# Introduction

* The aim was to test the biotite (Kunzmann & Hamacher, 2018) package, focus on one small aspect and compare with local code (Torda, 2020).
* Comparison of a big general package written in python with a small, purpose-built package written in Go.
* Sequence conservation, relationship with sequence logos.  
  The entropy at each site   
  where the summation runs over all amino acid types (Schneider, Stormo, & Gold, 1986).  
  Note that it is traditional to use a small-number correction in sequence logos, but the idea of entropy is similar.  
  The methods differ in that biotite uses to gives bits of information, whereas our code sets to the expected number of amino acid types so as to scale values between 0 and 1.

# Methods

* List version from biotite
* Say our code is version 0.1
* Profiling was done with xxx.
* Sequences were read from fasta format and the same input files were used for both codes in all comparisons.
* Random sequences were generated from local code – what sizes ?
* More interesting sequences (from Alan), with gaps – how big was the data set ?
* Both codes treat gaps as missing data, rather than a 21st type of amino acid.

# Results

* Agreement of calculated values. Difference probably due to biotite using a small numbers correction. No obvious bugs.
* Speed
* Results from profiling
* Repairs to biotite
  + Biotite bug in temporary files, now fixed
  + Better error messages

# Discussion

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

# References

Kunzmann, P., & Hamacher, K. (2018). Biotite: a unifying open source computational biology framework in Python. *BMC Bioinformatics, 19*, 346.

Schneider, T. D., Stormo, G. D., & Gold, L. (1986). Information content of binding sites on nucleeotide sequences. *J. Mol. Biol., 188*, 415-431.

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