SatuRation v1.0 — Manual

Lars Jermiin

10 December, 2022

Correspondence: Systems Biology Ireland, School of Medicine, University College Dublin, Belfield, Dublin 4, Ireland (lars.jermiin@ucd.ie).

Summary

This manual describes how to use three programs developed to (a) compute the strength of the historical signal (λ) in multiple sequence alignments (MSAs) of molecular data and (b) display sets of λ values from MSAs, as shown in Jermiin et al. (2023). Although the aim is to make you familiar with SatuRation and its input and output, the manual also explains the usage of two Perl scripts, SatuRationHeatMapper and RedundancyHeatMapper, which allow you to generate publication-ready heat maps with colour-coded λ values.

Downloads

The three pieces of software are available from:

- `https://www.github.com/lsjermiin/SatuRation.v1.0/`
- `https://www.github.com/ZFMK/SatuRationHeatMapper/`
- `https://www.github.com/ZFMK/RedundancyHeatMapper/`

and are described in more detail below.

SatuRation v1.0

SatuRation is a data-surveying tool. It is written in C++ and released under a GNU General Public License v3.0. It computes δ_{obs}, δ_{ran}, and λ values for all pairs of sequences in an MSA of nucleotides or amino acids (for definitions of δ_{obs}, δ_{ran}, and λ, see Jermiin et al. (2023)), and is executed using two command-line commands:

```
saturation <infile> <a|v> <b|f> <1|...|31>
```

or

```
saturation <infile> <a|v> <b|f> <1|...|31> > README
```

where `infile` refers to a text file with the MSA of characters in the fasta format, `a|v` refers to whether 'all sites' or just 'variant sites' should be used, `b|f` refers to whether a 'brief' or 'full' report of the output is to be printed, and `1|...|31` refers the data type and how the data should be analysed.

If `b` is used, SatuRation prints the summary statistics on a single line in the terminal. If `f` is used, it prints a file with the values of δ_{obs}, δ_{ran}, and λ (in the `.csv` format), two files with the δ_{obs} values (in the `.dis` and `.csv` formats), and a file with the λ values (in the `.csv` format). If `v` is used, SatuRation also prints a file with an alignment of the variant sites. This file may be needed if a user wishes to analyse this sub-MSA. A summary of the results is also printed to the terminal or to the `README` file (doing the latter is useful if multiple alignments are surveyed).

SatuRation is designed to analyse alignments of nucleotides, subsets of codons (e.g., di-nucleotides), codons, 10- and 14-state genotypes, and amino acids. When the `infile` contains sequences of:

- Single nucleotides (4-state alphabet), the sequences may be recoded into six 3-state alphabets or seven 2-state alphabets (recoding of nucleotides may be required as, for example, in Vera-Ruiz et al. (2022),

- Di-nucleotides (16-state alphabet; i.e., $AA, AC, \ldots, TG, TT$), the sequences may be divided into two alignments with 1st and 2nd positions,

- Codons (a 64-state alphabet; i.e., $AAA, AAC, \ldots, TTG, TTT$), the sequences may be divided into three alignments with di-nucleotide sequences and three alignments with single-nucleotide sequences,

- Amino acids (a 20-state alphabet), the letters may be recoded to a 6-state alphabet. This type of recoding was used in a study of early evolution of animals (Feuda et al. 2017). Other types of recoding amino acids have been used (Kosiol et al. 2004; Susko and Roger 2007) but are not considered here.

The 10- and 14-state genotype data cater for diploid and triploid genomes. For example, if a locus in a diploid genome contains the nucleotides A and G, then the genotype sequence will have an R at that locus. There are 10 distinguishable genotypes for each locus in diploid genomes and 14 for each locus in triploid genomes.

For alternative information on data types and how the data may be analysed, simply type:

```
saturation
```

on the command line and follow the instructions.

The output of SatuRation falls into two categories: `.csv` files and `.dis` files. The `_table.csv` file contains the estimates obtained for all pairs of sequences. It can be opened using, for example, Microsoft Excel. The `_dobs.csv`, and `_lambda.csv` files respectively contain the δ_{obs} and λ values, set in a format that can be read by SatuRationHeatMapper and RedundancyHeatMapper (see below). The `.dis` file contains the δ_{obs} values, and can be analysed using FastME (Lefort et al. 2015) and SplitsTree (Huson and Bryant 2006). Lastly, a `_sites_used.fst` file is printed if the `v` option it used; it contains the alignment of the sites used and may be needed in other analyses not mentioned here.

SatuRationHeatMapper v1.0

SatuRationHeatMapper is designed to generate a colour-coded heat map from the `_lambda.csv` file. The colours cover a range from white ($\lambda < 0.64$) to black ($\lambda < 1.0$), with values of $\lambda \geq 1.0$ highlighted in red. SatuRationHeatMapper is written in Perl and can be executed using the following command:

```
SatuRationHeatMapper -i <infile> -<t|f>
```

where infile must be the _lambda.csv file produced by SatuRation, and where t and f stand for 'triangle' and 'full', respectively. The output is an .svg file with a heat map of colour-coded λ values, in the scalable vector graphics format. The file can be opened using Adobe Illustrator.

RedundancyHeatMapper v1.0

RedundancyHeatMapper is designed to generate a colour-coded heat map from the _lambda.csv file. The colours cover a range from white ($\lambda \geq 0.01$) to black ($\lambda \geq 0.37$), with values of $\lambda < 0.01$ highlighted in red. RedundancyHeatMapper is written in Perl and can be executed using the following command:

```
RedundancyHeatMapper -i <infile> -<t|f>
```

where infile must be the _lambda.csv file produced by SatuRation, and where t and f stand for 'triangle' and 'full', respectively. The output is an .svg file with a heat map of colour-coded λ values, again in the scalable vector graphics format. The file can be opened using Adobe Illustrator.

References

Feuda R., Dohrmann M, Pett W, Philippe H, Rota-Stabelli O, Lartillot N, Wörheide G., Pisani D. 2017. Improved modeling of compositional heterogeneity supports sponges as sister to all other animals. Curr. Biol. 27, 3864–3870.

Huson D.H., Bryant, D. 2006. Application of phylogenetic networks in evolutionary studies. Mol. Biol. Evol. 23, 254–267.

Jermiin L.S., Meuseman K., Misof B., Shields D.C. 2023. Quantifying the strength of the historical signal in multiple sequence alignments of phylogenetic data. Syst. Biol. (in prep).

Kosiol C., Goldman N., Buttimore N.H. 2004. A new criterion and method for amino acid classification. J. Theor. Biol. 228, 97–106.

Lefort V., Desper R., Gascuel O. 2015. FastME – A comprehensive, accurate and fast distance-based phylogeny inference program. Mol. Biol. Evol. 32, 2798–800.

Susko E., Roger A.J. 2007. On reduced amino acid alphabets for phylogenetic inference. Mol. Biol. Evol. 24, 2139–2150.

Vera-Ruiz V.A., Robinson J, Jermiin L.S. 2022. A likelihood-ratio test for lumpability of phylogenetic data: Is the Markovian property of an evolutionary process retained in recoded DNA? Syst. Biol. 71, 660–675.