Summary of Activities (Quarter 1)

Haoling Zhang and Yuting Chen

Overview

Our *PyMOL Fellowship* Project is to establish high-level interfaces based on *PyMOL* framework[1] called *PyMOL-advance*. Its repository is set out in https://github.com/BGI-SynBio/PyMOL-advance and will be made public after basic completion (i.e. about early March). These implemented high-level interfaces effectively establish the channel from molecular structure data to manuscript-level figures. Predictably, as shown in Figure 1, they can help bioinformaticians, algorithm scientists and computer engineers to create figures for their manuscripts in a simpler and more rapid way.

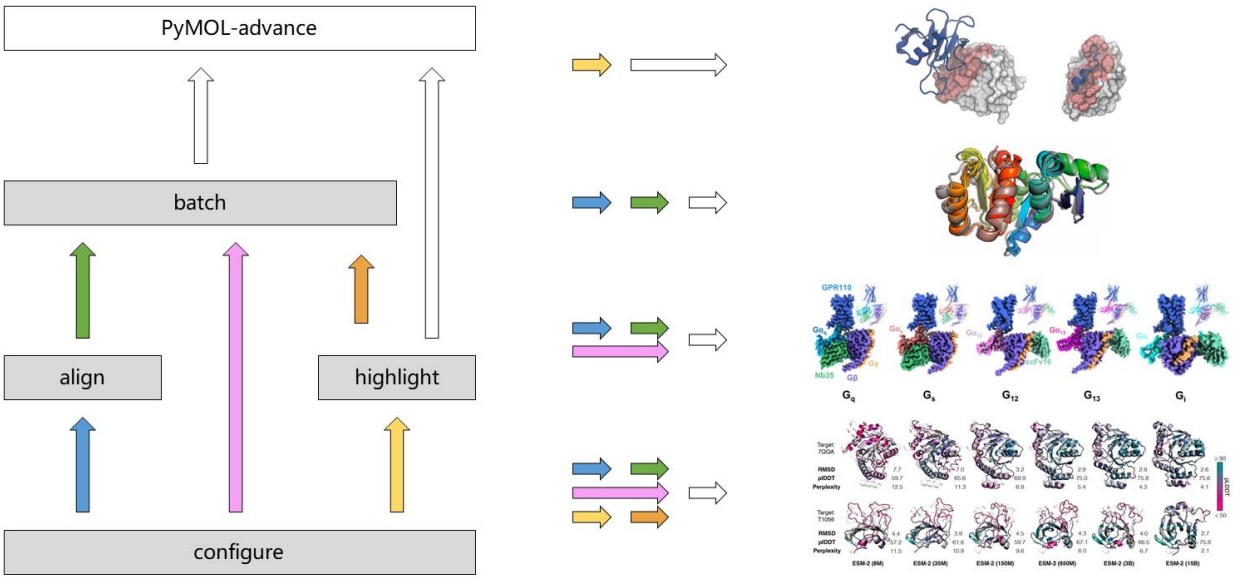

Figure 1. core requirement architecture and target cases[2-5].

As we mentioned in the proposal, we need to program scripts of alignment module and functional coloring module in our first quarter. Besides, we need to collect and reproduce valuable cases from figures in high-quality academic journals (like the right part of Figure 1) and provide unit tests of the established interfaces.

Quarterly Goal

1. **alignment module:** In the process of molecule structure alignment, structure rotating methods[6], scoring methods[7-9] and similarity metrics[10] are highly related to this module. To improve the practicability and reliability of this module, it is essential to clarify the limitations of the above-mentioned methods. Furthermore, users may also have pre- and post-processing requirements for this module. For example, they may need to align parts of the structure[11] or cluster the aligned structures[9]. If we meet these requirements, we could further reduce user development costs and have a positive impact on user experience.
2. **functional coloring module:** This module can regulate the coloring rule of the molecule structure to highlight the core information that users want to emphasize. Basic emphasis requirements include amino acids (or nucleotides)[12], short segments[13], secondary structures[14] and functional domains[15] from micro to macro perspective. Taking amino acids as a

type of unit, a broader requirement is the coloring of each unit corresponds to the provided value one by one. These values could be the physicochemical properties of units[16] or results of structure alignment[13]. Obvious properties include polarity, electronegativity, conformational flexibility, etc. Meanwhile, a classical case in alignment results is root-mean-square deviation (RMSD) of each unit. When such alignment value for each unit in the predicted structure is displayed, the figure can effectively demonstrate which parts of a particular structure are worse predicted by trained models.

3. **supporting functions:** As we mentioned above, some supporting functions are considered to improve the integrity of the project. The general pre-processing functions are regarded as the bridge connecting the aforementioned two modules, which include loading/saving molecule structures from files, disassembling the entire chain into sub-segments of the given k-mer length[17] and setting physicochemical properties for chains. To create manuscript-level figures through the well-established library namely *matplotlib*[18], 3-dimensional structure data and coloring data could be converted through the provided interfaces. Besides, format requirements (e.g. typeface, figure size, dots per inch, width limitation, etc) of different journals and conferences are set when users initialize the figures through this project. Last but not least, some suggestive statements, methods and classes can make the friend interaction between users and our project.

Solutions

1. **discuss limitations with proposers:** In the past three months, we have consulted several teams about the limitations of their designed methods. As one of the most well-accepted methods, TM-score[8] is affected by sequence lengths and model types (i.e. observed position number of each residue) jointly. Therefore, as shown in Table 1, we have established a long-term discussion relationship with Prof. Yang Zhang's group (especially Prof. Yang Zhang at University of Michigan and Dr. Chengxin Zhang at Yale University).

Table 1. Summary of TM-score discussion.

date	question	answer from Prof. Yang Zhang's group
Jun 23, 2021	There seems to be an obvious watershed of evaluating display performance in TM-score between 3-dimensional molecule structures with 4 amino acids and those with 5 amino acids. As the score threshold (0.7) of TM-score mentioned in the AlphaFold1 paper, we assume that the two 3D structures are highly similar and could be grouped in a cluster if TM-score is greater than the score threshold. However, as shown in attached figure, oligopeptide -[LAAA]- and oligopeptide -[QVEYA]-, their performance is different after clustering by TM-score with 0.7. We want to find the reasons for this difference.	I am afraid you are venturing into an area where TM-score is not designed for. TM-score is for well folded protein structures with non-trivial length. For example, in our study, we only obtain statistics for structures and fragments with more than 20 residues. Whether or not fragments shorter than 20 residues, follows the same statistics is unknown. In other words, while proteins with longer than 20 residues and TM-score > 0.7 are highly structurally similar, it is unknown whether highly similar structures with less than 20 residues should also have TM-score > 0.7.
Dec 08, 2022	We are using US-align for predicting structures of nucleic acids with 3SPN model, and have a question on its scope. In your supporting material, the observed atom of TM-score is C3' (or other single atoms). Could we use TM-score with 3SPN model directly or do some modifications (e.g. multiply the length by 3)?	TM-score is trained to use only one atom per nucleotide, even if each nucleotide has three or more atoms.

Early in 2021, we found that TM-score is not as effective as RMSD for structures of short sequence length. After discussing, we clarified the application scope of TM-score in length, and use RMSD instead of TM-score for the sequence with less than or equal to 20 residues. This length limitation also provides an impetus for the development of TM-score, the latest version of TM-score (i.e. US-align[9]) provides the corresponding solution while further supporting nucleic acid molecules. However, it only supports sampling models of one atom per residue. To better apply to different sampling models, we consider choosing one atom at a time for our calculations, using three parallel calculations to train models.

2. **collect physicochemical properties of different molecule type:** More and more scientists realize that the structure is highly related to physicochemical properties[16]. Providing these properties can help users to establish the relationship between biological functions and properties. Meanwhile, the association between properties and prediction results may help to further optimize trained models. Table 2 shows the collected properties for amino acids.

Table 2. collection of physicochemical properties.

property	description
polarity[19]	the distribution of electric charge.
electronegativity[19]	the attraction a bonded atom has for electrons.
hydrophobicity[20]	the association of nonpolar groups or molecules in an aqueous environment.
turn propensity[20]	the propensity of residues in turn structure.
helix propensity[20]	the propensity of residues in alpha-helix structure.
sheet propensity[20]	the propensity of residues in beta-sheet structure.
folding propensity[20]	the propensity of residue be folded in protein structures, determined by hydrophobicity, secondary-structure propensity, and electrostatic charge.
flexibility[21]	the conformational flexibility of residue, a crucial factor in determining their biological activity, including binding affinity, antigenicity, and enzymatic activity.

Besides, other physicochemical properties for deoxyribonucleic acids and ribonucleic acids are still being collected.

3. **collect manuscript formats:** Table 3 lists some common manuscript formats. Users can configure figure formats when initializing. By doing so, qualified figures can be created directly without further adjustments to meet the requirements of target journals or conferences.

Table 3. collection of journal, conference or publisher figure formats.

target	font	math font	max dpi	columns	width under column occupy (inches)		
					1	2	3
Nature	*Arial*	*Linux Libertine* *Lucida Calligraphy*	300	2	3.54	7.08	-
Science	*Helvetica*		300	3	2.24	4.76	7.24
Cell	*Arial*		300	2	3.35	6.85	-
				3	2.17	4.49	6.85
PNAS	*Helvetica*		600	2	3.43	7.08	-
ACS	*Arial*		600	2	3.30	7.00	-
Oxford	*Arial*		350	2	3.35	6.70	-
PLOS	*Arial*		300	1	5.20	-	-
IEEE	*Times New Roman*		300	2	3.50	7.16	-
ACM	*Linux Libertine*		300	2	2.50	6.02	-

Significantly, in order to represent as many mathematical symbol types as possible, the mathematical font is uniformly executed according to *ACM* mathematical font format, i.e. *Linux Libertine* format for normal, italic and bold fonts, and *Lucida Calligraphy* for calligraphy fonts.

4. **implement codes:** *PyMOL-advance* framework consists of three parts: 'mola' (abbreviation of PyMOL-advance), 'tests', and 'cases'. Among them, 'mola' provides core codes, the completion status of which is shown in Figure 2.

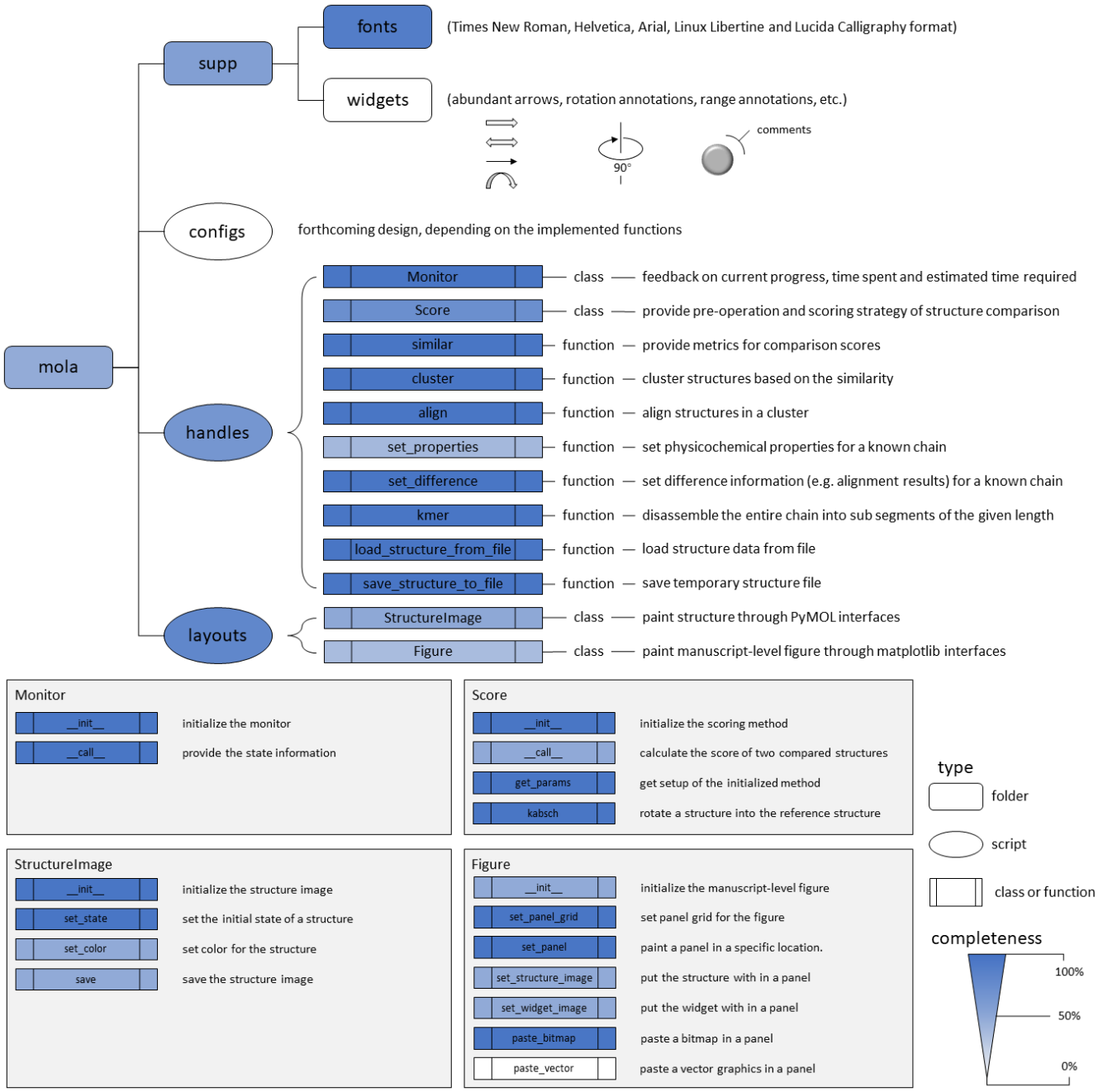

Figure 2. core code architecture, detailed description at function level and project completion status.

Ideally, to reduce the coupling within the framework, these high-level interfaces should be implemented as functions. Since partial visualization requirements depend on fixed variables (e.g. grid layout information in figures) in actual operations, the frequent transmission of these variables may have an unpredictable impact on the framework reliability. Therefore, after multiple adjustments, monitor, scoring methods, structure image operations and figure operations are implemented as classes and initialized as objects.

5. **execute unit tests:** We verify the smallest testable unit in our code by executing unit tests with CircleCI framework. Through the design of multi-aspect test cases, the feasibility and accuracy of codes in the 'mola' folder are demonstrated. Specifically, the response of inputting specific data is demonstrated and alpha testing of alignment module and partial supporting functions are completed.

Remaining Issues

1. **automatic verification for qualified figure outputs:** As we provide a visualization framework, we consider verifying the qualification of our results by figure output. If we stubbornly make our output figures completely equivalent to exquisite publications using assertion strategies, considering the debugging cost at the pixel level, the time is immeasurable. A potential solution is to create a project-made figure similar to the target figure and use this project-made figure as the reference for the automatic output verification. The reliability of this automatic verification strategy will be confirmed after the implementation of most of our interfaces.
2. **vector structure image by PyMOL framework:** Currently, our pipeline is to paste the final structure image in a *matplotlib* subplot. To our knowledge, the supported output image formats of PyMOL framework do not contain vector graphics formats[22] like Adobe Illustrator Artwork file format. If the structure image is a bitmap, it may be distorted after pasting. In order to provide better display effect, we wish the coming version has relevant interfaces or we may jointly investigate how to map 3-dimensional structure data to 2-dimensional vector graphics under a given camera angle.
3. **vector graphics painting scheme for matplotlib:** As the receiver of vector structure images, 'pyplot.imshow' interface of *matplotlib* can only paint bitmap. Although combining with Python-based vector graphics editors like *svgutils* and *skunk* can complete the painting task through complex operations (see Table 4), the usage is easy to lose flexibility.

Table 4. Painting vector graphics by repeatedly overlaying.

statement	description

`from svgutils import compose`
`from matplotlib import pyplot`
`from numpy import random`
	load 'compose' interface of *svgutils* module
	load 'pyplot' interface of *matplotlib* module
	load 'random' interface of *numpy* module

`width, height = 10, 5`
`pyplot.figure(figsize=(width, height))`
`pyplot.scatter(random.random(size=(10,)), random.random(size=(10,)))`
`pyplot.savefig("non-structure part.svg", transparent=True)`
	set figure size with inch format.
	create figure with the given size.
	complete some painting tasks by matplotlib.
	save painting data as vector graphics.

`width, height = str(width * 2.54) + "cm", str(height * 2.54) + "cm"`
`lower_layer = compose.Panel(compose.SVG("non-structure part.svg"))`
`upper_layer = compose.Panel(compose.SVG("structure part.svg"))`
`merged_figure = compose.Figure(width, height, lower_layer, upper_layer)`
`merged_figure.save("compose.svg")`
	adjust size of figure since *svgutils* has no inch format.
	create the lower layer.
	create the upper layer.
	assemble two vector graphics using *svgutils*.
	save the merge vector graphics.

Next Quarter

In the next quarter, we will focus on widget paintings and the layout setting of structure parts in target figures. The latter one will complete after confirming the strategy of painting structure to *matplotlib* subplot (i.e. issue 2). If the automatic

verification for qualified figure outputs is feasible (i.e. issue 1), we will provide a large number of case tests. Concurrently, we will let our repository go public and invite potential users to test it. The beta testing can be scheduled to start in early March. So far, we have invited several users who are interested in *PyMOL-advance* such as Dr. Chen Lin at University of Oxford. We also welcome you to invite users to give us more feedback later.

Reference

1 DeLano, W. L. Pymol: An open-source molecular graphics tool. *CCP4 Newsl. Protein Crystallogr* **40**, 82-92 (2002).

2 Corbi-Verge, C. & Kim, P. M. Motif mediated protein-protein interactions as drug targets. *Cell Communication and Signaling* **14**, 1-12 (2016).

3 Yang, J. *et al.* Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences* **117**, 1496-1503 (2020).

4 Zhu, X. *et al.* Structural basis of adhesion GPCR GPR110 activation by stalk peptide and G-proteins coupling. *Nature Communications* **13**, 5513 (2022).

5 Lin, Z. *et al.* Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv* (2022).

6 Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* **32**, 922-923 (1976).

7 Petitjean, M. On the root mean square quantitative chirality and quantitative symmetry measures. *Journal of Mathematical Physics* **40**, 4587-4595 (1999).

8 Zhang, Y. & Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic acids research* **33**, 2302-2309 (2005).

9 Zhang, C., Shine, M., Pyle, A. M. & Zhang, Y. US-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature methods* **19**, 1109-1115 (2022).

10 Xu, J. & Zhang, Y. How significant is a protein structure similarity with TM-score= 0.5? *Bioinformatics* **26**, 889-895 (2010).

11 Mosca, R., Brannetti, B. & Schneider, T. R. Alignment of protein structures in the presence of domain motions. *BMC bioinformatics* **9**, 1-17 (2008).

12 Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583-589 (2021).

13 Frontzek, K. *et al.* A conformational switch controlling the toxicity of the prion protein. *Nature Structural & Molecular Biology* **29**, 831-840 (2022).

14 Pan, X. *et al.* Molecular basis for pore blockade of human Na+ channel Nav1. 2 by the μ-conotoxin KIIIA. *Science* **363**, 1309-1313 (2019).

15 Gao, S., Yao, X. & Yan, N. Structure of human Cav2. 2 channel blocked by the painkiller ziconotide. *Nature* **596**, 143-147 (2021).

16 Skolnick, J. & Gao, M. The role of local versus nonlocal physicochemical restraints in determining protein native structure. *Current opinion in structural biology* **68**, 1-8 (2021).

17 Compeau, P. E., Pevzner, P. A. & Tesler, G. How to apply de Bruijn graphs to genome assembly. *Nature biotechnology* **29**, 987-991 (2011).

18 Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing in science & engineering* **9**, 90-95 (2007).

19 Cooper, G. M. & Hausman, R. E. The cell: A molecular approach. 2000. *ASM Press, Washington DC*, 467-519 (2007).

20 Tartaglia, G. G. & Vendruscolo, M. Proteome-level interplay between folding and aggregation propensities of proteins. *Journal of molecular biology* **402**, 919-928 (2010).

21 Huang, F. & Nau, W. M. A conformational flexibility scale for amino acids in peptides. *Angewandte Chemie International Edition* **42**, 2269-2272 (2003).

22 Barr, A. H. Global and local deformations of solid primitives. *ACM Siggraph Computer Graphics* **18**, 21-30 (1984).