Journal of Bioinformatics and Computational Biology

aCES: ACo-Evolution Simulator Generates Co-Varying Protein And Nucleic Acid Sequences..

--Manuscript Draft--

Manuscript Number:	JBCB-1113R1
Full Title:	aCES: ACo-Evolution Simulator Generates Co-Varying Protein And Nucleic Acid Sequences..
Article Type:	Research Paper
Corresponding Author:	Devin Camenares, Ph.D., Alma College Alma, MI UNITED STATES
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Alma College
Corresponding Author's Secondary Institution:	
First Author:	Devin Camenares, Ph.D.,
First Author Secondary Information:	
Order of Authors:	Devin Camenares, Ph.D.,
Order of Authors Secondary Information:	
Abstract:	Sequence-specific and consequential interactions within or between proteins and/or RNAs can be predicted by identifying co-evolution of residues in these molecules. Different algorithms have been used to detect co-evolution, often using biological data to benchmark a methods ability to discriminate against indirect co-evolution. Such a benchmark is problematic, because not all the interactions and evolutionary constraints underlying real data can be known a priori . Instead, sequences generated in silico to simulate co-evolution would be preferable, and can be obtained using aCES, the software tool presented here. Conservation and co-evolution constraints can be specified for any residue across a number of molecules, allowing the user to capture a complex, realistic set of interactions. Resulting alignments were used to benchmark several co-evolution detection tools for their ability to separate signal from background as well as discriminating direct from indirect signals. This approach can aid in refinement of these algorithms. In addition, systematic tuning of these constraints sheds new light on how they drive co-evolution between residues. Better understanding how to detect co-evolution and the residue interactions they predict can lead to a wide range of insights important for synthetic biologists interested in engineering new, orthogonal interactions between two macromolecules.

Journal of Bioinformatics and Computational Biology

ACES: A CO-EVOLUTION SIMULATOR GENERATES CO-VARYING PROTEIN AND NUCLEIC ACID SEQUENCES.*

DEVIN CAMENARES†

Department of Biochemistry, Alma College, 614 West Superior St
Alma, Michigan 48801, United States of America
camenaresd@alma.edu

For Review – 20-03-09

Sequence-specific and consequential interactions within or between proteins and/or RNAs can be predicted by identifying co-evolution of residues in these molecules. Different algorithms have been used to detect co-evolution, often using biological data to benchmark a methods ability to discriminate against indirect co-evolution. Such a benchmark is problematic, because not all the interactions and evolutionary constraints underlying real data can be known a priori. Instead, sequences generated in silico to simulate co-evolution would be preferable, and can be obtained using aCES, the software tool presented here. Conservation and co-evolution constraints can be specified for any residue across a number of molecules, allowing the user to capture a complex, realistic set of interactions. Resulting alignments were used to benchmark several co-evolution detection tools for their ability to separate signal from background as well as discriminating direct from indirect signals. This approach can aid in refinement of these algorithms. In addition, systematic tuning of these constraints sheds new light on how they drive co-evolution between residues. Better understanding how to detect co-evolution and the residue interactions they predict can lead to a wide range of insights important for synthetic biologists interested in engineering new, orthogonal interactions between two macromolecules.

Keywords: Coevolution, Mutual Information, Protein-Protein interaction.

1. Introduction

Many important cellular behaviors rely on protein-protein, protein-RNA, or RNA-RNA interactions. While highly conserved residues are often important, residues that are poorly conserved may still be critical for proper function if they help form a sequence-specific interaction.[1,2] These residues tolerate evolutionary changes coupled to compensatory changes in the interacting partner residue, found either within the same molecule or across two different molecules. Thus, identifying co-varying residue pairs using a bioinformatics approach may help uncover unappreciated sequence-specific interactions important for function. Subsequent experimental validation of these residue pairs can lead to new insights into the structure and function of these molecules and can facilitate synthetic biologists in re-engineering molecular interactions to create orthogonal

*A pre-print version of this article is available on bioXriv.org, and the software files and raw data for simulations here is present via the article, the author's website (http://bioinformatics.org/aces/), or by email request.
†Present address of the author.

systems or produce novel behaviors.[3,4] In addition, recent studies have employed this approach on a large scale, analyzing co-variation and co-evolution among many proteins in concert to uncover new interactions.[5]

Co-evolution can be calculated as the amount of mutual information (MI, Eq. 1) for every residue pair within and between two molecules.[6]

$$MI(i,j) = \sum_{X,Y}^{n} P(X_i, Y_j) \cdot \log\left(\frac{P(X_i,Y_j)}{P(X_i)\cdot P(Y_j)}\right) \quad (1)$$

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). The mutual information (MI) value 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 Adenine and the j-th residue as Cytosine; the i-th residue as Adenine and the j-th residue as Guanine, and so forth, for all combinations. For each calculation, the probability of a particular pair is represented as $P(X_i, Y_j)$ and the probabilities for the individual residues is represented as $P(X_i)$ and $P(Y_j)$.

For a given pair of residues, this score might reflect a true interaction. It may also be the result of indirect co-evolution, in which residues A and B both interact and co-evolve with the same third residue, C. Furthermore, differences in the amount of conservation among these residues and phylogenetic effects may also obscure the true nature of the scores obtained from this bioinformatics analysis.[7]

In order to better discriminate between direct co-evolution and interfering signals, correction factors for MI calculations and alternative approaches have been developed.[2,8] These different approaches are often assessed and compared through the use of benchmark datasets. Well characterized biological sequences are typically chosen as benchmarks, and detection methods are often evaluated on their ability to infer interactions between residues that are in close proximity in structural models. However, using biological data as a benchmark is problematic, as even well-studied proteins may not have been exhaustively characterized and might exist in multiple conformations or higher order structures.[2,9]

An ideal dataset for benchmarking co-evolution detection methods can be created by generating sequences in silico given appropriate constraints. Such a simulated dataset avoids the uncertainty that high scoring residue pairs are either a false positive or a previously unappreciated interaction between the two molecules.[10] Unlike previous attempts to create such simulated datasets, the new software tool described here, aCES, generates sets of sequences that can be specifically tuned to simulate both intramolecular and intermolecular co-evolution, as well as direct and indirect signals.[11-13]

This software, alongside related tools, are available for download at http://bioinformatics.org/aces/ (or by request). This includes example data, source code, and multiple versions of the software.

2. Software Description and Features

aCES was written in Java and available as an executable JAR file. To create a dataset that simulates co-evolution, the user must provide constraints for the sequences to be generated, such as the molecule's length, type of residues, conservation of key residues, and the strength and direction of the co-evolution interactions. These constraints can be provided via the graphical user interface; another version of this program is also available for input via an command-line interface. All of these features need not be defined precisely, as wildcard inputs are possible, allowing the user to capture stochastic co-evolution as well. Provided an input text file with these constraints, aCES will generate sequences for five molecules, designated A through E. This allows for enough complexity to simulate two molecules that share co-evolution directly (i.e. A and B, Fig. 1), co-evolve indirectly via a third interacting partner (C, Fig. 1), or are constrained by co-evolving partners that are not shared (D or E, Fig. 1).

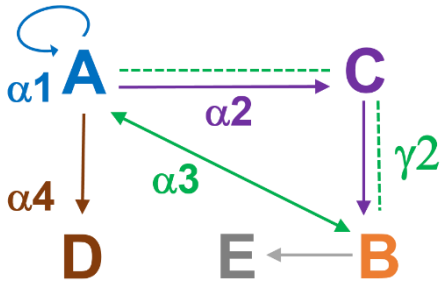


Fig. 1. Model of complex co-evolutionary relationships. The aCES software can model evolutionary relationships between five proteins or molecules, A-E. They are defined by their shared (A, B and C) or exclusive relationships (A is exclusive with D, B is exclusive with E). Relationships involving residues in molecule a are indicated in the figure by an alpha designation. These include intramolecular co-evolution (1), intermolecular between A and C (2), intermolecular between A and B (3), and intermolecular between A and D (4). Indirect co-evolution between A and B results from the shared relationships each molecule has with C; this indirect signal is represented by a dotted line (gamma-2). Other relationships, such as B and E, are present but not labelled.

To create a realistic simulation of co-evolution, distributions of identities are generated for every targeted residue pair. This distribution is adjusted using a Monte Carlo approach, selecting for those which have the desired MI score (Eq. 2).

$$P(X_i, Y_j) = (\varphi \cdot c) + P(X_i) \cdot P(Y_j) \quad (2)$$

A given distribution of residue identities for a particular pair, with each combination represented as $P(X_i, Y_j)$, is created by the expected occurrence in the absence of co-evolution, which is the product of $P(X_i)$ and $P(Y_i)$, adjusted by the product of a random variable, phi, and term proportional to the desired level of co-evolution, c. In the event that no co-evolution is specified, $c = 0$, and $P(X_i, Y_j)$ matches the product of independent probabilities, $P(X_i)$ and $P(Y_i)$. The possible probabilities are adjusted during creation of the identity distribution to ensure that the sum of a particular row or column is always equal

to $P(X_i)$ or $P(Y_i)$, respectively. Together these distributions across all interacting pairs are subsequently used to drive the creation of a simulated set of sequences.

By default, this fixed set of residue identities are randomly assigned in each sequence that is created, which mimics a 'star' phylogeny. However, users can specify the degree to which they want the result to mimic bifurcating evolution. This is a more realistic phylogenetic relationship for sequences, although it does tend to obscure co-evolution signals within small sequence sets.

In addition to the generated multiple sequence 'alignments', the software can also report the underlying distributions, and how close they match the input constraints. The resulting sequences should feature co-evolution only for those residues in which this was initially specified, and therefore these sequences are an ideal dataset to benchmark existing co-evolution detection methods.

3. Evaluation of Existing Co-Evolution Tools

To demonstrate the functionality of aCES, I created a dataset that simulates a set of protein-protein interactions to be used as a benchmark. This simulated dataset features varying levels of conservation and constraints for distinct intramolecular and intermolecular co-evolution (Fig 2, Text S1 – See appendix).

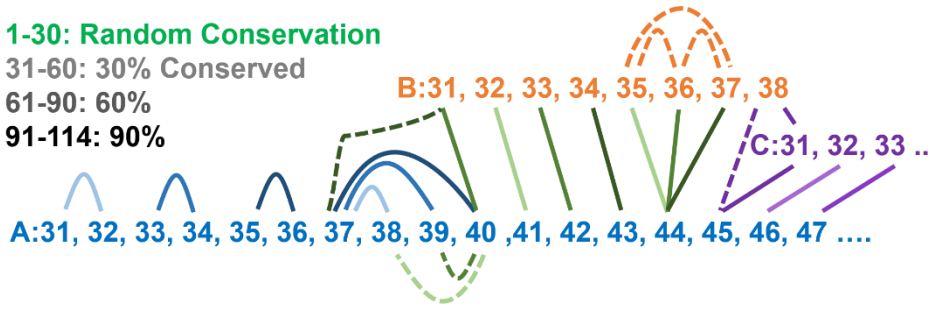


Fig. 2. Simulation and detection of Co-evolution in protein sequences. An overview of selected constraints used in the preliminary simulation. Lines indicate co-evolution constraints, shading indicates strength, while dotted lines indicate expected indirect co-evolution. Color indicates the nature of the relationship: blue is intramolecular co-evolution within molecule A (alpha-1 in Fig. 1), green is co-evolution between A and B (alpha-3 in Fig. 1), purple is co-evolution between A and C (alpha-2 in Fig. 1). The constraints are present in a repeating pattern, with each repeat featuring a different overall level of conservation. Please refer to text S1 for a more complete description.

The simulated dataset features alignments of 500 examples of 5 separate proteins, with repetitive constraints to test strong co-evolution against a variable, low, or high level of sequence conservation. This repetitive pattern of adjacent constraints is not required, however, and long-range interactions across the primary sequence of a molecule can be simulated just as effectively. Three independent replicates of this dataset were created using the same constraints, both to determine the consistency of the program and to capture stochastic events for some residues (pairs in the range 1-10 featured a variable amount of conservation in each replicate).

To verify that the software produced datasets that are faithful to the input constraints, the difference between the targeted amount of mutual information and the final product was examined (Fig 3). This analysis showed that in most cases, the program created both a distribution that featured the desired level of mutual information, and faithfully used this distribution of probabilities to create the simulated sequences. In rare cases, the program failed to create a distribution with the target level of mutual information within the specified limit of iterations (Fig 3, bottom, pair 65:66). Raising this limit, which would increase computation time, would achieve an even higher fidelity, but was not considered necessary in this preliminary analysis.

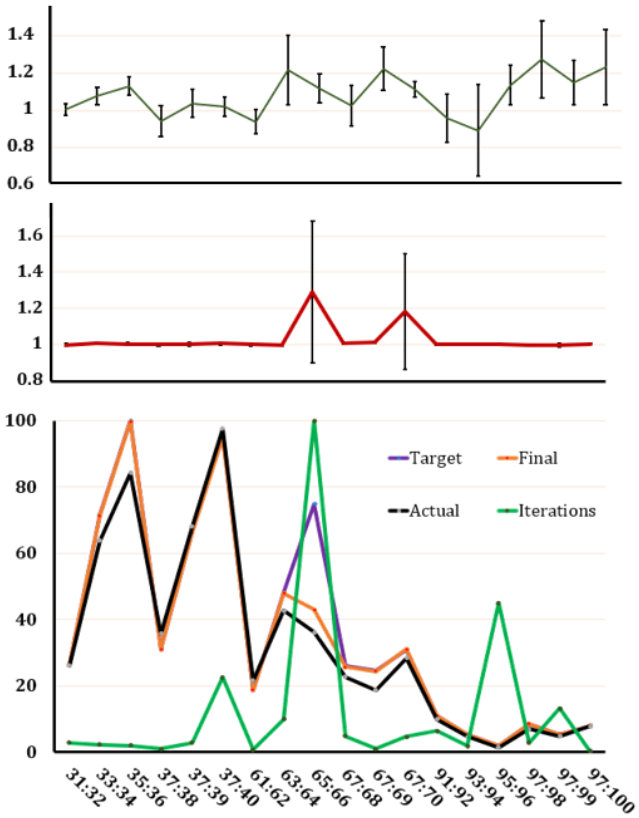


Fig. 3. Results of the simulation closely match the input constraints. Sequences generated as a benchmark for co-evolution detection methods were analyzed for how close they match their input constraints (Text S1). On the X-axis for all three graphs are a select group of residue pairs from molecule A that should display varying levels of intramolecular co-evolution. Top: Ratio of final MI distribution values before sequence generation versus actual measured MI values in the final sequences, Middle: ratio of target MI values to final MI distribution values, and Bottom: normalized target, final, and actual MI values for particular intramolecular pairs, including the number of iterations to create the MI distribution. Top and middle graphs are averages of three independent replicates of sequences created with the same constraints; the bottom graph is representative data from this set.

These verified datasets were used to evaluate several co-evolution detection methods: the MISTIC2 server, GREMLIN, a server hosted by the Gerstein lab, and my own Java-based program.[14-16] Across this panel of software, different computational approaches were tested. This included the calculation of MI, Pearson Correlation, and Direct Coupling Analysis. In each case, I evaluated the ability of each method to distinguish between the residues that have initial co-evolution constraints and all other residue pairs. A measure of how well the method distinguished between direct and indirect co-evolution was also calculated (Fig. 4).

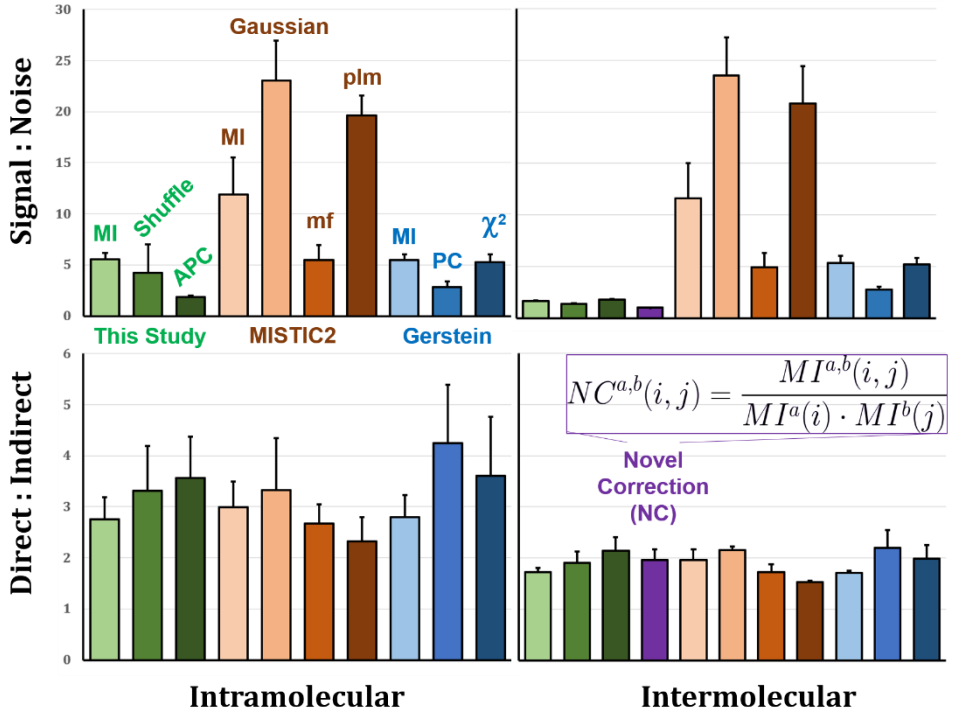


Fig. 4. Evaluation of Co-Evolution detection methods using the benchmark dataset. Up to 11 different detection methods and correction factors, implemented from my own software (This Study - miBio2, Green, Bars 1-3, or 1-4 for Intermolecular), the MISTIC2 server (Orange, Bars 4-7, or 5-8 for Intermolecular), or the co-evolution server provided by the Gerstein lab (Blue, Bars 8-10, or 9-11 for Intermolecular), which included average product correction (APC), Pearson Correlation (PC), and three different implementations of direct coupling analysis (DCA) – mean field (mf), pseudo-likelihood maximization (plm), and Gaussian modelling, were used to analyze protein coevolution in three independent simulated datasets. Gerstein's ELSC method and the GREMLIN server were also used, but not shown due to inconsistent results and data cutoff issues. Top: the Signal:Noise ratio for each detection method was calculated as the average score for all pairs with co-evolution constraints versus the average score for all other pairs. This was calculated for both intramolecular (left) and intermolecular (right) co-evolution. Higher Signal:Noise ratios indicate a better performing detection method. Bottom Left: Direct versus indirect intramolecular co-evolution was calculated as the ratio between the score for residue pair A37- A40, which is a direct interaction, and A37-A39, which is an indirect yet expected interaction. Bottom Right: Direct

versus indirect intramolecular co-evolution was calculated as the ratio between the score for residue pair A40-B31, which is a direct interaction, and A37-B31, which is an indirect yet expected interaction. Higher Direct:Indirect ratios indicate a better performing detection method. Equation Inset: The Novel Correction Factor minimizes any intermolecular covariation signals that are generated between residues that have strong intramolecular covariation. The novel correction value (NCa,b) for the pair of the i-th residue in molecule A and the j-th residue in molecule B is calculated as the intermolecular mutual information score for that pair (MIa,b) divided by the product of the sum of the intramolecular values for residue i (MIa) and the sum of the intramolecular values for residue j (MIb).

This analysis demonstrates that some co-evolution detection methods tend to be more reliable for highlighting the strongest interactions (in particular, the Gaussian modelling and PLM methods in the MISTC2 server), while most tools have a similar ability to discriminate between direct and indirect signals (See Tables 1 and 2). It is interesting to note that although aCES uses a MI-based method to generate co-evolution, the MI-based detection methods do not enjoy a consistent advantage over more recently developed algorithms based on direct coupling analysis.

Table 1. Top ranked co-evolving intermolecular pairs across different methods. Representative data from three independent trials.

	Ranked Pair for Intermolecular Coevolution (A-i:B-j)					
Method	1	2	3	4	5	6
MISTIC2-PLM	**44:37**	**43:34**	70:61	44:36	14:7	40:31
MISTIC2-MF	**44:37**	**43:34**	42:33	40:31	12:3	44:36
MISTIC2-GAUSS	**44:37**	**43:34**	44:36	14:7	70:61	40:31
MISTIC2-MI	**44:37**	**43:34**	42:33	40:31	44:36	12:3
ACES-NOVEL	74:67	44:37	73:64	43:34	74:66	103:94
ACES-APC	**44:37**	**43:34**	40:31	44:36	42:33	12:3
ACES-SHUFFLE	**44:37**	**43:34**	42:33	12:3	40:31	44:36
ACES-MI	**44:37**	**43:34**	42:33	12:3	40:31	44:36
GERSTEIN-CHI	**44:37**	**43:34**	42:33	12:3	40:31	44:36
GERSTEIN-ELSC	**44:37**	42:33	14:7	40:31	74:67	14:6
GERSTEIN-COR	70:61	10:1	103:94	112:98	44:37	92:98
GERSTEIN-MI	**44:37**	**43:34**	42:33	12:3	40:31	44:36
GREMLIN	44:37	43:34	70:61	44:36	14:7	40:31
Most Common	44:37	43:34	42:33	12:3	40:31	44:36
Occurrence / 13	11	10	6	4	4	5

In Table 1, the top scoring 6 pairs of residues from each analysis is displayed as the i-th residue from molecule A followed by the j-th residue of molecule B. Methods employed are abbreviated in similar fashion to Fig. 4 (ACES corresponds to the 'This Study'). The most commonly occurring residue across all methods in a particular rank is in bold and shown at the bottom of the graph. While some methods are in strong agreement, it is worth noting that pairs that score highly in a minority of methods, were designed to

have a high degree of co-evolution (90%, the same as pair 44:37) but an overall lower amount of variation than other pairs (the case for pair 74:67) or a higher number of co-evolving partners (the case for pair 70:61). Thus, some methods may be better at taking this effect into account than others.

A similar pattern can be seen in Table 2, which shows top pairs for the detection of intramolecular pairs, or pairs between the i-th and j-th residue, both from molecule A. In this case, the i-th residue is always set to be the lower number; this is why the same pair is repeated in some cases twice, it is detected in some methods from either direction. Noteworthy is that pair 37:40 features both direct and indirect co-evolution constraints; pair 35:36 is constrained to have the same amount of direct co-evolution, and the same level of overall variation, but no indirect co-evolution is specified. A heatmap for the same dataset of intermolecular and intramolecular contacts allows for a complementary visualization (Fig. 5, legend continued on following page).

Table 2. Top ranked co-evolving intermolecular pairs across different methods. Representative data from three independent trials.

	Ranked Pair for Co-evolution (A-i:A-j)					
Method	1	2	3	4	5	6
MISTIC2-PLM	37:40	63:64	37:39	35:36	33:34	7:9
MISTIC2-MF	37:40	35:36	37:39	33:34	7:9	1:2
MISTIC2-GAUSS	37:40	63:64	35:36	33:34	37:39	65:66
MISTIC2-MI	37:40	35:36	37:39	33:34	7:9	63:64
ACES-APC	37:40	37:40	35:36	35:36	37:39	37:39
ACES-SHUFFLE	37:40	37:40	35:36	35:36	37:39	37:39
ACES-MI	37:40	37:40	35:36	35:36	37:39	37:39
GERSTEIN-CHI	37:40	35:36	37:39	7:9	33:34	1:2
GERSTEIN-ELSC	35:36	37:40	7:9	37:39	33:34	9:23
GERSTEIN-COR	37:40	3:4	91:92	94:112	92:112	37:39
GERSTEIN-MI	37:40	35:36	37:39	7:9	33:34	1:2
GREMLIN	37:40	67:70	7:10	97:100	60:83	65:74
Most Common	37:40	35:36	37:39	35:36	33:34	37:39
Occurrence / 13	11	4	5	4	4	4

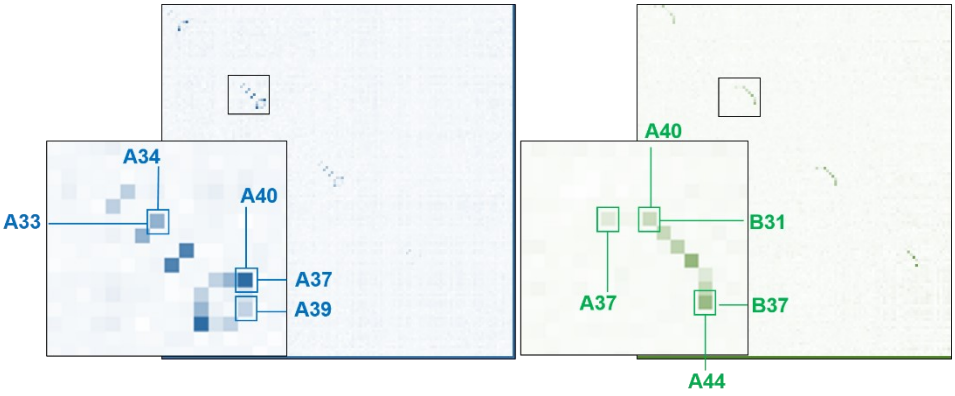


Fig. 5. Realistic co-evolution between two proteins can be simulated. Sequences generated as a benchmark for co-evolution detection methods were analyzed for the amount of mutual information present (Text S1). MI Scores are displayed over all residue pairs as a heatmap, with more intense shading indicating a higher MI score. These MI scores were computed using the miBio2 Java program included in the supplementary information. Left (Blue): a heatmap of intramolecular co- evolution, for residue pairs within molecule A. Background signals are minimized through subtraction of shuffled sequences. Inset: A particular region is enlarged to make visualization of the co-evolving pairs more apparent. Note that some pairs, such as A35-A36, are not labelled but nonetheless match the constraints, and can be followed by counting from the labelled cells in the heatmap. Right (Green): a heatmap of intermolecular co-evolution, for residue pairs between molecule A and B. Background signals are minimized through the use of the novel correction factor. Inset: A particular region is enlarged to make visualization of the co-evolving pairs more apparent.

The above analysis, as well as additional simulations involving co-evolution interactions with nucleic acids (Fig 6), validates aCES as a useful tool for producing a benchmark dataset for co-evolution studies. This tool has a unique approach to creating distributions of residue pairs with realistic co-evolution. Further refinement of the program is possible and ongoing – an improved version can model each organism by first creating a pool of sequences using standard mutation models (such as Jukes-Cantor), followed by selection which adheres to the co-evolution constraints specified by the user. While more computationally intensive and less robust, preliminary analysis indicate that this approach also better mimics bifurcating phylogeny. A beta version of this updated tool is also available at the aforementioned site, with supporting documentation.

As demonstrated here, the aCES tool allows a user to systematically evaluate co-evolution detection methods, improve these methods, and tease apart the relative contributions made by different interactions and residues. This should aid studies that use co-evolution analysis to determine intermolecular interactions, and also find similar utility for those trying to engineer protein function via directed evolution.[2,8,17]

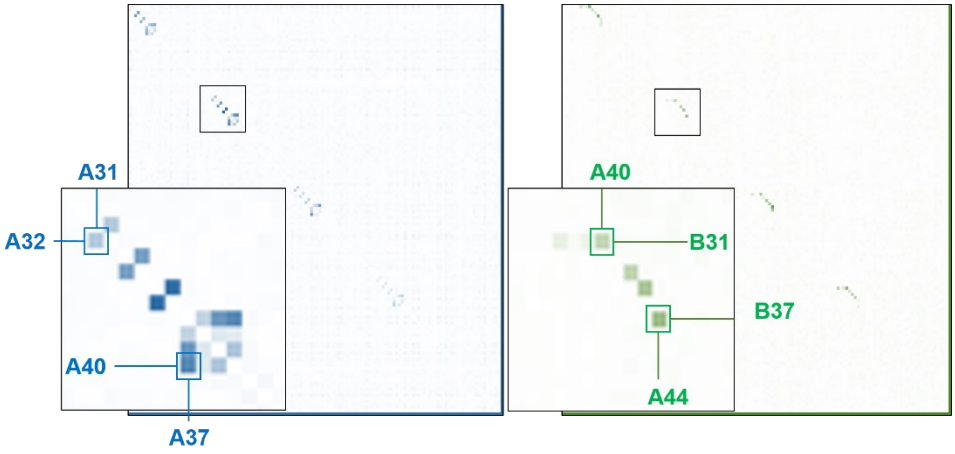


Fig. 6. Realistic co-evolution between a nucleic acid and a protein can be simulated. Sequences generated as a benchmark for co-evolution detection methods were analyzed for the amount of mutual information present (Text S1), using the same constraints with the single modification that molecule A was specified here to have only 4 categories of residues, characteristic of a nucleic acid. The legend for this figure is identical to that of S5, as (left) intramolecular co-evolution is displayed in blue, while (right) intermolecular co-evolution is displayed in green, and (insets) selected regions of the heatmaps are enlarged to make it easier to visualize the high scoring pairs.

Acknowledgments

A special thanks to Dr. Jim Mazzuca and Dr. Brad Westgate for careful reading of this manuscript, advice and insights. Thanks to Christopher Camenares for assistance with programming.

Appendix A. Supplementary Files and Information

Software.zip Available as supplementary information, contains both the executable JAR files and source code for the co-evolution simulation software (ACES) as well as a mutual information-based co- evolution detection tool (miBio2). Instructions on how to properly setup input information and utilize each program, as well as example inputs, are also included. This is available directly from http://bioinformatics.org/aces/files/Software.zip

Preliminary_Analysis.zip This includes sequences generated for the preliminary analysis of protein- protein co-evolution, used as a benchmark against existing detection methods, and some of the results of this analysis (truncated due to size constraints; full set available upon request). Within the main folder, there are three subdirectories for each independent analysis of sequences generated using the same constraints. Within these replicates, you will find a folder ending in "_aces" containing the sequences, and other folders that contain

the results of co-evolution detection on sequences from molecules A and B. This is available directly from http://bioinformatics.org/aces/files/Preliminary_Analysis.zip

Text S1: Description of constraints used in preliminary analysis. Conservation: Five protein molecules were generated, with a length varying between 120 and 135 residues. For each residue, there were 20 identities (all the natural amino acids) possible. In each molecule, the conservation of the first 29 and the last 5-20 residues were variable; residues 30-59 featured 30% conservation, residues 60-89 featured 60% conservation, and residues 90-115 featured 90% conservation. The level of conservation is expected to modulate the amount of co-evolution that is detectable – nearly invariant residues cannot display significant evolution at all, nor with a partner residue. Co-evolution: Across each set of 30 residues in each molecule, repeating patterns of co-evolution were featured. In this pattern (exemplified by Fig 1, Top) there was direct intramolecular co-evolution within A between residue pairs 1-2, 3-4, 5-6, 7-8, 7-9, and 7-10. Thus, this pattern repeated for pairs 31-32, 33-34, 35-36, 37-38, 37-39, and 37-40, and so on. In this pattern, the 1-2 and 7-8 pairs featured weak co-evolution constraints (+), the 3-4 and 7-9 pairs featured intermediate co-evolution constraints (++), and the 5-6 and 7-10 pair featured strong co-evolution constraints (+++). Likewise, there was intermolecular constraints for residue pairs A10-B1 (++), A11-B2 (+), A12-B3 (++), A13-B4 (+++), A14-B5 (+), A14-B6 (++), A14-B7 (+++). Inclusion of a third molecule adds the constraints A15-C1 (+++), A16-C2 (+), A17-C3 (++), B8-C1 (+++). This pattern of intermolecular constraints also repeated every ~30 residues. Due to this network of direct constraints, some indirect co-evolution is also expected. For example, residues A8 and A10 both share a co-evolving residue, and thus should display indirect co-evolution. Sequence Generation: For every co- evolving pair, a distribution of identities for the two residues must be created which features the desired mutual information score. Tunable parameters here included the maximum number of iterations for creating each distribution (1000), a multiplier for the random adjustment factor in each iteration (10), and the deviation from the target MI score that was acceptable to stop the iterations and select a distribution (1%).

References

1. C.M. Buslje, E. Teppa, T. Di Doménico, J.M. Delfino, M. Nielsen, Networks of high mutual information define the structural proximity of catalytic sites: Implications for catalytic residue identification, PLoS Comput. Biol. 6 (2010). doi:10.1371/journal.pcbi.1000978.
2. F. Morcos, J.N. Onuchic, The role of coevolutionary signatures in protein interaction dynamics, complex inference, molecular recognition, and mutational landscapes, Curr. Opin. Struct. Biol. 56 (2019) 179–186. doi:10.1016/J.SBI.2019.03.024.
3. J. Franceus, T. Verhaeghe, T. Desmet, Correlated positions in protein evolution and engineering, J. Ind. Microbiol. Biotechnol. 44 (2017) 687–695. doi:10.1007/s10295-016-1811-1..
4. M.A. Stiffler, F.J. Poelwijk, K.P. Brock, R.R. Stein, A. Riesselman, J. Teyra, S.S. Sidhu, D.S. Marks, N.P. Gauthier, C. Sander, Protein Structure from Experimental Evolution, Cell Syst. (2019). doi:10.1016/J.CELS.2019.11.008.
5. Q. Cong, I. Anishchenko, S. Ovchinnikov, D. Baker, Protein interaction networks revealed by proteome coevolution., Science. 365 (2019) 185–189. doi:10.1126/science.aaw6718.

6. J. Hummel, N. Keshvari, W. Weckwerth, J. Selbig, Species-specific analysis of protein sequence motifs using mutual information., BMC Bioinformatics. 6 (2005) 164. doi:10.1186/1471-2105-6-164.
7. D. Talavera, S.C. Lovell, S. Whelan, Covariation Is a Poor Measure of Molecular Coevolution., Mol. Biol. Evol. 32 (2015) 2456–68. doi:10.1093/molbev/msv109.
8. M.B. Zerihun, A. Schug, Biomolecular coevolution and its applications: Going from structure prediction toward signaling, epistasis, and function, Biochem. Soc. Trans. 45 (2017) 1253–1261. doi:10.1042/BST20170063.
9. I. Anishchenko, S. Ovchinnikov, H. Kamisetty, D. Baker, Origins of coevolution between residues distant in protein 3D structures, Proc. Natl. Acad. Sci. 114 (2017) 9122–9127. doi:10.1073/pnas.1702664114.
10. L.C. Martin, G.B. Gloor, S.D. Dunn, L.M. Wahl, Using information theory to search for co-evolving residues in proteins., Bioinformatics. 21 (2005) 4116–24. doi:10.1093/bioinformatics/bti671.
11. A. Pang, A.D. Smith, P.A. Nuin, E.R. Tillier, SIMPROT: using an empirically determined indel distribution in simulations of protein evolution., BMC Bioinformatics. 6 (2005) 236. doi:10.1186/1471-2105-6-236.
12. S.H. Ackerman, E.R. Tillier, D.L. Gatti, Accurate simulation and detection of coevolution signals in multiple sequence alignments., PLoS One. 7 (2012) e47108. doi:10.1371/journal.pone.0047108.
13. A. Low, N. Rodrigue, A. Wong, COMPASS: the COMPletely Arbitrary Sequence Simulator., Bioinformatics. 33 (2017) 3101–3103. doi:10.1093/bioinformatics/btx347.
14. E.A. Colell, J.A. Iserte, F.L. Simonetti, C. Marino-Buslje, MISTIC2: comprehensive server to study coevolution in protein families, Nucleic Acids Res. 46 (2018) W323–W328. doi:10.1093/nar/gky419.
15. S. Balakrishnan, H. Kamisetty, J.G. Carbonell, S.-I. Lee, C.J. Langmead, Learning generative models for protein fold families, Proteins Struct. Funct. Bioinforma. 79 (2011) 1061–1078. doi:10.1002/prot.22934.
16. K.Y. Yip, P. Patel, P.M. Kim, D.M. Engelman, D. McDermott, M. Gerstein, An integrated system for studying residue coevolution in proteins, Bioinformatics. 24 (2008) 290–292. doi:10.1093/bioinformatics/btm584.
17. J.M. Nicoludis, R. Gaudet, Applications of sequence coevolution in membrane protein biochemistry, Biochim. Biophys. Acta - Biomembr. 1860 (2018) 895–908. doi:10.1016/J.BBAMEM.2017.10.004.

Devin Camenares received a Doctorate in Molecular and Cellular Biology from Stony Brook University in 2013, working in the laboratory of Dr. Wali Karzai on *trans-translation*.

From 2014 to 2018, he was an Assistant Professor in the Department of Biological Sciences at Kingsborough Community College, part of the CUNY system in New York City. Since 2018, he has been teaching as an Assistant Professor and iGEM coordinator in the Department of Biochemistry at Alma College, a small liberal arts college in Mid-Michigan. He is passionate about synthetic biology and proud to be primary mentor for the Alma College iGEM team, who successfully earned a silver medal in their inaugural season in 2019. Go Scots!

Response to Reviewers, Camenares

Below are the reviewer comments (in blue) and my response– usually a summary of the changes made the manuscript and/or the software in accordance with their comments. I would like to thank the editor and reviewer for their feedback, which helped me to strengthen this work. In fact, some of the comments have inspired me to create additional tools that complement the software here, such as one that will convert PDB file information into a set of constraints. These are not quite ready, and so I did not mention them in the manuscript, but will provide them on the website when they are complete.

In many instances, I felt that a brief mention of the improvement in the manuscript was sufficient, and that demonstration of how these points were address are more appropriate here or in the software manual.

> Specific comments:
> (1) First, thanks to the author for making the software available for review - while it was a bit of a pain to find the right mechanism to share the software, I think that being able to run the software is an essential part of the review process.

No problem! To further increase access, I have provided these tools available for download at http://bioinformatics.org/aces/

> (2) On the software - the user manuals for both ACES and miBIO were clear and useful, and the software ran without a problem on my computer (a somewhat out of date MacBook Air).

Thanks!

> (3) One minor issue with the output - the output files were not put into the output folder, but rather into files with the folder name appended with a \. That is, the default output directory name was 1588082672944, and the output files ended up with the names like "1588082672944\Molecule_A_1588082672944.txt". Possibly a DOS/Unix issue?

This was by design – the output is put into a folder that has the name provided by the user for that run. That way, multiple runs can be carried out in succession, each placing sequences into a separate folder.

However, the reviewer's point indicates that this may not be ideal for everyone. Therefore, I have modified the program to include an option to output to common output folder. The files will still be named with the title of the run (which is the date in milliseconds by default).

This has been addressed – an option has now been included to place results into a common output folder, named output (created if not already present). This is not noted specifically in the manuscript, but indicated in the program's manual.

> (4) It is certainly convenient to have a java GUI. It might also be useful to implement a command line interface for larger automation tasks, such as simulation over a wide range of parameters.

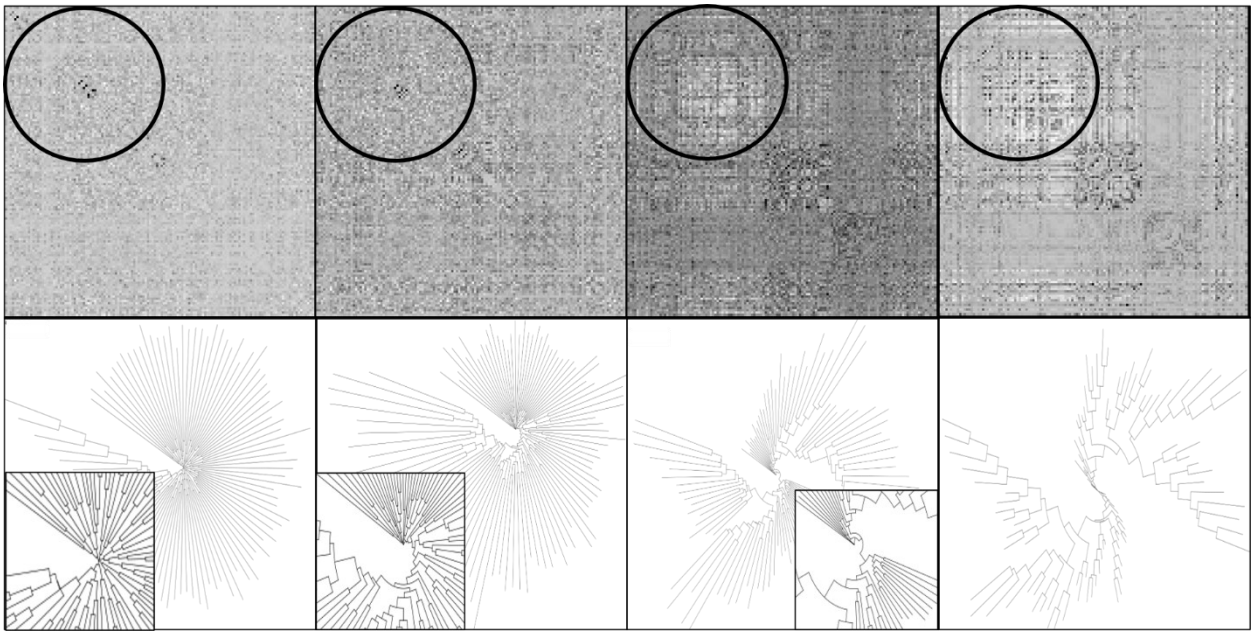
I have addressed this by creating a command-line interface version of the software. It noted in the manuscript, and related instructions are included in the manual. This version of the software will simply request an input filename, and so can be automated over a large set of

files using standard command-line "<" function for a text file that contains the address of all files to be analyzed. Supporting documentation is also provided alongside this version, available online.

> (5) My main conceptual concern is the lack of phylogenetic framework. I presume that ACES treats each of the evolving lineages as independent, and thus assumes a star phylogeny. However, most real datasets are generated from a bifurcating process, not a star phylogeny. It is not clear to me that benchmarking results obtained using alignments with an underlying star phylogeny will translate to alignments with more realistic evolutionary histories. Ideally the author would include the option to evolve sequences on a phylogeny. Minimally, this limitation of the software should be acknowledged up front, and implications noted.

This is a correct interpretation of the underlying working of the program as delivered to the reviewer: it uses essentially a star phylogeny, since it is creating each sequence independently using rules and constraints determined at the outset. This default condition is now noted in the manuscript.

I have added the ability to mimic, to a degree desired by the user, a bifurcating phylogeny. This was accomplished by constraining the program to keep residues similar across organisms as much as possible, while still adhering to the constraints. While this lead to the desired phylogenetic structures, it predictably obscured the co-evolution signals. Examples and demonstration of this effect has been included below, and in the manual for the program, and not the manuscript itself (I can change this if necessary, but felt that the manual was a more appropriate document – it seemed forced to work in a demonstration of this feature into the manuscript).



In the above figure, four different settings for phylogeny are shown, indicating the % of time a bifurcating process was used. From left to right, each vertical panel is 0%, 50%, 80%, and 100%. The top panels show a heatmap for co-evolution, with a peak signal expected in the circled region (and very prominent in the 0% bifurcating setting). The bottom panels shown

a phylogenetic tree for the sequences, with an inset zoomed onto the root or center of the tree.
The difference shown here is a drastic one – this is intentional, as a small sequence set makes the tree structure easier to visualize. However, a consequence of this is a strong obfuscation of the co-evolution signal. With larger sequence sets and moderate levels of bifurcating phylogeny, the result should be somewhere between what is seen above for 0 and 50%.s

The heatmaps were generated using my own mutual information program, and the bottom trees were generated using a combination of Clustal Omega (https://www.ebi.ac.uk/Tools/msa/clustalo/) and iTOL (https://itol.embl.de/).

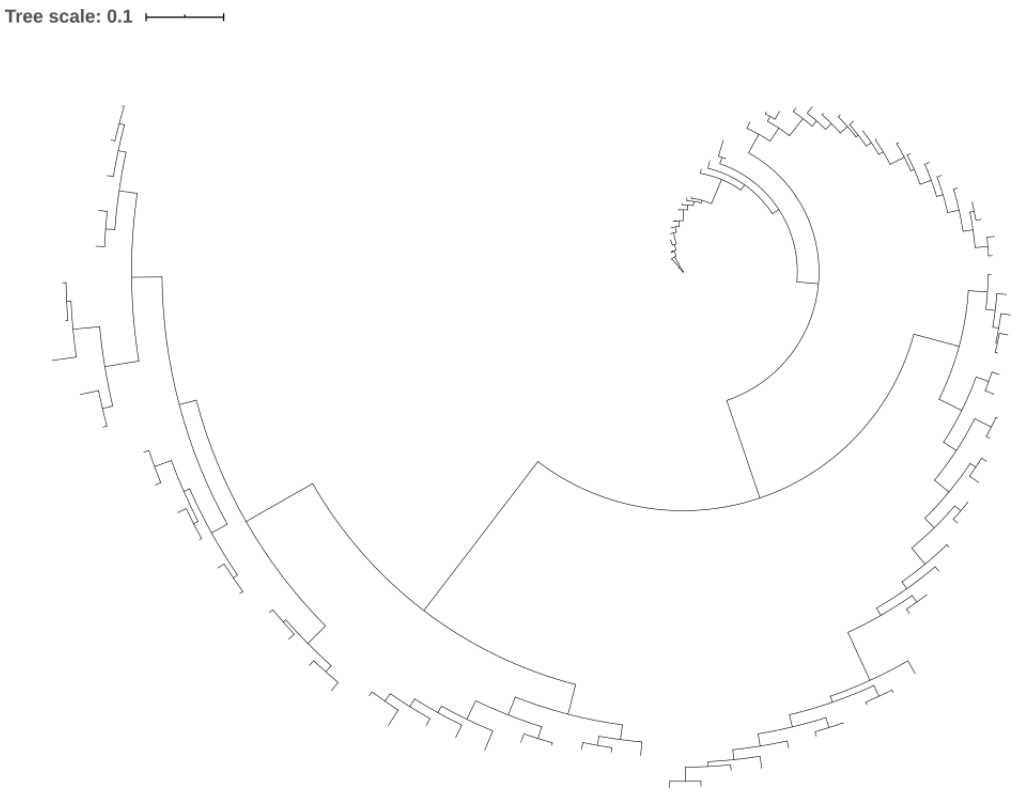
Moreover, this is also addressed in the alternative version of the program that am creating, ACES2, mentioned below.

> (6) In other respects, it is unclear how the simulation parameters match up to those typically used in molecular evolution. For example, under what model are the sequences evolved (e.g., Jukes-Cantor, HKY85, etc…)? with what parameter values (kappa for HKY85, for example)?. How much sequence divergence is generated in a given simulation? Can this be altered?

The simulation does not use typical models of evolution. The amount of sequence divergence can be specified, and this together with the co-evolution matrix that is determines what mutations will be made. This has been made more explicit in the manuscript.

Moreover, I am in the process of developing a refined version of the program (ACES2) in which rounds of mutation and selection are present. For each organism to be simulated in the program, a pool of sequences is generated, scored based on how well they match the constraints, and a top scoring sequence is chosen to be fixed. In this way, a true evolution process is simulated. However, to ensure that the resulting sequence collection is faithful to the input constraints, additional culling of the sequences is implemented.

This refined program relies on a Jukes-Cantor model, using signal mutation rate which can be user-defined. This approach also better replicates a bifurcating model of phylogeny, if desired by the user, as exemplified by the phylogenetic tree below generated from these sequences (see the below figure – representing 100 separate 'organisms', with a strongly bifurcating phylogenetic relationship).



A beta-version of this software is available, and indicated as such in the manuscript, although it does not reproduce co-evolution in as robust a manner. Future iterations of the program will function more effectively and allow for different models of sequence mutation and evolution, such as the GTR model of Tavare, with constants to be specified by the user.