The Journal of Open Source Software

Genetic code finder:

Stefanie Mühlhausen[1]

1 Theoretical Computer Science and Algorithmic Methods Group, Institute of Computer Science, University of Göttingen, Goldschmidtstr. 7, 37077 Göttingen, Germany

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

Summary

Yeasts belonging to the so-called CTG-clade and a related clade deviate from the canonical translation of DNA into protein. Instead of translating the CUG codon into standard leucine, they translate CUG as serine, alanine or even stochastically into both leucine and serine. To test any given yeast for its CUG usage, its proteome needs to be sequenced.

MaxQuant is a widely used, quantitative software package for analysing mass spectrometry data [@Cox2008], precise enough to determine single amino acids differences. Here, we present a set of scripts to (i) prepare an unbiased database for MaxQuant to match mass spectra against and to (ii) search and process MaxQuant output for CUG translations.

<todo: mass spectrometry of cell lysate; gene prediction.>

Statement of need

MaxQuant will build a custom database to search spectra against based on FASTA input file. To provide an unbiased database containing CUG codons translated into all amino acids, one should therefore provide a gene prediction with each CUG codon position translated into each amino acid as input. MaxQuant will then digest protein sequences *in silico* and match mass spectra against the resulting peptides. The resulting so-called peptide spectrum matches (PSMs) need to be mapped back onto underlying cDNA sequence to reveal measured CUG translations.

Various private, custom scripts exist to perform afore-mentioned steps. Here, we present ours as open source software. Our scripts are tested, basis for several publications [@Muehlhausen:2018; @Muehlhausen2016] and include useful improvements of both database design and PSM evaluation compared to naïve implementations. The scripts require minimum computational skills to determine CUG translation of a given yeast.

Usage

genetic-code-finder is a set of Ruby scripts than can be invoked from the command line and configured via command line options. The scripts are named in their intended order of execution.

First, MaxQuant's input database is prepared. Compared to the naïve implementation outlined above we found several improvements to be useful in practice: We opt against providing translation into isoleucine, as both isoleucine and leucine are of same mass and thus indistinguishable in mass spectrometry (also, MaxQuant favours otherwise identical peptides containing leucine over those containing isoleucine). Also, we pre-empt *in silico* protein digestion. This allows us to add only those parts of each protein actually containing CUG codons in 19 copies to the database and all other parts only once. If we would add full proteins in 19 copies, we would then end up with 19 identical copies of all those peptides not containing

CUG codons, creating lots of redundancy. Cleavage sites for *in silico* protein digestion, codon of interest and isoleucine treatment can be configured via command line options.

```
ruby 01_create_maxquant_db.rb --input sample_data/Clavispora_cDNA_excerpt.fasta --
```

The next step is to start MaxQuant on the previously generated sequence database. Rather than inspecting MaxQuant output visually, genetic-code-finder programmatically compiles CUG usage values from MaxQuant output tables. As tables may contain millions of rows and take up to several GB of disk space, we chose to pre-format MaxQuant tables `evidence.txt` and `msms.txt`, which contain information about all identified peptides and their MS/MS profile. Peptides in evidence table are merged with the actual peak species found in their fragmentation spectrum (obtained from msms table) and with underlying codon sequence (obtained by mapping peptides back onto the original gene prediction). In addition, we filter out rows belonging to peptides from decoy and reverse database matches. If configured in the first step, the codon under investigation must be specified here, too.

```
ruby 02_combine_maxquant_tables.rb --evidence sample_data/evidence.txt --msms samp
```

Third, the obtained peptide spectrum matches are analysed for codon usage. The script reports measured translation frequencies of CUG codons into all of the tested 19 amino acids. To ensure a CUG codon has indeed been translated *in vivo* as suggested by the matched peptide, we test if peaks have been measured on both sides of the CUG codon allowing to derive the amino acid's mass. This filtering serves as an additional quality measurement on top of MaxQuant's global FDR filtering, allowing to determine *in vivo* CUG usage with high confidence. To complement statistics about CUG usage, we also report overall numbers on both mass spectrometry analysis in general and gene recovery. The latter has proved useful to determine expression patterns of genes containing CUG codons vs those genes not containing any CUG codons.

```
ruby 03_make_statistics.rb --input Clavispora_enriched_evidence.txt --cdna sample_
```

Lastly, this software contains to two R scripts for plotting CUG translation and gene recovery data. Both are easily adjustable to specific needs. Resulting plots for a CTG-clade species might look like section , section .

```
Rscript 04_plot_translation.R Clavispora_statistics.txt [CTG]
Rscript 04_plot_genes.R Clavispora_gene_statistics.csv [CTG]
```

<todo: figures> Caption for example figure. Caption for example figure.

This plots have been obtained by running genetic-code finder on CTG-clade species *Clavispora lusitaniae ATCC 42720* [@Muehlhausen:2018]. The full Clavispora dataset can be found at ProteomeXchange.

Acknowledgements

[todo] Software package wouldn't exist without martin kollmar, who initiated the first project resulting in our own naïve implementation which we've gradually refined in every cug codon analysis since. Also thank xxx (postdoc bei henning urlaub), samir, uwe, henning for help understanding maxquant and building our pipeline.

References