demask Documentation

Release 1.1

Dan Munro

Oct 25, 2021

CONTENTS:

CHAPTER

ONE

INSTALLATION

To install, clone the repository or download and unzip. To get Python dependencies and be able to run or import the modules from any directory, install with pip:

```
pip install -e DeMaSk/
```

Unless you're supplying your own aligned homologs, you'll need to have the blastp program. It can be downloaded as part of BLAST+.

The blastp step requires a formatted sequence database. To use UniRef90, which demask.princeton.edu uses, download the zipped fasta file, unzip, and use makeblastdb from BLAST+ to format the database.

To avoid having to specify the location of the blastp binary and the database for every DeMaSk run, put them in a config file, e.g.:

```
blastp=/usr/local/ncbi/blast/bin/blastp
db=/path/to/database/uniref90.fasta
```

By default, DeMaSk will look for the config file DeMaSk/config.ini, which can also contain any other command line arguments, such as nseqs, threads, and matrix.

CHAPTER TWO

RUNNING FROM THE COMMAND LINE

Once the demask package is installed, you can run run it from anywhere. If you don't have aligned homologs for your query yet, run the demask.homologs module:

```
python3 -m demask.homologs -s myquery.fa -o myquery_homologs.a2m
```

The above command will also produce a file myquery.blast.json containing the intermediate blastp output.

Then, generate fitness impact predictions for all single-residue variants of the query sequence:

```
python3 -m demask.predict -i myquery_homologs.a2m -o myquery_predictions.txt
```

Full options for finding homologs:

```
usage: homologs.py [-h] -s SEQFILE [-c FILE] --blastp BLASTP --db PATH
                  [-t THREADS] [-e EVALUE_CUTOFF] [-b BITSCORE_CUTOFF]
                  [-n NSEQS] (-o OUTFILE | -d OUTDIR)

-h, --help            show this help message and exit
-s SEQFILE, --seqfile SEQFILE
                        Name of a fasta file with one or more query sequences.
-c FILE, --config FILE
                        Configuration file (e.g. the same configuration file
                        used for running DeMaSk). Defaults to 'config.ini' in
                        the demask directory.
--blastp BLASTP       The full path of your blastp executable.
--db PATH             Path of the BLAST database in which supporting
                        sequences will be searched. Must be formatted using
                        the makeblastdb program.
-t THREADS, --threads THREADS
                        Number of threads (CPUs) blastp will use. Defaults to
                        1.
-e EVALUE_CUTOFF, --evalue_cutoff EVALUE_CUTOFF
                        E-value threshold for blastp hits. Defaults to 1e-5.
-b BITSCORE_CUTOFF, --bitscore_cutoff BITSCORE_CUTOFF
                        Bits per query residue threshold for blastp hits.
                        Default is no threshold
-n NSEQS, --nseqs NSEQS
                        Maximum number of top hits to include per query
                        sequence. Defaults to 500.
-o OUTFILE, --outfile OUTFILE
                        Name of output file (only used for single-query runs).
```

(continues on next page)

(continued from previous page)

```
-d OUTDIR, --outdir OUTDIR
                        Name of directory to write output files in a2m (FASTA)
                        format. The sequence titles (after the '>' and before
                        any whitespace) will be used for file names.

For each query sequence, an output file in a2m (FASTA) format will be produced
containing the query sequence and the most similar homologs up to a specified
number (default 500).
```

Full options for getting DeMaSk predictions:

```
usage: predict.py [-h] -i INFILE [-o OUTFILE] [-c FILE] [-m FILE]
                  [--coefs FILE] [-n NSEQS] [-w WEIGHT_THRESHOLD]

-h, --help            show this help message and exit
-i INFILE, --infile INFILE
                        Name of the file containing a sequence alignment in
                        A2M (FASTA) format, with the query protein as the
                        first sequence.
-o OUTFILE, --outfile OUTFILE
                        Name of new file to write scores to.
-c FILE, --config FILE
                        Configuration file. Defaults to 'config.ini' in the
                        demask directory.
-m FILE, --matrix FILE
                        File containing the directional substitution matrix.
                        Defaults to the file at 'DeMaSk/data/matrix.txt'.
--coefs FILE          File containing the intercept, entropy, log2f_var, and
                        matrix coefficients. Defaults to the file at
                        'DeMaSk/data/coefficients.txt'.
-n NSEQS, --nseqs NSEQS
                        Maximum number of supporting sequences per query
                        sequence. Defaults to 500.
-w WEIGHT_THRESHOLD, --weight_threshold WEIGHT_THRESHOLD
                        Sequence identity threshold used for sequence
                        weighting, e.g. 0.8. Sequences are weighted by the
                        inverse of the number of sequences within this percent
                        identity. If None (default), sequence weighting is not
                        used.
```

2.1 Running in Python

Corresponding functions can be run in Python code:

`demask.homologs.`**`find_homologs`**(*seqfile*, *blastp*, *db*, *threads=1*, *evalue_cutoff=1e-05*, *bitscore_cutoff=None*, *nseqs=500*, *outfile=None*, *outdir=None*)

Find protein homologs using blastp.

For each query sequence, an output file in a2m (FASTA) format will be produced containing the query sequence and the most similar homologs up to a specified number (default 500).

Parameters

- **seqfile** (str) – Name of fasta file with query sequence/s.
- **blastp** (str) – The full path of your blastp executable.
- **db** (str) – Path of the BLAST database in which supporting sequences will be searched. Must be formatted using the makeblastdb program.
- **threads** (int) – Number of threads (CPUs) blastp will use. Defaults to 1.
- **evalue_cutoff** (float) – E-value threshold for blastp search. Defaults to 1e-5.
- **bitscore_cutoff** (Optional[float]) – Bitscore (bit_score / query_len) threshold for blastp search. If None (default), no threshold.
- **nseqs** (int) – Number of top hits to include per query sequence.
- **outfile** (Optional[str]) – Name of output file (only used for single-query runs).
- **outdir** (Optional[str]) – Name of directory to write output files. The sequence titles (after the ">" and before any whitespace) will be used for file names.

demask.predict.**run_demask**(*infile*, *matrix*, *coefs*, *nseqs=500*, *weight_threshold=None*)

Predict fitness for all substitutions in a query sequence.

Parameters

- **infile** (str) – Name of the file containing a sequence alignment in A2M (FASTA) format, with the query protein as the first sequence.
- **matrix** (str) – Name of the file containing the directional substitution matrix.
- **coefs** (str) – Name of the file containing the intercept, entropy, log2f_var, and matrix coefficients, with one name and value, separated by a tab, per line.
- **nseqs** (int) – Maximum number of sequences in the alignment to use.
- **weight_threshold** (Optional[float]) – Sequence identity threshold used for sequence weighting, e.g. 0.8. Sequences are weighted by the inverse of the number of sequences within this percent identity. If None (default), sequence weighting is not used.

Return type list

Returns A list of tuples, each containing the position, WT AA, variant AA, score, and the three feature values for each prediction.

CHAPTER

THREE

USER-GENERATED MATRIX AND COEFFICIENTS

DeMaSk comes with a directional substitution matrix computed from a collection of deep mutational scanning datasets, as well as corresponding linear model coefficients. Additional commands are included in case you want to fit the model to a custom matrix, or even calculate a matrix from a custom data collection and then fit the linear model to it.

For example, the default matrix was generated like this:

```
python3 -m demask.matrix -d DeMaSk/data/datasets -o DeMaSk/data/matrix.txt
```

Then, linear model coefficients were calculated:

```
python3 -m demask.fit -d DeMaSk/data/datasets -a DeMaSk/data/alignments -m DeMaSk/data/
↪matrix.txt -o DeMaSk/data/coefficients.txt
```

Full options for computing a matrix:

```
usage: matrix.py [-h] -d DATADIR -o OUTFILE [--columns COLUMNS] [-c FILE]

-h, --help            show this help message and exit
-d DATADIR, --datadir DATADIR
                        Name of directory containing (only) DMS data files,
                        one per protein. If a file name is supplied, only that
                        file will be used.
-o OUTFILE, --outfile OUTFILE
                        Name of file to write matrix to.
--columns COLUMNS     An optional comma-separated list of column names to
                        read in place of 'pos', 'WT', 'var', and 'score',
                        respectively. For example,
                        'Position,wt,mutant,Fitness'. Must apply to all files
                        provided.
-c FILE, --config FILE
                        Configuration file. Defaults to 'config.ini' in the
                        DeMaSk directory.

Loads a collection of DMS datasets and averages scores by WT + variant
identity. Data should already be normalized as desired. For the default DeMaSk
matrix, variant scores were rank-normalized per protein, and then the wild-
type fitness levels subtracted, so that scores represent delta fitness.
Dataset files must be tab-delimited with column names. Columns containing
residue position, WT residue, variant residue, and score must be present in
any order, and other columns will be ignored. Entries for which the WT and
variant residues are not among the 20 canonical amino acids or are the same
are ignored.
```

Full options for calculating coefficients:

```
usage: fit.py [-h] -d DATADIR -a ALNDIR [-m FILE] -o OUTFILE
            [--columns COLUMNS] [-c FILE] [-n NSEQS] [-w WEIGHT_THRESHOLD]

-h, --help            show this help message and exit
-d DATADIR, --datadir DATADIR
                        Name of directory containing (only) DMS data files,
                        one per protein.
-a ALNDIR, --alndir ALNDIR
                        Name of the directory containing alignment files,
                        eachof which must be named as the basename of a DMS
                        data filefollowed by the '.a2m' extension.
-m FILE, --matrix FILE
                        File containing the directional substitution matrix.
                        Defaults to the file at 'demask/matrix/matrix.txt'.
-o OUTFILE, --outfile OUTFILE
                        Name of the file in which to save the coefficients.
--columns COLUMNS     An optional comma-separated list of DMS data column
                        names to read in place of 'pos', 'WT', 'var', and
                        'score', respectively. For example,
                        'Position,wt,mutant,Fitness'. Must apply to all files
                        provided.
-c FILE, --config FILE
                        Configuration file. Defaults to 'config.ini' in the
                        demask directory.
-n NSEQS, --nseqs NSEQS
                        Maximum number of supporting sequences per query
                        sequence. Defaults to 500.
-w WEIGHT_THRESHOLD, --weight_threshold WEIGHT_THRESHOLD
                        Sequence identity threshold used for sequence
                        weighting, e.g. 0.8. Sequences are weighted by the
                        inverse of the number of sequences within this percent
                        identity. If None (default), sequence weighting is not
                        used.

For a set of DMS datasets, load the DMS scores, aligned homologs, and pre-
computed substitution matrix, and fit a linear model, saving the coefficients
to a file for later prediction on new proteins. Dataset files must be tab-
delimited with column names. Columns containing residue position, WT residue,
variant residue, and score must be present in any order, and other columns
will be ignored. Entries for which the WT and variant residues are not among
the 20 canonical amino acids or are the same are ignored. DMS scores should
already be normalized as desired. For the default DeMaSk matrix, variant
scores were rank-normalized per protein, and then the wild-type fitness levels
subtracted, so that scores represent delta fitness.
```

3.1 Running in Python

Corresponding functions can be run in Python code:

`demask.matrix.prepare_matrix`(*datadir*, *outfile*, *columns=['pos', 'WT', 'var', 'score']*)

Compute a directional substitution matrix from DMS data.

Loads a collection of DMS datasets and averages scores by WT + variant identity. Data should already be normalized as desired. For the default DeMaSk matrix, variant scores were rank-normalized per protein, and then the wild-type fitness levels subtracted, so that scores represent delta fitness.

Dataset files must be tab-delimited with column names. Columns containing residue position, WT residue, variant residue, and score must be present in any order, and other columns will be ignored. Entries for which the WT and variant residues are not among the 20 canonical amino acids, or are the same, are ignored.

Parameters

- **datadir** (`str`) – Name of directory containing (only) DMS data files, one per protein. If a file name is supplied, only that file will be used.
- **outfile** (`str`) – Name of the file to which the matrix will be written.
- **columns** (`list`) – An optional list of column names to read in place of ‘pos’, ‘WT’, ‘var’, and ‘score’, respectively. Must apply to all files provided.

`demask.fit.fit_model`(*datadir*, *alndir*, *matrix*, *outfile*, *columns=['pos', 'WT', 'var', 'score']*, *nseqs=500*, *weight_threshold=None*)

Fit the DeMaSk model coefficients.

For a set of DMS datasets, load the DMS scores, aligned homologs, and pre-computed substitution matrix, and fit a linear model, saving the coefficients to a file for later prediction on new proteins.

Dataset files must be tab-delimited with column names. Columns containing residue position, WT residue, variant residue, and score must be present in any order, and other columns will be ignored. Entries for which the WT and variant residues are not among the 20 canonical amino acids or are the same are ignored.

DMS scores should already be normalized as desired. For the default DeMaSk matrix, variant scores were rank-normalized per protein, and then the wild-type fitness levels subtracted, so that scores represent delta fitness.

Parameters

- **datadir** (`str`) – Name of the directory containing (only) DMS data files, one per protein.
- **alndir** (`str`) – Name of the directory containing alignment files, each of which must be named as the basename of a DMS data file followed by the .a2m extension.
- **matrix** (`str`) – Name of the file containing the directional substitution matrix.
- **outfile** (`str`) – Name of the file in which to save the coefficients.
- **columns** (`list`) – An optional list of DMS data column names to read in place of ‘pos’, ‘WT’, ‘var’, and ‘score’, respectively. Must apply to all files provided.
- **nseqs** (`int`) – Maximum number of sequences in the alignment to use.
- **weight_threshold** (`Optional[float]`) – Sequence identity threshold used for sequence weighting, e.g. 0.8. Sequences are weighted by the inverse of the number of sequences within this percent identity. If None (default), sequence weighting is not used.

CHAPTER

FOUR

FULL DOCUMENTATION

You won't need to use most of these functions directly, but they are described here for custom use and curiosity.

4.1 matrix.py

Code for computing a substitution matrix from DMS data. The matrix can be used for variant impact prediction in place of the provided matrix.

`demask.matrix.`**`get_subs_matrix`**`(`*data*`)`

Compute a substitution matrix from DMS data.

Values will be averaged by WT + variant identity. Diagonal (synonymous) matrix elements will be set to 0 since the matrix represents expected substitution impact.

Parameters **`data`** (`list`) – The output of `load_dataset()`: A list of lists, each containing the residue position, WT residue, variant residue, and score for a DMS measurement. Position information is not used.

Return type `dict`

Returns A dictionary in which keys are (AA1, AA2) tuples.

`demask.matrix.`**`prepare_matrix`**`(`*datadir*, *outfile*, *columns=['pos', 'WT', 'var', 'score']*`)`

Compute a directional substitution matrix from DMS data.

Loads a collection of DMS datasets and averages scores by WT + variant identity. Data should already be normalized as desired. For the default DeMaSk matrix, variant scores were rank-normalized per protein, and then the wild-type fitness levels subtracted, so that scores represent delta fitness.

Dataset files must be tab-delimited with column names. Columns containing residue position, WT residue, variant residue, and score must be present in any order, and other columns will be ignored. Entries for which the WT and variant residues are not among the 20 canonical amino acids, or are the same, are ignored.

Parameters

- **`datadir`** (`str`) – Name of directory containing (only) DMS data files, one per protein. If a file name is supplied, only that file will be used.
- **`outfile`** (`str`) – Name of the file to which the matrix will be written.
- **`columns`** (`list`) – An optional list of column names to read in place of 'pos', 'WT', 'var', and 'score', respectively. Must apply to all files provided.

`demask.matrix.`**`write_matrix`**`(`*matrix*, *fname*`)`

Write a substitution matrix to file.

Parameters

- **matrix** (dict) – A dictionary in which keys are (AA1, AA2) tuples, as provided by `get_subs_matrix()`.
- **fname** (str) – The file name to write to.

4.2 homologs.py

Functions for generating homolog alignments.

`demask.homologs.`**`find_homologs`**(*seqfile*, *blastp*, *db*, *threads=1*, *evalue_cutoff=1e-05*, *bitscore_cutoff=None*, *nseqs=500*, *outfile=None*, *outdir=None*)

Find protein homologs using blastp.

For each query sequence, an output file in a2m (FASTA) format will be produced containing the query sequence and the most similar homologs up to a specified number (default 500).

Parameters

- **seqfile** (str) – Name of fasta file with query sequence/s.
- **blastp** (str) – The full path of your blastp executable.
- **db** (str) – Path of the BLAST database in which supporting sequences will be searched. Must be formatted using the makeblastdb program.
- **threads** (int) – Number of threads (CPUs) blastp will use. Defaults to 1.
- **evalue_cutoff** (float) – E-value threshold for blastp search. Defaults to 1e-5.
- **bitscore_cutoff** (Optional[float]) – Bitscore (bit_score / query_len) threshold for blastp search. If None (default), no threshold.
- **nseqs** (int) – Number of top hits to include per query sequence.
- **outfile** (Optional[str]) – Name of output file (only used for single-query runs).
- **outdir** (Optional[str]) – Name of directory to write output files. The sequence titles (after the ">" and before any whitespace) will be used for file names.

`demask.homologs.`**`get_aligned_AAs`**(*infile*, *bitscore_cutoff=None*)

Extract aligned amino acids from BLAST output.

Parameters

- **infile** (str) – Name of blastp output file in json format (made using blastp argument '-outfmt 15').
- **bitscore_cutoff** (Optional[float]) – Bitscore (bit_score / query_len) threshold for blastp search. If None (default), no threshold.

Return type dict

Returns A dictionary in which the keys are the query sequence names and the values are lists of aligned hits. Each hit in the list is a dictionary with the hit's 'id' and 'seq'.

`demask.homologs.`**`passes_filters`**(*homolog*, *query*)

Check whether a homolog has >= 20% identity with query and whether the alignment is < 10% gaps.

Parameters

- **homolog** (str) – The aligned homolog sequence. It should be the same length as the query sequence, including '.' for end gaps and '-' for internal gaps.
- **query** (str) – The query sequence.

Return type `bool`

Returns True if both conditions are met, otherwise false.

`demask.homologs.`**`process_query`**`(`*query*, *bitscore_cutoff=None*`)`

Extract alignment of hits for one query from BLAST output.

Parameters

- **`query`** (`dict`) – A dictionary obtained by parsing JSON-formatted BLAST output into a nested Python structure and selecting one element of the 'BlastOutput2' list.
- **`bitscore_cutoff`** (`Optional[float]`) – Bitscore (bit_score / query_len) threshold for blastp search. If None (default), no threshold.

Return type `list`

Returns A list of dictionaries, each containing a hit's 'id' and 'seq'.

`demask.homologs.`**`run_blastp`**`(`*seqfile*, *blastp*, *db*, *evalue*, *nseqs*, *outfile*, *threads=1*`)`

Execute blastp.

Executes blastp with the given parameterss and writes output in json format (using flag '-outfmt 15'). Throws an exception if the blastp run fails.

Parameters

- **`seqfile`** (`str`) – Name of fasta file with query sequence/s.
- **`blastp`** (`str`) – The full path of your blastp executable.
- **`db`** (`str`) – Path of the BLAST database in which supporting sequences will be searched. Must be formatted using the makeblastdb program.
- **`nseqs`** (`int`) – Maximum top hits to include per query sequence.
- **`outfile`** (`str`) – Name of results file.
- **`threads`** (`int`) – Number of threads (CPUs) to use. Defaults to 1. If 0, it will use a remote database, so db should be the name of an NCBI protein database, e.g. 'nr'.

4.3 profiles.py

Functions for computing sequence profiles from homologs, i.e. proportions of each amino acid at each position.

`demask.profiles.`**`get_aligned_profile`**`(`*alns*, *weights=None*, *pseudocount=0.0001*`)`

Create a sequence profile from alignments and optional weights.

Weights cannot be negative but need not sum to one per position, since the profiles will be renormalized per position. All characters other than the 20 canonical amino acids will be ignored, so the 20 fractions will always sum to one.

Parameters

- **`alns`** (`ndarray`) – An array with dimensions (query length, num. homologs) containing query + aligned AAs and gaps.
- **`weights`** (`Optional[ndarray]`) – An optional array with same dimensions as *alns* containing corresponding weights.
- **`pseudocount`** – A small value added to all frequencies before normalizing to proportions, allowing the proportions to later be log-transformed.

Return type `ndarray`

Returns An array with dimensions (query length, 20) containing the proportions of AAs per position.

`demask.profiles.`**`get_weights`**`(`*alns*, *ident_cutoff=0.8*`)`

Get weights for aligned AAs.

Weights are produced for individual aligned AAs, but currently all weights for a given aligned sequence are the same. The weight for a sequence is proportional to the inverse of the number of sequences (including itself) above a percent identity cutoff. Similar to clustering, this normalizes for uneven phylogenetic depths in the database.

Parameters

- **`alns`** (`ndarray`) – An array with dimensions (query length, nseqs + 1) containing query + aligned AAs and gaps.
- **`ident_cutoff`** (`float`) – Sequences will be weighted relative to the inverse of the number of sequences (including itself) with at least this percent identity.

Return type `ndarray`

Returns An array with same dimensions as *alns* containing corresponding weights.

4.4 fit.py

Functions for fitting a linear model to relate entropy, variant frequency, and a substitution matrix to DMS scores.

`demask.fit.`**`coefficients`**`(`*dms*, *entropy*, *log_profile*, *matrix*`)`

Compute coefficients for the DeMaSk linear model.

Parameters

- **`dms`** (`dict`) – A dictionary in which keys are dataset names and values are DMS data as returned by `load_dataset()`.
- **`entropy`** (`dict`) – A dictionary in which keys are dataset names and values are iterables with entropy for each protein position.
- **`log_profile`** (`dict`) – A dictionary in which keys are dataset names and values are the sequence profile 2D arrays returned by `get_aligned_profile()`.
- **`matrix`** (`dict`) – A dictionary in which keys are (AA1, AA2) tuples.

Return type `dict`

Returns A dictionary with keys “intercept”, “entropy”, “log2f_var”, and “matrix”, and corresponding coefficients as values.

`demask.fit.`**`fit_model`**`(`*datadir*, *alndir*, *matrix*, *outfile*, *columns=['pos', 'WT', 'var', 'score']*, *nseqs=500*, *weight_threshold=None*`)`

Fit the DeMaSk model coefficients.

For a set of DMS datasets, load the DMS scores, aligned homologs, and pre-computed substitution matrix, and fit a linear model, saving the coefficients to a file for later prediction on new proteins.

Dataset files must be tab-delimited with column names. Columns containing residue position, WT residue, variant residue, and score must be present in any order, and other columns will be ignored. Entries for which the WT and variant residues are not among the 20 canonical amino acids or are the same are ignored.

DMS scores should already be normalized as desired. For the default DeMaSk matrix, variant scores were rank-normalized per protein, and then the wild-type fitness levels subtracted, so that scores represent delta fitness.

Parameters

- **`datadir`** (`str`) – Name of the directory containing (only) DMS data files, one per protein.

- **alndir** (str) – Name of the directory containing alignment files, each of which must be named as the basename of a DMS data file followed by the .a2m extension.
- **matrix** (str) – Name of the file containing the directional substitution matrix.
- **outfile** (str) – Name of the file in which to save the coefficients.
- **columns** (list) – An optional list of DMS data column names to read in place of 'pos', 'WT', 'var', and 'score', respectively. Must apply to all files provided.
- **nseqs** (int) – Maximum number of sequences in the alignment to use.
- **weight_threshold** (Optional[float]) – Sequence identity threshold used for sequence weighting, e.g. 0.8. Sequences are weighted by the inverse of the number of sequences within this percent identity. If None (default), sequence weighting is not used.

4.5 predict.py

Functions for generating variant fitness predictions for a query sequence.

demask.predict.**compute_scores**(*wt*, *entropy*, *log_profile*, *matrix*, *coefs*)

Compute fitness predictions for all possible substitutions.

Parameters

- **wt** (str) – The wild-type protein sequence as a string, list, or array of characters.
- **entropy** (list) – A list the length of the query sequence containing per-position entropy from the homolog sequence profile.
- **log_profile** (ndarray) – A sequence profile 2D array as returned by get_aligned_profile(), but subsequently log2-transformed.
- **matrix** (dict) – A dictionary in which keys are (AA1, AA2) tuples.
- **coefs** (dict) – A dictionary with keys "intercept", "entropy", "log2f_var", and "matrix", and corresponding coefficients as values.

Return type list

Returns A list of tuples, each containing the position, WT AA, variant AA, score, and the three feature values for each prediction.

demask.predict.**demask_score**(*WT*, *var*, *entropy*, *logf_var*, *matrix*, *coefs*)

Calculate the fitness scores for a list of substitutions.

Parameters

- **WT** (str) – The wild-type amino acid.
- **var** (str) – The variant amino acid.
- **entropy** (float) – The Shannon entropy for *WT*'s position in the homolog profile.
- **logf_var** (float) – The log2-transformed variant AA proportion.
- **matrix** (dict) – A dictionary in which keys are (AA1, AA2) tuples.
- **coefs** (dict) – A dictionary of coefficients with keys "intercept", "entropy", "log2f_var", and "matrix".

Return type float

Returns The variant's predicted delta fitness.

`demask.predict.`**`read_coefficients`**`(`*fname*`)`

Read DeMaSk model coefficients from file.

Parameters **`fname`** (`str`) – File name.

Return type `dict`

Returns A dictionary with keys "intercept", "entropy", "log2f_var", and "matrix", and corresponding coefficients as values.

`demask.predict.`**`run_demask`**`(`*infile*, *matrix*, *coefs*, *nseqs=500*, *weight_threshold=None*`)`

Predict fitness for all substitutions in a query sequence.

Parameters

- **`infile`** (`str`) – Name of the file containing a sequence alignment in A2M (FASTA) format, with the query protein as the first sequence.
- **`matrix`** (`str`) – Name of the file containing the directional substitution matrix.
- **`coefs`** (`str`) – Name of the file containing the intercept, entropy, log2f_var, and matrix coefficients, with one name and value, separated by a tab, per line.
- **`nseqs`** (`int`) – Maximum number of sequences in the alignment to use.
- **`weight_threshold`** (`Optional[float]`) – Sequence identity threshold used for sequence weighting, e.g. 0.8. Sequences are weighted by the inverse of the number of sequences within this percent identity. If None (default), sequence weighting is not used.

Return type `list`

Returns A list of tuples, each containing the position, WT AA, variant AA, score, and the three feature values for each prediction.

4.6 utils.py

Utility functions for the demask package.

`demask.utils.`**`entropies`**`(`*profile*`)`

Calculate Shannon entropy (log base 2) for each position of a sequence profile.

Parameters **`profile`** (`ndarray`) – A 2D array of AA proportions as returned by `get_aligned_profile()`.

Return type `list`

Returns A list of entropy values corresponding to the rows of `profile`.

`demask.utils.`**`extract_DMS_WT`**`(`*datadir*, *output*`)`

Infer query sequence/s based on substitutions listed in DMS data.

Amino acids specified by the 'pos' and 'WT' columns will be assembled, and any missing positions will be represented as 'X' in the sequence. Sequences will be written to a fasta file.

This method is not used by DeMaSk, but is provided for convenience if the WT sequence for a DMS dataset is missing.

Parameters

- **`datadir`** (`str`) – Name of the directory containing (only) DMS data files, one per protein. If a file name is supplied, only that file will be used.
- **`output`** (`str`) – Name of the output (fasta) file.

demask.utils.**get_filenames**(*location*)

Get file names from user-specified directory.

If the input is actually the name of an existing file, only that file name will be returned.

Parameters **location** (str) – The name of the directory or file.

Return type list

Returns A list of file names.

demask.utils.**index**(*x*)

Get array of indices for True elements in 1D array.

Return type ndarray

demask.utils.**load_dataset**(*fname*, *cols=['pos', 'WT', 'var', 'score']*)

Load a DMS dataset from a file.

File must be tab-delimited with column names. Columns containing residue position, WT residue, variant residue, and score must be present in any order, and other columns will be ignored. Entries for which the WT and variant residues are not among the 20 canonical amino acids or are the same are removed.

Parameters

- **fname** (str) – Name of the file.
- **cols** (list) – An optional list of column names to read in place of 'pos', 'WT', 'var', and 'score', respectively.

Return type list

Returns A list of lists, each containing the residue position, WT residue, variant residue, and score for a data point.

demask.utils.**read_fasta**(*filename*, *as_dict=True*)

Read one or more sequences from a fasta file.

Names are extracted from the header line after '>' and before any whitespace.

Parameters

- **filename** (str) – Name of the fasta file to read.
- **as_dict** (bool) – If set to false, will return a list of sequences.

Return type dict

Returns A dictionary with names as keys and sequence strings as values, or a list of sequences.

demask.utils.**read_matrix**(*fname*)

Read a substitution matrix from file.

The file must be tab-delimited with column names in the first line. Row names should not be provided and will be copied from the column names. If the matrix is not symmetric, rows should correspond to AA1 and columns to AA2.

Parameters **fname** (str) – The file name.

Return type dict

Returns A dictionary in which keys are (AA1, AA2) tuples.

demask.utils.**seq_matrix**(*seqs*)

Get a residue matrix from a list of sequences.

Returns a numpy array of characters with dimensions (seq length, nseqs). The rows correspond to letters in the first (query) sequence, so positions in which the first sequence has a gap ('-' or '.') are removed. Lowercase (masked) residues are capitalized so that they are counted as valid AAs.

Return type `ndarray`

`demask.utils.`**`write_fasta`**(*seqs*, *filename*)

Write one or more sequences to a fasta file.

Parameters

- **`seqs`** (`list`) – A list of dictionaries, each containing a sequence's 'id' and 'seq'.
- **`filename`** (`str`) – Name of fasta file to write to.

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

INDEX