Evolutionary Biology (2020) 47:240–245
https://doi.org/10.1007/s11692-020-09507-9

TOOLS AND TECHNIQUES

Link Your Sites (LYS) Scripts: Automated Search of Protein Structures and Mapping of Sites Under Positive Selection Detected by PAML

Lys Sanz Moreta[1] · **Rute R. da Fonseca**[2]

Received: 14 January 2020 / Accepted: 9 June 2020 / Published online: 30 June 2020

Abstract

The visualization of the molecular context of an amino acid mutation in a protein structure is crucial for the assessment of its functional impact and the understanding of its evolutionary implications. Currently, searches for fast evolving amino acid positions using codon substitution models like those implemented in PAML (Yang and Nielsen in Estimating synonymous and nonsynonymous substitution rates under realistic evolutionary models. Mol Biol Evol 17(1):32–43, 2000; Zhang et al. in Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. Mol Biol Evol 22(12):2472–2479, 2005) are done in almost complete proteomes, generating large numbers of candidate proteins making the analysis of individual protein structures and models very time-consuming. Here we present the package Link Your Sites (LYS) that can be used to reduce the number of analysed targets to those for which structural information can be retrieved. LYS consists of two python wrapper scripts, where the first one (i) mines the RCSB database (Berman et al. in The protein data bank. Nucleic Acids Res 28(1):235–242, 2000) using the BLAST alignment tool to find the best matching homologous sequences, (ii) fetches their domain positions by using Prosites (Hamelryck and Manderick in Pdb file parser and structure class implemented in python. Bioinformatics 19(17):2308–2310, 2003; Sigrist et al. in Prosite: a documented database using patterns and profiles as motif descriptors. Brief Bioinf 3(3):265–274, 2002; Sigrist et al. in New and continuing developments at prosite. Nucleic Acids Res 41(D1):D344–D347, 2012), (iii) parses the output of PAML extracting the positional information of fast-evolving sites and transforms them into the coordinate system of the protein structure, (iv) outputs one file per gene with the equivalence among the positions in the input sequence and homologous structure. The second script produces figures to be used in publications highlighting the positively selected sites mapped on regions that are known to have functional relevance.

– *Motivation* Automatizing the search for protein structures to assess the functional impact of sites found to be under positive selection by codeml, implemented in PAML (Yang and Nielsen in Estimating synonymous and nonsynonymous substitution rates under realistic evolutionary models. Mol Biol Evol 17(1):32–43, 2000). Building publication-quality figures highlighting the sites on a protein structure model that are within and outside functional domains. Reduces the workload associated with selecting proteins for which a functional assessment of the impact of substitutions can be done using a protein structure. This is especially relevant when analyzing almost complete proteomes which is the case of large comparative genomic studies.
– *Software* LYS scripts are executed in the command line. They automatically search for homologous proteins at the RSCB database (Nielsen in Molecular signatures of natural selection. Annu Rev Genet 39:197–218, 2005), determine the functional domain locations and correlate the positions pointed by the M8 model (Yang and Nielsen in Estimating synonymous and nonsynonymous substitution rates under realistic evolutionary models. Mol Biol Evol 17(1):32–43, 2000), and output a data frame that can be used as the input by PyMOL (Schrodinger in The pymol molecular graphics system. *Version 1* in 2010) to generate a visualization of the results.
– *Availability* LYS is easy to install and implement and they are available at https://github.com/LysSanzMoreta/LYS_Automatic_Search.

Keywords Functional domain · Positive selection · BLAST · PDB · Codeml · Homologous proteins · Prosites · PyMOL

Extended author information available on the last page of the article

Introduction

One of the goals in comparative genomics studies is to find regions of the genomes that evolve at elevated rates, which can potentially indicate that they involved in promoting adaptation to new environments. Such regions are said to be evolving under positive selection (Nielsen 2005). It is possible to infer positive selection occurring in individual protein sequences by assessing the rates of substitutions at specific codons (sets of three nucleotides that correspond to an amino acid) thanks to site models such as those implemented in PAML (Yang and Nielsen 2000; Yang 2007). Positive selection is detected using the value that corresponds to the ratio between the amount of non-synonymous substitutions per non-synonymous site and the amount of synonymous substitutions per synonymous site. Non-synonymous substitutions can be relevant if the amino acid switch introduced generates a change in the physicochemical properties of the residue and consequently affects the protein function. A first step in the evaluation of the impact of these substitutions consists on identifying their location on a protein structure (which could be the structure of a closely related homologous protein) and verify whether they are located within known functional domains. In a protein structure the amino acids form a backbone that is folded into a specific conformation, with the folding patterns being dictated by a series of non-covalent bonds (hydrogen bonds, ionic bonds and van der Waals attractions) directed by the residue's side chains. If the residues in the functional domain are exchanged with an amino acid with different properties, these interactions will be modified together with the structure and its binding attributes will be affected (Alberts et al. 2002). Substitutions in the functional domain are more likely to affect the protein's function when compared to those located in other parts of the structure. In order to easily assess which proteins in a large selection scan can be analyzed at the structural level, we present a Python wrapper that reads a file containing the sequences to analyze and the paths to the output files from M8 codeml model, and performs an automatic search of homologous proteins by BLASTing the query sequences to the RCSB database (Nielsen 2005). The results from BLAST are ranked according to the percentage of identity, the coverage and finally, resolution of the crystallographic protein information file. The selected PDB files are further analyzed via the Prosites (Sigrist et al. 2002, 2012) software implemented in Biopython (Hamelryck and Manderick 2003) to find the domains positions. Next, the positions correspondence algorithm is implemented among the query sequence and the homologous protein sequence. Position correspondence refers to the equivalent position among the input sequence and the best protein match. This correspondence is outputted as a data frame that is then used in a second script to create the visualization of the protein structure with highlighted functional domains and positively selected sites in PyMOL (Schrodinger 2010).

Methods

Design of the Algorithm to Perform the Positions Correspondence

The main algorithm finds the correspondent positions among the query gene sequence and the crystallography file sequences. These are the main steps (see also Fig. 3) followed in the script:

1. Creation of two lists: (i) list A containing the positions in the alignment (Biopython's (Schrodinger 2010; Cock et al. 2009) global alignment) where there are no gaps in any of the sequences and (ii) list B, which has the length of the gene sequence, filled with 'nan' values.
2. Counting the amount of gaps between each segment, bounded by the i and i + 1 positions contained in list A, in the aligned sequences. This step is performed for both sequences. Two output lists are generated (C and D) with the reciprocal correspondence of the positions where there are not gaps in the alignment of chain A and B.
3. Lists C and D are used to fill in list B with the correspondent positions. Furthermore, the correspondent positions of the gene in the PDB sequence are substituted by the actual residue ID numbers from the PDB file, which follow their own numbering settings (Fig. 1).

Materials

LYS consists of a series of Python version 3 scripts available in a Github repository (https://github.com/LysSanzMoreta/LYS_Automatic_Search) and licensed under an Apache Version 2 License. All of the scripts require the freely available packages of pandas, numpy, pymol and Biopython, whose installation is highly recommended through anaconda version 3. The scripts that call PyMOL (Schrodinger 2010) can be also used freely under educational purposes. A simple video tutorial for the two main scripts is available at https://youtu.be/ZUxUHWfQ9kw (Tables 1, 2).

LYS has been tested on Unix platforms like Ubuntu 18.04. To be able to make use of the scripts that call the PyMOL GUI, make sure that the PyMOL Educational version is the in the command line path. The input files for the main script LYS_PDB_Search.py are, a file containing all the sequences (whose formats can be specified with the flag-format, fasta is default and recommended) and a tab

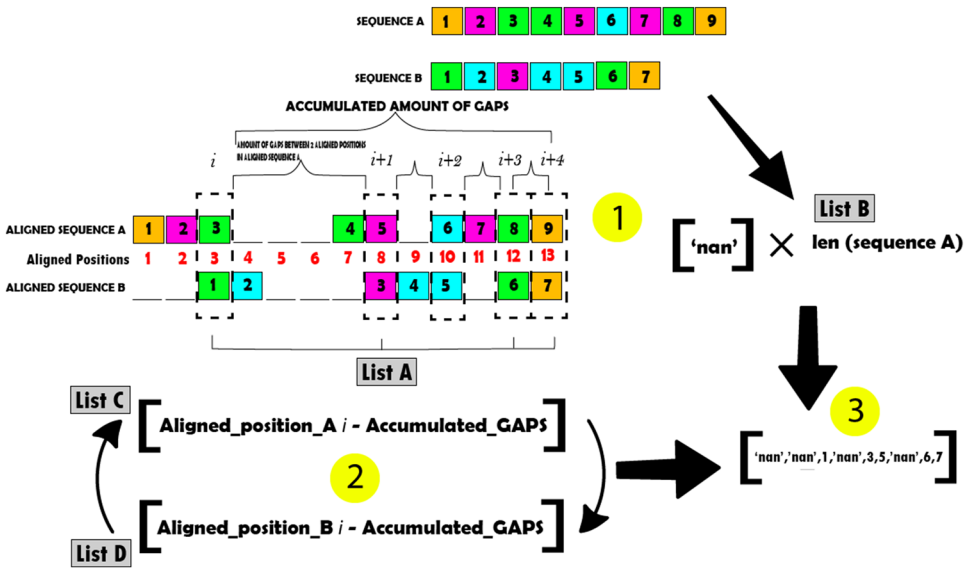

Fig. 1 Graphical explanation of the algorithm that matches the coordinates of 2 sequences by using their unaligned and aligned versions (local or global alignment in Biopython, 2009). The numbers indicate the residues positions in the chain/sequence

Table 1 LYS table output of correspondence among the coordinates/residues of the studied sequences. These dataframes are directed to the Positions_Dataframe folder

Gene_Position	PDB_Position	Label
1	Nan	Not
2	− 1	Domain
3	0	Selected
4	432	Selected_and_Domain

separated file containing rows with the name of the sequence (containing the exact same sequence name as in the first file, for example the Fasta headers) and its path to the codeml M8 output results. The complete list of available arguments is shown in Table 3. The outputs, which will be stored in the Positions_Dataframe folder, are dataframes containing the positions number equivalence among the input sequence and its best matching sequences as seen in Table 3. The script returns a first script, "Full_Blast_results_against_PDB" , where the user can visualize the percent id, coverage and resolution of all the matched proteins."Full_Blast_results_against_PDB_filtered" is also returned and contains only the

Table 2 LYS_PDB_Search.py script list of arguments

Argument	Required	Help	Default value
–Proteins	True	Path to File containing the coding sequences (Recommended: Fasta format)	
–Codeml	True	Path to file containing rows with: "Gene name" + '\ t' + 'Path to codeml M8/bsA1 output file'. Remember: Gene name needs to match the Gene name in the Sequences file	
–format	False	Sequence or Multiple Alignment File Format	Fasta
–prob	False	Choice of level of posterior probability on the sites, 95% or 99% from M8 Out file	99%
–missing_data	False	Decide if the missing data (labeled as 'N') should be kept from the nucleotide sequence. It might affect the final alignment, is recommended to check the alignment scores in both options (activate print_alignment to do so).	Yes
–print_alignment	False	Choose to visualize the PDB file sequence aligned with the gene	No
–number_homologous	False	Select the maximum number of homologous to be analyzed according to their structure resolution, where smaller resolution is better.	3

top 3 matched proteins according to their resolution. This number can be changed with the flag –number_homologous. Alongside a folder where the crystallography protein files are downloaded is created (PDB_files).

Once the data frames have been created navigate to that folder and find, for example through grep -rl "Selected_and_Domain", which ones have determined that the homologous protein displays positively selected residues in the domain. Following, call the LYS_PyMOL_input_Dataframe.py GUI interface, Fig. 3, to plot in a personalized approach the proteins, check for customizable features in Table 3, that display the result of interest, "Testing the Scripts". The list of available scripts is the following:

Main Scripts

– LYS_PDB_Search.py: Performs a BLAST search against RSCB database to find and download the best PDB files for the query sequences. The results are saved to the files "Full_Blast_results_against_PDB.tsv" and the reduced version containing the best scoring results,"Full_Blast_results_against_PDB _Filtered.tsv". This is followed by the generation of a data frame of the correspondent positions among each query sequence and the homologous sequence. Simultaneously these positions are assigned a label that indicates whether: (a) "Domain" they belong to the domain residues (using Prosites (Sigrist et al. 2002, 2012)), "Selected" they are positively selected (given by the codeml (Yang and Nielsen 2000) output), "Selected_and_Domain" both or "Not" none, positions not found to have equivalence and therefore are not highlighted.
– LYS_PyMOL_input_Dataframe.py: Takes the output data frame of LYS_PDB _Search.py and generates a customizable graphic visualization.

Complementary Scripts

– LYS_PyMOL_Prosites.py: Inputs individual sequence and a chosen PDB file, and allows personalized configuration. The domain positions can be assigned using various methods, for example via Prosites (Sigrist et al. 2002, 2012), a list of "\ n" separated positions (referring to the query sequence) or by using the desired Uniprot's

Table 3 LYS_PyMOL_input_Dataframe.py customizable features inside the script or GUI

Settings	Options
Background, Residues and Font Colours	Choose colours from the palette: https://pymolwiki.org/index.php/Color_Values
Residues Shapes (GUI)	Choose from: https://pymolwiki.org/index.php/Show
Select and Remove Chains	Choose if any of the chains should be removed in the visualization
Legend: Font Size and Placement (GUI)	Change the values of the axes, cyl_text and cmd.set
Carbon-alpha residues labelling	Activate cmd.label accordingly: Designed to highlight only alpha carbons of selected sites

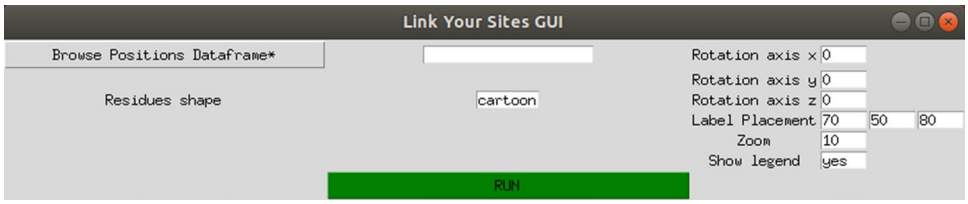

Fig. 2 LYS_PyMOL_input_Dataframe.py's interface. The compulsory files for the GUI to work are marked with a* . Tutorial at https://www.youtube.com/watch?v=8ui1TxpOd6M

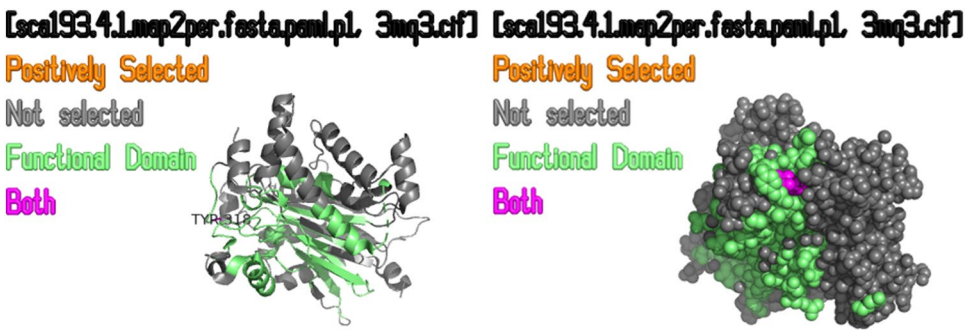

Fig. 3 LYS's visual output examples of protein coloured according to its evolutionary positively selected amino acid residues and domain positions. Cartoon (left) and spheres (right) modes

domain sequences clustered in a fasta file. They will be locally aligned to the PDB file sequence.
– LYS_PyMOL_GUI_Prosites.py: GUI version of LYS_PyMOL_Prosites.py see Fig. 2.

Results

Testing the Scripts

The scripts were tested in a Unix server on 5 protein coding sequences of 438, 244, 183, 122 and 61 amino acids long, which are available at https://github.com/LysSanzMoreta/LYS_Automatic_Search/tree/master/TestSequences, together with their corresponding codelm results. The LYS_PDB_Search.py script running time was measured and the results are 1m56.113s for real, 0m9.880s for user and 0m0.296s in sys times. These sequences contain several types of examples, such as some sequences that do not show homologous proteins, some only show one or several matches in the PDB database and one that contains positively selected residues that are present in the functional domain of the homologous protein (see "Testing the Scripts") (Fig. 3).

Discussion

After detecting regions of the genome under fast evolution, one of the goals of molecular evolution studies is to understand the functional impact of the substitutions in those regions. It is already possible to pinpoint the positions in a certain protein that seem to be evolving at a fast rate, but to infer the impact of a mutation in the protein function in silico it is important to first map it to a protein structure, when available, or an adequate template corresponding to a homologous protein. LYS automates the search for protein structures, depicts them in PyMOL together with the information on known functional domains, and incorporates the information from PAML's M8 output providing a publication-ready representation of the results. It also creates easy to parse tables with all the results, facilitating further analyses of the end user.

Acknowledgements The authors gratefully acknowledge the following for supporting their research: Villum Fonden Young Investigator Grant VKR023446 (R.R.F. and L.S.M.); the Danish National Research Foundation for its support of the Center for Macroecology, Evolution, and Climate—Grant DNRF96 (R.R.F.); Novo Nordisk Foundation grant NNF16OC0023494 (L.S.M.); Programa Operativo de Empleo Juvenil FSE 2104-2020—Grant CCI 2014ES05M9OP001 (L.S.M.).

Author Contributions L.S.M. and R.R.F. designed the study; L.S.M. wrote the software with input from R.R.F.; L.S.M. wrote the manuscript with contributions from R.R.F.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2002). The extracellular matrix of animals. In M. Anderson & S. Granum (Eds.), *Molecular Biology of the Cell* (4th ed.). New York: Garland Science.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., et al. (2000). The protein data bank. *Nucleic Acids Research*, *28*(1), 235–242.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., et al. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423.

Hamelryck, T., & Manderick, B. (2003). Pdb file parser and structure class implemented in python. *Bioinformatics*, *19*(17), 2308–2310.

Nielsen, R. (2005). Molecular signatures of natural selection. *Annual Review of Genetics*, *39*, 197–218.

Schrodinger, L. (2010). The pymol molecular graphics system. *Version 1*.

Sigrist, C. J., Cerutti, L., Hulo, N., Gattiker, A., Falquet, L., Pagni, M., et al. (2002). Prosite: A documented database using patterns and profiles as motif descriptors. *Briefings in Bioinformatics*, *3*(3), 265–274.

Sigrist, C. J., De Castro, E., Cerutti, L., Cuche, B. A., Hulo, N., Bridge, A., et al. (2012). New and continuing developments at prosite. *Nucleic Acids Research*, *41*(D1), D344–D347.

Yang, Z. (2007). Paml 4: Phylogenetic analysis by maximum likelihood. *Molecular Biology and Evolution*, *24*(8), 1586–1591.

Yang, Z., & Nielsen, R. (2000). Estimating synonymous and nonsynonymous substitution rates under realistic evolutionary models. *Molecular Biology and Evolution*, *17*(1), 32–43.

Zhang, J., Nielsen, R., & Yang, Z. (2005). Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. *Molecular Biology and Evolution*, *22*(12), 2472–2479.

Affiliations

Lys Sanz Moreta[1] · **Rute R. da Fonseca**[2]

✉ Lys Sanz Moreta
moreta@di.ku.dk

[1] Computer Science - Image Section, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark

[2] Center for Macroecology, Evolution and Climate (CMEC), GLOBE Institute, University of Copenhagen, 1350 Copenhagen, Denmark