Title: Selectoscope: A Modern Web-App for Positive Selection Analysis of Genomic Data

Authors: Zaika, AV; Davydov, II ; Gelfand, MS [1,2]

Journal: BIOINFORMATICS RESEARCH AND APPLICATIONS

Year: 2016

Volume: 9683

Pages: 253-257

DOI: 10.1007/978-3-319-38782-6_21

Selectoscope: a modern web-app for positive selection analysis of genomic data.

Andrey V. Zaika[1], Iakov I. Davydov[3,4], and Mikhail S. Gelfand[1,2]

[1] A.A. Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia
[2] Faculty of Bioengineering and Bioinformatics, Lomonosov Moscow State University, Moscow, Russia
[3] Department of Ecology and Evolution, University of Lausanne, Switzerland,
[4] Swiss Institute of Bioinformatics, Switzerland

Abstract. Selectoscope is a web application which combines a number of popular tools used to infer positive selection in an easy to use pipeline. A set of homologous DNA sequences to be analyzed and evaluated are submitted to the server by uploading protein-coding gene sequences in the FASTA format. The sequences are aligned and a phylogenetic tree is constructed. The `codeml` procedure from the `PAML` package is used first to adjust branch lengths and to find a starting point for the likelihood maximization, then `FastCodeML` is executed. Upon completion, branches and positions under positive selection are visualized simultaneously on the tree and alignment viewers. Run logs are accessible through the web interface. Selectoscope is based on the `Docker` virtualization technology. This makes the application easy to install with a negligible performance overhead. The application is highly scalable and can be used on a single PC or on a large high performance clusters. The source code is freely available at https://github.com/anzaika/selectoscope.

Keywords: positive selection, codeml, fastcodeml

1 Introduction

Positive selection is the major force standing behind the innovation during the evolution. Methods based on the ratio of non-synonymous (dN) to synonymous (dS) mutations allow the detection of ancient positive selection in protein-coding genes. A number of methods have been developed over the years, they differ in their data partitioning approach, power, and computational performance [3–6] and many others.

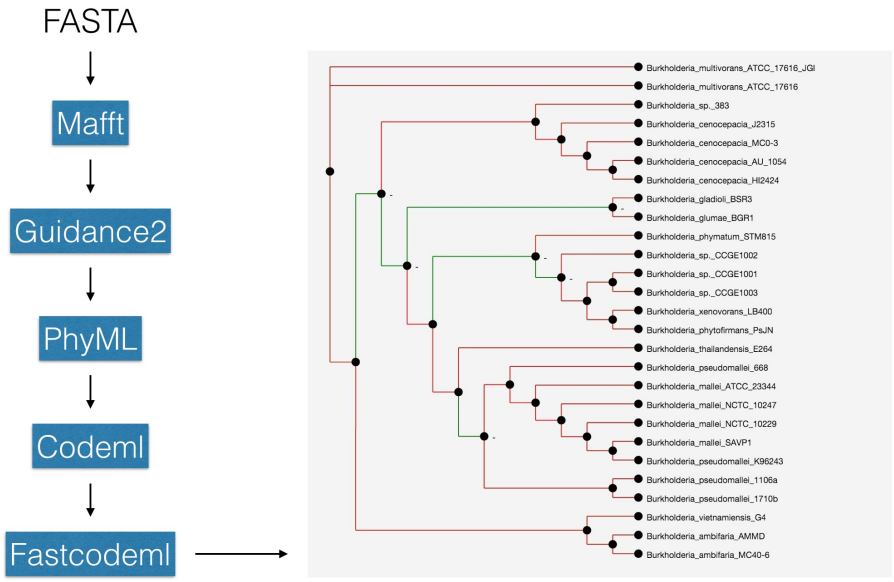
Not only the method choice has a strong impact on the results, but also the intermediate stages of data processing. It has been shown that positive selection codon models are sensitive to the quality of multiple sequence alignments [7, 8] and the gene tree [9]. All this makes detection of positive selection on a genomic scale not only computationally intensive, but also difficult to perform properly.

We created a web application that does not require prior experience with tools for positive selection analysis, while allowing more experienced users to

reconfigure parts of the pipeline without writing any code and to share easily their successful designs with colleagues.

The pipeline is based on the branch-site model of positive selection, as this model is widely used, highly sensitive, and straightforward to interpret [10]. We use a fast implementation of the method in `FastCodeML` [1].

2 Implementation



One starts with uploading FASTA files containing nucleotide sequences of orthologous protein-coding genes. The files can be uploaded one by one or in batches of up to hundreds.

The sequences are aligned with `mafft` [13], and low-quality regions are filtered out using `guidance2` [11].

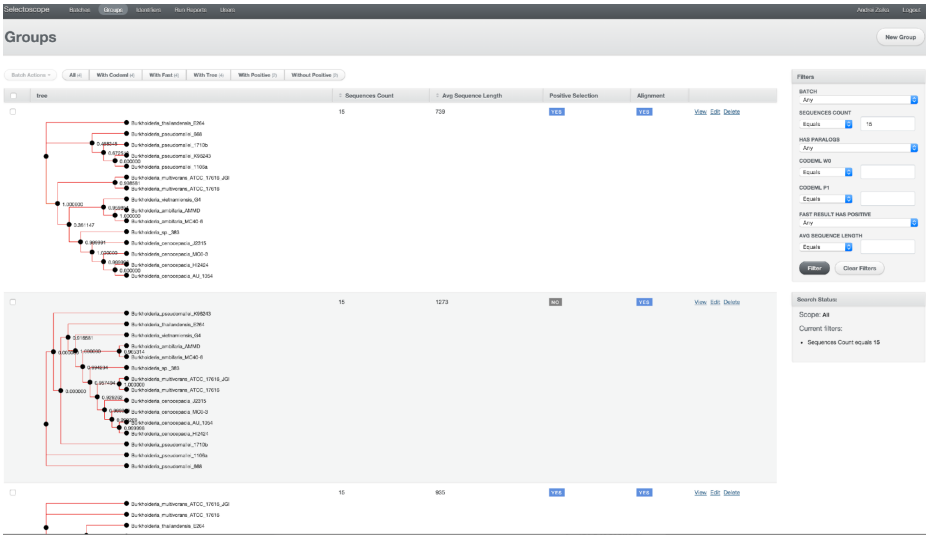
The pipeline includes `phyml` [12] to construct multiple phylogenetic trees. By default a tree is constructed for every orthologous group independently, but it is also possible to create a single tree based on the concatenated alignment. This approach should provide a better tree in the case of homogeneous substitution rates over the genes and in the absense of horizontal gene transfers.

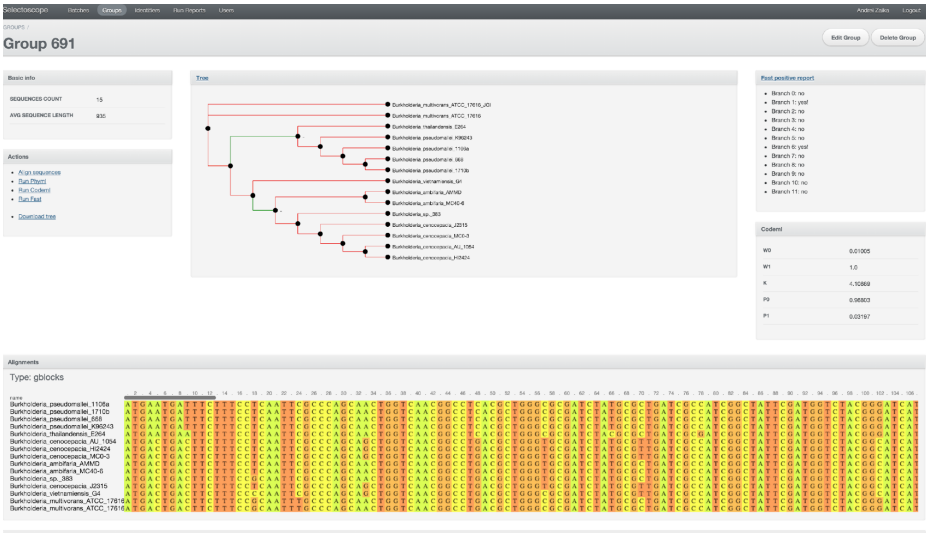
We use `codeml` with model M1a [14] to refine the branch lengths and estimate the transition to transversion ratio (κ).

`FastCodeML` uses the branch-site model [10] to infer branches and sites under positive selection. The phylogenetic tree estimated during the `codeml` run is used. By default the branch lengths are not optimized during the `FastCodeML` for the sake of computational performance.

After the computations are finished, the phylogenetic tree is displayed. All branches, for which positive selection has been detected, are highlighted in the tree, and positions under positive selection are highlighted in the alignment viewer.

Our application uses pure JavaScript libraries provided by BioJS [15] project for displaying the alignments and phylogenetic trees. Thus it does not require any installation procedures on the user's computer.





3 Tools

The core of the pipeline is `FastCodeML`, an extremely optimized software for detecting positive selection using Yang's branch-site model of positive selection. `FastCodeML` uses highly optimized matrix computation libraries (BLAS, LAPACK), supports multicore (OpenMP) and multihost (MPI) parallelization.

The sequence alignment is performed using `mafft`. Low-quality regions of the multiple sequence alignment are detected using `guidance2` and filtered out according to the threshold (0.93 by default).

To construct the phylogenetic tree we first use `phyml` with the `GTR+Gamma+I` model with four gamma rate site classes. This tree is used as an initial tree for the M1a model. We use `codeml` from the `PAML` package [2], to refine transition/transversion ratio (*kappa*) and branch lengths.

All of the application code runs in the Docker containers. Docker is a virtualisation technology that allows one to run a complete operating system within a container without sacrifices in the computational power as opposed to other virtualization technologies such as Virtualbox. This is achieved using resource isolation features of the Linux kernel - cgroups and kernel namespaces. This allows the processes running in the container to run on the host machine without being able to interact with host processes in any way. Docker provides the following advantages:

- Easy dependency management. No need to install or configure any part of the pipeline. Docker is available for all major operating system and its installation process is described in detail on its website [16].
- The container can be launched both as a web frontend server or as a worker that runs parts of the pipeline. This provides a considerable flexibility in launching the application on a single workstation or on a fleet of machines in a computational cluster.

4 Acknowledgements

This study was supported by the Scientific & Technological Cooperation Program Switzerland-Russia (RFBR grant 16-54-21004 and Swiss National Science Foundation project ZLRZ3_163872).

References

1. Valle M., Schabauer H., Pacher C., Stockinger H., Stamatakis A., Robinson-Rechavi M., Salamin N.: Optimisation strategies for fast detection of positive selection on phylogenetic trees. Bioinformatics 30(8): 1129.1137.
2. Yang Z.: PAML 4: phylogenetic analysis by maximum likelihood. Mol Biol Evol. 2007 Aug;24(8):1586-91
3. Yang Z, Nielsen R, Goldman N, Pedersen AM. Codon-substitution models for heterogeneous selection pressure at amino acid sites. Genetics. 2000 May;155(1):431-49.

4. Zhang J, Nielsen R, Yang Z. Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. Mol Biol Evol. 2005 Dec;22(12):2472-9. Epub 2005 Aug 17.
5. Kosakovsky Pond SL, Murrell B, Fourment M, Frost SD, Delport W, Scheffler K. A random effects branch-site model for detecting episodic diversifying selection. Mol Biol Evol. 2011 Nov;28(11):3033-43. doi: 10.1093/molbev/msr125. Epub 2011 Jun 13.
6. Murrell B, Wertheim JO, Moola S, Weighill T, Scheffler K, Kosakovsky Pond SL. Detecting individual sites subject to episodic diversifying selection. PLoS Genet. 2012;8(7):e1002764. doi: 10.1371/journal.pgen.1002764. Epub 2012 Jul 12.
7. Redelings B. Erasing errors due to alignment ambiguity when estimating positive selection. Mol Biol Evol. 2014 Aug;31(8):1979-93. doi: 10.1093/molbev/msu174. Epub 2014 May 27.
8. Fletcher W, Yang Z. The effect of insertions, deletions, and alignment errors on the branch-site test of positive selection. Mol Biol Evol. 2010 Oct;27(10):2257-67. doi: 10.1093/molbev/msq115. Epub 2010 May 5.
9. Diekmann Y, Pereira-Leal JB. Gene Tree Affects Inference of Sites Under Selection by the Branch-Site Test of Positive Selection. Evol Bioinform Online. 2016 Jan 18;11(Suppl 2):11-7. doi: 10.4137/EBO.S30902. eCollection 2015.
10. Zhang J, Nielsen R, Yang Z. Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. Mol Biol Evol. 2005 Dec;22(12):2472-9. Epub 2005 Aug 17.
11. Sela I, Ashkenazy H, Katoh K, Pupko T. GUIDANCE2: accurate detection of unreliable alignment regions accounting for the uncertainty of multiple parameters. Nucleic Acids Res. 2015 Jul 1;43(W1):W7-14. doi: 10.1093/nar/gkv318. Epub 2015 Apr 16.
12. Guindon S., Dufayard J.F., Lefort V., Anisimova M., Hordijk W., Gascuel O. New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. Systematic Biology, 59(3):307-21, 2010.
13. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol. 2013 Apr;30(4):772-80. doi: 10.1093/molbev/mst010. Epub 2013 Jan 16.
14. Yang Z, Nielsen R, Goldman N, Pedersen AM. Genetics. 2000 May;155(1):431-49. Codon-substitution models for heterogeneous selection pressure at amino acid sites.
15. BioJS, the leading and open-source JavaScript visualization library for life sciences https://www.biojs.net/
16. Docker installation guide https://docs.docker.com/engine/installation/