



PyLasso

a PyMOL plugin to identify lassos

Contents

1. INTRODUCTION.....	2
2. INSTALLATION.....	3
2.1. PREREQUISITES	3
2.2. INSTALLATION OF PYLASSO.....	3
3. PROTEIN.....	4
3.1. INPUT FILES	4
3.2. INTERFACE.....	6
3.3. LASSO INFORMATION WINDOW	8
3.4. LASSO VISUALIZATION	9
4. TRAJECTORY	10
4.1. INPUT FILES	10
4.2. INTERFACE.....	12
4.3. TRAJECTORY INFORMATION WINDOW	12
5. ADVANCED OPTIONS.....	15
5.1. SMOOTHING.....	15
5.2. CORRECT MODE	15
5.3. GAUSSIAN LINKING NUMBER.....	16
6. BIBLIOGRAPHY	17

1. Introduction

Entanglement in proteins is an important phenomenon and a subject of multidisciplinary research. Lassos are new entangled motifs discovered recently in proteins [1]. Lassos occur in structures with disulfide (or other) bridges, where at least one terminus of a protein backbone pierces through a surface spanned on a covalent loop (closed by such a bridge). Lassos have been identified in around 4% of known protein structures [2]. The PyLasso plugin enables to detect lassos and to analyze their geometric structure.

The PyLasso is a multiplatform, user-friendly plugin for PyMOL. The PyLasso detects all types of lassos based on techniques developed in [1], which involve analysis of surfaces of minimal area spanned on closed loops, and the number and directions of segments piercing such surfaces. The plugin enables to detect different types of covalently closed loops. A user can also define news loops, either by selecting a part of the backbone forming a loop, or typing sequential numbers of two atoms that form a bridge. Detected lasso configurations, including the minimal surface and piercings, are represented graphically in the PyMOL environment. The plugin is equipped with additional tools that facilitate analysis of lassos, in particular structure smoothing (which simplifies the structure by ignoring e.g. helices and sharp turns) and minimal surface smoothing.

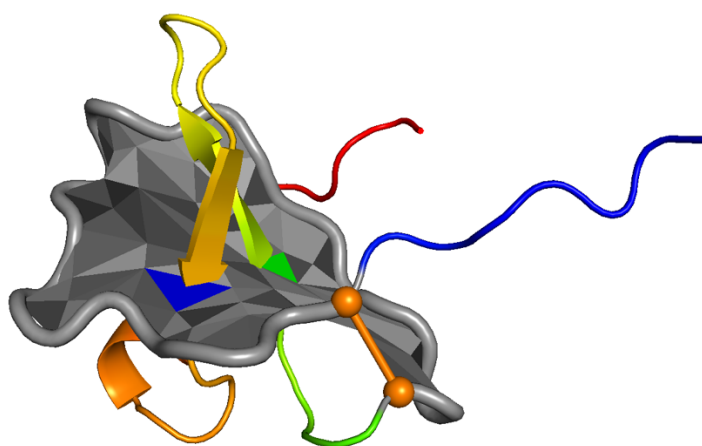


Fig. 1 Crystal structure of viral macrophage inflammatory protein II (PDB code 1cm9, chain A) with a lasso spanned on a closed loop, between atoms of indices 14 and 38.

The PyLasso is also equipped with advanced tools, such as a manual parameterization of the algorithm to detect lassos, as well as an alternative method to detect entanglement by means of the Gaussian Linking Number (GLN), which provides additional information about lasso geometry. The analysis of detected loops involves both single frames and whole trajectories (representing e.g. time evolution). In the trajectory mode two charts are generated, which describe the change in the topology along consecutive frames, and enable detailed analysis of entanglement in a chosen frame (as in the single frame mode).

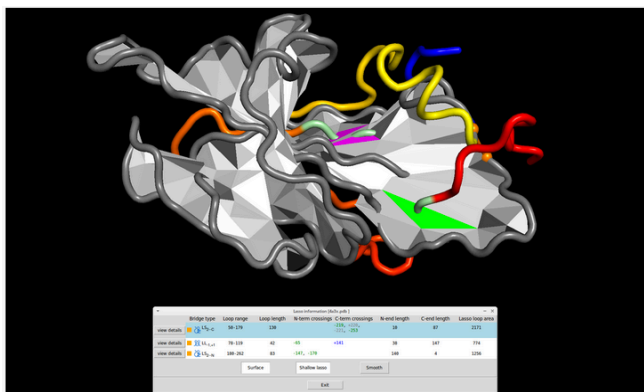


Fig. 2 Structure of oxidoreductase (PDB code 4ax3): a view with a displayed surface of minimal area spanned on a closed loop between atoms of indices 50 and 179.

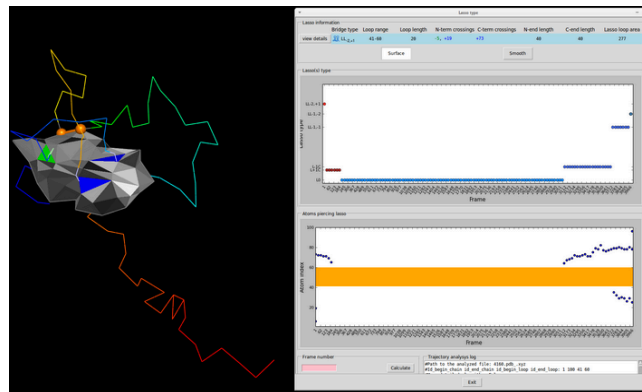


Fig. 3 Trajectory result window involving the visualization of a certain frame and the chart of topology and piercings change during the simulation. For details see trajectory analysis section.

The PyLasso has been developed based on the server <http://lassoprot.cent.uw.edu.pl> [2]. It has been tested and is available on platforms such as Linux, Mac OS and Windows (7/8/10).

2. Installation

2.1. Prerequisites

The PyMOL is a software that enables molecular visualization and modeling in structural biology. It can be downloaded from [PyMOL official site](#) or [SourceForge](#). The PyLasso uses the PyMOL environment to depict lassos and minimal surfaces.

The PyMOL uses scripts written in Python 2.6+ high-level programming language, so it is vital to have it installed. The newest version can be downloaded from [Python official site](#).

Matplotlib is a 2D plotting library, which uses Python. It enables the plugin to produce high-quality charts. The library can be downloaded from [Matplotlib official site](#).

2.2. Installation of PyLasso

After unpacking the compressed file, the plugin can be installed via selecting in PyMOL *Plugin > Plugin Manager > Install New Plugin > Choose file* and then choosing `__init__.py`. Subsequently, an installation directory should be chosen (we recommend installation in the first chosen path). The process of installation is depicted in this video.

On Linux and Mac OS X platforms the mode of `detect_lassos` (a file responsible for calculating the minimal surface) as well as other python files (files with `.py` extension) should be changed. To change the access permissions to it, a user can type the system call as shown below.

```
chmod a+x detect_lassos
chmod a+x *.py
```

Remark – users of Windows shall append a path to python compiler in their environment variables PATH (the simplest way to do it is described [here](#)). Doing so will enable running a python program in the command prompt via typing python there.

The PyLasso is started by selecting tab *Plugin > PyLasso*. Following this a new window, as shown in Fig. 4, is displayed. This window enables a user to pick a file for further operations. A directory containing the file will be the place where all necessary files will be generated, so it is advised to keep a structure in a separate directory. The available types of files acceptable by the PyLasso are PDB (including NMR structures) and XYZ. After proper selection the user should click Open. This allows the interface to adapt to the current file extension (different interfaces are displayed for proteins (single frames) and for trajectories). The file name, including the extension, should contain only letters in lower case.

There can be only one structure loaded and displayed in PyMOL viewer. Also, the user should remember to close each PyLasso session in order to prevent undesirable behavior of the plugin.

Remarks:

- The names of directories where files to be analyzed are stored should not contain spaces – their presence will result in wrong invocation of the program to detect lassos and consequently in the crash of the plugin or in printing wrong data (e.q. “No lasso found in a protein”).
- Be sure that you installed Matplotlib and PyMOL, not a MacPymol.

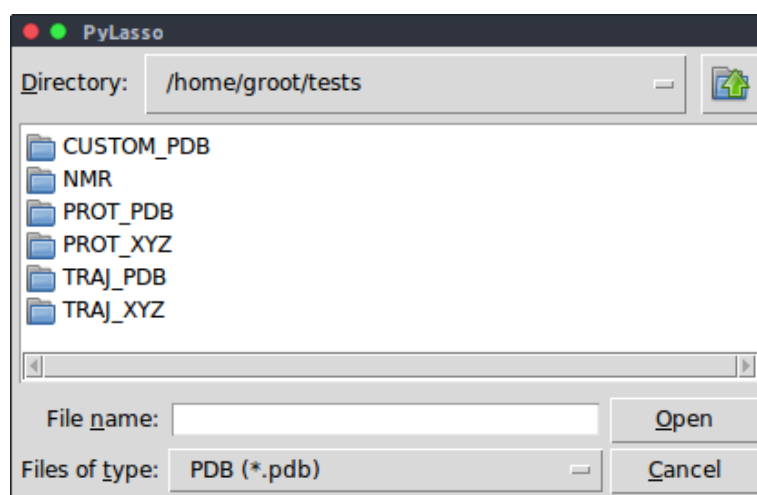


Fig. 4 PyLasso – a window that enables to choose a file according to its extension – Files of type – for which the lasso analysis will be conducted.

3. Protein

3.1. Input files

A user can download a file via the following command in the PyMOL terminal:

```
>> fetch [protein_pdb_code], type=pdb
```

(for example: fetch 3sum, type=pdb) or just download it directly from the PDB database. Next the downloaded file should be loaded to the PyLasso plugin via “open command”, as shown in Fig. 4.

Two formats of input files with a (single frame) configuration of a protein (or other polymer-like structure) are supported by the PyLasso: **PDB** and **XYZ**.

PDB is a standard file format from the PDB database (including NMR data) or a custom PDB file. For automatic detection of closed loops in the protein structure, a header of the PDB file should also be uploaded. The PyLasso uses only Ca atoms, whose positions represent the backbone chain. The chain index or indices and loop ranges need to be specified in the submit form.

```
[...]
SSBOND  1 CYS A    1    CYS A   13              1555  1555  2.03
SSBOND  2 CYS A    7    CYS A   19              1555  1555  2.03
LINK      N  CYS A    1              CG  ASP A    9      1555  1555  1.36
[...]
ATOM    24  CA  GLU A   24      14.212  33.180   1.863  1.00  64.63          C
ATOM    25  CA  GLU A   25      16.086  31.678  -1.097  1.00  59.03          C
ATOM    26  CA  GLU A   26      15.156  34.903  -2.877  1.00  41.85          C
ATOM    27  CA  GLY A   27      17.541  36.952  -0.744  1.00  46.12          C
ATOM    28  CA  VAL A   28      15.113  37.849   2.017  1.00  38.98          C
ATOM    29  CA  GLU A   29      16.989  38.371   5.242  1.00  43.20          C
ATOM    30  CA  TRP A   30      15.083  39.568   8.265  1.00  23.42          C
ATOM    31  CA  GLY A   31      16.419  41.023  11.489  1.00  19.92          C
ATOM    32  CA  TYR A   32      16.084  43.613  14.220  1.00  20.17          C
ATOM    33  CA  GLU A   33      18.064  46.492  12.735  1.00  20.57          C
[...]
```

Fig. 5 An example of an original PDB file with SSBOND and LINK.

```
[...]
ATOM    24  CA  GLU A   24      14.212  33.180   1.863  1.00  64.63          C
ATOM    25  CA  GLU A   25      16.086  31.678  -1.097  1.00  59.03          C
ATOM    26  CA  GLU A   26      18.766  29.646  -0.945  1.00  55.63          C
[...]
```

Fig. 6 An example of a custom PDB file.

PyLasso also analyzes PDB files with NMR data. These data are recognized via a line which contains words “SOLUTION NMR”. In this case the user is requested in an additional window, see Fig. 7, to decide how to analyze further data as:

- A single structure. In this case chemical bridges, e.g. cysteine bridges, can be detected automatically.
- A trajectory (as it contains multiple frames – NMR structures). In this case the user will be requested to provide the number of atoms to determine a closed loop. All calculations will be performed on the first chain in the PDB file.

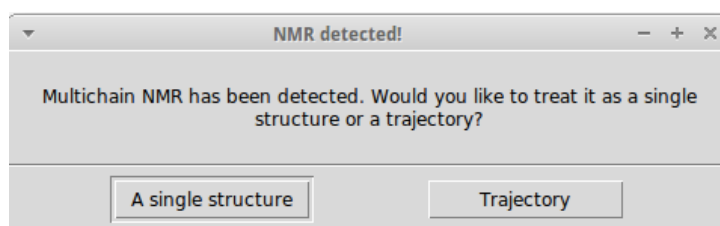


Fig. 7 An information window which appears when NMR data are detected. Here the user can choose how to analyze data: as a single PDB structure or as a trajectory.

XYZ is a simple format, which contains only sequential numbers and Cartesian coordinates of amino acids (or C α atoms) forming the backbone chain. Such a file cannot contain additional columns.

1	65.749	-49.197	15.303
2	69.453	-49.413	16.193
3	71.017	-51.127	19.196
4	72.074	-54.316	20.982
5	70.209	-57.190	22.570
[...]			

Fig. 8 An example of a file in XYZ format.

3.2. Interface

After loading the data the PyLasso window appears, as shown in Fig. 9. In this window a method to detect lasso can be chosen, after specifying the following options.

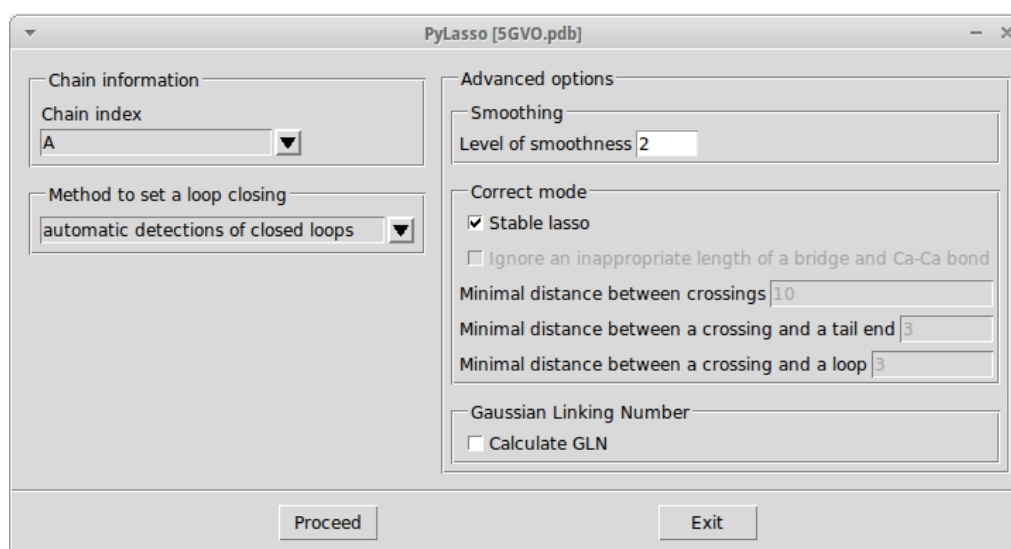


Fig. 9 PyLasso – an interface for protein structures.

- Chain index** – when the loaded file contains more than one chain, this option enables to choose a chain to be analyzed. The chains are retrieved from the input file, in case it has .pdb extension. By default the “Chain index” is set to the first chain found in the standard PDB file (typically chain A). If the input file is in XYZ format, chain A is automatically displayed.
- Method to find bridges** – enables to find covalent bonds in protein structures or to design new bridges, which define closed loops in a chain. Identification of closed loops is a necessary step to determine lasso topology. There is no restriction on the number of bridges. There are three ways to determine a bridge:
 - *Automatic detection of closed loops* – this option allows automatic detection of chemical covalent bridges in the original PDB file (a structure has to be loaded with all headers, in particular SSBOND and LINKS lines).
 - *Choose a bridge type* – this option works for original PDB files and displays five types of chemical bridges (disulfide/SS, amide, ester, thioester or other – Fig. 10).



Fig. 10 Types of bridges detected by the PyLasso.

More information about classification of bridges can be found under [this](#) link. A warning is shown in case a chosen bridge type has not been found (Fig. 11).

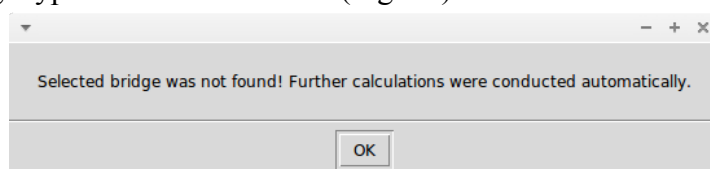


Fig. 11 A warning showed when selected bridge type has not been found in a given protein.

- *Choose two atoms to form a bridge* – allows manual selection of a pair of atoms forming a bridge, which are joined by an additional segment, see Fig. 12. Since the direction of the chain influences its lasso type, the first and the second number represent respectively loop opening and loop closing atoms. Note that in case of proteins a selection is set to Ca (internal GUI) in PyMOL. This method to identify bridges is available in both PDB and XYZ file formats.

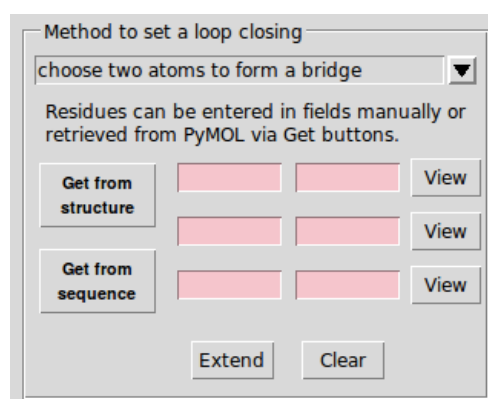


Fig. 12 Interface used to choose manually a pair of atoms forming a bridge (and closing a loop).

Amino acids (atoms) forming a bridge can be specified as follows:

- **Manually** – their sequential numbers should be typed in red fields. In case these numbers do not correspond to any pair of indices in the uploaded file the PyMOL displays an error, see Fig. 13.

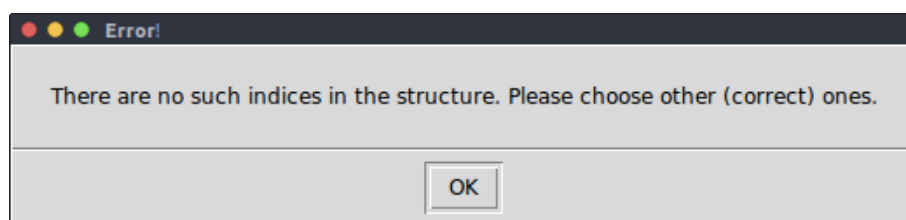


Fig. 13 An error displayed once atom(s) with numbers provided do not exist in the structure.

- **Get from sequence** – a user can use the cursor to choose a fragment of a backbone in the sequential structure in the top of the PyMOL window; the selection must consist of atoms from the specified chain in “Chain index” field. As a result, the sequential numbers of the first and the last (chosen) atoms along the sequence are inserted automatically in the first empty field.
- **Get from structure** – by pointing the cursor on the structure and choosing two atoms.

The *View* button next to each filled row displays the corresponding closed loop in grey color along the 3D structure in the PyMOL viewer. The segment (bridge) connecting two chosen atoms is shown in orange. If the distance between these atoms is bigger than 10Å (the maximal observed distance between interacting amino acids in the native state of proteins), a warning “too big!” is displayed (Fig. 14).

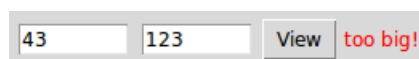


Fig. 14 A warning that the distance between atoms chosen to form a bridge is bigger than 10 Å.

The *Extend* button adds a new row and enables to define additional closed loops, once previous three rows of fields are filled with data (otherwise an error occurs). The *Clear* button deletes all data.

3. **Advanced** – these options are described in chapter “Advanced” below.

3.3. Lasso information window

After detection of closed loops and identification of minimal surfaces and lassos, all relevant data is presented in a new window “Lasso information”, see Fig. 15. Each row in this window contains information about one detected loop. The action of each button in this window is described in section Lasso visualization. Files generated in the process of lasso identification are saved in a new folder created in the working directory.


Lasso information window [4A3X.pdb chain A]								
	Lasso type	Loop range	Loop length	N-term piercings	C-term piercings	N-end length	C-end length	Lasso surface area
view details	LS ₂ -C	50-179	130		-219, -253	10	87	2171
view details	LL _{-1,+1}	78-119	42	-65	+141	38	147	774
view details	LS ₂ -N	180-262	83	-147, -170		140	4	1256
<div> <div>Surface</div> <div>Shallow lasso</div> <div>Smooth</div> </div>								
<div>Exit</div>								

Fig. 15 Lasso information window – an array with results for protein structure with PDB code 4a3x chain A (as indicated in the top of the window) with buttons *Surface* and *Shallow lasso* pressed. Each row in the array corresponds to one closed loop.

The following geometric and topological information about each closed loop is provided:

1. **Lasso type** – detailed signature of the identified lasso type; the subscript denotes the number of times the loop is pierced; the letter indicates which lasso tail (or tails, C or N) crosses the surface with assigned direction (+ or -). L₀ indicates an unpierced loop, L₁ lasso with one piercing, L₂ a double lasso (a loop that is pierced twice), LS a supercoiling, LL_{*i,j*} – both tails pierce the surface, respectively *i* and *j* times. A schematic representation of a given

lasso type is shown in the beginning of each line, in blue. A colored square shows a type of chemical interaction closing the loop. The square is displayed only for an input structure with SSBOND or LINK lines.

2. **Loop range** – sequential numbers of the atoms forming a closed loop. Proteins with incomplete chains are denoted by an additional symbol , and in this case undetermined fragments are replaced by straight segments. Note that this can generate additional piercings, so that detected lasso can be an artifact.
3. **Loop length** – the length of a closed loop (the number of amino acids in this loop).
4. **N- and C-term crossings** – a list of indices of amino acids that pierce the loop, with assigned direction of crossing: „+” (in blue) or „-” (in green). Shallow piercings are shown in grey. The numbers indicate sequential numbers of segments which pierce the surface spanned on a chosen covalent loop.
5. **N-end length and C-end length** – the numbers of amino acids in the chain before and after the loop, respectively for N- and C-tail of the lasso.
6. **Lasso loop area** – an area of the (minimal) surface spanned on the loop (in Å²).

The *view details* button on the left of each row displays the corresponding surface spanned on the closed loop in the PyMOL viewer, with all crossings in relevant colors. If only one lasso is found in a given structure, it is automatically displayed in the PyMOL viewer.

If detected lasso is non-trivial, i.e. it has at least one piercing, two PyMOL objects are created, POS_PIERC and NEG_PIERC highlighted in respectively green and blue color, representing the piercings. In addition a structure SEQ is created, which contains a sequence of atoms forming a lasso. Finally, there is an object BR_beginofloop_endofloop, which displays a bridge formed between chosen atoms.

3.4. Lasso visualization

Once a lasso is displayed (after clicking *view details* button), the following additional 2, 3 or 4 buttons are shown (depending on previously chosen options).

1. **Surface** – displays/hides the bridge and the minimal surface spanned on selected loop (Fig. 16).

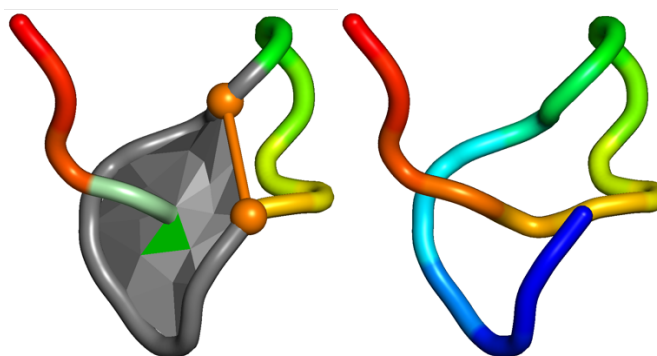


Fig. 16 Displayed/hidden surface spanned on atoms 1-8 (protein with PDB code 2lx6). The pierced triangle is shown in green.

2. **Smooth** – displays/hides the smoothed chain and the minimal surface (Fig. 17). Smoothed chain is not available if smoothing would change topology and for structures detected as artificial.

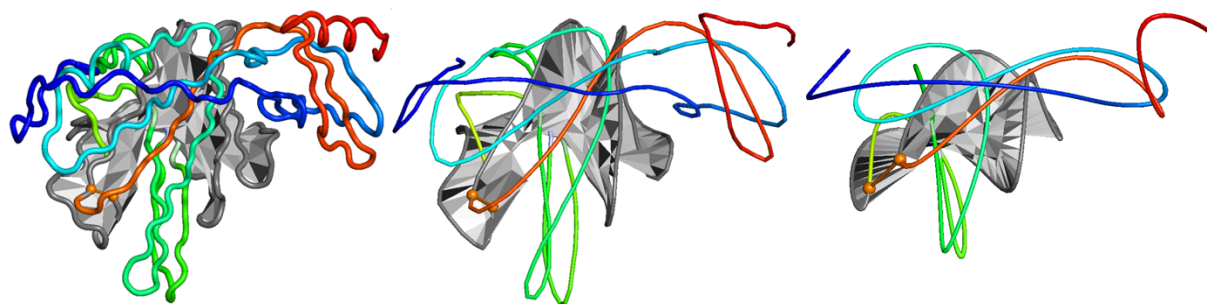


Fig. 17 Protein with PDB code 4h14 (chain A) in original (left) and two smoothed (to different extent) versions (middle and right).

- Shallow lasso** – appears only if shallow crossings have been detected in the structure, and shows or hides these crossings (Fig. 18). After clicking the button, it is advised to click *view details* once again in order to display piercings on the surface spanned on a chosen loop.

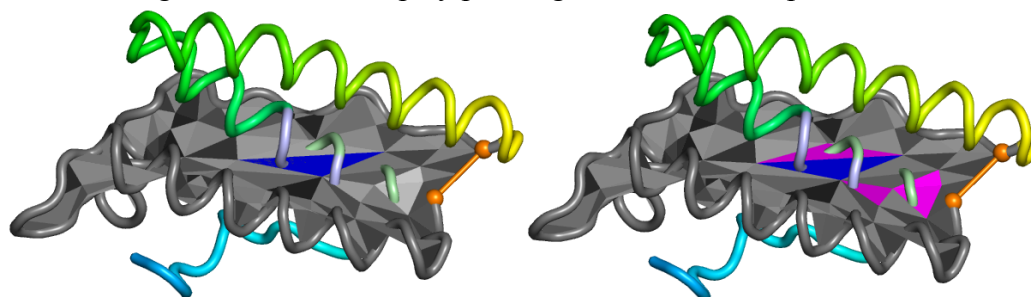


Fig. 18 Protein with PDB code 1ax8 (chain A). Hidden (left panel) and displayed (right panel) shallow piercings. When the Shallow Lasso button is not pressed, piercings on the surface are not displayed.

- GLN** – this option is described in chapter “Advanced”.

4. Trajectory

4.1. Input files

A trajectory can be saved in a multi-structure file in either of the two formats. The first is multi-structure PDB format. This format requires correct positioning of information in each line. Moreover, the chain name is required in each frame. In the PDB file, the models (structures) should be separated by both “ENDMDL” and “MODEL n” lines (where “n” is the number of the frame); models separated by “END” lines are not valid. The PyLasso uses only C α atoms. Note that the first column coincides with the beginning of the row.

TITLE	t= 7289.00000										
MODEL	1										
ATOM	1	CA	LEU	A	1	-2.440	-1.220	15.860	1.00	0.00	C
ATOM	2	CA	VAL	A	2	-5.490	0.610	15.860	1.00	0.00	C
ATOM	3	CA	PHE	A	3	-4.880	3.050	12.810	1.00	0.00	C
ATOM	4	CA	GLY	A	4	-3.660	-0.610	10.980	1.00	0.00	C
ATOM	5	CA	ALA	A	5	-6.710	-1.830	11.590	1.00	0.00	C
ATOM	6	CA	TRP	A	6	-8.540	1.830	10.370	1.00	0.00	C
ENDMDL											
TITLE	t= 7290.00000										

```

MODEL          2
ATOM          1 CA LEU A 1 -2.440 -1.220 15.860 1.00 0.00 C
ATOM          2 CA VAL A 2 -6.100 0.610 15.860 1.00 0.00 C
ATOM          3 CA PHE A 3 -4.880 2.440 12.810 1.00 0.00 C
ATOM          4 CA GLY A 4 -3.660 -1.220 10.980 1.00 0.00 C
ATOM          5 CA ALA A 5 -7.320 -1.830 11.590 1.00 0.00 C
ATOM          6 CA TRP A 6 -8.540 1.220 10.370 1.00 0.00 C
ENDMDL

```

Fig. 19 An example of a multi-structure PDB file format.

It is also possible to analyze PDB file with NMR data as a trajectory. In this case the PDB file should contain “SOLUTION NMR” information, and the models (structures) should be separated by both “ENDMDL” and “MODEL n” lines (where “n” is interpreted as the number of the frame); models separated by “END” lines are not valid. The PyLasso uses only Ca atoms. Example of such a format is shown in Fig. 20 based on protein with PDB code 5gvo.

```

...
EXPDTA      SOLUTION NMR
NUMMDL      15
...
MODEL          1
ATOM          1 N GLY A 1 1.714 3.159 -1.844 1.00 0.00 N
ATOM          2 CA GLY A 1 3.041 3.738 -1.950 1.00 0.00 C
...
TER          301 ILE A 18
ENDMDL
MODEL          2
ATOM          1 N GLY A 1 1.859 3.309 -1.983 1.00 0.00 N
ATOM          2 CA GLY A 1 3.166 3.897 -2.213 1.00 0.00 C
...
TER          301 ILE A 18
ENDMDL

```

Fig. 20 NMR format example based protein with PDB code 5gvo.

The second file format is multi-structure XYZ format, with consecutive time frames separated by letter “t” followed by a number specifying moment of time. Numbers 1, 2, ..., 6 represent consecutive beads. Note that there is no additional space between the beginning of the row and the first letter or number.

```

t 0.00000
1 309.520 300.740 308.220
2 307.970 302.000 304.930
3 309.710 303.310 301.740
4 311.370 299.900 301.510
5 288.870 295.970 282.810
6 288.490 293.410 280.000
t 1.25000
1 311.350 305.300 304.570
2 309.880 304.440 301.160
3 310.700 304.940 297.380
4 312.410 301.520 297.640
5 298.250 283.500 285.870
6 296.820 280.060 286.010

```

Fig. 21 An example of custom XYZ file format with trajectory.

4.2. Interface

The analysis of multi-structure files is similar to files with a single structure (Fig. 22). The PyLasso offers basic and more advanced tools to analyze lassos in consecutive frames. A user can determine the level of calculational complexity, and the level of output specificity (“More detailed algorithm” and “More detailed output file”). A user can also define the “Step”, i.e. number of frames, which are skipped when analyzing the trajectory. The most important tools are summarized below.

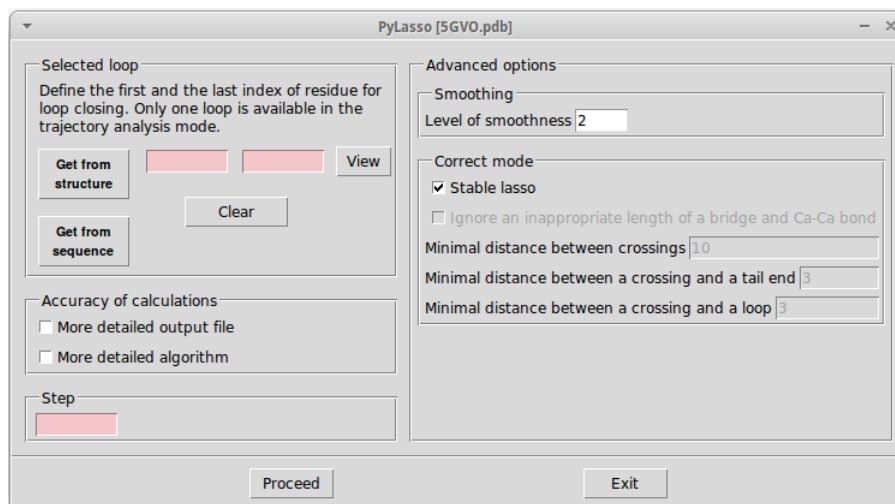


Fig. 22 PyLasso – an interface for trajectory structures.

1. **Selected loop** – enables to select a pair of atoms forming a bridge (closing the loop). In case of a trajectory a user is always prompted to select only one bridge. Atoms can be selected in the same way as in the single structure mode.
2. **Accuracy of calculations:**
 - *More detailed algorithm* – defines complexity of calculations.
 - *More detailed output* – defines how extensive information should be included in the output file. There are two options:
 - Less detailed output: provides information about numbers of crossings and their positions (after reduction of shallow piercings),
 - More detailed: provides additional information including radius of gyration, numbers and positions of reduced crossings (before reduction), and surface area.
3. **Step** – the number of frames skipped during the analysis of the trajectory. By default the step is set to 1. Increasing this value speeds up calculations. It is recommended to analyze data first with a large step and then repeat the calculation with smaller step only for frames where non-trivial entanglement is found. Note that even if step bigger than 1 is chosen, the data in lasso information window in Fig. 23 is calculated starting from the first frame from an uploaded file.
4. **Advanced** – this option is described in chapter “Advanced”.

4.3. Trajectory information window

A type of lasso and its geometry determine physical and biological properties of (bio)polymers, e.g. their stability. These properties can be determined by analyzing lasso dynamics. Time evolution can be used to create the free energy landscape. Output from the multi-structure input file is suited for analysis of the lasso entanglement dynamics, e.g. the change in lasso type and

piercing residue index can serve as new reaction coordinates. The following information summarizing trajectory analysis is provided.

1. **Lasso information** – for a specific frame chosen (see point 4 below), it is the same as described in section 3.3 “Lasso information window”.
2. **Lasso(s) type** – a chart that displays the change of lasso type in consecutive frames, see Fig. 19. The chart shows lasso types in each time frame. Clicking on any point displays detailed information: the number of this frame, the type of lasso, and the geometry of the piercing (N or C terminal, +/- orientation). Different lasso types are presented in different colors. The conformation of the structure in selected frame, including the minimal surface spanned on the closed loop, is displayed in the PyMOL viewer.

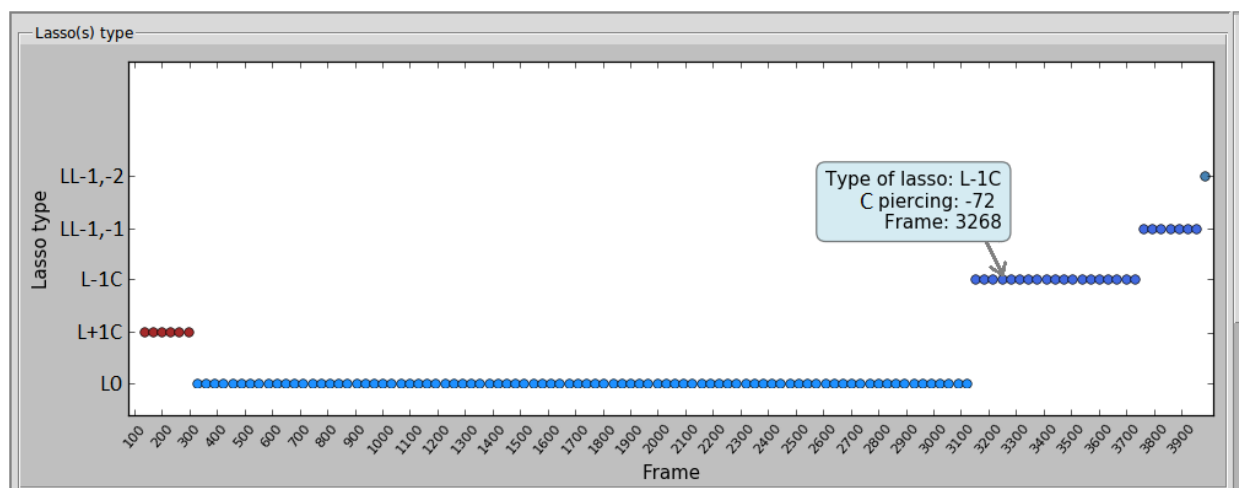


Fig. 23 The results of the trajectory analysis displayed in the chart. The chart presents detected lasso types (the vertical axis) at a given time frame (the horizontal axis). Upon clicking the mouse above a selected frame, details about detected lasso are presented in the blue box.

3. **Atoms piercing lasso** – a chart depicting which atoms in each frame pierce the surface spanned on the lasso loop (Fig. 24). This loop is marked as an orange rectangle, spanned across the whole time evolution. A chart below shows a single piercing. In general, the surface can be pierced several times.

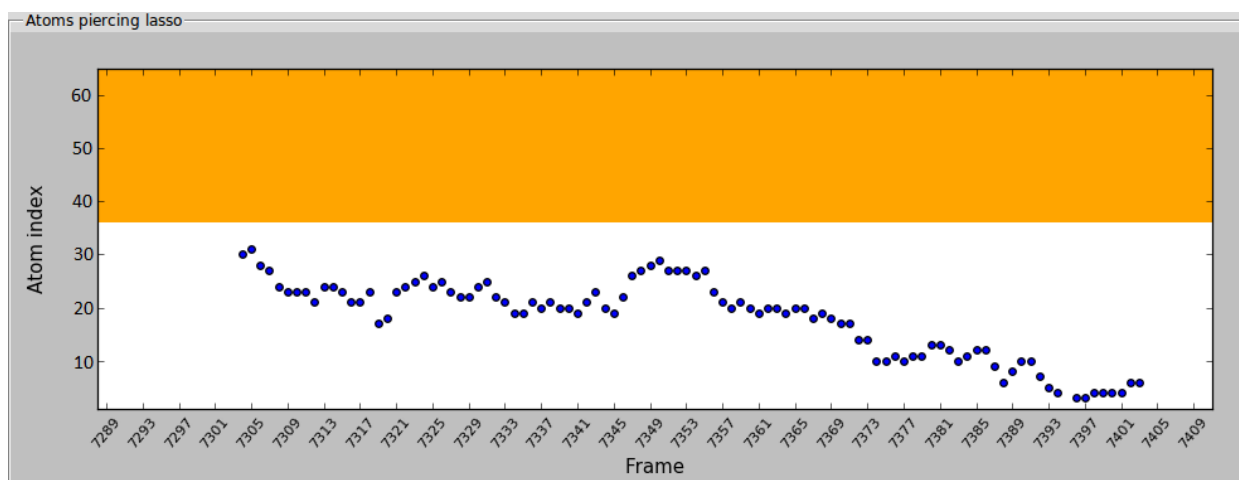


Fig. 24 The results of the trajectory analysis displayed by the chart. This chart presents sequential numbers of piercings (in blue) of a closed loop (orange). The orange rectangle represents atoms which form a closed loop. A single piercing at each frame corresponds to lasso of type L_1 in this case.

4. **Frame number** – this window enables to conduct detailed (as in the single frame mode) analysis of lasso configuration in several (maximum 6) frames (Fig. 25). *Extend* button enables to specify additional frames. *Clear* button deletes all the information. *Calculate* button initiates the analysis, whose results are presented in the “*Lasso information*” window. If the array contained some previous results, they are removed and replaced with new ones.

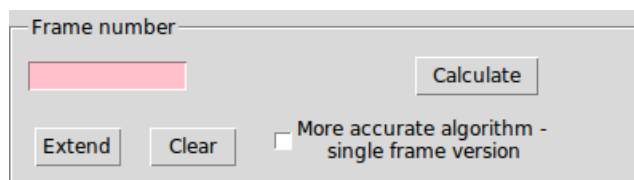


Fig. 25 Frame number window.

5. **Trajectory analysis log** – a window with information about analyzed trajectory. In first 8 lines (starting with #) a description of the file content is shown. The first line contains the name of the file. The second line contains indices of the N- and C-terminal residues and indices of bridge-forming residues. The following lines summarize which options have been chosen. The content of the box differs depending on the complexity of the output chosen (as discussed in section *Interface for trajectory*), e.g. see the difference between Fig. 26 and Fig. 27. The notation is explained in more detail under [this](#) link.

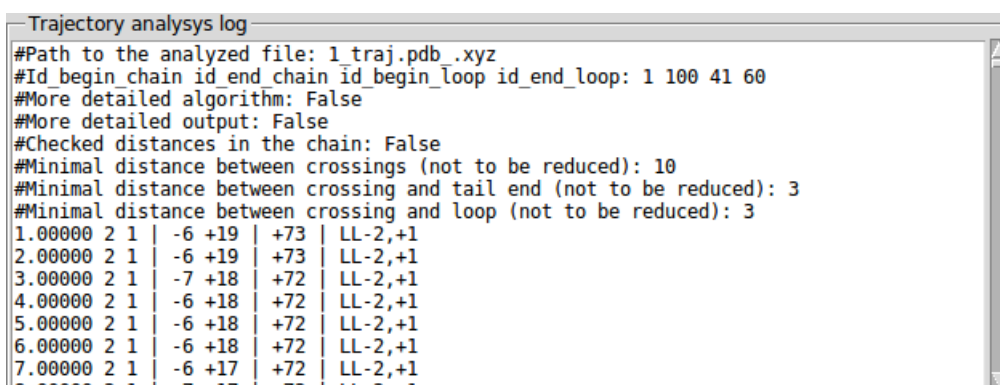


Fig. 26 Trajectory analysis log – less detailed version.

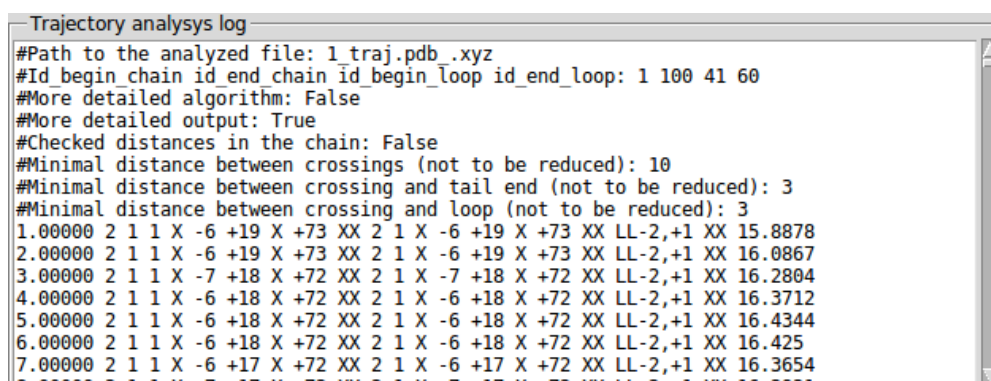


Fig. 27 An example of trajectory analysis log – more detailed version.

5. Advanced options

5.1. Smoothing

This option smoothens the backbone chain and the surface spanned on the lasso loop. The level of smoothness can be adjusted, by setting its value between 1 and 100, see Fig. 28. A refined chain is created using the running average. Note that smoothing is not available for structures recognized as artificial.

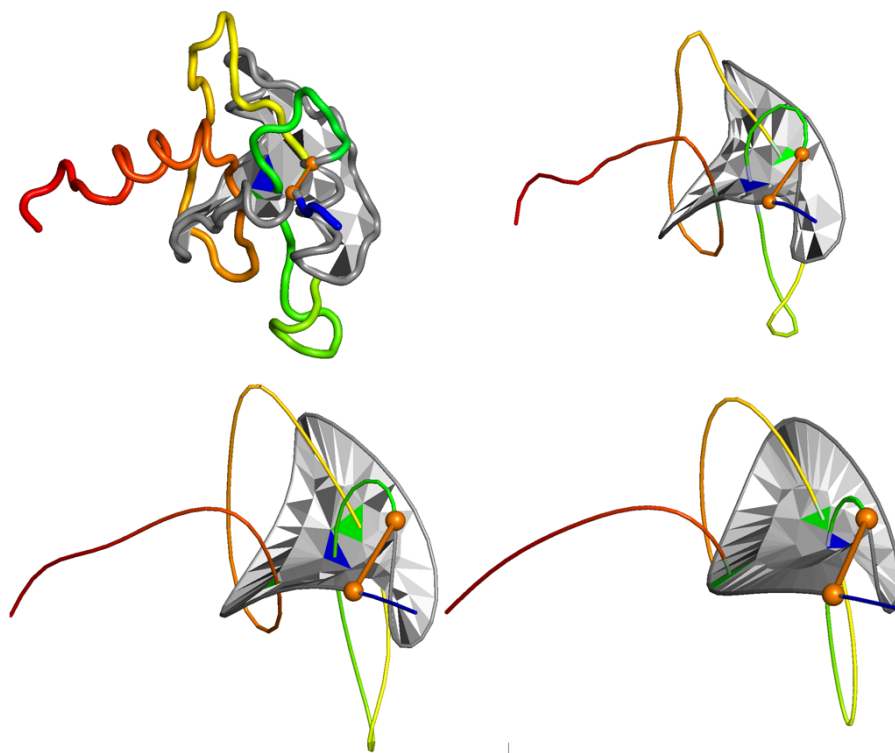


Fig. 28 Examples of a smoothed chain, based on protein with PDB code 2msx (chain A). The value of smoothing is set respectively to 2, 8 and 24.

5.2. Correct mode

This option sets geometrical conditions used to determine surfaces spanned on closed loops, and distinguishes between artificial (uncertain) and proper (stable) piercings.

1. **Stable Lasso** – unticking this option allows to conduct analysis in the following 4 approaches, with advanced parameters set by the user.
2. **Ignore an inappropriate length of a bridge and Ca-Ca bonds** – ignores checking if these lengths take reasonable values. This option should be ticked for polymer studies.
3. **Minimal distance between crossings** – the number of residues between piercings that defines a deep crossing. By default it is set to 10. This option enables to avoid “shallow” lassos, which can be unthreaded by thermal fluctuations.
4. **Minimal distance between a crossing and a tail end** – the number of residues between a piercing and the end of a protein chain, that defines a deep crossing. By default it is set to 3. This option enables to avoid shallow lassos, which may be artifacts of thermal fluctuations.
5. **Minimal distance between a crossing and a loop** – the number of residues between a piercing and a loop (disulfide or other bridge), that defines a deep crossing. By default it is set to 3. This enables to avoid artificial thermal fluctuations.

If those three values are increased, more piercings will be considered as shallow. On the other hand, if those values are decreased, more complicated lasso types are identified. In particular, if all three values are set to zero, none of the piercings is recognized as shallow.

5.3. Gaussian Linking Number

Gaussian Linking Number (GLN) option enables computation of Gauss linking integrals, which are then presented graphically. GLN is yet another measure of entanglement between the loop and the tail, which indicates how many times (and in which direction) the tail winds around the loop. For instance, the value of GLN close to 1 means that the tail binds around the loop more or less once, thus probably it is pierced once through the loop and the lasso is of type L_1 . Additional information provided by GLN may be helpful to identify the geometry of the piercing, which may strongly influence e.g. stability of the structure. This option is unavailable for trajectories.

The button “GLN” appears in “Lasso information” window only if the option to calculate GLN had been clicked before. After clicking this button, two GLN matrices are generated and presented under the array with results, and a protein structure in the PyMOL is colored according to GLN values (Fig. 29). The matrices are interactive: clicking on any square under the diagonal results in coloring corresponding sequence between selected atoms.

Remarks

- for long chains (represented by more than 300 amino acids or beads) the process of generating GLN matrices can take a few minutes.
- The GLN matrix is not generated when the length of the terminus is less than 5 amino acids.

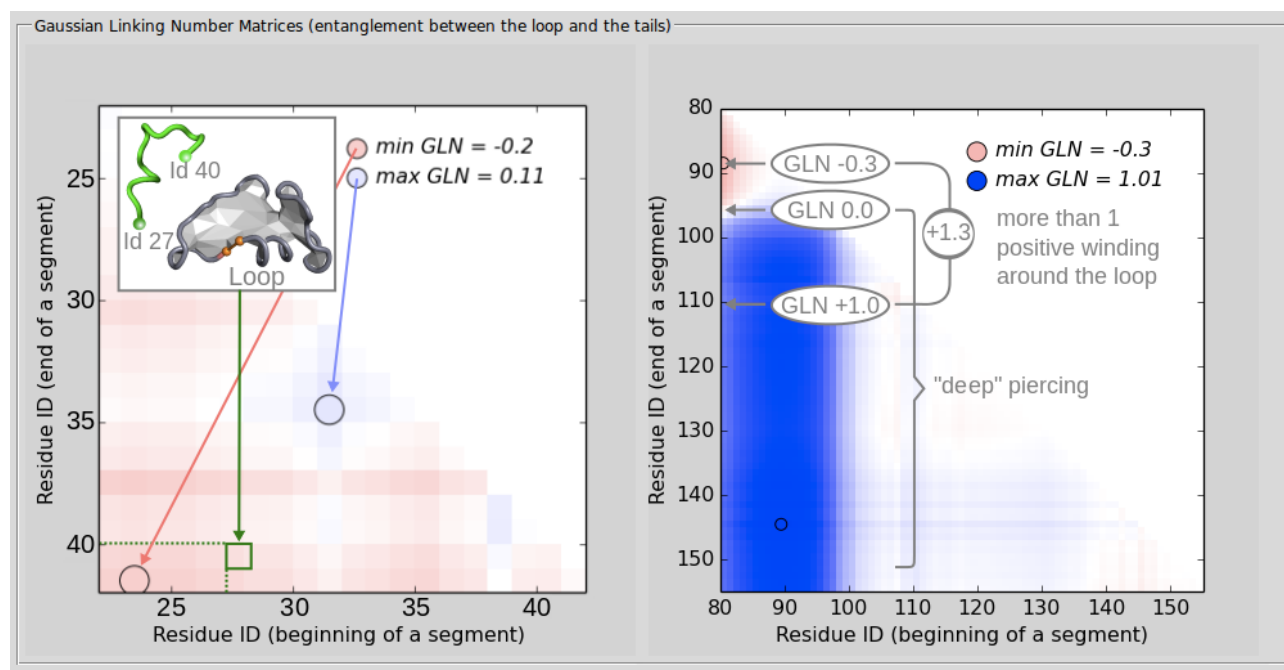


Fig. 29 Protein with PDB code 3sum. Two GLN matrices for N-tail (left) and C-tail (right) and a lasso spanned on amino acids with sequential numbers 43 and 80. Each point in the matrix corresponds to a specific fragment of a tail (as an example, a sequence between ids 27 and 40 is shown in the left panel) and is colored proportionally to the GLN value between it and the whole loop. Extreme values in the left figure are low (-0.2 and 0.11), which indicates no significant binding around the loop, so that the N-tail is not pierced through the loop. The maximum value of GLN in the right figure is as high as 1.01, which indicates the presence of one positive crossing and a complex lasso structure of type L_{+1C} . Moving along the left edge of the right matrix – from the top

to the bottom – more details of this particular lasso geometry are revealed: first, while the color becomes slightly red (up to the minimum GLN value -0.3), the beginning of the tail binds slightly around the loop in the negative direction (probably without any piercings). The next color is whitening and then it becomes deep blue (GLN value rises up to +1), while the middle part of a tail (ids 88-108) binds around the loop more than once, which indicates positive “strong” piercing in that region. Finally the blue color does not change much to the bottom of the column, which indicates that the long last part of the tail remains on one side of the loop and probably does not come back close to the surface.

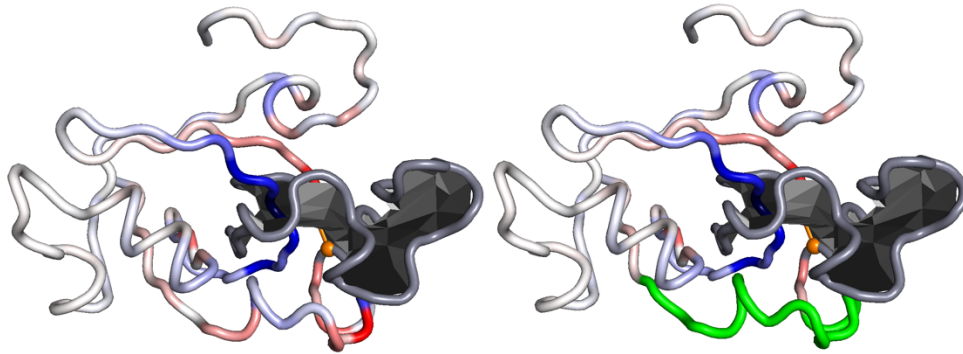


Fig. 30 Protein with PDB code 3sum. Left: protein structure colored according to GLN values – the color of each segment of the tail corresponds to the GLN value between that segment and whole loop. The most intensive colors are shown in the piercings regions. Right: a sample sequence between atoms 27-40 marked in green, once the corresponding point in the GLN matrix is clicked.

6. Bibliography

- [1] Niemyska, W., Dabrowski-Tumanski, P., Kadlof, M., Haglund, E., Sułkowski, P. and Sulkowska, J. I. *Complex lasso: new entangled motifs in proteins*. Scientific Reports 6 (2016) 36895.
- [2] Dabrowski-Tumanski, P., Niemyska, W., Pasznik, P. and Sulkowska, J. I. *Lassoprot: server to analyze biopolymers with lassos*. Nucleic acids research 44 (2016) W383–W389.