Sequence analysis

The EVcouplings Python framework for coevolutionary sequence analysis


1Department of Systems Biology, 2Department of Cell Biology, Harvard Medical School, Boston, MA 02115, USA, 3cBio Center, Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Boston, MA 02215, USA, 4Center for Bioinformatics, University of Tübingen, 72076 Tübingen, Germany, 5Applied Bioinformatics, Department of Computer Science, 72076 Tübingen, Germany, 6Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139, USA, 7Computational Biology Center, Memorial Sloan Kettering Cancer Center, New York, NY 10065, USA, 8Department of Biomedical Engineering, University of Virginia, Charlottesville, VA 22904, USA and 9Department of Informatics, Technische Universität München, 85748 Garching, Germany

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first three authors should be regarded as Joint First Authors.
‡The authors wish it to be known that, in their opinion, the last two authors should be regarded as Joint Senior Authors.

Abstract

Summary: Coevolutionary sequence analysis has become a commonly used technique for de novo prediction of the structure and function of proteins, RNA, and protein complexes. We present the EVcouplings framework, a fully integrated open-source application and Python package for coevolutionary analysis. The framework enables generation of sequence alignments, calculation and evaluation of evolutionary couplings (ECs), and de novo prediction of structure and mutation effects. The combination of an easy to use, flexible command line interface and an underlying modular Python package makes the full power of coevolutionary analyses available to entry-level and advanced users.

Availability and implementation: https://github.com/debbiemarkslab/evcouplings
Contact: chris@sanderlab.org or debbie@hms.harvard.edu

1 Introduction

Coevolutionary sequence analysis presents a promising new approach to the long-standing problem of de novo prediction of the 3D structure of proteins and RNAs. In this approach, pairwise graphical models are used to identify evolutionary couplings (ECs) between sites, which frequently correspond to physical contacts in the molecule’s 3D structure. ECs have been used to successfully predict the residue contacts (Balakrishnan et al., 2011; Ekeberg et al., 2013; Marks et al., 2011; Morcos et al., 2011) and full 3D structure of proteins (Hopf et al., 2012; Marks et al., 2011; Ovchinnikov et al., 2015), RNAs (Weinreb et al., 2016), complexes (Hopf et al., 2014; Ovchinnikov et al., 2014; Weigt et al., 2009), as well as effects of mutations (Figliuzzi et al., 2015; Hopf et al., 2017). However, these applications require integrating multiple tools, data sources and extensive data processing. Available software in this field provides high-performance reimplementations of EC inference tools (Kaján et al., 2014; Seemayer et al., 2014; Weinreb et al., 2016), integration of multiple signals to improve prediction
accuracy (Jones et al., 2015; Skwark et al., 2014), and a library targeted at format conversion between the different approaches (Simkovic et al., 2017). To make these methods accessible to a general biological audience, we present a flexible, open source application and Python package for end-to-end evolutionary coupling analysis. EVcouplings, making use of external tools, covers all necessary functionality, including alignment generation, EC calculation, de novo structure and mutation effect prediction, visualization of results, and comparison of predictions to experimental structures.

2 EVcouplings framework

The EVcouplings framework integrates the functionality of the previously published methods EVMfold (Hopf et al., 2012; Marks et al., 2011), EVcomplex (Hopf et al., 2014) and EVmutation (Hopf et al., 2017). It provides (i) an easy-to-use command-line application and (ii) a modular Python package containing all functions, data structures and pipelines that comprise the application.

Command-line application: The command-line application allows users to obtain predictions for their proteins and complexes of interest by running the respective EVcouplings pipelines (Fig. 1). Each pipeline is comprised of a series of modular stages that can be configured using a YAML file, which aids reproducibility by documenting all parameters. The pipelines are parallelized and support local multi-process execution as well as commonly used cluster systems, and automatically handles job submission and monitoring. The steps of the prediction pipelines are: align, which generates and processes sequence alignments, concatenate, which pairs putatively interacting sequences for the protein complex pipeline, couplings, which calculates ECs, compare, which compares ECs to experimental structures, mutate, which predicts the effects of mutations, and fold, which generates de novo 3D models.

EVcouplings Python package: The command-line application is built on the underlying evcouplings Python package, whose modular architecture and comprehensive documentation facilitate the development of new stages and pipelines. Additionally, the package serves as a toolbox for handling and analyzing EC-related data. Examples for interactive usage are provided in Jupyter notebooks (Kluyver et al., 2016) distributed with the package, and extensive documentation is available on the web (http://evcouplings.readthedocs.io).

3 Conclusion

EVcouplings is an open source, integrated pipeline for evolutionary couplings analyses. The underlying API serves as a modular basis for data analysis and will allow developers to rapidly create new workflows.

Acknowledgements

We thank the members of Marks and Sander labs for helpful comments and testing, and HMS Research Computing for computational resources and support.

Funding

This work has been supported by NSF GRFP DGE1144152 (AGG), DOE CSGF fellowship DE-FG02-97ER25308 (AJR), Pathway Commons U41 HG006623, NRNB P41 GM103504 and R01 GM106303.

Conflict of Interest: none declared.

References


