MethodsX 9 (2022) 101786

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/mex

Method Article

OBI: A computational tool for the analysis and systematization of the positive selection in proteins

Julián H. Calvento[a], Franco Leonardo Bulgarelli[b], Ana Julia Velez Rueda[a,*]

[a] Departamento de Ciencia y Tecnología, CONICET, Universidad Nacional de Quilmes, Argentina
[b] Mumuki.org, Argentina

A B S T R A C T

There are multiple tools for positive selection analysis, including vaccine design and detection of variants of circulating drug-resistant pathogens in population selection. However, applying these tools to analyze a large number of protein families or as part of a comprehensive phylogenomics pipeline could be challenging. Since many standard bioinformatics tools are only available as executables, integrating them into complex Bioinformatics pipelines may not be possible.

We have developed OBI, an open-source tool aimed to facilitate positive selection analysis on a large scale. It can be used as a stand-alone command-line app that can be easily installed and used as a Conda package.

Some advantages of using OBI are:

- It speeds up the analysis by automating the entire process
- It allows multiple starting points and customization for the analysis
- It allows the retrieval and linkage of structural and evolutive data for a protein through

We hope to provide with OBI a solution for reliably speeding up large-scale protein evolutionary and structural analysis.

A R T I C L E I N F O
Method name: OBI
Keywords: Python library, Proteins evolution, Structural bioinformatics pipeline
Article history: Available online 16 July 2022

* Corresponding author.
E-mail address: avelezrueda@uvq.edu.ar (A.J.V. Rueda).

https://doi.org/10.1016/j.mex.2022.101786

Specifications Table

Subject Area	Bioinformatics
More specific subject area:	3: Biochemistry, Genetics and Molecular Biology 7: Chemistry 8: Computer Science
Method name:	OBI
Name and reference of original method	HyPhy
Resource availability	https://anaconda.org/jcalvento/obi

Method details

Introduction and general background

Despite their robustness, proteins exhibit remarkable evolutionary adaptability, and new functionalities have emerged throughout the history of the planet [1,2]. We now know that new enzyme functions can evolve in a matter of a few decades, as has happened with enzymes that break down synthetic chemicals that first appeared on this planet during the 20th century [3,4], and the alarming evolution of drug resistance. There is evidence that evolution operates by selecting functional dynamic movements or restricting structural movements that are detrimental to protein function [5,6]. Selection processes then allow for a better adaptation of organisms to their environment. Therefore, identifying sites of a protein subject to positive selection can enrich studies of evolutionary biology and functional characterization.

Positive selection analysis is a bioinformatic prediction technique with multiple applications, including, for example, vaccine design or the detection of new drug-resistant pathogenic variants [7,8]. However, efficient detection of positive selection could be problematic since selection often operates on only a few sites in a short evolutionary time frame [9–11]. Consequently, choosing the appropriate method for its detection and making the correct interpretations for its results is critical.

Here we present OBI, a tool that integrates several bioinformatics tools, optimized for making evolutionary inferences and positive selection analysis. In addition, OBI maps such sequential information to the protein structure. By just receiving a protein's FASTA sequence, our tool retrieves the homologous proteins [12], and gene sequences using Entrez [13] and performs the positive selection analysis using Hyphy [14]. Furthermore, OBI links the evolutionary information with the structural data available for the protein of interest, allowing the user to easily detect positive selection cases related to structural changes and their possible association with the activity and function of proteins.

An extra complication could be applying these analyses on a large scale, for a big number of protein families, or as part of a bigger pipeline. This kind of analysis requires automation and optimization in computing speed and interoperability between technological tools, which makes it hard to achieve. OBI is an open-source tool that facilitates the analysis of positive selection on a large scale. We have implemented a stand-alone command-line app, developed entirely in Python, that can be easily installed and used as a Conda[1] package.

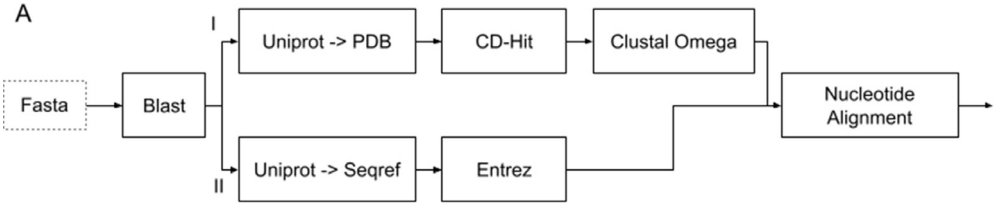
Package structure and user interface

OBI presents a pipeline architecture [15], in which a protein sequence is processed hierarchically until reaching a positive selection analysis report. In each stage, in-house developed utilities are combined with frequently used Python bioinformatics tools such as Biopython, Blast [16], or Uniprot [17].

The whole pipeline can be run through a command-line interface, which allows the specification of analysis parameters such as the min-coverage for getting the targets or the e-value used for filtering the hits obtained (Fig. 1A). All the parameters information and their usage can be accessed by running the *obi –help* option (Fig. 1B).

[1] OBI Conda Package: https://anaconda.org/jcalvento/obi.

A



B

```
(obi) obi@obi:~$ obi -help

There are the available commands, each one has its own parameters. Use obi <command> --help to learn more about the usage

obi fetch-db              Downloads Swissprot DB and Uniprot <-> PDB mappings from NCBI FTP server
obi run [<args>]          Runs the whole analysis or first part depending on the arguments
obi analysis [<args>]     Runs positive selection analysis, which includes running Hyphy and getting final results
obi resume-analysis [<args>]  Runs the whole analysis or first part depending on the arguments
```

C

```
(obi) obi@obi:~$ obi run -help

usage: main.py [-h] [--fasta FASTA] [--email EMAIL] [--output-path OUTPUT_PATH] [--blast BLAST] [--min-identity MIN_IDENTITY] [--max-evalue MAX_EVALUE] [--min-coverage
MIN_COVERAGE] [--max-gaps MAX_GAPS] [--include-analysis]
            [--mode MODE]

optional arguments:
  -h, --help          show this help message and exit
  --fasta FASTA       Path of the Fasta file containing the problem sequence
  --email EMAIL       Email to register while using Entrez
  --output-path OUTPUT_PATH
                      Directory where the program creates output files. Default: ./results/<fasta-filename>
  --blast BLAST       Path to Blast Swissprot DB. Default: ./swissprot/swissprot
  --min-identity MIN_IDENTITY
                      Minimum identity percentage for Blast query expressed as float. Default: 0.4
  --max-evalue MAX_EVALUE
                      Max evalue permitted for Blast results. Default: 0.005
  --min-coverage MIN_COVERAGE
                      Min coverage permitted for Blast results. Default: 90
  --max-gaps MAX_GAPS   Maximum number of gaps permitted for Blast results. Default: 6
  --include-analysis    If present, runs positive selection analysis, which includes running Hyphy and getting final results. If not present, result includes until nucleotide
alignment
  --mode MODE         Only used when --include-analysis is present. "local" if you want to run Hyphy locally, "remote" to use Datamonkey instead. Default: local
```

Fig. 1. A. The data preparation pipeline flow includes I) the homologous proteins search using BLAST, the sequences clustering using CD-Hit, and the sequences alignment using Clustal; II) finally produce a nucleotide alignment guided by the amino acid alignment; B. Obi command general information and usage can be accessed by using *–help* flag; C. Obi provides different running configurations, that allow the users to customize the pipeline running according to their preferences.

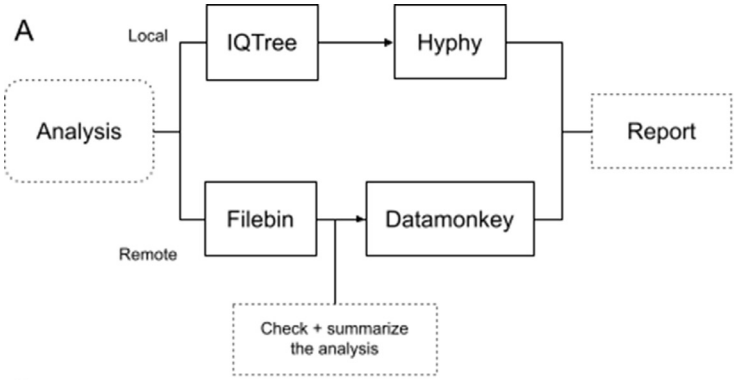
Data preparation

OBI exposes several configurable entry points, and also its usage in different contexts, and users can make a manual curation of data if needed. When running the complete pipeline, the query sequence introduced by the user is fully processed in three steps: data preparation, positive selection analysis and an output with the information necessary for the positive selection analysis is obtained.

In the initial stage, the software retrieves the homologous proteins for the query sequence provided by the user in a FASTA file using the Python implementation of BLAST[2] [18]. These results can be filtered by the user preferences, to obtain the most appropriate construction of the sequence alignment necessary to obtain reliable results in evolutionary inferences (see Fig. 1C) [19,20].

After retrieving the homologous sequences, they are clustered to reduce redundancy and improve the performance of the following steps [21] (see Fig. 1A). For this step, the CDHIT algorithm [22] is used. The outputs generated in this step include a FASTA file with the query protein and the non-redundant homologous sequences, which are subsequently aligned using CLUSTAL-Omega (or ClustalO) [23] for feeding the evolutionary reconstruction software.

[2] Bio Blast Package: https://biopython.org/docs/1.75/api/Bio.Blast.html.



B

```
(obi) obi@obi:~$ obi analysis –help

usage: resume_analysis.py [-h] [--input-path INPUT_PATH]

optional arguments:
  -h, --help            show this help message and exit
  --input-path INPUT_PATH
                        Path of alignment preparation result
```

C

```
(obi) obi@obi:~$ obi resume –help

usage: resume [-h] [--input-path INPUT_PATH]

optional arguments:
  -h, --help            show this help message and exit
  --input-path INPUT_PATH
                        Path of alignment preparation result
```

Fig. 2. A. Alternative Local or remote positive selection analysis flow: OBI provides users with two different strategies for running the positive selection analysis; B. Positive selection analysis can be run separately from the alignments construction and homolog proteins retrieving, by using the first step output files; C. When running Obi in the remote mode, the analysis can be resumed with the resume command.

In this step, OBI also retrieves the coding gene sequences for all the homologous proteins through ENTREZ [13]. This database provides the linkage between the gene-oriented and genome information and the protein information. From the information provided by Entrez and guided by the proteins alignment previously obtained, OBI builds an equivalent nucleotide alignment of the coding regions to feed the evolutionary reconstruction software in the following step.

When alignments curation is required, users can omit the *–include-analysis* parameter, so the positive selection analysis won't be executed. After the first step's output manual curation, users will be able to resume the pipeline, using this stage outputs after the manual revision, by running the analysis command (Fig. 2B).

Positive selection analysis

The OBI pipeline provides two alternative execution workflows for the positive selection analysis (Fig. 2A). When executing the pipeline locally, extra installations are required, which OBI solves for the

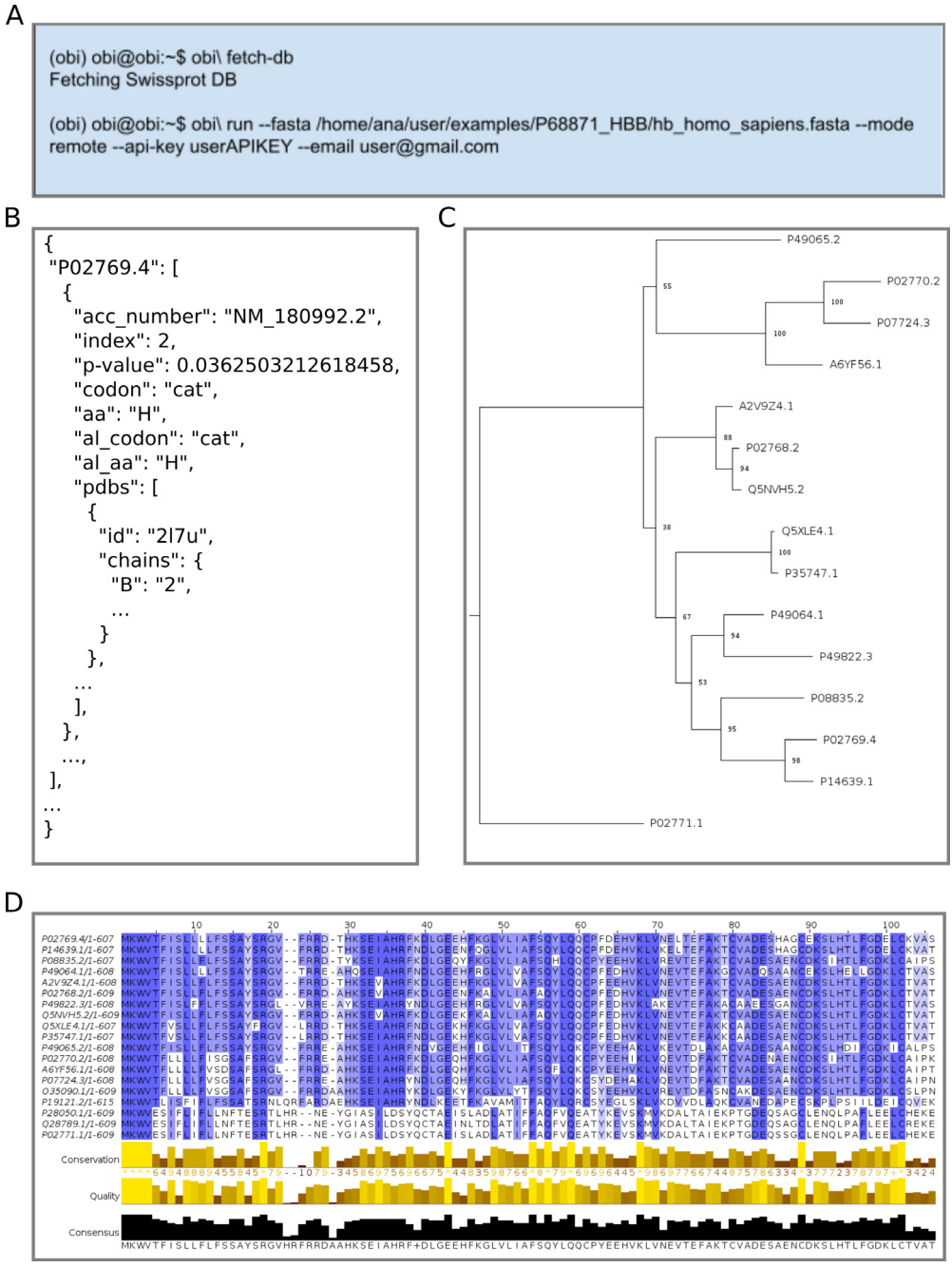


Fig. 3. Report and deliverables example: A. OBI's command-line interface running analysis command; B. JSON report file content example; C. Phylogenetic tree generated by OBI example; D. Protein sequences alignment example.

users during its initial setup. It implements the phylogenetic inference by using the IQTree [24,25] software, which finds the best maximum likelihood tree [26] guided by a heuristics search. This phylogenetic tree serves as input for positive selection analysis with HyPhy [14]. In particular, the OBI uses the MEME method [27], which is a computational technique aimed to identify instances of episodic and pervasive positive selection at the level of an individual site. It has been shown to have superior performance over other models under a broad range of scenarios [28–30].

OBI also offers users the ability to run HyPhy remotely by using the Datamonkey API REST [31]. As an initial result of remote execution, the response from the server is persisted to a *datamonkey_response.json* file within the output directory, so that the user can resume the work by using the *resume* command (Fig. 2C). By running this command, OBI will find the answer previously saved and consult the status of the analysis at the HyPhy server and, in case it has finished, get the results to continue with the rest of the pipeline execution.

Report and deliverables

Our tool allows the user to obtain all its intermediate results individually. Both the proteins and nucleic acids alignments are provided as deliverables, to make the analysis reproducible. The evolutionary analysis generates multiple deliverables such as the blast search result and the phylogenetic trees.

With the positive selection analysis results, OBI generates a report summarizing the results obtained. This report contains for each codon its gene sequence id, the codon's sequence, the position in the codon's alignment, positive selection analysis p-value, protein's corresponding amino acid for this codon, proteins' alignment position, and protein's related PDB information. The structural information of each analyzed protein is automatically mapped to the protein sequence using SIFTS database [32]. This report is written into the chosen results directory with the name *positive_selection.json* (Fig. 3B). A complete input and output example could be found in the Obi project's repository, as well as the commands to be run for executing the pipeline for the human Hemoglobin protein.

Software distribution

The OBI software is distributed via Conda, an open-source package manager and environment management system commonly used for bioinformatics and research projects. OBI is a multiplatform tool, meaning that can be installed and used in Windows, Linux, and macOS. OBI can be used as a stand-alone tool for automated bioinformatic analysis, which may be useful for users without coding skills. Alternatively, it can be also used as a Python library, to allow easy integration into other bioinformatics pipelines. Also, this is a key aspect of OBI, since many standard bioinformatics tools - such as HyPhy and BLAST - are only available as executables, thus reducing interoperability.

The OBI project is open to contributions and thus can be downloaded and installed from the code source on GitHub (https://github.com/jcalvento/obi).

Conclusion

Here we presented OBI, an open-source tool built-in Python, that aims to ease the protein's positive selection analysis. It provides a starting point for several specific pipelines and future works. It is an open-source code tool that can be easily merged in Bioinformatics pipelines as a Conda package or even to an initial source to be adapted.

Our software allows not only the full analysis for a query protein but also a user-customized analysis with different entry points. The OBI software automatically retrieves all the homologous sequences for the analysis and maps the positions under positive selection to all the PDB structures available for the query protein. The high-level approach for retrieving the structural and evolutive data for a protein through OBI facilitates its application to large-scale analysis.

Our tools present significant contributions to bioinformatics since it solves a problem of great interest to the field, by applying software architecture techniques that maximize robustness and

flexibility. We hope to provide with OBI a tool that reliably speeds up the evolutionary and structural analysis of proteins on a large scale.

Declaration of Competing Interest

The authors have no conflicts of interest to declare.

Data Availability

No data was used for the research described in the article.

Fundings

AJVR is a Postdoctoral fellow from CONICET. This work was supported by Universidad Nacional de Quilmes (PUNQ 1004/11), ANPCyT (PICT-2014-3430, PICT-2013-0232). The funders had no role in the study design, data collection, analysis, decision to publish, or preparation of the manuscript.

Authors contributions

AJVR supervised the work and was in charge of conceptualization, and project administration. JHC carried out the software development and FLB made the technological supervision. AJVR and FLB did the manuscript writing with input from all authors.

References

[1] A.L. Hughes, The evolution of functionally novel proteins after gene duplication, Proc. Biol. Sci. 256 (1346) (1994) 119–124.
[2] A. Aharoni, L. Gaidukov, O. Khersonsky, S. McQ Gould, C. Roodveldt, D.S. Tawfik, The “evolvability” of promiscuous protein functions, Nat. Genet. 37 (1) (2005) 73–76.
[3] A. Fernández, D.S. Tawfik, B. Berkhout, R. Sanders, A. Kloczkowski, T. Sen, et al., Protein promiscuity: drug resistance and native functions–HIV-1 case, J. Biomol. Struct. Dyn. 22 (6) (2005) 615–624.
[4] T. Zou, V.A. Risso, J.A. Gavira, J.M. Sanchez-Ruiz, S.B. Ozkan, Evolution of conformational dynamics determines the conversion of a promiscuous generalist into a specialist enzyme, Mol. Biol. Evol. 32 (1) (2015) 132–143.
[5] D. Granata, L. Ponzoni, C. Micheletti, V. Carnevale, Patterns of coevolving amino acids unveil structural and dynamical domains, Proc. Natl. Acad. Sci. USA. 114 (50) (2017) E10612–E10621.
[6] J. Marchetti, A.M. Monzon, S.C.E. Tosatto, G. Parisi, M.S. Fornasari, Ensembles from ordered and disordered proteins reveal similar structural constraints during evolution, J. Mol. Biol. 431 (6) (2019) 1298–1307.
[7] V. Duvvuri, B. Duvvuri, W.R. Cuff, G.E. Wu, J. Wu, Role of positive selection pressure on the evolution of H5N1 Hemagglutinin, Genom. Proteom. Bioinform. 7 (1–2) (2009) 47–56.
[8] L. Chen, A. Perlina, C.J. Lee, Positive selection detection in 40,000 human immunodeficiency virus (HIV) type 1 sequences automatically identifies drug resistance and positive fitness mutations in HIV protease and reverse transcriptase, J. Virol. 78 (7) (2004) 3722–3732.
[9] J. Zhang, Frequent false detection of positive selection by the likelihood method with branch-site models, Mol. Biol. Evol. 21 (7) (2004) 1332–1339.
[10] S. Yokoyama, T. Tada, H. Zhang, L. Britt, Elucidation of phenotypic adaptations: Molecular analyses of dim-light vision proteins in vertebrates, Proc. Natl. Acad. Sci. U.S.A. 105 (36) (2008) 13480–13485.
[11] J. Chen, Y. Sun, Variation in the analysis of positively selected sites using nonsynonymous/synonymous rate ratios: an example using influenza virus, PLoS One 6 (5) (2011) e19996.
[12] G.S. Chang, Y. Hong, K.D. Ko, G. Bhardwaj, E.C. Holmes, R.L. Patterson, et al., Phylogenetic profiles reveal evolutionary relationships within the “twilight zone” of sequence similarity, Proc. Natl. Acad. Sci. U.S.A. 105 (36) (2008) 13474–13479.
[13] D. Maglott, J. Ostell, K.D. Pruitt, T. Tatusova, Entrez Gene: gene-centered information at NCBI, Nucleic. Acids. Res. 33 (Database issue) (2005) D54–D58.
[14] S.L.K. Pond, S.D.W. Frost, S.V. Muse, HyPhy: hypothesis testing using phylogenies, Bioinformatics 21 (5) (2005) 676–679.
[15] G. Hohpe, B. Woolf, Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, 1st ed., Addison-Wesley Professional, Boston, 2003.
[16] C.E. Jones, U. Baumann, A.L. Brown, Automated methods of predicting the function of biological sequences using GO and BLAST, BMC Bioinform. 6 (2005) 272.
[17] E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, P. Bansal, A.J. Bridge, et al., UniProtKB/Swiss-prot, the manually annotated section of the uniprot knowledgebase: how to use the entry view, Methods Mol. Biol. 1374 (2016) 23–54.
[18] M. Johnson, I. Zaretskaya, Y. Raytselis, Y. Merezhuk, S. McGinnis, T.L. Madden, NCBI BLAST: a better web interface, Nucleic. Acids. Res. 36 (Web Server issue) (2008) W5–W9.
[19] E.V. Koonin, M.Y. Galperin, in: Principles and methods of sequence analysis. Sequence — evolution — function, Springer US, Boston, MA, 2003, pp. 111–192.

[20] A. Di Franco, R. Poujol, D. Baurain, H. Philippe, Evaluating the usefulness of alignment filtering methods to reduce the impact of errors on evolutionary inferences, BMC Evol. Biol. 19 (1) (2019) 21.
[21] K. Sikic, O. Carugo, Protein sequence redundancy reduction: comparison of various method, Bioinformation 5 (6) (2010) 234–239.
[22] Y. Huang, B. Niu, Y. Gao, L. Fu, W. Li, CD-HIT Suite: a web server for clustering and comparing biological sequences, Bioinformatics 26 (5) (2010) 680–682.
[23] F. Sievers, D.G. Higgins, Clustal Omega for making accurate alignments of many protein sequences, Protein Sci. 27 (1) (2018) 135–145.
[24] J. Trifinopoulos, L.-T. Nguyen, A. von Haeseler, B.Q. Minh, W-IQ-TREE: a fast online phylogenetic tool for maximum likelihood analysis, Nucleic. Acids. Res. 44 (W1) (2016) W232–W235.
[25] B.Q. Minh, H.A. Schmidt, O. Chernomor, D. Schrempf, M.D. Woodhams, A. von Haeseler, et al., IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era, Mol. Biol. Evol. 37 (5) (2020) 1530–1534.
[26] S.H. Jacobson, E. Yücesan, Analyzing the performance of generalized hill climbing algorithms, J. Heuristics 10 (4) (2004) 387–405.
[27] B. Murrell, J.O. Wertheim, S. Moola, T. Weighill, K. Scheffler, S.L. Kosakovsky Pond, Detecting individual sites subject to episodic diversifying selection, PLos Genet. 8 (7) (2012) e1002764.
[28] J.D. Bloom, Identification of positive selection in genes is greatly improved by using experimentally informed site-specific models, Biol. Direct 12 (1) (2017) 1.
[29] J.M. Moreno, T.F. Jesus, M.M. Coelho, V.C. Sousa, Adaptation and convergence in circadian-related genes in Iberian freshwater fish, BMC Ecol. Evo. 21 (1) (2021) 38.
[30] L. Picard, Q. Ganivet, O. Allatif, A. Cimarelli, L. Guéguen, L. Etienne, DGINN, an automated and highly-flexible pipeline for the detection of genetic innovations on protein-coding genes, Nucleic. Acids. Res. 48 (18) (2020) e103.
[31] S. Weaver, S.D. Shank, S.J. Spielman, M. Li, S.V. Muse, S.L. Kosakovsky Pond, Datamonkey 2.0: A modern web application for characterizing selective and other evolutionary processes, Mol. Biol. Evol. 35 (3) (2018) 773–777.
[32] J.M. Dana, A. Gutmanas, N. Tyagi, G. Qi, C. O'Donovan, M. Martin, et al., SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins, Nucleic. Acids. Res. 47 (D1) (2019) D482–D489.