Logo_UHH_rgb.tif

Protocol

**Comparison of Biotite and Entropy.exe for calculating Sequence-Entropies**

Name: Miriam Busch

Matriculation number: 6926928

Degree course: M.Sc. Molecular Life Sciences

Table of Contents

[1. Introduction 1](#_Toc60437189)

[2. Methods 2](#_Toc60437190)

[2.1. Software and input files 2](#_Toc60437191)

[2.2. Analysis procedure 3](#_Toc60437192)

[3. Results 3](#_Toc60437193)

[3.1. Comparison of runtime 3](#_Toc60437194)

[3.2. Profiling of *Biotite* code 4](#_Toc60437195)

[3.3. Comparison of calculated entropies 4](#_Toc60437196)

[3.4. Repairs to *Biotite* 5](#_Toc60437197)

[4. Discussion 5](#_Toc60437198)

[5. References 6](#_Toc60437199)

# 1. Introduction

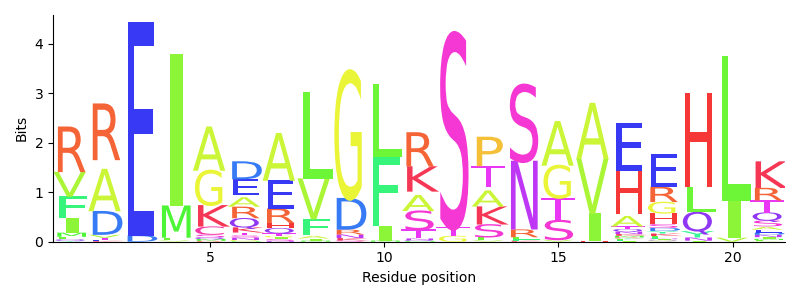
The aim of this labrotation was to compare the entropy package *entropy.exe* (Torda, 2020), which calculates the per-site entropy in a multiple sequence alignment, to the *Python* package called *Biotite* (Kunzmann & Hamacher, 2018). *Biotite* is a large general package written in Python that can be used to handle a major part of the typical workflow for sequence and biomolecular structure data, whereas the purpose-built entropy package is very small and written in Go. In order to compare these two programs, the *Biotite* source code had to be adapted to be left with the same functions as the entropy package.

The calculation of per-site entropy for a multiple sequence alignment allows an assessment of sequence conservation in nucleotide sequences as well as protein sequences (the latter were used for comparison in this labrotation). The entropy for each site is calculated as follows:

is the number of different residues. At each site the summation runs over all amino acid types and gives the fraction of each residue that can be found (Schneider, Stormo & Gold, 1986). Hence the entropy range lies between for (only one type of residue occurs) and the maximal value that results from all residues being equally likely. In this case different residues only show up once and equals (Sen et al., 2019). A point in which *Biotite* and *entropy.exe* differ is the value inserted for . *Biotite* uses to give bits of information, whereas the code of the *entropy.exe* sets to the expected number of residue types to scale values between 0 and 1.

High entropy scores imply a low sequence conservation while low entropy scores indicate a higher conservation. To find high sequence conservation in certain protein regions is of interest since it may reflect on them having essential functions and structural roles. If the sequence was altered in those areas, through mutations for example, it could lead to the protein being non-functional. Identifying conserved sequences can therefore be used for functional annotation and has applications in drug design. Furthermore, sequence conservation can be used to generate phylogenetic trees (Torda, 2020).

For graphical representation of sequence composition and conservation sequence logos are frequently used (see figure 1). At each position all symbols that occur are displayed stacked on top of each other in increasing order of their frequencies. The relative height of the symbols depicts their relative frequency. The height of each stack derives from the maximum possible entropy of the alphabet subtracted by the positional entropy and can therefore be used as an indicator for its conservation (Schneider & Stephens, 1990).



**Figure 1: Example for a sequence logo**  
The sequence logo shows the conservation of LexA DNA-binding site of several gram-positive species. All occurring symbols are shown for each residue position (y-axis). In addition to that the degree of conservation of each symbol is reflected through their height in bits (x-axis).

Sequence logos are preferably used to have a look at specific to look at a certain section of the aligned sequences, e.g. the binding motif in some gram-positive bacteria. But limited data in sequence alignments can result in a systematic underestimation of the entropy. This small sample bias becomes significant if the alignment comprises less than 20 nucleotide or 40 protein sequences (Crooks et al., 2004). Therefore, it is traditional to use a small-number correction in sequence logos which has influences the height of the symbols. The idea of entropy, however, stays similar (Schneider, T et al., 1990).

2. Methods  
  
2.1. Software

Table 1: Overview of used software

|  |  |
| --- | --- |
| **Software** | **Version** |
| Biotite | 0.23.0 |
| entropy.exe | 0.1 |
| Visual Studio Code | 1.52 |
| Visual Studio 2019 | 16.8.3 |
| Microsoft Office Excel | 2016 |

## 2.2. Analysis procedure

The analysis of the entropy calculation with *Biotite* was carried out with the help of *Visual Studio Code (VS code)*. The python source code of the application example “Conservation of *LexA* DNA-binding site” was downloaded from the *Biotite*-website (https://www.biotite-python.org/examples/gallery/index.html; latest access 01.01.2021). This example includes code for entropy calculations and code that is used to generate sequence logos. With the help of the *Debug* feature in *VS Code* the code could be adapted to meet the requirements needed to compare the entropy calculations of *Biotite* to the calculations conducted with *entropy.exe*. Following steps were included in both programs:

1. Reading sequences from FASTA format
2. Calculation of frequencies and entropies
3. Writing results to a csv-file

Both programs treat gaps as missing data, rather than a 21st type of amino acid. In addition to that, the time taken to execute the three steps mentioned above was measured and both codes used the same input files for all comparisons.

One step in the analysis procedure was to compare if the entropies the two programs calculate are similar. On that account a test file was generated with an application called *randseq.exe.* The file included 10 random generated sequences with 100 residues each. The results of the entropy computations were saved into one Excel-file and a diagram showing the entropy per residue was created. Because the code of the entropy package scales values between 0 and 1, whereas *Biotite* gives the values in bits, a secondary y-axis had to be added to compare the numbers despite their value difference. Additionally, the entropy calculations of both programs were executed with a set of biologically relevant sequences. These sequences originate from different *Burkholderia* species and hold the code for a protein called BimC of which the exact function is yet to be determined. The set comprised of 4158 sequences per 441 residues (2,13 MB).

The runtime analysis required test files of different sizes to be read in. They all contained random generated sequences (each 10,000 residues long) and ranged from 500-10,000 sequence counts (4,77 – 95,4 MB). The files were again created with the *randseq.exe* application. The time both programs needed to conclude step 1-3 was determined ten times with each test file and mean values were calculated to create comparable time measures. A diagram showing *Biotite’s* and *entropy.exe´s* runtime depending on sequence count was created in *Microsoft Office Excel*. Moreover, this software was used to perform regression analysis.

In order to assess runtime differences between *Biotite* and *entropy.exe*, profiling was performed with the help of a CPython-based interpreter in *Visual Studio 2019*.

* Something about debugging in VS?

# 3. Results

## 3.1. Comparison of calculated entropies

* **entropy.exe**
* **Biotite**

**Figure 2: Calculated entropies of entropy.exe and Biotite**

## 3.2. Comparison of runtime

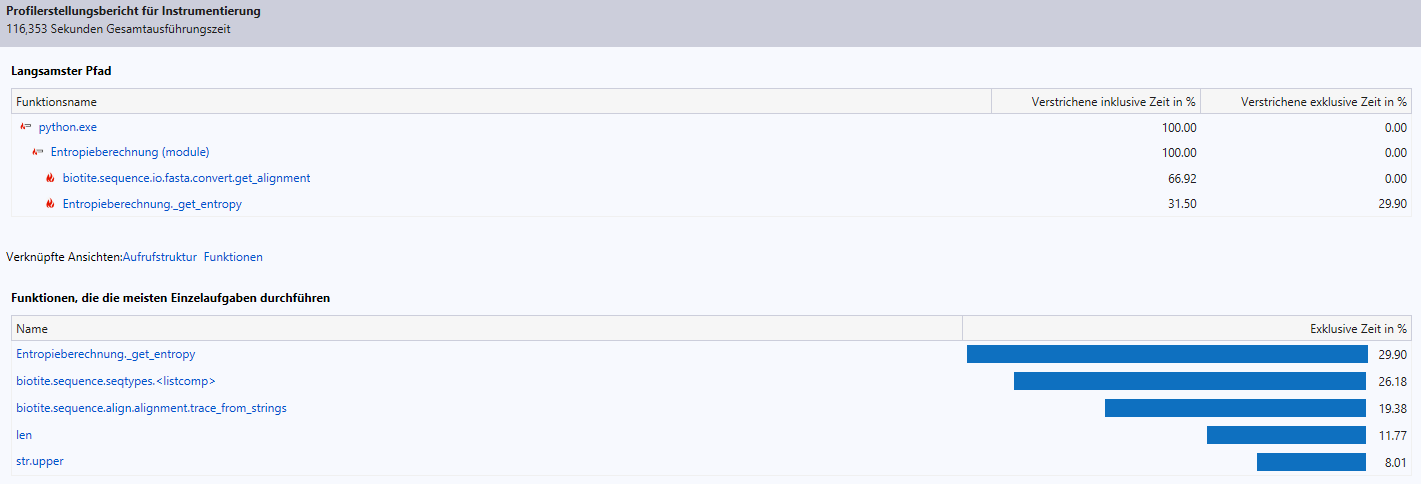
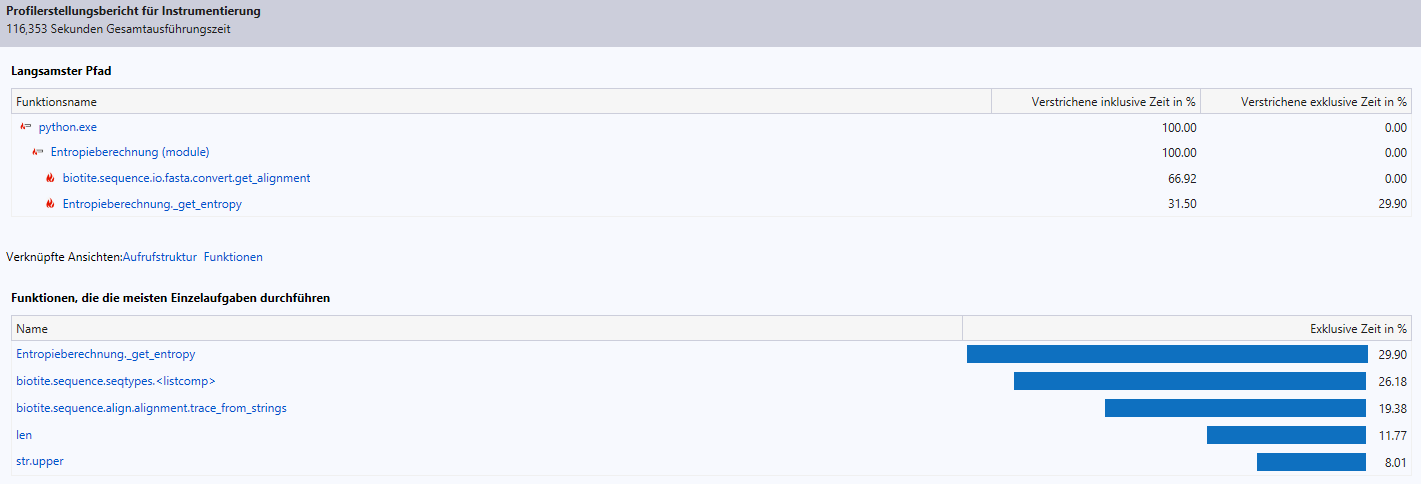
* **entropy.exe**
* **Biotite**

**Figure 4: Runtime of Biotite and entropy.exe depending on sequence count**

y = 0,0207x + 0,2464  
R² = 0,9998

y = 0,0002x + 0,0749  
R² = 1

## 3.3. Profiling of *Biotite* code



**Figure 4: Results of profiling Biotite´s code in Visual Studio 2019**

## 3.4. Repairs to *Biotite*

# 4. Discussion

hgvh

How did you feel about the design of biotite ?  
Error messages.

What should be changed in our entropy code ?  
 \* accommodating csv files for german excel  
 \* other output formats ?  
Would it be helpful to add an option to write a full script for R or gnuplot ?

Why biotite is not bad  
 \* bigger package that tries to do everything (für laien ist es einfacher code anzupassen, als sich ein neues programm zu schreiben bzw. Neue funktionen einzubauen: As a result the user can skip writing code for basic functionality (like file parsers) and can focus on what their code makes unique - from small analysis scripts to entire bioinformatics software packages <https://www.biotite-python.org/index.html>,

On the one hand side, working with Biotite should be computationally efficient, with the help of the powerful packages NumPy and Cython. On the other hand it aims for simple usability and extensibility, so that beginners are not overwhelmed and advanced users can easily build upon the existing system to implement their own algorithms <https://www.biotite-python.org/tutorial/target/index.html#tutorial>)

\* makes plots

# 5. References

Kunzmann, P., & Hamacher, K. (2018). Biotite: a unifying open source computational biology framework in Python. BMC Bioinformatics, 19, 346. https://doi.org/10.1186/s12859-018-2367-z

Schneider, T. D., Stormo, G. D., & Gold, L. (1986). Information content of binding sites on nucleotide sequences. J. Mol. Biol., 188, 415-431. https://doi.org/10.1016/0022-2836(86)90165-8

Torda, A. (2020). seq\_compat. doi:10.5281/zenodo.433

Sen, S., Dey, A., Chowdhury, S. et al. (2019). Understanding the evolutionary trend of intrinsically structural disorders in cancer relevant proteins as probed by Shannon entropy scoring and structure network analysis. BMC Bioinformatics 19, 549. https://doi.org/10.1186/s12859-018-2552-0

Torda, A. (2020). Genome-wide association studies [lecture notes]. University of Hamburg. https://lernen.min.uni-hamburg.de/pluginfile.php/84808/mod\_resource/content/1/4\_gwas\_1-52.pdf

Crooks, G. E., Hon, G., Chandonia, J. M. & Brenner, S. E. (2004). WebLogo: a sequence logo generator. *Genome research*, *14*(6), 1188–1190. https://doi.org/10.1101/gr.849004

Schneider, T. D. & Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. Nucleic acids research, 18(20), 6097–6100. <https://doi.org/10.1093/nar/18.20.6097>

# 6. Appendix

