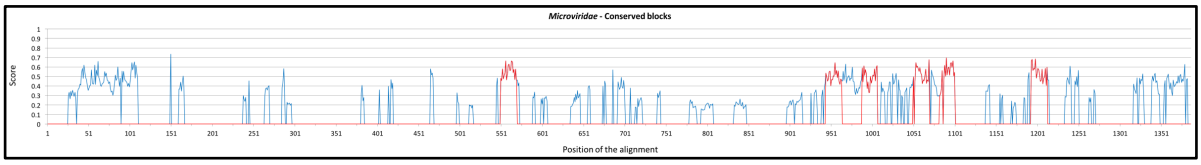
In a first step, the program estimates the conservation level across all sequences (parameter `-m`) in all positions of the MSA, using the algorithm described by Capra & Singh (2007). Following scoring, `TABAJARA` runs a 15-base sliding window (parameter `-w`) where at least 50% of the bases (parameter `-p`) must have scores higher than 0.5 (parameter `-t`). If two contiguous alignment blocks are located up to 3 bases from each other (parameter `-md`), they are joined into a single block, otherwise they are treated separately. Finally, the program scans the selected blocks using a 20-base sliding window (parameter `-mb`) and calculates the score sums base by base. The window presenting the highest score sum is then used to select the final alignment region from each previously designated block. This second sliding window step allows to refine and restrict the selected region to a short and specific alignment block. This short alignment block is then extracted from the MSA. Gap-only columns are deleted and sequences presenting more than 20% gap characters (parameter `-gs`) are discarded. The extracted MSA block is used as input by `hmmbuild` to construct a profile HMM. Finally, all output files are stored within the specified output directory (parameter `-o`).

6.1.3 Inspecting the output files

Once `TABAJARA` finishes the processing, an output directory is created. Within this directory, you will find several files and subdirectories created by `TABAJARA`:

- `logfile.txt`: a log file created by `TABAJARA` on every run. This file stores all the parameters utilized to run `TABAJARA`. A list of all identified alignment blocks and selected regions are also stored.
- `scores.csv`: this file is composed of tab-separated columns listing all positions of the MSA and their corresponding scores, respectively. Score values, ranging from zero to one, are ascribed according to the chosen scoring method. An additional column lists those scores restricted to positions of the alignment blocks selected by the chosen method (Conservation, Mutual Information or Sequence Harmony), with all remaining positions presenting a score of zero. If `–m b` option is used, three additional columns are presented, corresponding to scores of Mutual Information, Sequence Harmony and the combination of both methods, respectively. This file can be used as input for a spreadsheet, to easily generate a graph depicting scores along the MSA, with selected blocks highlighted in a different color, as exemplified in Figure 2.

A



B

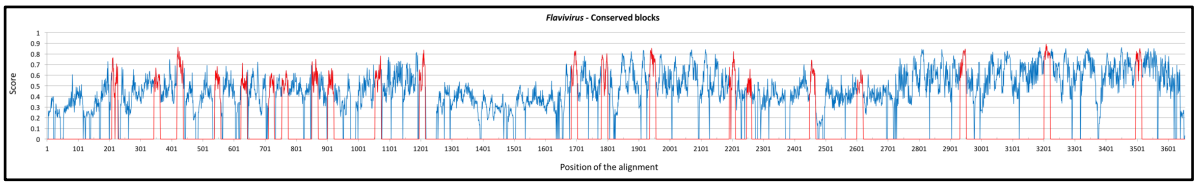


Figure 2 – Position-specific scores along multiple sequence alignments (MSA) of 83 VP1 protein sequences of *Microviridae* phages (A) and 100 *Flavivirus* polyprotein sequences (B). Selected regions conserved across all sequences of the MSA, according to Jensen–Shannon divergence, are indicated in red.

- `selected_blocks.txt`: this text file lists the MSA coordinates of all selected alignment blocks.
- `original_blocks`: this directory contains multiple-sequence FASTA files, each one containing sequences restricted to the coordinates of extracted alignment blocks. These files may contain identical sequences.
- `blocks_without_redundancy`: this directory contains files that are similar to those within `original_blocks` directory, but without identical sequences and gap-only columns.
- `hmms`: contains the profile HMM files generated by `TABAJARA` from the selected alignment blocks. Files are named with three numbers, separated by underlines, which correspond to location order in the MSA and respective initial and end coordinates of the source blocks.
- In case `TABAJARA` finds identical sequences and/or gap-only columns in the original MSA file, the program also stores an MSA without these occurrences. The naming is composed of the original file name followed by `_no_redundancy.fasta`.

6.2 Example 2 – Finding alignment blocks discriminative for a group of sequences/taxa

6.2.1 Finding discriminative regions for *Alpavirinae* and *Gokushovirinae* (*Microviridae* family).

The command below executes `TABAJARA` using the MSA file `microviridae.fasta` as input. The parameter `-m mi` is used for Mutual Information. The parameter `–c` with the option `Alpavirinae` indicates that all sequences starting with the prefix `Alpavirinae` should be treated as a group and all remaining sequences as another group.

```
tabajara.pl -i data/microviridae.fasta -t 0.5 -w 15 -p 50 -b
15 -m mi –c Alpavirinae -md 3 -gs 20 -mb 20 –o alpa_mi_dir
```

Parameter `–m` can also be used with options `sh` for Sequence Harmony and `b` for a combination of both methods, respectively. Suggested commands:

```
tabajara.pl -i data/microviridae.fasta -t 0.5 -w 15 -p 50 -b
15 -m sh –c Alpavirinae -md 3 -gs 20 -mb 20 –o alpa_sh_dir
```

and...

```
tabajara.pl -i data/microviridae.fasta -t 0.5 -w 15 -p 50 -b
15 -m sh –c Alpavirinae -md 3 -gs 20 -mb 20 –o alpa_sh_dir
```

You can also run `TABAJARA` to select *Gokushovirinae*-specific regions (parameter `–c Gokushovirinae`) using either MI or SH, or a combination of both. Suggested commands:

```
tabajara.pl -i data/microviridae.fasta -t 0.5 -w 15 -p 50 -b 15 -m
mi –c Gokushovirinae -md 3 -gs 20 -mb 20 –o gokush_mi_dir
```

```
tabajara.pl -i data/microviridae.fasta -t 0.5 -w 15 -p 50 -b 15 -m
sh –c Gokushovirinae -md 3 -gs 20 -mb 20 –o gokush_sh_dir

tabajara.pl -i data/microviridae.fasta -t 0.5 -w 15 -p 50 -b 15 -m b
–c Gokushovirinae -md 3 -gs 20 -mb 20 –o gokush_mi_sh_dir
```

As can be seen in Figure 3, alignment blocks specific for *Alpavirinae* and *Gokushovirinae* have been found. Is it worth mentioning that some most of these regions are common to both subfamilies. This is not contradictory, but rather reflects the fact that some alignment regions are conserved within members of the same taxonomic groups, but are divergent from each other, allowing for discrimination between groups.

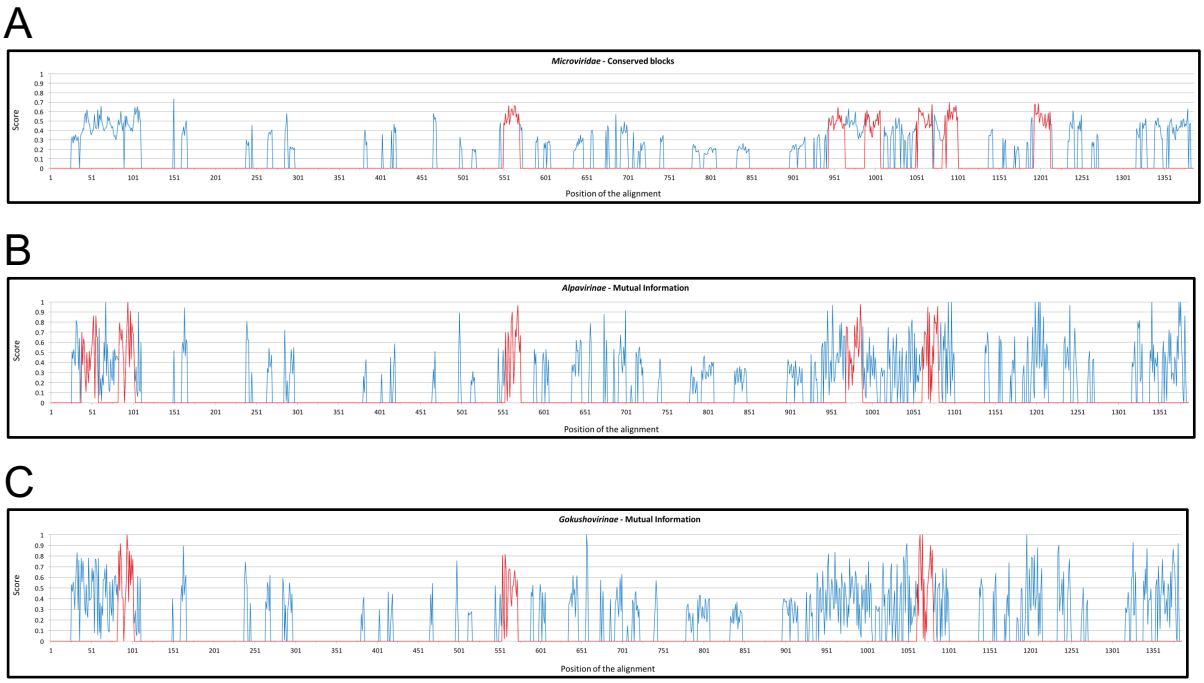


Figure 3 – Position-specific scores along a multiple sequence alignment (MSA) of 83 VP1 protein sequences of *Microviridae* phages. `TABAJARA` selected regions (indicated in red color) that are conserved across all *Microviridae* sequences based on Jensen–Shannon divergence (A), or regions specific to member of *Alpavirinae* and (B), *Gokushovirinae* (C) subfamilies, according to Mutual Information (Adami, 2004; Cover & Thomas, 2006).

6.2.2 Finding discriminative regions for Dengue, Zika and Yellow Fever viruses (genus *Flavivirus*)

Using the MSA provided in this tutorial (file `flavivirus.fasta`), it is possible to select regions that are specific for DENV, ZIKV and YFV. As can be seen in Figure 1, Members of the genus *Flavivirus* (Fig. 4A) are much more conserved than viruses of the *Microviridae* family (Fig. 3A). This means that sometimes we need to use more relaxed parameters (e.g. `–t 0.45` instead of `0.5`), with the consequence of possibly higher occurrences of cross-reactivity. We suggest the following parameters for DENV, using Mutual Information, Sequence Harmony, or a combination of both methods, respectively:

```
tabajara.pl -i data/flavivirus.fasta -t 0.5 -w 15 -p 50 -b 15
-m mi -md 3 -gs 20 -mb 20 -c DENV -o DENV_MI_dir
```

```
tabajara.pl -i data/flavivirus.fasta -t 0.5 -w 15 -p 50 -b 15
-m sh -md 3 -gs 20 -mb 20 -c DENV -o DENV_SH_dir
```

```
tabajara.pl -i data/flavivirus.fasta -t 0.5 -w 15 -p 50 -b 15
-m b -md 3 -gs 20 -mb 20 -c DENV -o DENV_MI_SH_dir
```

For YFV, just replace the option of parameter –c to YFV and change the name of the output directory. For Mutual Information, the suggested command is:

```
tabajara.pl -i data/flavivirus.fasta -t 0.5 -w 15 -p 50 -b 15
-m mi -md 3 -gs 20 -mb 20 -c YFV -o YFV_MI_dir
```

For ZIKV, replace the option of parameter `–c` to `ZIKV`, use `–t 0.45` instead of `0.5` (more relaxed value) and change the name of the output directory. For Mutual Information, the suggested command is:

```
tabajara.pl -i data/flavivirus.fasta -t 0.45 -w 15 -p 50 -b 15
-m mi -md 3 -gs 20 -mb 20 -c ZIKV -o ZIKV_MI_dir
```

Figure 4 shows the results obtained for *Denguevirus* using Mutual Information, Sequence Harmony or a combination of both methods. A list of the coordinates of the selected blocks is shown in Table 1. If a region selected by Mutual Information overlaps with another region obtained by Sequence Harmony, `TABAJARA` collapses both into one single region, scans this

region with a sliding window and then selects the sub-region which presents the highest score sum. For this reason, the coordinates listed in the MI+SH columns of Table 1 can be slightly different from the original coordinates found for Mutual Information (MI) and Sequence Harmony (SH).

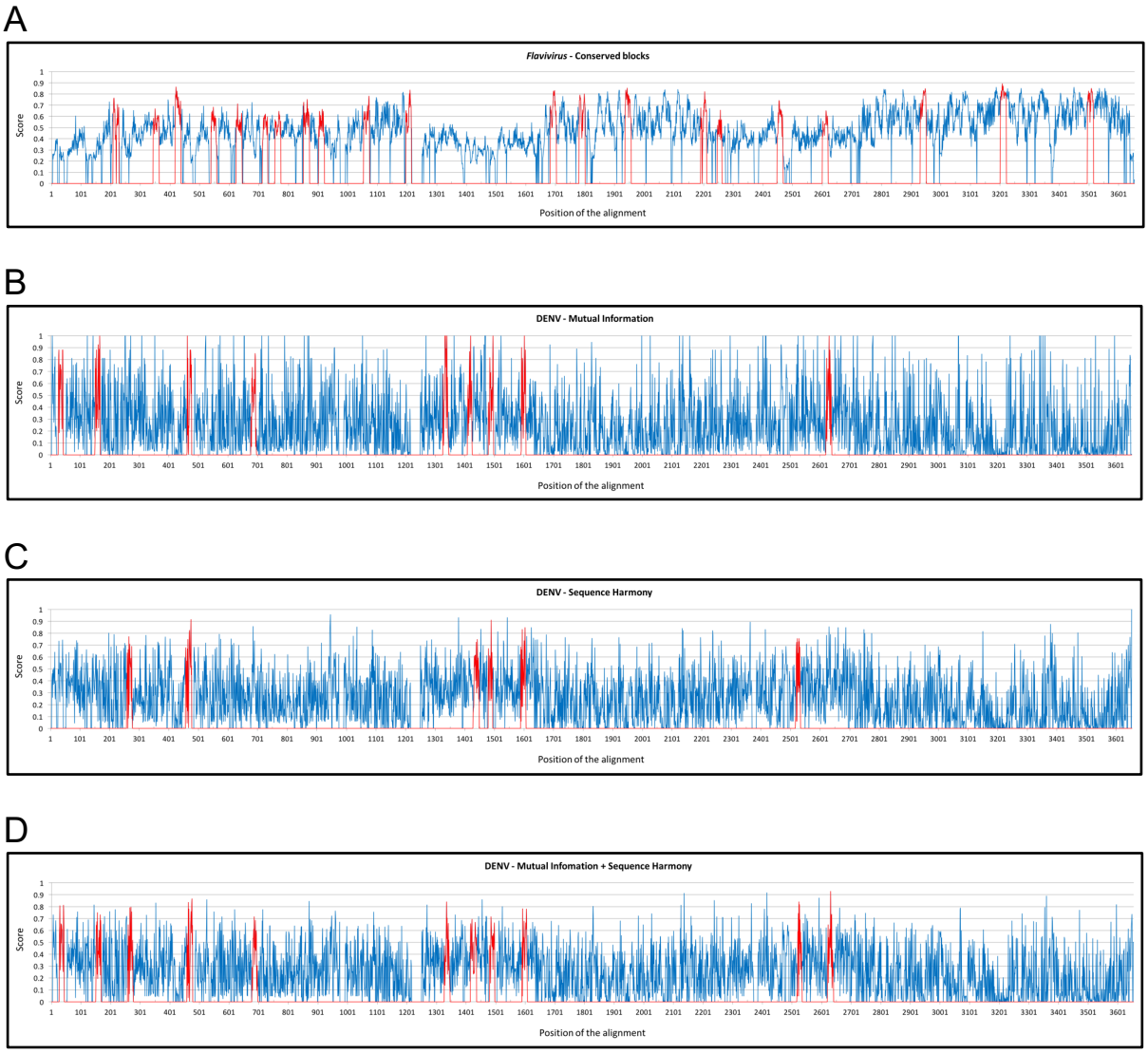


Figure 4 – Position-specific scores along a multiple sequence alignment (MSA) of 100 polyprotein sequences of members of the genus *Flavivirus*. TABAJARA selected regions (indicated in red color) that are conserved across all *Flavivirus* sequences based on Jensen–Shannon divergence (A), or regions specific to *Denguevirus* using Mutual Information (B), Sequence Harmony (C) or a combination of both methods (D).

We could also run TABAJARA and simultaneously select regions specific for DENV, ZIKV and YFV, and even use all discrimination methods (Mutual Information, Sequence Harmony and a combination of the methods). In this case, a suggested command is:

```
tabajara.pl -i data/flavivirus.fasta -t 0.45 -w 15 -p 50 -b 15
-m b -md 3 -gs 20 -mb 20 -c DENV,ZIKV,YFV -o all_MI_SH_dir
```

Please notice that, depending on the inter- and intraspecific divergence of each taxonomic group, very discrepant numbers of regions can be found by each method for any particular group. In this case, we recommend running the categories individually, using the most adequate set of parameters for each one.

7 Using profile HMM seeds with GenSeed-HMM

You can use profile HMMs generated by TABAJARA as seeds for progressive assembly using metagenomic datasets and the program GenSeed-HMM (Alves *et al*., 2016). GenSeed-HMM is publicly available at https://sourceforge.net/projects/genseedhmm/. The program is fully documented and tutorial, available on the site, provides information on some public metagenomic datasets of human fecal samples that can be used to reconstruct *Microviridae* phage genomes using profile HMMs and GenSeed-HMM. In case you find that profile HMMs build by TABAJARA are not specific to the chosen taxonomic group, try a higher stringency (e.g. increasing the values of parameters –t, –w and –g).

Table 1 – Coordinates of alignment blocks selected by TABAJARA from a multiple sequence alignment (MSA) of 100 polyprotein sequences of members of the genus *Flavivirus*. with specificity to *Denguevirus*. TABAJARA was run to select regions that are conserved across all *Flavivirus* sequences (C) or specific to *Denguevirus* using Mutual Information (MI), Sequence Harmony (SH) or a combination of both methods (MI+SH).

Method			
C	MI	SH	MI+SH
209-228	27-43	258-277	27-43
344-363	151-168	458-477	151-168
418-437	462-479	1429-1448	258-277
539-558	678-696	1480-1494	458-477
624-643	1327-1346	1590-1606	678-696
712-731	1410-1425	2518-2534	1327-1346
755-774	1481-1500		1416-1435
851-870	1591-1606		1480-1497
902-921	2621-2640		1590-1606
1054-1073			2518-2534
1196-1215			2621-2640
1684-1703			
1781-1800			
1936-1955			
2191-2210			
2247-2262			
2449-2468			
2601-2620			
2932-2951			
3202-3221			
3496-3515			

8 References

Adami C. Information theory in molecular biology. (2004). Phys Life Rev. 1: 3–22.

Alves JM, de Oliveira AL, Sandberg TO, Moreno-Gallego JL, de Toledo MA, de Moura EM, Oliveira LS, Durham AM, Mehnert DU, Zanotto PM, Reyes A & Gruber A. (2016) GenSeed-HMM: A Tool for Progressive Assembly Using Profile HMMs as Seeds and its Application in *Alpavirinae* Viral Discovery from Metagenomic Data. Front. Microbiol. 7: 269.

Capra JA & Singh M (2007). Predicting functionally important residues from sequence conservation. Bioinformatics 23: 1875–1882.

Cover TM & Thomas JA. (2006). Elements of Information Theory. Second edition. John Wiley & Sons, Inc., Hoboken, New Jersey.

Feenstra KA, Pirovano W, Krab K & Heringa J. (2007). Sequence harmony: detecting functional specificity from alignments. Nucleic Acids Res. 35(Web Server issue): W495-8.

Kumar, U., Kumar, V. & Kapur, J. N. (1986). Normalized measures of entropy. Int. J. Gen. Syst. 12: 55-69.

Lin, J. (1991) Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory 37: 145–151.

Pirovano, W., Feenstra, K.A. & Heringa, J. (2006) Sequence comparison by sequence harmony identifies subtype specific functional sites. Nucleic Acids Res. 34: 6540–6548.

Roux S, Krupovic M, Poulet A, Debroas D & Enault F. (2012). Evolution and diversity of the *Microviridae* viral family through a collection of 81 new complete genomes assembled from virome reads. PLoS One 7(7): e40418.

Shannon, C.E. (1948). A Mathematical Theory of Communication. The Bell System Technical Journal 27: 379–423, 623–656.

Shannon, C.E. & Weaver, W. (1949). The Mathematical Theory of Communication. The University of Illinois Press, Urbana.

Shenkin, P.S., Erman, B. & Mastrandrea, L.D. (1991). Information-theoretical entropy as a measure of sequence variability. Proteins 11: 297–313.