Mutual Information Co-evolution Detection Software, User Manual

From "aCES: A Co-Evolution Simulator generates co-varying proteins and nucleic acids"

Authored by Devin Camenares
Assistant Professor, Department of Biochemistry
Alma College
614 W. Superior St, Alma MI 48801
camenaresd@alma.edu

Purpose:
This tool will take two multiple sequence alignments (MSA) and calculate, for each residue pair, the mutual information. This property is defined as follows:

$$MI(i,j) = \sum P(X_i, Y_j) \cdot \log\left(\frac{P(X_i, Y_j)}{P(X_i) \cdot P(Y_j)}\right)$$

Here, the mutual information (MI) of the i-th residue from sequence #1 and the j-th residue from sequence #2 is calculated as the sum of coincidence of different residue identities in the pair. For example, if two DNA sequences are being compared, MI(i,j) is the sum of the calculations for the i-th residue as A and the j-th residue as C; the i-th residue as A and the j-th residue as G, and so forth, for all combinations.

For each calculation, the probability of a particular pair is represented as P(Xi, Yj) and the probabilities for the individual residues is represented as P(Xi) and P(Yi).

Mutual information reports the degree of coincidence, or in this case, co-evolution, between two residues either within one gene / protein (intramolecular) or between two separate molecules (intermolecular). Such co-evolution predicts that these residues form a sequence specific, functionally important interaction. Once verified experimentally, knowledge about these interactions informs the design of orthogonal molecule sets and may aid the field of synthetic biology (among others).

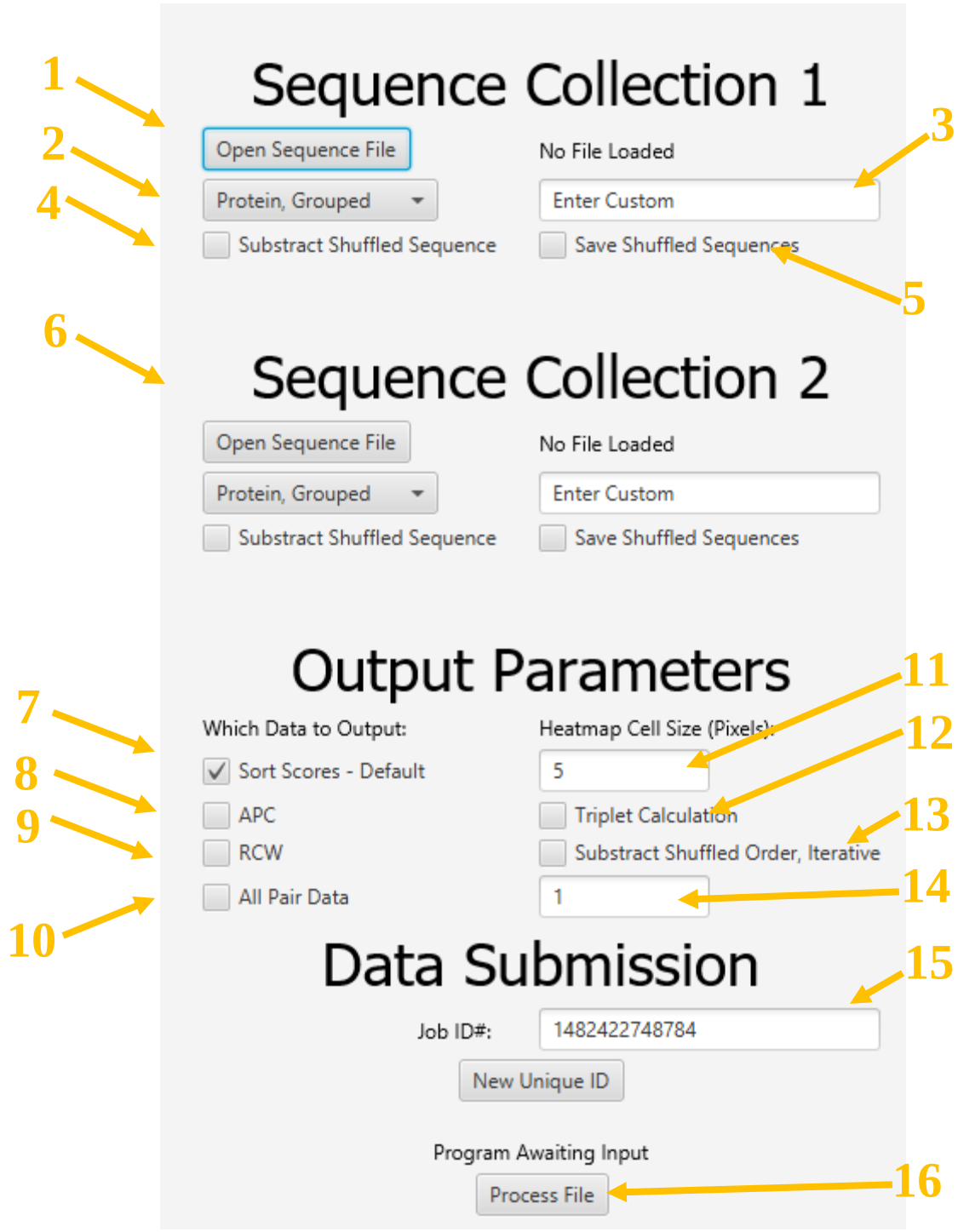
To aid in analysis of the data, the program will also calculate the intramolecular co-evolution present in each MSA that is loaded, and output this data as well.

In addition, this program allows for correction factors to be applied to the data: both average product correction (APC) or row-column weighting (RCW). For more information about mutual information and these correction facts, please view Buslje et al, 2009: "Correction for phylogeny, small number of observations and data redundancy improves the identification of co-evolving amino acid pairs using mutual information". (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2672635/)

Input:
This tool requires between one and two text files (If only one file is loaded, only intramolecular co-evolution can be calculated) containing FASTA formatted multiple sequence alignments. Each FASTA entry must have a header that starts with the ">" character and ends with "|", followed by the aligned sequence.

In order to produce meaningful results, the sequences in a file must all be the same size (typical for an alignment) and the sequence position or row in each file must share the same source organism (i.e. the first sequence in each file must be a gene or genes from *E. coli*, the second in each file must be from *F. tularensis,* etc)

Interface:



1. **Open MSA File Button**: This button will open a .txt or .fa file with an MSA in a FASTA format; please see the Input section above for more information on requirements for file format and content.

2. **Identity Categories Presets**: This will govern how the residues or identities are processed by the program. The Protein, Grouped preset (default) will group amino acids by chemical properties as follows (groups separated by commas): MC, DE, KR, VIAL, FWY, H, G, P, TSNQ. In this example, an Arginine residue (R) and a Lysine residue (K) will be treated as equivalent. In the Amino Acid preset, all amino acids are treated separately. The DNA and RNA grouping treats nucleotides separately: grouping is either A, T, C, G or A, U, C, G, respectively.

3. **Custom Identity Categories:** If you desire, you can input your own grouping or categories for the residue identity. Each group must be an unbroken string, and groups are separated by commas. For example, you may want a variation on the Protein Grouped preset above; perhaps you want to group Histidine with basic amino acids, as well as grouping Glutamine and Asparagine with Glutmate and Aspartate. To accomplish this, you would enter MC, DENQ, KRH, VIAL, FWY, G, P, TS (the order of the groups or characters is not important, only the position of the commas and spaces).

4. **Subtract Shuffled Sequences:** If this option is selected, the program will engage a correction routine after calculation of the initial MI values. This routine will take the original inputs, scramble the identities for each amino acid (i.e. randomizing the order of each residue), recalculate the mutual information, and then subtract this value from the original mutual information. This is a correction method that is intended to reduce signals by subtracting background noise. This process can be iterated to average out all random background (see option **14** below for control of iteration).

5. **Save Shuffled Sequences:** Together with option 4, if this is selected then the data generated for each shuffled sequence will be saved for further analysis.

6. **Sequence Collection #2:** This section of the interface contains items 1-5 as they apply to the 2nd sequence collection / MSA.

7. **Sort Scores:** If this option is selected, then any set of raw data will have a corresponding file in which the residue pairs are sorted, in descending order, based upon their value. This is a useful option if you want to quickly determine the areas that have the strongest degree of co-evolution.

8. **APC (Average Product Correction):** Selecting this option will apply, both to intramolecular MI calculations and intermolecular MI calculations, the average product correction. This correction method is designed to reduce background noise, and is calculated as follows:

$$APC(i,j) = MI(i,j) - \frac{MI(i) \bullet MI(j)}{MI(n)}$$

The APC of residue pair i & j is equal to the mutual information for the pair (MI-i,j), minus a factor that is the sum of residue i MI values (MI-i) times the sum of residue j MI values (MI-j), divided by the overall sum of MI from the entire analysis (MI-n).

9. **RCW (Row Column Weighting):** Selecting this option will apply, both to intramolecular MI calculations and intermolecular MI calculations, row-column weighting. This correction method is designed to reduce background noise, and is calculated as follows:

$$RCW(i,j) = \frac{MI(i,j) \cdot 2}{MI(i) \cdot MI(j)}$$

The RCW value of residue pair i & j is equal to double the mutual information for the pair (MI-I,j), divided by the

product of the sum of residue i MI values (MI-i) and the sum of residues j MI value (MI-j).

10. **All Pair Data:** If this option is selected, then alongside the MI score for any residue pair will be the actual counts for every possible pair of identities for those two residues. Selecting this option will increase run time and also increase the resulting file size, but it will allow instant determination for what identities work in concert for any co-evolving pairs of interest.

11. **Heat Map Size:** A graphical heat-map is generated for any and all data sets and corrections when the program is run; here, you can set the dimensions of every square representing a single residue pair (i.e. if set to 5, then a comparison between a MSA that is 100 characters in length and another that is 300 in length will lead to a graphic being produced that is 500 x 1500 pixels in size).

12. **Triplet Calculation:** Selecting this option will apply a novel correction method to the intermolecular co-evolution signal. The 'Triplet' correction method is a novel correction method that seeks to minimize any intermolecular covariation signals that are generated between residues that have strong intramolecular covariation. Such residues are likely to display intermolecular signals that represent indirect coevolution; in other words, these residues do not make any functionally significant, sequence specific contacts, but rather they covary due to the constraints they share with other residues. The formula used to apply this correction is as follows:

$$Triplet^{a,b}(i,j) = \frac{MI^{a,b}(i,j)}{MI^{a}(i) \cdot MI^{b}(j)}$$

The 'Triplet' calculation for the intermolecular (a,b) signal between residues i & j is calculated by dividing the intermolecular mutual information of this residue pair (MI-a,b-i,j) by the product of the sum of the **intra**molecular values for residue i (MI-a-i) by the sum of the intramolecular values for residue j (MI-b-j).

13. **Subtract Shuffled Order:** Selecting this option will shuffle the order of organisms within a sequence collection. In contrast to option **4**, which shuffles individual amino acids, this option keeps individual sequences intact. The resulting shuffled sequences are subtracted from the original value. This routine can be iterated, set by the number field below it (option **14**).

14. **Shuffling Iterations:** This number field can be used to set the number of times the shuffle routine, set in either option **4, 13,** or both, will be run.

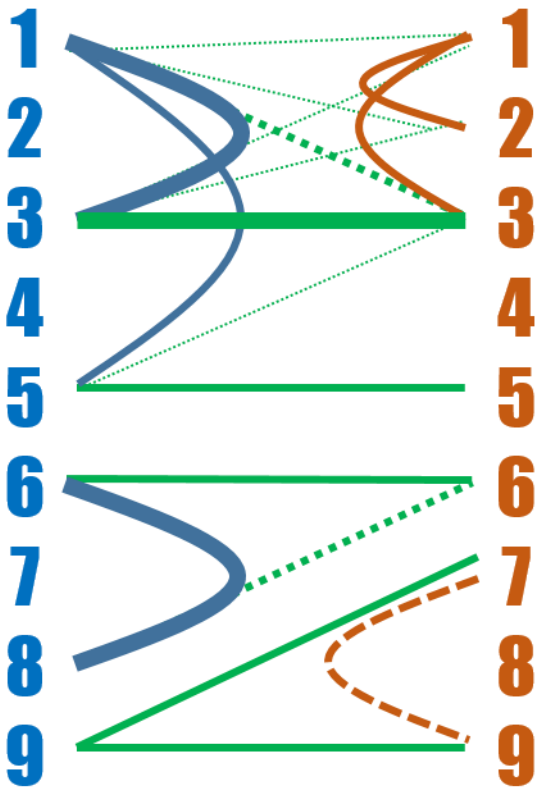
15. **Job ID#:** This textfield, with a timestamp generated number, will be used to name the directory in which the output files will be placed. You may rename the folder by changing this value, or generate a new ID (so that previous work is not overwritten) by hitting the 'Generate New ID' button. Note that a new ID is automatically generated each time the program is loaded anew.

16. **Processing Button:** Selecting this button will process the inputs; an error message will be displayed if the one or more inputs are missing.

Test Data (Note: the below pertains to the old example data. Newer example input can also be tested, and is described in the main text of the paper).

To test the program, you can run the sample data, which includes **msaCoEvSim_molA.txt** and **msaCoEvSim_molB.txt**. This will only generate I recommend that you use the intramolecular option when performing this analysis. If successful, the program should generate the similar data to the files present in the **Example_Output** folder.

The sample data are short, simulated MSAs of proteins, each 9 residues long and from 500 species, that feature co-evolution between certain pairs. In order to ensure that there is some co-evolution between certain residues, the following constraints were applied when the sequences were generated:



In the diagram above, blue and orange numbers represent positions 1-9 in sequences A and B, respectively. Solid lines indicate constraints for co-evolution, dashed lines indicate expected indirect co-evolving pairs. Line color is blue or orange to represent intramolecular co-evolution (consistent with text color for sequence A or B), and green to represent co-evolution.

When run, the program will generate a new folder, in the same directory as the executable JAR file. Inside this folder will be a text file with runtime information (# of sequences analyzed, # of residue pairs, time to process, etc). A raw data output is unsorted: the residue pairs are given in order alongside the MI score. A heatmap will also be included. Depending on parameters set a number of other files may be present, as discussed in the Interface section above.