

HeliXplore: A Python package for analyzing multi-strand helix deformations

2025-11-24

Ronnie Mondal [1,2] [ORCID: 0000-0003-4393-018X]

Valerie Vaissier Welborn [1,2,*] [ORCID: 0000-0003-0834-4441]

1 Department of Chemistry, Virginia Tech, Blacksburg, VA 24061

2 Macromolecules Innovation Institute, Virginia Tech, Blacksburg, VA 24061

*Email: vwelborn@vt.edu

Summary

Multi-stranded helices are a recurring structural motif in biomolecular systems, most prominently found in DNA and collagen (Watson and Crick 1953; Ramachandran and Kartha 1954). They also appear in synthetic polymers and macromolecules (ref). Multi-stranded helices can undergo local and global deformations that directly impact their function. Helix properties depend on strand length, strand composition, mutations and change in the environment, which can be modeled using classical Molecular Dynamics (MD). However, MD lacks a tool to systematically quantify multi-strand helix deformation across systems and conditions. In the past, people have used polysaccharide atom distances (Khatami et al. 2021), collagen cross-sectional triangles (Ravikumar and Hwang 2008), pitch, or principal axis measures (Zhang et al. 2006; Koneru et al. 2019). Although adequate for the system under investigation, these metrics are not generally useful and miss key local or collective distortions. Without local or inter-strand descriptors, comparisons across systems remain largely qualitative.

We present **HeliXplore**, an open-source Python package for the systematic and quantitative analysis of multi-strand helix deformation. Inspired by collagen but generalizable to any helical bundle, **HeliXplore** measures how helices deviate from their ideal geometry using backbone markers. **HeliXplore** runs calculations in three sections: Section 1 for intra-strand deformations in rise, radius and twist per atom or group of atoms and per time frame; Section 2 for axial shifts, inter-strand deformations, local deformations and helical regularity; and Section

3 for triple-helix deformations, using the size and shape of the cross-sectional triangle.

In practice, one only needs `HeliXplore.py`, the MD trajectory file (in TINKER `.arc` format or the standard RCSB `.pdb` format) and the number of strands to be able to run the code. The number of units in each strand is an optional input. Users can also input the atom names or atom types (for TINKER `.arc` format) to select backbone markers. `read_tinker_arc()` and `read_traj_pdb()` functions can be replaced to cater to other trajectory file formats. `HeliXplore` checks for the four required Python dependencies (`numpy`, `scipy`, `pandas` and `matplotlib`) before running the main code, and no other installations are required. For a detailed description of the inputs and examples, see the `README` file on GitHub. A shorter description is provided with `python HeliXplore.py --help`.

Statement of Need

`HeliXplore` provides the first open-source, Python-based implementation of a quantitative framework for analyzing poly-helix deformations. By resolving intra- and inter-helix deformations, `HeliXplore` enables atomic resolution of structural distortions that have previously eluded MD studies. More broadly, `HeliXplore` establishes a transferable methodology for analyzing helices, positioning it as a foundational tool for systematic comparisons across systems, conditions and force fields. Users of `HeliXplore` are free to modify the code and underlying mathematical formulations to adapt to their specific research needs.

Mathematics

Section I:

For a helix of length N , with $\mathbf{p}^{i,m}(t)$ coordinates of atom i in strand m over time t , the **helical axis vector** $\mathbf{v}_1^m(t)$ is determined via principal component analysis:

$$\left[\frac{1}{N} \sum_{i=1}^N (\mathbf{p}^{i,m}(t) - \mathbf{c}^m(t))(\mathbf{p}^{i,m}(t) - \mathbf{c}^m(t))^T \right] \mathbf{v}_1^m(t) = \lambda_1^m(t) \mathbf{v}_1^m(t)$$

where $\mathbf{c}^m(t)$ is the average coordinates of strand m at time t defined as:

$$\mathbf{c}^m(t) = \frac{1}{N} \sum_{i=1}^N \mathbf{p}^{i,m}(t)$$

and λ_1^m indicates the principal eigenvalue. The unit vector of $\mathbf{v}_1^m(t)$ is hereafter referred to as $\hat{\mathbf{v}}_1^m(t)$.

For intra-helix metrics, two consecutive units of a strand are chosen for rise and three units for the radius and twist. **Deviations in rise** for unit i at strand m are calculated as:

$$\delta_{\text{Rise}}^{i,m}(t) = ((\mathbf{p}^{i+1,m}(t) - \mathbf{p}^{i,m}(t)) \cdot \mathbf{v}_1^m(t) - \text{Rise}^{i,m}(0)) / \text{Rise}^{i,m}(0)$$

Deviations in radius are calculated using the circumradius $R^{i,m}$ as:

$$\delta_{\text{Radius}}^{i,m}(t) = (\mathbf{R}^{i,m}(t) - \mathbf{R}^{i,m}(0)) / R^{i,m}(0)$$

where:

$$R^{i,m}(t) = \frac{\|\mathbf{p}^{i,m}(t) - \mathbf{p}^{i-1,m}(t)\| \|\mathbf{p}^{i+1,m}(t) - \mathbf{p}^{i-1,m}(t)\| \|\mathbf{p}^{i,m}(t) - \mathbf{p}^{i+1,m}(t)\|}{4 \times \frac{1}{2} \|(\mathbf{p}^{i,m}(t) - \mathbf{p}^{i-1,m}(t)) \times (\mathbf{p}^{i+1,m}(t) - \mathbf{p}^{i-1,m}(t))\|}$$

Deviations in twist for consecutive set of three atoms on strand m are calculated as the angle between the normals of the planes defined by units $(i-1, i, i+1)$ ($\mathbf{n}_a^{i,m}(t)$) and $(i, i+1, i+2)$ ($\mathbf{n}_b^{i,m}(t)$) as:

$$\delta_{\text{Twist}}^{i,m}(t) = \left(\arccos \left(\frac{\mathbf{n}_a^{i,m}(t) \cdot \mathbf{n}_b^{i,m}(t)}{\|\mathbf{n}_a^{i,m}(t)\| \|\mathbf{n}_b^{i,m}(t)\|} \right) - \text{Twist}^{i,m}(0) \right) / \text{Twist}^{i,m}(0)$$

Intra-strand deviations ($\kappa^{i,m}(t)$) can then be calculated from the output files as:

$$\langle \kappa^{i,m}(t) \rangle = w_{\text{Rise}} * \langle \delta_{\text{Rise}}^{i,m}(t) \rangle + w_{\text{Radius}} * \langle \delta_{\text{Radius}}^{i,m}(t) \rangle + w_{\text{Twist}} * \langle \delta_{\text{Twist}}^{i,m}(t) \rangle$$

where w_{Rise} , w_{Radius} and w_{Twist} are user-defined weights during post-processing and the average can be taken over units, strands, time or a combination of the three.

Section II:

Local deformations are calculated using a sliding window approach which quantifies local deviations after optimal superposition. A window around unit i is defined and then aligned to the reference structure using the Kabsch algorithm (Kabsch 1976) as:

$$\mathbf{q}^{i,m}(t) = \mathbf{K}(\mathbf{p}^{i,m}(t), \mathbf{p}^{i,m}(0))$$

where $\mathbf{q}^{i,m}(t)$ is the position of atom i on strand m at time t after alignment, and \mathbf{K} is Kabsch alignment operator, which returns the optimally rotated and

translated version of $\mathbf{p}^{i,m}(t)$ aligned to $\mathbf{p}^{i,m}(0)$. The local deformation is then the scalar positional deviation, calculated as

$$d_{\text{local}}^{i,m}(t) = \|\mathbf{q}^{i,m}(t) - \mathbf{p}^{i,m}(0)\|$$

Helical regularity for a strand m is calculated as:

$$\mathcal{R}^m(t) = \frac{\sigma_{\text{Rise}}^i(t)}{\langle |\text{Rise}^{i,m}(t)| \rangle_i} + \frac{\sigma_{\text{Radius}}^i(t)}{\langle |\text{Radius}^{i,m}(t)| \rangle_i} + \frac{\sigma_{\text{Twist}}^i(t)}{\langle |\text{Twist}^{i,m}(t)| \rangle_i}$$

where σ^i is the standard deviation of the metric over i .

Axial shift of one strand m with respect to another strand n along the central axis is calculated by first reorienting consistently to prevent sign ambiguity of direction. $\tilde{\mathbf{v}}_1^m(t)$ is defined by enforcing $\hat{\mathbf{v}}_1^m \cdot \hat{\mathbf{v}}_1^n > 0 \forall m, n$ and $m \neq n$. If found negative, $\tilde{\mathbf{v}}_1^n(t) = -\hat{\mathbf{v}}_1^n$, otherwise $\tilde{\mathbf{v}}_1^n(t) = \hat{\mathbf{v}}_1^n$. The axial shift is calculated as:

$$s^{mn}(t) = |\langle \tilde{\mathbf{v}}_1^m(t) \rangle_m \cdot (\mathbf{c}^m(t) - \mathbf{c}^n(t))|$$

Deviations in axis angles are calculated between the unit vectors of the principal axes of the two strands m and n as:

$$\delta_\theta^{mn}(t) = \frac{|\arccos(\hat{\mathbf{v}}_1^m(t) \cdot \hat{\mathbf{v}}_1^n(t)) - \theta^{mn}(0)|}{\theta^{mn}(0)}$$

Deviations in axis distances are calculated using the perpendicular distance between the axes of strands m and n as:

$$\delta_d^{mn}(t) = (|(\mathbf{c}^m(t) - \mathbf{c}^n(t)) \cdot (\hat{\mathbf{v}}_1^m(t) \times \hat{\mathbf{v}}_1^n(t))| - d^{mn}(0)) / d^{mn}(0)$$

Deviations in centroid distances are calculated using the distance between the average coordinates of strands m and n as:

$$d_c^{mn}(t) = (\|\mathbf{c}^m(t) - \mathbf{c}^n(t)\| - c^{mn}(0)) / c^{mn}(0)$$

Section III:

For triple helices, one atom on each strand is taken to form a triangular cross-section at that unit.

Deviations in area ($\text{Area}^i(t)$) are calculated from the area of the triangle as

$$\text{Area}^i(t) = \left(\frac{1}{2} \|(\mathbf{p}^{i,2}(t) - \mathbf{p}^{i,1}(t)) \times (\mathbf{p}^{i,3}(t) - \mathbf{p}^{i,1}(t))\| - \text{Area}^i(0) \right) / \text{Area}^i(0)$$

Deviations in shape ($\text{Shape}^i(t)$) are calculated from the normalized isoperimetric ratio (IP) as

$$\text{Shape}^i(t) = \left(\frac{4\pi \text{Area}^i(t)}{P^i(t)^2} - \text{IP}^i(0) \right) / \text{IP}^i(0)$$

where $P^i(t)$ is the corresponding perimeter.

Inter-strand deviations ($\zeta^i(t)$) can then be calculated for a triple helix from the output files as

$$\zeta^i(t) = w_{\text{Area}} * \langle \text{Area}^i(t) \rangle + w_{\text{Shape}} * \langle \text{Shape}^i(t) \rangle$$

where w_{Area} and w_{Shape} are user-defined weights during post-processing and the average can be done over units or time or both.

The first frame is taken as the reference. The reference can be changed by appending a new reference frame to the beginning of the input MD trajectory. All mathematical details are also outlined as comments in the code.

Acknowledgements

The authors thank the National Institute of Health, National Institute of General Medical Sciences, under award number R35-GM150409 and Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper.

References

- Kabsch, Wolfgang. 1976. “A Solution for the Best Rotation to Relate Two Sets of Vectors.” *Foundations of Crystallography* 32 (5): 922–23. <https://doi.org/10.1107/S0567739476001873>.
- Khatami, Mohammad Hassan, William Barber, and Hendrick W de Haan. 2021. “Using Geometric Criteria to Study Helix-Like Structures Produced in Molecular Dynamics Simulations of Single Amylose Chains in Water.” *RSC Advances* 11 (20): 11992–2002. <https://doi.org/10.1039/D1RA00071C>.
- Koneru, Jaya Krishna, Xiao Zhu, and Jagannath Mondal. 2019. “Quantitative Assessment of the Conformational Heterogeneity in Amylose Across Force Fields.” *Journal of Chemical Theory and Computation* 15 (11): 6203–12. <https://doi.org/10.1021/acs.jctc.9b00630>.
- Ramachandran, Go No, and Gopinath Kartha. 1954. “Structure of Collagen.” *Nature* 174 (4423): 269–70. <https://doi.org/10.1038/174269c0>.

- Ravikumar, Krishnakumar M, and Wonmuk Hwang. 2008. “Region-Specific Role of Water in Collagen Unwinding and Assembly.” *Proteins: Structure, Function, and Bioinformatics* 72 (4): 1320–32. <https://doi.org/10.1002/prot.22026>.
- Watson, James D, and Francis HC Crick. 1953. “Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid.” *Nature* 171 (4356): 737–38. <https://doi.org/10.1038/171737a0>.
- Zhang, Qingmin, Zhenyu Lu, Hao Hu, Weitao Yang, and Piotr E Marszalek. 2006. “Direct Detection of the Formation of v-Amylose Helix by Single Molecule Force Spectroscopy.” *Journal of the American Chemical Society* 128 (29): 9387–93. <https://doi.org/10.1021/ja057693+>.