

Klink - clustering of ESTs with k-links

Lauren Bragg

March 27, 2009

Contents

1 License	1
2 Preamble	1
3 Purpose	2
4 Compilation	2
5 Usage	3
6 Output	4
7 Contact	4
8 References	4

1 License

Klink is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Klink is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

For more details, please see <http://www.gnu.org/licenses/>.

2 Preamble

The *Klink* program was implemented to illustrate issues with different levels of clustering linkage in ESTs. This is but one way to implement the K-Link algorithm. I anticipate that improvements can be made in terms of speed and memory usage. Please feel free to improve the software.

3 Purpose

The *Klink* program uses the K-Link clustering algorithm to cluster ESTs. As input it takes a file in *Klink* seed format (in some respects, similar to the output format of *wcd* [Hazelhurst *et al.*, 2008]). Each EST is represented by an integer id. ESTs are numbered from 0 to N. Each line of the input describes one cluster seed for the K-Link algorithm. The format is strictly

```
[EST SEED ID] : [MEMBER1] [MEMBER2] [MEMBER3] .
```

For example, the following may be the beginning of a *Klink* cluster seed file.

```
0:0 1 2.  
1:0 1.  
2:0 2.
```

where the members must be ordered from smallest to largest. Note that even though the seed sequence itself is implicitly a member of its own cluster, it must be included in the member list for the cluster.

In our study (Bragg and Stone, 2009, submitted to Bioinformatics), sequences were given integer identifiers using a Perl script, and were then compared in an all-against-all BLAST. The alignment results were filtered using a similarity threshold (in our case, 40 bases @ 90% identity), and seeds were prepared in the format described above.

4 Compilation

I have only compiled the *Klink* program on a 64-bit SUSE linux server. I can't see any reason why it wouldn't compile on other linux distributions. I cannot guarantee that it compiles on Windows or Mac. Please let me know if you have difficulty getting it to compile.

The *Klink* distribution requires the *thread* static library, and the `lexical_cast` library from Boost (<http://www.boost.org/>), as well as libraries from the GNU General Scientific Library (GSL) (<http://www.gnu.org/software/gsl/>). I have only tested *Klink* compiles using Boost version 1.36.0.

To use the provided Makefile, it requires that the BOOST environmental variable be set to the location of the directory containing the `/lib` and `/include` directories of the Boost distribution.

This can be done in BASH by:

```
BOOST=/path/to/boost  
source BOOST
```

If the location of the GSL installation is not in the include path, you will need to modify the Makefile. You will also need to customise the Makefile for compilation on other platforms.

5 Usage

There are a number of parameters for the K-link clustering algorithm. The only compulsory parameters are the cluster seed file and the number of threads to use.

Command line options:

Usage: `-f <file> -p <number of threads> [options]`

Options:

- `-f file`: file containing clusters in Klink format. See documentation.
- `-p int`: number of threads to use
- `-i int`: <optional> max number of iterations. Default is 5
- `-k int`: <optional> minimum number of links to use in initial loose clustering for probability calculation <optional> default is 6
- `-c double`: <optional> frequency of chimeras in the dataset. Default is 0.01
- `-P double`: <optional> probability of at least one chimeric merge (cutoff is defaulted to 0.01)
- `-m int`: <optional> the maximum number of co-occurrences to calculate up to while identifying optimal k (default 5)
- `-n int`: <optional> the minimum number of co-occurrences to start calculating from when identifying the optimal k (default 1)
- `-E`: <optional> Do NOT estimate the links parameter. Use the links parameter specified by `-k` (default of 6)

The parameter estimation for the number of links (k) is generally the slowest part of the program. Our original implementation was much slower, as it calculated the probability of seeing l or less chimeras. The current implementation calculates a mathematical interval for this probability, and while it is not necessarily the smallest interval to contain the probability, it contains it 100% of the time. Taking the conservative approach, we compare the probability cutoff parameter to the upper bound of this interval.

Make sure that if you do want to use the parameter estimation, you set the number of links (`-k`) (for the rough clustering) to a value at least as large as you expect the optimal k to be. The parameter estimation can be avoided if you have a good idea of the number of links to use. Please refer to the paper for further information on how to choose the number of links.

The probability calculation will be computed for varying levels of co-incidence (ie. the maximum number of times we expect to see any chimera in particular). Under the default settings, this starts from the value specified by `-n`. Under the default settings, the probability calculations would begin with $= 1$ (this is the probability of seeing a chimera once in the dataset). If this probability is larger than the cut-off probability (`-P`), the software will proceed to calculate the co-occurrence for 2 chimeras (probability of seeing the same chimera twice). This continues (incrementing by one each time) until the probability calculated is smaller than the chimeric merge cut-off or will halt after the maximum co-occurrence rate has been reached. The program terminates if the probability of

a chimeric merge at the maximum co-occurrence level (-m, default of 5) is still larger than the chimeric merge cutoff (default 0.01).

If the probability of a co-occurrence at a given level falls below the cut-off, the number of links (k) used during clustering will be the co-occurrence level (l) + 1. For instance, say the calculated maximum probability of seeing one chimera is 0.5, and the max probability of seeing a particular chimera twice is 0.005 (given a 0.01 probability cutoff), then k , the number of links used, is $2 + 1 = 3$ links.

As described in the paper, a rough clustering is obtained by setting an arbitrarily high k (ie. between 6 and 10 - larger than the optimal k for the dataset) and flagging no estimation of the links parameter (-E). If unexpectedly large clusters are forming in the data, this may be an indication of inadequate sequence masking.

6 Output

Klink produces three output files, *.clusters, *.removed_clusters and *.chimeras (where * is the name of the supplied cluster seed file). The *.clusters file contains the resulting clusters in wcd output format.

The *.removed_clusters file contains those clusters which were removed in the first iteration (in wcd output format). It is possible for some sequences to be removed from the resulting clustering altogether after the first round. This is rare, however it is most likely to occur when the sequence belongs only to clusters which are less than the number of links used.

The *.chimeras file contains all the sequences (or id's of sequences, to be precise) that belong to multiple clusters after the last round of clustering (ie. the remaining putative chimeras).

7 Contact

The *Klink* software was written by Lauren Bragg and Glenn Stone, of CSIRO Mathematical and Information Sciences. We can be contacted at lauren.bragg@csiro.au, glenn.stone@csiro.au.

8 References

Hazelhurst, S., Hide, W., Liptak, Z., Nogueira, R. and Starfield, R. An overview of the wcd EST clustering tool. *Bioinformatics*. **24(13)**, July, pp. 1542-1546. doi: 10.1093/bioinformatics/btn203