Evolutionary Pathways

The shortest pathway between two sequences, along a fitness landscape, can be found efficiently using the A* algorithm. The algorithm iterates over a single round's sequence population as a graph, with each sequence present as its own node and the edit distances between sequences as edges with distance. The python script `peak_pather_v01.py` finds the *N* best pathways between two sequences, ranked by: 1) shortest total path length, 2) smallest maximum step size, 3) smallest average step size, 4) largest minimum sequence count, using a sequence counts file as the reference map. Among equal-length pathways, the ones with the highest number of total steps are prioritized.

How to use the script to calculate evolutionary pathways:

The pathways described in this publication are computed as follows:

```
python peak_pather_v01.py input_file output_file start_sequence end_sequence --min_count [variable] --max_step [variable] -n [variable] --max_length [variable] -p
```

where `input_file` is R5c-counts.txt. The code requires a "counts" file as input, consisting of three lines of metadata followed by one line per unique sequence in the pool in the following format: sequences in the first column and counts (an integer number) in the second column. Such files are produced by our Galaxy tools, currently available at the Chen Lab website. The file corresponding to the round used to produce the pathways reported in the publication (R5c-counts.txt) can be found in this repository (in Count reads files for original selection).

Minimum sequence count (`min_count`) was set at 3 for the highly-populated pathways between Motif 1A and 1B, and set at 2 for all other pathways. For all pairs of sequence endpoints investigated, maximum step size (`max_step`) was set at 1, then incremented by 1 until 5 pathways were found. The script was then run again with min_step increased 1 further, to generate 5 additional pathways with larger step tolerance.

The number of best pathways to be generated with each run (`n`) was set to 5. The maximum length of any path searched before the path is abandoned (`max_length`) was set to 35.

For more information on usage, read below or run `python peak_pather_v01.py -h` in the terminal.

Advance usage information

Goal:

The peak_pather script searches for the shortest pathway between two sequences, along a fitness landscape, using the A* algorithm. The algorithm iterates over a single round's sequence population as a graph, with each sequence present as its own node and the edit distances between sequences as edges with distance. The python script `peak_pather_v01.py`finds the *N* best pathways between two sequences, ranked by 1) shortest total path length, 2) smallest maximum step size, 3) smallest average step size, 4) largest minimum sequence count, using a sequence counts file as the reference map (this could be replaced with highest minimum fitness, if the reference file is a list of sequences and their fitnesses instead).

Input:

The script can be generally used as follows:

```
python peak_pather_v01.py input_file output_file start_seq end_seq
```

Required arguments (positionally dependent):

`input` - File containing sequence counts for the round to be iterated over (e.g. `R5c-counts.txt`). The code requires a "counts" file consisting of three lines of metadata followed by one line per unique sequence in pool, in the following format: sequences in the first column and counts (an integer number) in the second column. Such files are produced by our Galaxy tools, currently available at the Chen Lab website.

`output` - Output file (e.g. `paths-output.txt`)

`start_seq` - Path start sequence (e.g. `'CTACTTCAAACAATCGGTCTG'`)

`end_seq` - Path end sequence (e.g. `'CCACACTTCAAGCAATCGGTC'`)

Optional arguments:

The following options can be utilized to adjust the behaviour of the script. Most options require an additional argument. A comma indicates that an option can be called multiple ways.

`-h`, `--help` - Show help message and exit

`-n NUM_PATHS`, `--num_paths NUM_PATHS` - Number of best pathways to be generated with each run (must be an integer). Default value is 1.

`--max_length MAX_LENGTH` - Maximum length of any path searched before the path is abandoned (must be an integer). Default value is twice the length of the starting sequence.

`--min_count MIN_COUNT` - Minimum count of sequences searched (must be an integer). Default value is 2.

`--max_step MAX_STEP` - Maximum step size allowed by search paths (must be an integer). Default value is 1.

`-d DIST_TYPE`, `--dist_type DIST_TYPE` - Distance metric used to find shortest pathway: `edit` distance or `hamming`distance. Default option is edit.

`-p`, `--track_progress` - If this flag is used, progress will be printed in the terminal.

Output:

The output file from this script contains a list of the `NUM_PATHS` top pathways. Each is provided as a list of comma-separated values, with the data on each column (denoted by a header) corresponding to:

`step #` - The number of steps from the sequence in this column to the start sequence.

`sequence` - The sequence of the node at this step along the pathway.

`step size` - The distance between this sequence along the pathway and the previous sequence.

`total distance` - The total distance of all steps up to this one.

`sequence count` - The count of this particular sequence in the counts file. May correspond to a different value, e.g. if the input file is a list of sequences and their fitness value instead of sequencing count.