

BAG: A Graph Theoretic Sequence Clustering Algorithm Manual

Sun Kim*

School of Informatics
Center for Genomics and Bioinformatics
Indiana University – Bloomington
sunkim@bio.informatics.indiana.edu

Contents

1	Introduction	2
2	Installing	2
3	How to use at a glance.	2
3.1	Output files	2
3.1.1	The format of <code>prefix.cluster.result</code> file	3
3.2	The file format for pairwise matches	4
3.3	Setting the cutoff score	4
4	Splitting and Merging Clusters	5
4.1	Splitting clusters with options <code>-O</code> and <code>-L</code>	5
4.2	Merging clusters	5
4.3	Issues in splitting and merging	6
5	More on BAG algorithm	6

*Partially supported by INGEN (Indiana Genomic Initiatives), IBM SUR grant, and NSF Career Award 0237901.

1 Introduction

This is for a brief manual for a sequence clustering algorithm BAG based on graph theory. Our algorithm clusters sequences using two properties of graph, biconnected component and articulation point. For more detail, see [2] or visit <http://bio.informatics.indiana.edu/sunkim/BAG>.

2 Installing

The BAG implementation uses LEDA [?] that can be obtained from <http://www.algorithmic-solutions.com>. As of now, BAG has been tested with LEDA 4.3 on Linux platform (RedHat 7.1) only.

1. Uncompress and untar the tile `bag.tar.Z`.
`$ uncompress bag.tar.Z; tar xvf bag.tar`
2. There are seven files: `compute.cutoffs.c`, `bag.c`, `Makefile`, `Make.pro`, `Make.lst`, `bag.manual.pdf`, and `merge.ecoli.h.cmp` that is a pairwise comparison file for *E. coli* and *H. influenzae*.
3. Set an environment variables `LEDAROOT` and `LD_LIBRARY_PATH` where the LEDA package is installed. For example, assuming `/home/leda/LEDA4.3` using a bash on Linux,
`export LEDAROOT=/home/leda/LEDA4.3`
4. Make two binaries, `compute.cutoffs` and `bag`.
`$ make compute.cutoffs`
`$ make bag`
5. Run `compute.cutoffs` and `bag`

3 How to use at a glance.

1. If you need to set a cutoff score, running
`$ compute.cutoffs pairwise-file -L minscore -M maxscore -d delta`
will print a series of clustering analysis using `pairwise-file` from `minscore` to `maxscore` with an interval `delta`. Setting the value of `cutoff` is explained in Section 3.3 and the format of `pairwise-file` is explained in Section 3.2. If you already decided a value for the cutoff score for `bag` clustering analysis, you can skip this step.
2. Once you set a value for `cutoff`, running
`$ bag pairwise-file cutoff -P prefix`
will produce five files with the `prefix`, `prefix.cluster.result`, `prefix.cluster.history`, `prefix.cluster.ranges`, `prefix.cluster.merge`, and `prefix.cluster.multidomain`

3.1 Output files

There are five files from running `bag`. This section explains the content of each file.

1. The file `prefix.cluster.result` lists clusters and their member sequence identifiers. See the next subsection for details of the file format.
2. The file `prefix.cluster.history` lists clusters that are split into multiple. The cluster that split are not considered as final clusters.
3. The file `prefix.cluster.ranges` lists ranges where all member sequences in a cluster share.
4. The file `prefix.cluster.merge` *suggests* which final clusters could be merged into one. This is only suggestions and is *not* performed automatically due to some technical difficulties (see Section 4.3).
5. The file `prefix.cluster.multidomain` lists a set of possible multidomain sequences. These sequences do *not* share common regions longer than the value that can be specified by `-a` option.

3.1.1 The format of `prefix.cluster.result` file

A part of the clustering analysis of predicted proteins from *E. coli* and *H. influenzae* are shown below. A cluster starts with a cluster heading (`CLUSTER` followed by a *cluster number*), a list of sequence identifiers, and `ENDCLUSTER`. If a cluster is split into multiple ones at the splitting step, the cluster heading contains `SPLIT INTO` followed by a list of cluster numbers as a result of splitting. A sequence that is articulation point is marked as either `ARTI`, when its neighbor sequences do *not* share common regions of a preset length that can be specified by `-a` option, or `arti`, when its neighbor sequences share common regions.

```

...
CLUSTER 6 size= 2
    1573353
    1786960 arti  6 9 10 1449
ENDCLUSTER

CLUSTER 7 size= 18 SPLIT INTO  1449 1450
    1572960
    1573560
    1573567
    1573634
    1573710
    1573881
    1574592
    1574742
    1786960 arti  6 9 10 1449
    1788922
    1789108
    1789559 ARTI  1449 1450
    1789737
    1789738
    1790302
    1790412
    1790835
    2367247

```

ENDCLUSTER

...

In the above example, the cluster 6 contain two sequences and the sequence 1786960 presents in clusters, 6, 9, 10, and 1449, which becomes candidates for merging (see file `cluster.merge`). The cluster 7 has a heading `SPLIT INTO 1449 1450`, indicating that it is split into two clusters, 1449 and 1450. The sequence 1789559 marked as `ARTI` belongs to two clusters, 1449 and 1450, but different parts of it involve in two clusters, thus the sequence becomes a candidate for multidomain proteins (see file `cluster.multidomain`).

3.2 The file format for pairwise matches

A set of pairwise matches should be prepared using a pairwise sequence alignment program such as FASTA and BLAST. Each line in the file summarizes a pairwise alignment and five fields are required, `seqid1`, `seqid2`, `match-score`, `overlapping regions of seqid1 and seqid2`. Below is an example, where the first line summarizes a pairwise match of 3212176 and 1574400 with matches scores (Zscore of 96.8, Evaluate of 0.311, and a Smith-Waterman score of 90), overlapping regions of 1,339 for 3212176 and 31,111 for 1574400.

```
$ cat pairwise-file
3212176 1574400 96.8 2.4 82 0.311 90 51,137 - 31,111
3212176 1574131 95.4 2.9 86 0.295 78 186,262 - 620,696
3212176 1574206 95.0 3 83 0.221 95 153,243 - 190,281
...
```

By default, the first and second columns are sequence identifiers, the third column a match score for clustering, the eighth and the tenth columns the overlapping regions. The score that will be used for clustering can be changed using `-S column-number` option. The score should be strictly increasing for more significant matches; Zscore, SW score, and bitscore can be used while Evaluate *cannot* be used. The column numbers for overlapping regions can be can changed using `-s column-number1` for the first sequence id. and `-t column-number2` for the second sequence id. The following two commands do the same clustering analyses.

```
$ bag pairwise-file cutoff -P output-file-prefix
$ bag pairwise-file cutoff -S 3 -s 8 -t 10 -P output-file-prefix
```

How to set `cutoff` will be explained in the next section.

3.3 Setting the cutoff score

The command `bag` performs a clustering analysis using a `pairwise-file`. To run `bag`, you need to specify a *cutoff score* C . Any pairwise match with a score below C will be discarded. The value of C should be determined in terms of the number of biconnected components and the number of articulation points while performing a series of clustering analyses with varying cutoff scores. The command `compute-cutoffs` can be used to set C . For example, the following command

```
compute-cutoffs merge.ecoli.h.cmp -L 100 -M1000 -d 100
```

performs 11 clustering analyses with cutoff scores from 100 to 1000 in an interval of 100 as shown below.

```
I read 59993 pairwise matches among 117584
cutoff = 100 : no of bccs = 115 with aps 111
cutoff = 200 : no of bccs = 1448 with aps 205
cutoff = 300 : no of bccs = 1435 with aps 131
cutoff = 400 : no of bccs = 1391 with aps 103
cutoff = 500 : no of bccs = 1330 with aps 98
cutoff = 600 : no of bccs = 1252 with aps 74
cutoff = 700 : no of bccs = 1129 with aps 47
cutoff = 800 : no of bccs = 1028 with aps 35
cutoff = 900 : no of bccs = 936 with aps 28
cutoff = 1000 : no of bccs = 831 with aps 25
```

A guide on setting the threshold can be found in Section 5.

4 Splitting and Merging Clusters

4.1 Splitting clusters with options `-O` and `-L`

Given a cluster F , `bag` tests if all members in F share common regions. You can specify the length of the common regions with options `-O` and `-L`. See RANGE-TEST in Section 5.

The option `-O number1` requires that the common regions should be the minimum `number1` percent of the longest overlapping region.

The option `-L number2` simply requires that the shared regions should be of the minimum `number2` characters (either nucleotides or amino acids).

The file `cluster.history` shows how a cluster is split into multiple ones iteratively.

Enforcing stricter conditions (higher values of `number1` or `number2`) results in clusters of more specific categories (see Section 4.3), but it may result in too many clusters. See Section 4.3 for more discussion on the use of splitting options.

4.2 Merging clusters

Merging clusters are not performed automatically due to the technical difficulties (see Section ??). However, `bag` suggests which clusters may be merged together into one. This is again done by computing biconnected components on a hypergraph where clusters are vertices and edges are articulation points (those with `arti` labels).

When a group contains more than two clusters, it may be considered for merging. However, it is tricky when a group contains only two clusters (see Section ?? for more details).

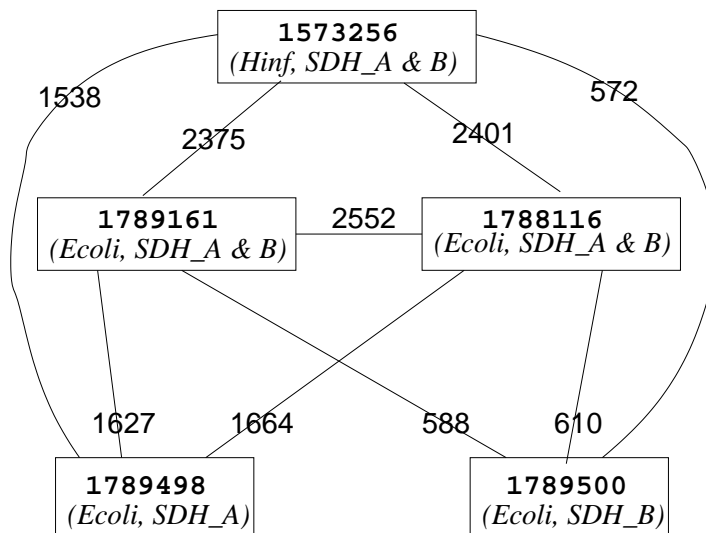


Figure 1: Sequence graphs with the Zscore cutoff threshold of 400. The numbers on the edges denote the Zscores between two sequences. SDH_{α} and SDH_{β} were detected by PFAM search.

4.3 Issues in splitting and merging

In this section, several issues in splitting and merging will be explained using 5 sequences with SDH_{α} and SDH_{β} domains.

A sequence graph is constructed with a Zscore cutoff value of 400 in Figure 1. The family begins to separate into two families at a stricter cutoff value of 600 as shown in Figure 2, *i.e.*, two BCCs, $\{1788116, 1789161, 1789498, 1573256\}$ and $\{1788116, 1789500\}$ with default settings. However, with a stricter clustering criterion, more specific clustering result can be obtained. For example, an option `-o 90` requires all sequences in a cluster should share 90% of the longest pairwise overlap among them, resulting in splitting sequences into three groups; of SDH_{α} only, of SDH_{β} only, and of both domains.

The above example, however, becomes a challenge while merging of several clusters is attempted. Consider merging of three clusters, of SDH_{α} only, of SDH_{β} only, and of both domains. Merging any of the first two and one with both domains will share a domain, but it may not be the result that we want as the resulting cluster loses the information on the three sequences with both domains. A better clustering result may be three clusters, $\{1788116, 1789161, 1789498, 1573256\}$ with SDH_{α} only, $\{1788116, 1789161, 1573256, 1789500\}$ with SDH_{β} , and $\{1788116, 1789161, 1573256\}$ with both domains. However, in case that there are more than two domains, it is not clear how we can represent clustering result. This is why merging is *not* performed automatically for the current version of BAG.

The file `cluster.merge` contains a list of merging suggestions. An option `-a number` can be used for controlling the merging suggestions, which requires an articulation point sequence to share common regions longer than `number` characters. See ARTI-TEST in Section 5.

5 More on BAG algorithm

We performed a series of clustering analyses with Zscore cutoff thresholds ranging from 100 to 1000 at 50 increment intervals for the pairwise comparisons from *E. coli* and *H. influenzae*. The plot on the left side in

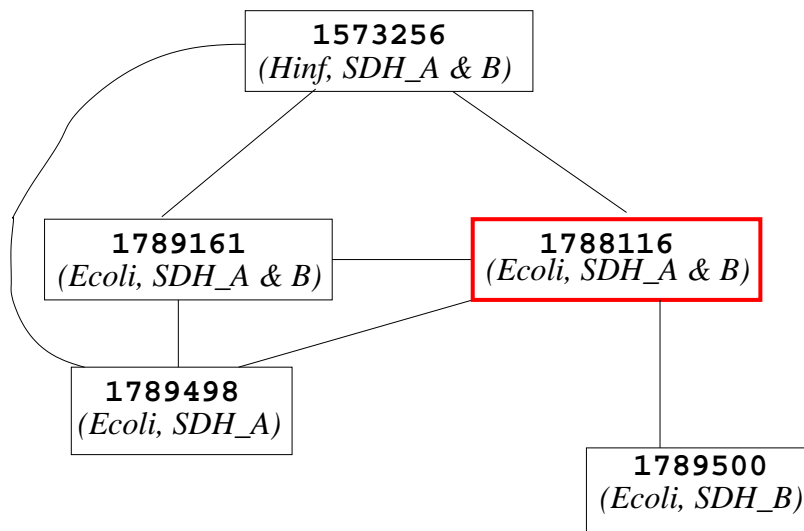


Figure 2: Sequence graphs with the Zscore cutoff threshold of 600. The graph has two BCCs, $\{1788116, 1789161, 1789498, 1573256\}$ and $\{1788116, 1789500\}$, and an articulation point, 1788116.

Figure 3 is the distribution of the number of biconnected components *vs.* the Zscore cutoff threshold. We will call this plot the *BCC plot*. As we can observe in the figure, the number of biconnected components increases up to a certain value, 150 for Zscore, and then continue to decrease. The increase in the number of biconnected components is intuitive as a higher cutoff value will remove more false positives, thereby families of large size due to false positives being separated into several families. The decrease in the number of biconnected components is also intuitive as a higher cutoff value will remove more true positives, thereby more vertices become singletons, *i.e.*, vertices without incident edges; note that singletons are not counted. We would expect that there exists a peak in the BCC plot if the scoring method like Zscore effectively models the pairwise sequence relationship. This observation is consistent with non-statistical scores like Smith-Waterman score [2].

Note that the basic clustering algorithm runs linear time in relation to the number of pairwise matches above a preset cutoff threshold after computing pairwise matches from a set of sequences. The series of clustering analysis with Zscore in Figure 3 took less than 6 seconds on a Pentium IV 1.7 GHz processor machine running Linux. This computational efficiency makes it possible to efficiently conduct the series of clustering analyses with varying cutoff thresholds to find the cutoff threshold, $C_{maxbiconn}$, that generates the maximum number of biconnected components. However, we need to consider the number of articulation points since articulation points are candidates for multidomain proteins. The right plot in Figure 3 shows the number of articulation points with respect to varying Zscore cutoff thresholds. The articulation points become candidates for multidomain proteins and need to be tested for having multidomain proteins: the test method will be briefly described in the following sections. We would avoid selecting the cutoff threshold with too many articulation points. Let NAP_C the number of articulation points at score C . One way to select the cutoff value is to use a ratio $r = \frac{NAP_{C+I}}{NAP_C}$ where I is the interval of the score for the series of clustering analysis.

1. **AP-TEST** tests an articulation point for having potential multidomains.
2. **RANGE-TEST** tests each biconnected component for being a single family.

These two tests are to see if there are common shared regions among the sequences. For example, Figure

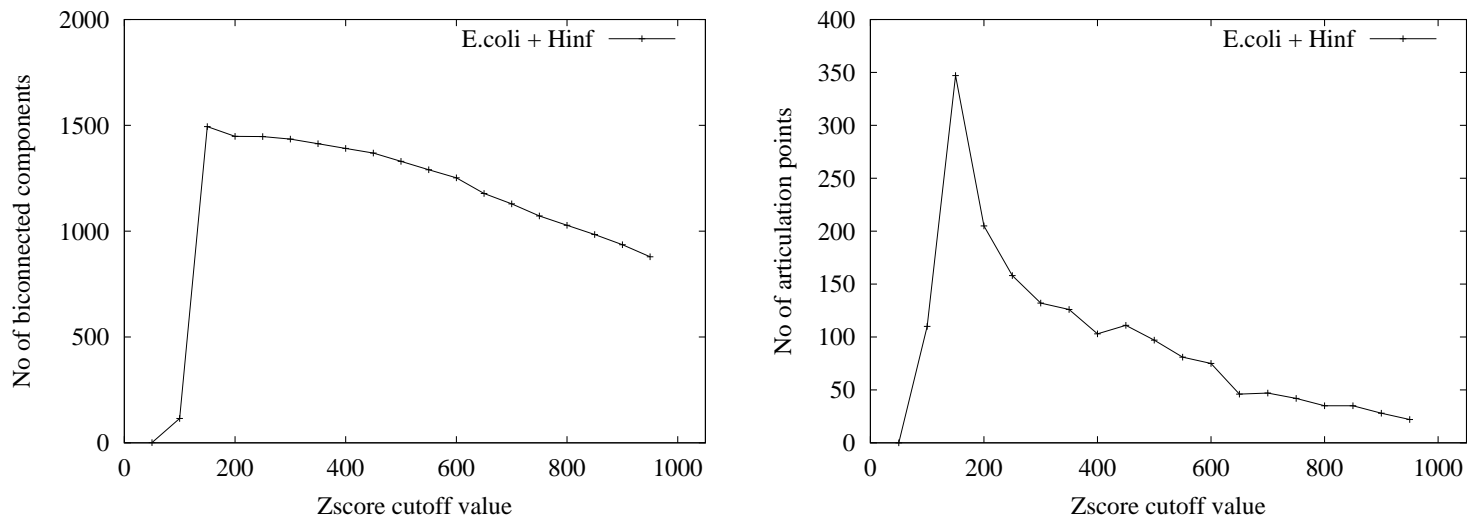


Figure 3: The distribution of the number of biconnected components *vs.* the Zscore cutoff threshold (left plot) and the distribution of the number of articulation points *vs.* the Zscore cutoff threshold (right).

4 shows that four sequences share common subsequence regions. Depending on the test result, splitting and merging operations are performed in a greedy fashion, *i.e.*, a resulting subgraph from a splitting or merging operation is not considered for further splitting or merging.

References

- [1] Bateman, A., Birney, E., Durbin, R., Eddy, S. R., Howe, K. L., and Sonnhammer, E. L. L., (2000) "The Pfam Protein Families Database," *Nucleic Acids Research*, **28** 263-266.
- [2] Kim, S. (2002), Graph theoretic sequence clustering algorithms and their applications to genome comparison, to appear in *Computational Biology and Genome Informatics*, edited by Cathy H. Wu, Paul Wang, and Jason T. L. Wang, World Scientific, 2002

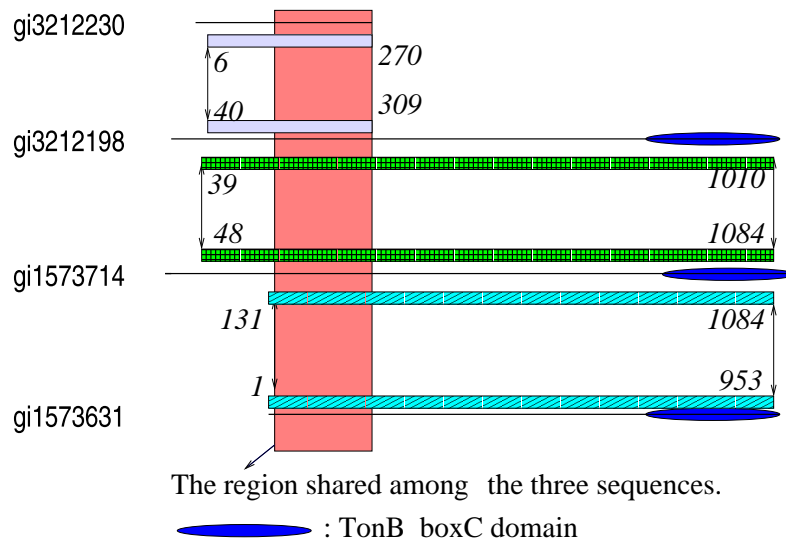


Figure 4: The region shared among gi1573631, gi1573714, gi3212198, and gi3212230 can be computed by chaining pairwise overlaps. All four sequences share domains, that are not present in the PFAM 7.1 database (A recent PFAM contains the domain as TonB_boxC domain and I do not know the TonB_boxC domain is related to the hemoglobin binding domain. A new manual will use another example for the multidomain case.) , and they are annotated to share the hemoglobin binding domain. The three sequences, gi1573631, gi1573714, and gi3212198, share the TonB_boxC domain in the PFAM database. This clustering result demonstrates that our algorithm correctly cluster sequences even when multiple domains are involved.