Manual for Coral: an integrated suite of visualizations for comparing clusterings

Darya Filippova

October 19, 2012

Contents

1 Overview

This tutorial will use the following definitions:

- *data item* - this a basic unit of data of which you may have thousands and you wish to study the relationships these units. They may be proteins interacting with each other, or genes, or protein/DNA sequences, or ... — you name it,
- *module* - a group of data items closely related to one another by some metric or property,
- *clustering* - a collection of modules, possibly overlapping.

Coral is a suite of visualizations that helps you compare multiple clusterings at a time. For example, modules can constitute a collection of genes that get co-expressed together or proteins forming a complex. A *clustering* usually is an output of a clustering algorithm. You may also choose to group data according to attributes that come with the data such as cellular component or molecular function GO terms and use that partition as a clustering. You may combine data from different experiments and across species so long as the data items that the user treats as homologous have the same IDs across the dataset.

2 Running Coral

Coral is a cross-platform desktop application written in Java. Start by downloading the application for your system from `www.cbcb.umiacs.umd.edu/kingsford-group/coral`. You should put the file in the directory of your choice and unzip it.

If you have downloaded Coral package for MacOS, you should double click on Coral's icon in the folder you've just downloaded. This will launch the application with allocating 1Gb of memory for Java's heap.

You can run Coral from a command line by issuing the following command in the top-level Coral directory:

```
$ ./coral
```

in Coral's directory. This will run a shell script that launches Coral, and tells Java to use 1Gb of memory for heap space. You can edit the script to allocate more memory for the heap which usually a good idea in case you have large clusterings or many of them.

3 Loading data

The input files for Coral should each describe a single clustering with every line in the file corresponding to a single module. Coral expects the first item on the line to be a module identifier, i.e.:

```
1 AT3G13710 AT1G74520 AT1G17080 ...
2 AT3G10340 AT3G53260 AT3G59940 ...
3 AT1G70700 AT2G01570 AT1G53510 ...
...
```

Here, 1, 2, and 3 are module identifiers followed by the protein IDs that form those modules. Module names within a single clustering file have to be unique; if a module ID is repeated within a file, Coral reports a parsing error. Coral assumes that each module should have an ID and at least one item in it; if a line describing a module only contains a module name or a single data item, Coral reports a parsing error. If the module name is omitted, and the line starts with data items in the module, the first data item would be parsed as the module name and will not be included as part of the module.

Users can use any latin and punctuation characters for their module and item names. Spaces and tabs can not be part of the name since they are reserved as separator characters. If you use spaces in your data item names, each part would be parsed as a separate data item.

Users can load the clustering files using a `File -> Open clusterings` menu option.

4 Using Coral views

Coral consists of 6 linked views: selection in one view causes the application to select corresponding items in a different view making it easy to track groups of items across the whole application.

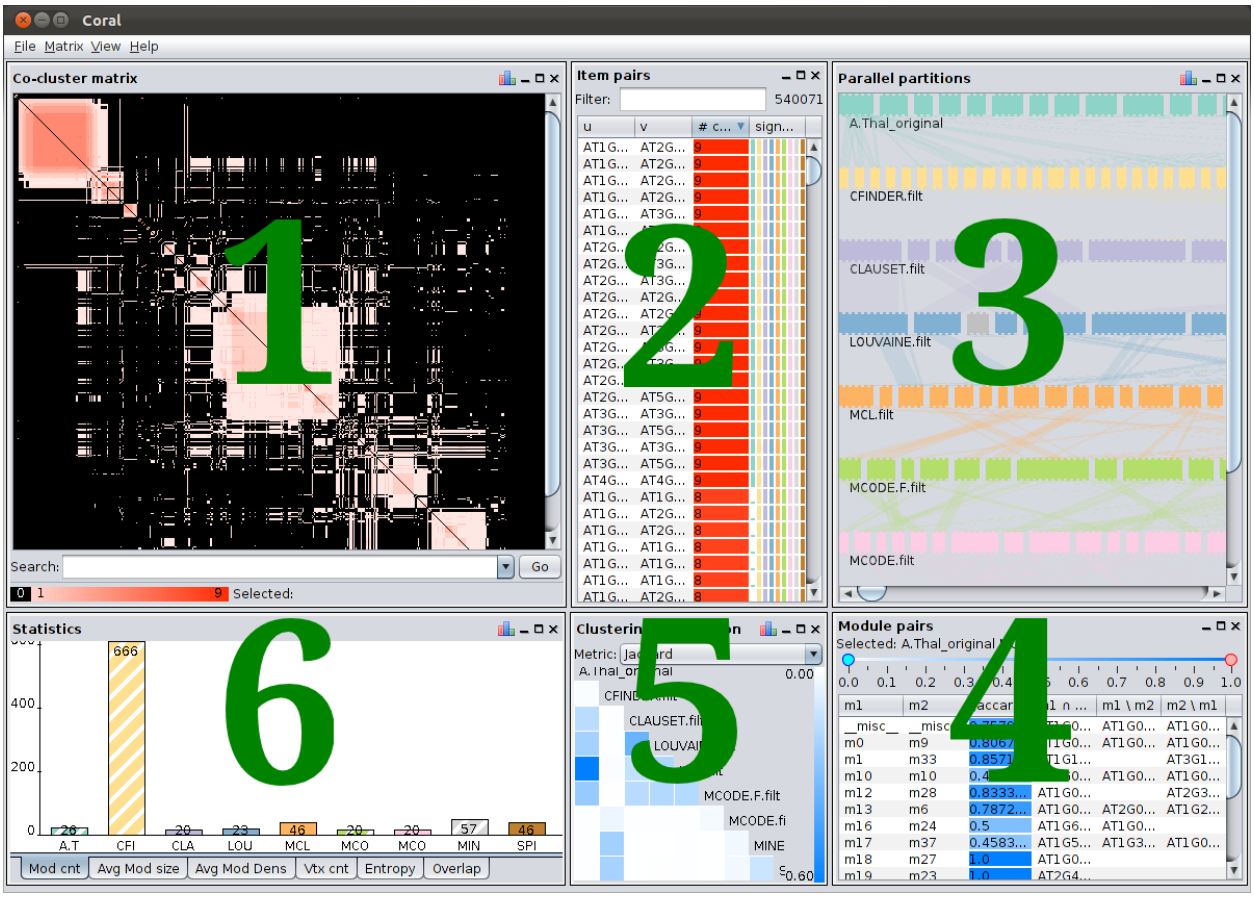


Figure 1: Coral consists of 6 linked views: co-cluster matrix, item pairs table, paralle partitions plot, modules table, comparison ladder, and overview statistics.

4.1 Co-cluster matrix

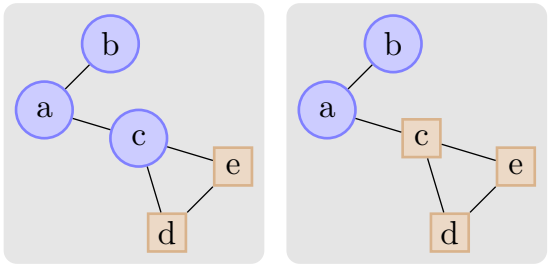
In Coral, pairwise co-cluster memberships are aggregated in a *co-cluster matrix*. Given a single clustering K, $n = |K|$, we define an $n \times n$ matrix A^K to be K's co-cluster matrix where its entries a_{ij}^K are:

$$a_{ij}^K = \begin{cases} 0 & v_i \text{ and } v_j \text{ are in different modules in } K \\ 1 & v_i \text{ and } v_j \text{ are in the same module in } K. \end{cases}$$

You can see a small example of two such co-cluster matrices for partitionings of a small network in Figure 2.

We sum up the co-cluster matrices from individual clustering to form a single matrix:

$$A^+ = \sum_{t=1}^{k} A^{K_t},$$



(a) (b)

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

Figure 2: An example of two different clusterings of a small network and their corresponding co-cluster matrices. Blue and brown nodes represent two modules in the partitioning.

where A^{K_t} is a co-cluster matrix for a clustering K_t, and k is the number of clusterings. Here, the a_{ij}^{+} entries equal k (the number of clusterings) for item pairs (v_i, v_j) that have co-clustered in all partitions suggesting a strong relationship between the items, and the low a_{ij}^{+} values correspond to pairs that co-clustered in only a few clusterings and are more likely to have been assigned to the same module by chance (see Figure 3). The cells are colored according to their values and vary from pale pink (low values) to red (high values).

You can zoom in and out on sections of the matrix with a scroll wheel. When matrix does not fit on the screen, the scrollbars appear, and you can recenter the matrix on the region of interest.

If the clusterings are identical, the co-cluster matrix would look like Figure 3a. A single clustering with overlapping modules would produce a matrix like in Figure 3b. Clusterings that have a high amount of disagreement may look like Figure 3c.

(a) (b) (c)

Figure 3: Various co-cluster matrices. (a) Several identical non-overlapping clusterings. (b) A single clustering with overlapping modules. (c) Two disagreeing non-overlapping clusterings.

4.1.1 Reordering matrix

The amount of information you can learn from the matrix depends on the order of its rows and columns. Coral uses a SPIN algorithm [**?**] for finding an ordering of rows and columns that pushes non-zero matrix elements towards the matrix diagonal. Greedy vs optimal. Iterations. More interations - better solution, but longer time.

To reorder your matrix, you can navigate to `Menu->Matrix->Reorder`. You will see a dialog box asking you to specify how many times to apply one of the reordering techniques. You can set the counter to 0 if

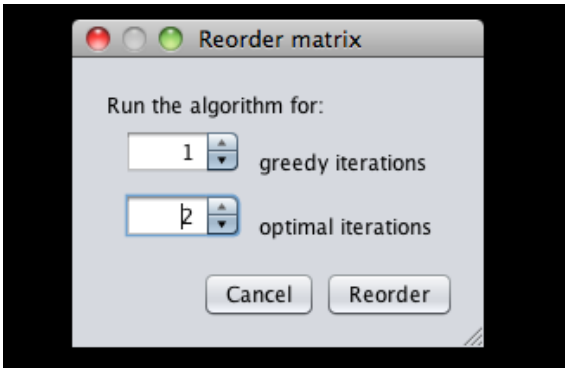


Figure 4: TODO

you do not want that reordering technique to run at all.

4.1.2 Selecting a base matrix

4.2 Item pairs table

An item pairs table facilitates sorting and search for particular item pairs that have co-clustered together at least once. In other words, every non-zero element in the co-cluster matrix gets a row in the pairs table. Each row in the table represents a pair of data items and displays the number of times the items co-clustered along with the pair's *signature*. The signature is a k-long vector where the t^{th} element is 1 when both data items, say, proteins, have been placed in the same module in clustering K_t. If the pair's items were not in the same module in K_t, the t^{th} element is set to 0.

Visually, the signature's elements that are 1 are drawn as tall solid columns and zeros are represented by the short stumps using the same color for each clustering as is used in the overview statistics (Subsection 4.6) and in the parallel partitions plot. Figure 5 shows an example of two such pairs that have different co-cluster signatures suggesting that the relationship between the last two *A. thaliana*proteins is stronger than that of the first pair. Users can sort the rows by either the item name, the number of shared clusterings, or by the co-clustering signature. Users can also filter by the signatures to display only the rows matching a user's pattern.

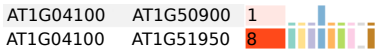


Figure 5: TODO

You can filter the table based on the signatures by defining a regular expression in the textbox above the table (Figure 6) and hitting the "Enter" key. Only rows that match the provided pattern would remain in the table. The row count (to the right of the pattern textbox in Figure 6) would reflect the number of rows that matched. Here are the rules for writing a pattern: if you want the signature to have a 1 in position t, then put a 1 in the pattern at that position. If you want t^{th} element to be 0, then substitute 1 for a 0. If you do not care whether it is a 1 or a 0, then put a dot (.) in the position. In addition, you do not have to specify all k elements, only the first few you are interested in. In the example below, the pattern "1.1111" means that I want to see all pairs of data items that have co-clustered together in clusterings 1, 3, 4 and 5, but it does not matter if data items have been placed together in clustering 2. Omitting the last digits in hte pattern is equivalent to writing "1.1111....", i.e. it is saying that it does not matter whether data items have co-clustered in the last four clusterings. The results reflect this: The first and second pairs' signatures show two ways to match the pattern. By scrolling futher down the table, we see more rows that matched for a total of 33477 rows with this pattern. For the same dataset, replacing a dot in the second position

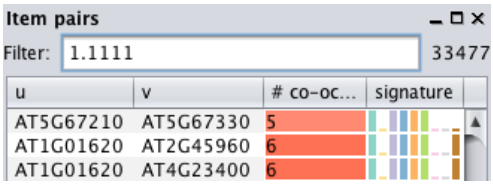


Figure 6: TODO

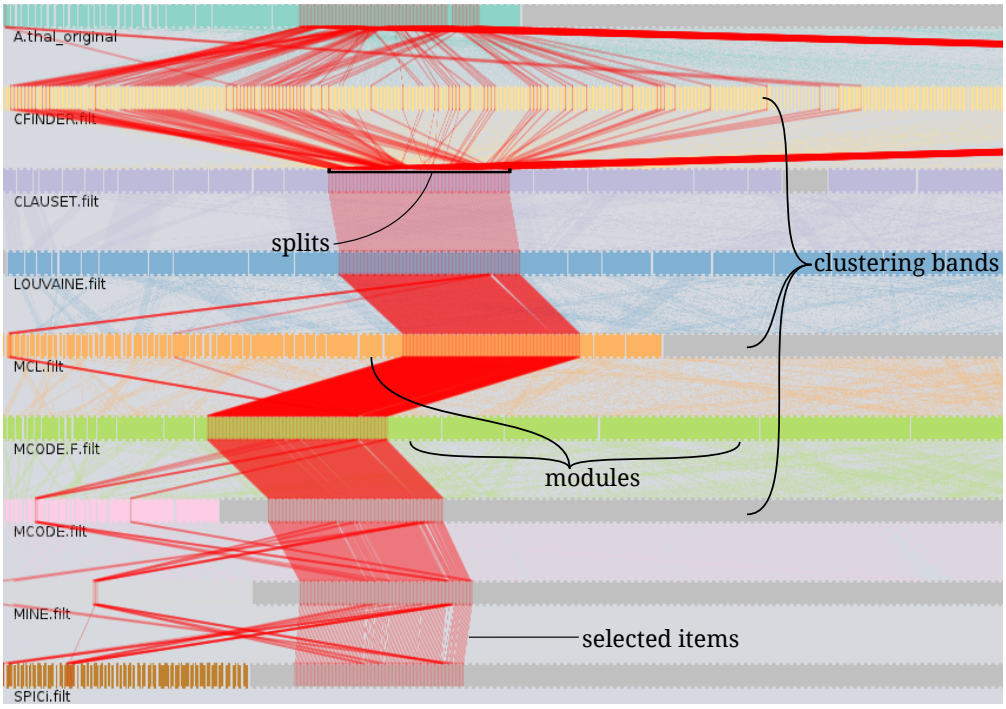


Figure 7

(111111) reduces the number of matched rows to 299, and only 21 rows match the pattern of all ones for each clustering — only 21 item pairs have co-clustered in all of the clusterings in the dataset.

4.3 Parallel partitions

The parallel partitions plot (P^3) represents each clustering as a horizontal band. The blocks comprising each bands represent modules, with the width of a block proportional to the number of items in that module. Semi-transparent parallelograms between clusterings connect data items with the same name. That is, each item in a clustering will be connected to its copy in the band immediately below it (see Figure 7).

Users can select a module with a mouse while holding a "Shift" key to highlight its members in every clustering in the plot (see red traces in Figure 7). Similarly, users may select individual items and trace them through every clustering band. The selections made in the parallel partitions plot propagate to other views making it easy to track the same group of items throughout the application. The plot is zoomable users may zoom in to focus on a few items at a time or zoom out to see global trends across the ensemble. When the zoom level permits it, the plot displays the item labels.

Gray bar in P^3

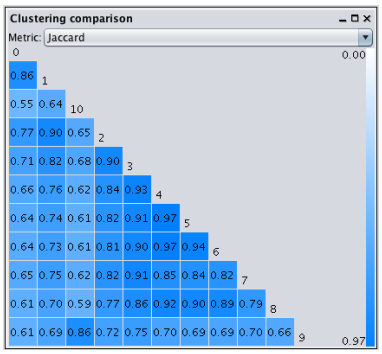


Figure 8

4.4 Module-to-module table

When users select a pair of clusterings using the ladder widget, the module table

4.5 Ladder widget

The ladder widget represents a lower triangle of an all-to-all comparison matrix between every pair of clusterings. For example, in Figure 8, the ladder shows Jaccard similarity between ten different clusterings of a small social network. Cells are color-coded with more intense blue cells corresponding to pairs of clusterings of higher similarity. Users may choose between several similarity metrics discussed in Section 5.

TODO: clicking

4.6 Overview statistics

Overview statistics are rendered as bar charts with each bar corresponding to the value of a statistics of a single clustering. If a clustering contains overlapping modules, some of the statistics (such as the average module size or entropy) may be skewed – the bars for such clusterings are drawn using a hashed pattern (see Figure 1, area 6: the tall yellow bar represents one such clustering).

Module count — shows the number of modules in each clustering. The `GrabBag` module is not included in this count.

Average module size — shows the average module size per clustering. Computed as $\sum_i m_i/m$ where the numerator represents the sum of aizes across all modules (excluding the `GrabBag`) and the denominator represents the number of modules considered in the numerator.

Average module density — XXX (e/v or e/chose2). Only available for the network modules and requires that the original network is loaded.

Item count* — displays the total number of data items that were assigned to some module by the given clustering. * in older versions of Coral "item count" was called "vertex count".

Entropy — displays the entropy value for each clustering giving an idea of how skewed the distribution of module sizes is. Entropy of a clustering K is computed as $-\sum_{m \in K} p_m \log_2 p_m$ where $p_m = |m|/|K|$ is a probability of a data item being assigned to module m.

Overlap — displays the percentage of data items included in the clustering that were assigned to more than one module.

5 Similarity metrics

The ladder widget offers users a choice of several metrics for evaluating similarity between pairs K_i, K_j of clusterings within an ensemble. The first four measures are based on the pair counting; the next two are

based on information theory; the last 3 measures are based on set matching.

There are four quantities on which pair counting measures are based:
a – number of item pairs placed in a module together in clustering K_i as well as in clustering K_j,
b – number of item pairs placed in a module together in clustering K_i, but not in clustering K_j,
c – number of item pairs placed in separate modules in K_i, but in the same module in K_j,
d – number of item pairs placed in separate modules in both K_i and K_j.

Jaccard
$a/(a+b+c)$,

Mirkin
blah

Rand
The third etc ...

Folkes-Mallows
The third etc ...

Mututal information
The third etc ...

Variation of information
The third etc ...

Purity
The third etc ...

Inverse purity
The third etc ...

F-measure
The third etc ...

6 Managing windows

7 Clustering algorithms

8 Errors and exceptions

9 Symbols

K – a single clustering,
m_i^j – an i^{th} module in a clustering j,
$n = |K|$ – number of data items considered in the clustering K,
A^K – co-cluster matrix for a single clustering K,
a_{ij}^K – an element of a co-cluster matrix for a clustering K, a 0 or a 1,
A^+ – a sum of co-cluster matrices for a collection of k clusterings $K_1, \ldots, K_k$,
a_{ij}^+ – an element of a co-cluster matrix, ranges from 0 to k,
k – number of clusterings in an ensemble,
E_{in} – the number of edges with both endpoints within a given module
E_{out} – the number of edges with only a single endpoint within a given module