

Visual Analytics for Macromolecular Science

Dissertation by

Deng Luo

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

King Abdullah University of Science and Technology

Thuwal, Kingdom of Saudi Arabia

© August 2025

Deng Luo

All rights reserved

 <https://orcid.org/0000-0003-4610-8730>

ABSTRACT

This dissertation presents a suite of scientific visual analytics systems that tightly couple large-scale computational modeling—including both AI-powered inference and traditional molecular simulations—with interactive, scalable visualization. By enabling experts to directly explore, interpret, and refine complex macromolecular data generated from intensive computation, these systems bridge the persistent “non-optimizable gap” that automation alone cannot overcome. This integrative approach empowers more effective human–machine collaboration, accelerates scientific discovery, and addresses challenges that demand nuanced human insight in molecular science.

Three core contributions anchor this work: DiffFit, a visually-guided, differentiable fitting framework that unites automated optimization with expert-driven inspection for assembling protein structures into cryo-EM volumes; ProteinCraft, an integrative visualization system that combines structural, interaction, and multivariate attribute visualization to accelerate and rationalize AI-driven protein binder discovery; and SynopFrame, a synchronized multiscale framework for analyzing dynamic DNA nanotechnology simulations, revealing temporal and spatial patterns that traditional methods often miss.

By orchestrating advances in information visualization, graph analytics, and molecular rendering, these systems bridge structural, functional, and temporal scales, empowering users to filter, compare, and iteratively refine molecular designs with greater accuracy and efficiency. Through comprehensive use cases and expert evaluations, this dissertation demonstrates that integrative visual analytics not only improves the involved workflows’ efficiencies but also transforms the process of exploration, validation, and interpretation in complex macromolecular systems. Ultimately, this work establishes visual analytics as a critical mediator in the AI era, enabling scientists to overcome the limitations of both black-box AI and

manual analysis, and accelerating the translation of computational advances into impactful scientific and biomedical discoveries.

ACKNOWLEDGEMENTS

It is with sincere gratitude that I present this dissertation, the culmination of my journey with the Nanovis Group at KAUST. These years have been marked by exploration, collaboration, and personal growth.

My deepest thanks go to my supervisor, Professor Ivan Viola, whose vision, guidance, and encouragement have profoundly shaped my development as a scientist and individual. I am also deeply grateful to my co-supervisor, Dr. Tobias Isenberg, for his insightful advice and unwavering support throughout this journey.

I thank the Nanovis Group for the culture of cooperation and innovation, and especially my close collaborator, Dr. Ondřej Strnad, for his technical expertise and steadfast camaraderie. I am thankful for the diverse and supportive community at KAUST, which has made my experience both intellectually stimulating and personally meaningful. Above all, I am indebted to my family—my parents for their unwavering encouragement, and my brother for his patience and support.

To all mentors, collaborators, friends, and family who have accompanied me along the way: THANK YOU. This dissertation reflects your generosity, wisdom, and belief in my potential.

PERSONAL STATEMENT

The Power of Seeing

Ever since my earliest days in school, I discovered a simple truth: if I could picture what I was reading in a textbook, I could understand it; if I could not, the words remained impenetrable. In the pages of biology books, vivid “movies” would form almost effortlessly in my mind—double helix Deoxyribonucleic Acid (DNA) unwinding for transcription, ribosomes attaching to messenger RNA (mRNA) to initiate translation, amino acids linking together and folding into functional proteins. When a concept could be imagined and visualized, comprehension often arrived in an instant, as if the invisible had suddenly become tangible.

Over time, this intuition crystallized into a conviction: great visualization is not merely a tool for communication, but a catalyst for discovery. It is an accelerant that transforms abstract concepts and data into “aha!” moments, not only for established scientists but also for students, educators, and anyone curious about the natural world. I have come to believe that the most profound insights and the most effective learning often arrive the moment we are able to see them.

This conviction was reinforced and deepened by my exposure to the work of Richard Feynman,¹ whose talent for explaining intricate science has long inspired me. In the BBC series *Fun to Imagine*,² Feynman confronts the fundamental limits of explanation through what I like to call his “alien challenge.” When asked why magnets repel,³ he unpacks the question by imagining how to explain it to someone completely unfamiliar with our world. His reasoning, which progresses

¹en.wikipedia.org/wiki/Richard_Feynman

²The official record on BBC is at bbc.co.uk/programmes/p0198zc1. One full video record on the Internet is at youtu.be/nYg6jzotiAc.

³It starts at 14:53 in the full video. This section is also available as a 7.5-minute standalone video at youtu.be/Q11L-hX027Q.

from the mundane (“Aunt Minnie is in the hospital”) to the deepest laws of physics, reveals a profound insight: we can only take things for granted because we live and interact within a world we can see and touch. In contrast, when faced with phenomena outside our lived experience—when we are, so to speak, “aliens” to a new scale or domain—imagination and visualization become our only bridges to understanding.

This realization, more than anything else, has shaped my academic journey and career. Years later, during an open house event, I encountered the atomic model of human immunodeficiency virus (HIV) presented by my future PhD supervisor, Prof. Ivan Viola. According to my friends who told me later, I stood transfixed before that exhibit far longer than at any other display—unaware of the time passing. The impact was immediate and profound. In that moment of seeing, my path became clear: I knew I wanted to pursue my PhD under his guidance, driven by the conviction that seeing leads to understanding and that visualization is not merely a supplement to science, but an indispensable foundation for insight and discovery.

Contents

Abstract	2
Acknowledgements	4
Personal Statement	5
List of Abbreviations	10
List of Figures	11
List of Tables	12
1 Introduction	13
1.1 My PhD Journey	13
1.2 Research Overview and Research Question	14
1.3 Technical Motivation: The Non-Optimizable Gap	14
1.4 Scope and Contributions	16
1.5 Authorship Statement	18
1.6 Dissertation Structure	20
2 Background	21
2.1 Structural Biology and Cryo-EM	21
2.2 Protein Binder Design	22
2.3 DNA Nanotechnology	23
3 State of the Art	25
3.1 Structural Biology: From Experimental Determination to Integra- tive Modeling	25
3.2 AI-Driven Protein Design: Generative Models and Human-AI Team- ing	31
3.3 DNA Nanotechnology and Molecular Dynamics Visualization . . .	34
3.4 Summary and Research Gaps	37
4 DiffFit: Visually-Guided Differentiable Fitting of Molecule Struc- tures to a Cryo-EM Map	39
4.1 Introduction	40

4.2	Method	42
4.3	Implementation	55
4.4	Use case scenarios	55
4.5	Feedback	61
4.6	Limitations and Future Work	63
4.7	Discussion	65
5	ProteinCraft: Integrative visualization of protein attributes and residue interactions in the AI era	67
5.1	Introduction	68
5.2	Method	74
5.3	Implementation	83
5.4	Use Case Scenarios	84
5.5	Discussion	87
6	SynopFrame: Multiscale time-dependent visual abstraction framework for analyzing DNA nanotechnology simulations	89
6.1	Introduction	90
6.2	Motivation, approach, and prerequisites	92
6.3	Design of the SynopFrame	97
6.4	DNA-nano molecular dynamics simulations (MDS) exploratory analysis case study	112
6.5	Further feedback	114
6.6	Limitations and future development	115
6.7	Discussion	116
7	Conclusions and Outlook	118
7.1	Reflections and Lessons Learned: Discover the Non-optimizable Gap Through Visualization	119
7.2	Generalization and Human-in-the-Loop Design: The Broader Impact of DiffFit	121
7.3	Differentiability in Visualization: The Perspective of SynopSpace .	122
7.4	Toward Standardization and Benchmarking in Visualization Research	123
7.5	Summary	125
7.6	Conclusion and Outlook	126
	References	126
	Appendices	141

A	Appendix for DiffFit	141
A.1	Detailed benchmark results for use case scenario 1—Fit a single structure	141
A.2	Details on the user feedback sessions	141
B	Appendix for SynopFrame	142
B.1	Detailed description of SynopPoints	142
B.2	DNA-nano design simulation	146
B.3	OxDNA2’s model geometry	146
B.4	Houdini-specific implementation	147
B.5	Transitions	149
B.6	SynopFrame performance	149
B.7	The <code>SynopSpace.hb</code> format	150
B.8	Case Study 2: An RNA tile design	151
B.9	Statistical scalar plots	152
B.10	Algorithms	153
B.11	User feedback details	153
B.12	Comparison to Miao et al.’s DNA origami abstraction space DimSUM	154

LIST OF ABBREVIATIONS

AF2ig	AlphaFold2 initial guess
CA	alpha-carbon
cryo-EM	cryogenic electron microscopy
CSV	Comma-separated values
CUDA	Compute Unified Device Architecture
DNA	Deoxyribonucleic Acid
GPU	graphics processing unit
HIV	human immunodeficiency virus
JSON	JavaScript Object Notation
MDS	molecular dynamics simulations
mRNA	messenger RNA
NT	nucleotide
PAE	predicted aligned error
PCA	principal component analysis
PDB	Protein Data Bank
RMSD	root-mean-square deviation
UMAP	Uniform Manifold Approximation and Projection

LIST OF FIGURES

1.1	Automation spectrum and the <i>non-optimizable gap</i>	14
2.1	DNA-nano concept	24
4.1	DiffFit workflow	42
4.2	DiffFit Clustering and filtering	45
4.3	DiffFit fitting result table browser based on ChimeraX	48
4.4	DiffFit fitting result interactive spatial viewer	53
4.5	Fitting a single structure for 6WTI	55
4.6	Compositing a protein	59
4.7	Unknown density identification	60
5.1	ProteinCraft system overview	75
5.2	Use ProteinCraft in protein binder design workflow	76
5.3	Distribution of inter chain predicted aligned error (PAE) values for generated designs in ProteinCraft in two rounds	85
5.4	ProteinCraft case study 2: Iterative binder design	86
6.1	SynopFrame dashboard shows an icosahedron design (6540nu- cleotide (NT)s) in various different representations	90
6.2	Inspirations for fundamental representations in SynopFrame from existing tools	95
6.3	Schematic view of the SynopSpace	100
6.4	Zoomed-in views in SynopFrame for an icosahedron design	102
6.5	Transformations in the Schematic3D algorithm	105
6.6	Three frames from an animation that showcase the break-up of a structure in SynopFrame.	109
6.7	Cube case study	110
6.8	User feedback for the effectiveness of SynopFrame	116
B.1	Example for the use of a different color map in SynopFrame for people with color deficiencies	148
B.2	SynopFrame case study for an RNA tile design	149
B.3	Statistical scalar plots in SynopFrame	152

LIST OF TABLES

4.1	DiffFit performance results for fitting a single structure	58
4.2	DiffFit performance results for identifying unknown structures . .	61
6.1	Summary of the six key representations in SynopSpace	104
A.1	DiffFit performance results for fitting a single structure	155
B.1	Comparison between SynopSpace and Miao et al.'s [67] DNA origami abstraction space.	159

Chapter 1

Introduction

1.1 My PhD Journey

Guided by a belief in the “power of seeing”, I transitioned from a bioscience master’s program into computer science for scientific visualization. To contribute meaningfully, I effectively completed a computer-science master’s curriculum while learning to connect biological questions with visualization methods. Throughout, I sought to bridge communities—biologists, computer scientists, and visualization researchers.

Early in my PhD, I helped build one of the first atomistic models of SARS-CoV-2,¹ introducing me to the mesoscale and to collaborations across modeling, structural biology, and art. Our follow-up visualization of the SARS-CoV-2 life cycle won the *Computer Graphics Forum* cover contest.² Mentoring high-school interns who modeled T4,³ chloroplasts,⁴ thylakoids,⁵ and more deepened my interest in education and produced assets later used in VOICE [46].

SynopSpace—a conceptual space for DNA visualization—spanned much of my PhD and seeded ideas that recur in later systems. Through collaborations on inverse procedural modeling, I encountered differentiable compositing [85] and adapted it to cryo-EM fitting (DiffFit). Finally, ProteinCraft unified multivariate attributes and 3D structure to support AI-driven binder design. Together, these projects form the backbone of this dissertation.

¹nanovis.org/SARS-CoV-2-model.html

²vcg.isti.cnr.it/cgf/winner.php?year=2021,
cg.tuwien.ac.at/news/2021-01-07-Nanographics-wins-CGF-Cover-Contest

³nanovis.org/T4-model.html

⁴nanovis.org/Chloroplast-model.html

⁵nanovis.org/Thylakoid-model.html

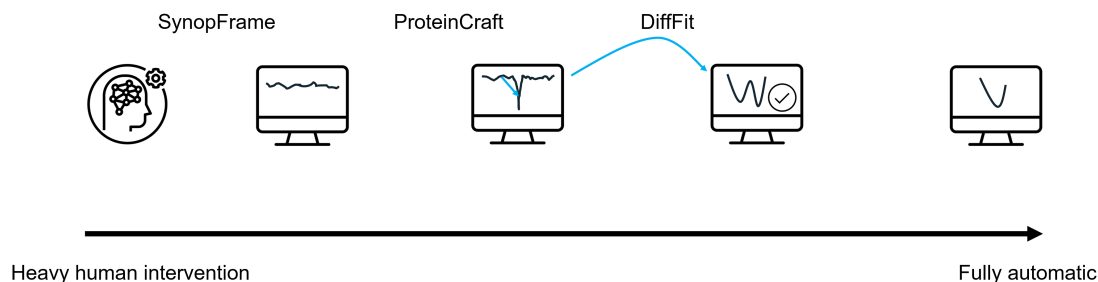


Figure 1.1: Automation spectrum and the *non-optimizable gap*. Toward the fully automatic end, a single smooth optimum can be reached by standard optimization. Moving left, multiple plausible minima require human judgment and selection. Further left, narrow basins and rough paths make robust initialization essential. At the other end, heavy human intervention is needed for open-ended questions without a single, fixed objective. Visual analytics helps humans steer algorithms and evaluate outcomes across this spectrum.

1.2 Research Overview and Research Question

Many problems in macromolecular science sit between automation and expert reasoning. I refer to this persistent region as the *non-optimizable gap*: the residual distance between what algorithms can deliver—given available objectives, priors, and data—and the scientific insight required to decide, refine, or explain results. In this gap, objectives are incomplete or multi-faceted, landscapes are rough with many acceptable minima, or the task itself is open-ended and context dependent. Visualization is therefore not an afterthought but the interface that enables human–AI teaming to bridge this gap [72].

Research question. *How can scientific visual analytics systems be designed to tightly couple automated modeling with interactive visualization so that experts can steer, audit, and refine computation—thereby bridging the non-optimizable gap (Figure 1.1) across diverse macromolecular problems?*

1.3 Technical Motivation: The Non-Optimizable Gap

Recent advances in AI and large-scale modeling have transformed macromolecular research, from structure prediction and generative design to mesoscale simulation. Yet these capabilities expose a region where automation alone is insufficient: objectives are ill-specified or competing, landscapes are non-convex with many

acceptable solutions, and data can be noisy or incomplete. In this *non-optimizable gap* (Figure 1.1), expert knowledge is essential to set context, impose constraints, adjudicate among alternatives, and ultimately extract meaning from large computational ensembles.

Three archetypal manifestations of the gap motivate the systems developed in this dissertation: (i) **Multiple good minima**—e.g., situations where several viable solutions coexist, as in selecting the fitting result from DiffFit; (ii) **Narrow basins and rough paths**—e.g., scenarios where reaching a workable region requires laborious manual intervention, as in cryo-EM model fitting prior to DiffFit and in the protein design workflows; and (iii) **Open-ended questions**—e.g., contexts where no single fixed objective exists, such as diagnosing failed DNA nanostructure assembly or exploring conformational switching.

How the three systems bridge the gap

DiffFit (from narrow basins & rough paths to decision making among multiple solution candidates). Before DiffFit, users manually rotated and roughly positioned subunits inside cryo-EM volumes to reach a basin where local refiners succeed. DiffFit replaces this manual coarse step with visually guided, differentiable optimization on GPUs, robust initialization sampling, and loss formulations that make difficult basins reachable. The expert remains in the loop where it matters most—*evaluating and selecting* among high-quality fits—thus shifting human effort from low-level manipulation to high-level judgment.

ProteinCraft (navigate through narrow basins & rough paths). In protein binder design, there is no single global objective that guarantees stability and affinity. ProteinCraft links multivariate attributes, residue–residue interactions, and 3D structures so experts can focus computation on promising regions of the landscape by selecting promising candidates, performing local “jittering,” and iteratively redesigning the structure. Visual analytics thus *moves the algorithmic focus from a rough landscape to the sweet spot of a good minimum* by steering

which candidates to expand and which evidence to trust.

SynopFrame (open-ended questions). When a DNA design fails to assemble or toggles between conformations, there is no single target pose or scalar objective. SynopFrame provides a *visualization space*—an array of synchronized, gradually abstracted views across granularity, visual idiom, and layout axes—that supports exploratory analysis, hypothesis formation, and localization of failure modes in large, heterogeneous simulations.

1.4 Scope and Contributions

The central vision of this dissertation is to advance the field of scientific visual analytics by developing integrative visualization systems that bridge the “non-optimizable gap” in contemporary macromolecular research. This gap arises wherever computational modeling and AI-driven inference reach their limits—where expert intuition, interactive exploration, and visual reasoning become indispensable for meaningful scientific progress.

The scope of my research lies at the intersection of structural biology, computational protein design, DNA nanotechnology, data visualization in general, and specifically molecular visualization. The technical solutions that I develop in this dissertation respond to the challenges of interpreting large-scale, heterogeneous, and high-dimensional molecular data that resist full automation. Throughout, the focus remains on empowering scientists to explore, validate, and refine complex molecular systems—combining the strengths of automated computation with human insight and creativity.

This dissertation integrates results from the following core research publications:

- D. Luo, Z. Alsuwaykit, D. Khan, O. Strnad, T. Isenberg, and I. Viola, “**DiffFit**: Visually-guided differentiable fitting of molecule structures to a cryo-em map,” IEEE Transactions on Visualization and Computer Graphics, vol. 31, no. 1, pp. 558–568, 2025.

The early development of this work was presented as a spotlight at the ICML 2024 Workshop on Differentiable Almost Everything. After journal publication, an extended version of DiffFit was also presented at the 2nd IAS Symposium on Biological Cryo-EM 2025 (Hong Kong), where it was awarded Best Short Talk.

- D. Luo, C. Feinauer, L. Song, T. Isenberg, and I. Viola, “***ProteinCraft***: *Integrative Visualization of Protein Attributes and Residue Interactions in the AI Era*,” In preparation.
- D. Luo, A. Kouyoumdjian, O. Strnad, H. Miao, I. Barisic, T. Isenberg, and I. Viola, “***SynopFrame***: *Multiscale time-dependent visual abstraction framework for analyzing DNA nanotechnology simulations*,” Computers & Graphics, in revision.

The visuals developed for SynopFrame were recognized with second place in the DESIGN X BIOINFORMATICS student competition.⁶

Core Contributions

The principal contributions of this dissertation are:

1. **A visually-guided, differentiable fitting framework (DiffFit) for cryo-EM model assembly.**

DiffFit integrates a novel gradient-based optimization algorithm on GPU with interactive, expert-driven inspection to accelerate and improve the accuracy of fitting protein subunits into experimental cryo-EM volumes. The approach introduces several technical innovations, including a preprocessing technique that generates an array of smoothed volumes, an efficient initialization sampling strategy, a negative space formulation, and a robust loss function. Combined with human-in-the-loop evaluation and seamless integration with established molecular visualization tools, DiffFit reduces

⁶<https://cellmicrocosmos.org/conferences/DesXBioInf2022/winners/>

manual effort while enhancing both speed and reliability of structural fitting workflows. The open-source plugin we implemented for ChimeraX has been downloaded more than 1,800 times,⁷ reflecting its broad adoption by the structural biology community.

2. **An integrative visual analytics system (ProteinCraft) for AI-driven protein binder design.**

ProteinCraft enables researchers to analyze, filter, and refine large pools of AI-generated binder designs through coordinated 2D/3D visualization, novel encodings for residue interactions, and multi-level selection and ranking. By linking structure, sequence, and functional attributes, ProteinCraft makes the generative process more transparent and guides the identification of high-affinity designs, significantly improving the in-silico success rate.

3. **A multiscale, time-dependent abstraction framework (SynopFrame) for DNA nanotechnology simulations.**

SynopFrame addresses the challenge of analyzing complex, dynamic simulation data by organizing representations along structural, schematic, and temporal axes. The dashboard system facilitates synchronized, multi-view analysis of design flaws and conformational changes, empowering experts to detect failure modes and interpret molecular dynamics at scale.

1.5 Authorship Statement

All manuscripts and systems presented in this dissertation were developed during my PhD research at King Abdullah University of Science and Technology (KAUST), under the supervision of Professor Ivan Viola and co-supervision of Dr. Tobias Isenberg. I am the primary contributor and first author for each of the core works described herein. Viola and Isenberg provided ongoing supervision and support, contributed to high-level research vision, and assisted in manuscript review and refinement.

⁷As of July 2025: <https://cxtoolshed.rbvi.ucsf.edu/apps/diffit>

DiffFit: This project originated from an inverse procedural modeling initiative at the mesoscale, which was co-conceived by Ivan Viola and myself. In the early exploratory phase, together with Zainab Alsuwaykit and Dawar Khan, we investigated large cryo-EM datasets of axonemes. Viola then introduced me to a key reference on differentiable compositing algorithms from the computer graphics community [85]. With his guidance, I studied the algorithm in detail and quickly recognized its potential for repurposing in the context of fitting protein structures into cryo-EM volumes. I subsequently developed the core differentiable optimization algorithm. Ondřej Strnad contributed to the development of the ChimeraX plugin. Alsuwaykit and Khan were responsible for the related work review and manuscript writing for that section. I authored the remainder of the paper, with Viola providing guidance on the mathematical formalism, and Tobias Isenberg offering careful revision and editorial input throughout the manuscript.

ProteinCraft: This project originated when Le Song, who leads a research team and company developing generative AI models for protein design, approached our group for collaboration. I explored state-of-the-art protein binder design methods and formulated the research concept through meetings and feedback with Christoph Feinauer, Song, and Viola. I then implemented the visualization system, integrating Tulip [6] and ChimeraX [80], and evaluated the approach by designing binders on benchmark datasets. I authored the full manuscript, with Isenberg providing careful revision and editorial feedback throughout. Viola oversaw the entire project.

SynopFrame: This project originated from a visit by Ivan Barišić to KAUST. In early discussions with Viola, Barišić, Haichao Miao, and myself, we explored multiple directions, including constructing a DNA nanostructure database, visualizing assembly processes, and developing different structural views for the dynamic behaviors of DNA nanostructures. Alexandre Kouyoumdjian and Ondřej Strnad assisted in creating the initial prototype demo visuals. I subsequently proposed the visualization space concept to integrate these ideas and later implemented

all views and interactions in Houdini,⁸ focusing on the visualization of molecular dynamics trajectories of DNA nanostructures. Kouyoumdjian contributed to the initial draft, specifically refining the introduction, related work, and parts of the methods section. Miao wrote the segment on abstraction spaces in visualization within the related work. I authored the remainder of the manuscript, with extensive revision and editorial feedback from Isenberg. Both Viola and Isenberg contributed to the restructuring and revision of the paper for resubmission. Viola oversaw the entire project.

1.6 Dissertation Structure

This dissertation is organized into three main parts. In the first part, I provide an overview of the research motivation, technical background, and the broader scientific context. I introduce the concept of the non-optimizable gap in macromolecular science and outline the scope, contributions, and structure of the thesis.

In the second part, I present three core research works, each as a dedicated chapter. These chapters are based on my original manuscripts—DiffFit [61], ProteinCraft, and SynopFrame—which I have adapted and slightly modified for narrative coherence and thematic consistency. I introduce and contextualize each chapter to highlight its relevance to the overarching theme of bridging the non-optimizable gap through scientific visual analytics.

In the final part, I synthesize key lessons and cross-cutting themes from the three systems, reflecting on the broader impact of visual analytics in enabling human–AI collaboration and accelerating discovery in molecular science. I conclude the dissertation with perspectives on future directions and the evolving role of visualization in computational biology.

⁸<https://www.sidefx.com/>

Chapter 2

Background

To provide the necessary context for the systems and applications explored in this dissertation, I review in this chapter the scientific and technical foundations of three domains central to my research: structural biology and cryo-EM, protein binder design, and DNA nanotechnology. Each section offers a concise background tailored to the workflows and challenges addressed by the visual analytics systems developed in subsequent chapters.

2.1 Structural Biology and Cryo-EM

We begin with an overview of structural biology and cryo-EM, which underpin much of the experimental data and modeling challenges addressed by the DiffFit system. Structural biology employs various techniques to understand how atoms are arranged in macromolecular complexes, ranging from 60 kDa (i.e., 4,472 atoms; [PDB 6NBD](#) [41]) to 50,000 kDa (3,163,608 atoms; [PDB 8J07](#) [120]). These techniques are essential for the study of processes in living cells—cryogenic electron microscopy (cryo-EM) being a particularly powerful one [8, 63, 58]. With cryo-EM, bioscientists can capture images of flash-frozen biological specimens using an electron microscope, preserving their natural structure without the interference of staining or fixing [21], which would otherwise interfere with the sample. These images are then used to construct cryo-EM 3D volumes or maps using the *single particle method*, which aligns thousands of projections from structurally identical molecular instances into a single map using the Fourier slice-projection theorem. This map represents the electron density of the sample, which can be used to infer the atom positions within the molecule.

Subsequently, the bioscientists need to build accurate atomistic or molecular models that match the electron density map obtained from the cryo-EM process to gain insight into molecular function and interactions. This process involves mapping or fitting known sub-molecules into their corresponding positions within the map. The objective is to achieve an optimal correspondence between the model and the experimental or simulated volume, revealing the organization of molecules in 3D space, including single molecules, complexes, and the placement of small molecules and ligands into binding sites. Molecular models are available in the Protein Data Bank (PDB, rcsb.org), accessible in various formats such as PDB, Crystallographic Information File (CIF), and mmCIF (macromolecular CIF). As of now, the fitting is typically achieved through manual placement, alignment, and comparison with the density maps. The manual nature of this process makes it time-consuming and tedious, and can only be performed by expert biologists. To address this challenge, numerous approaches have been developed to automate the fitting process, which largely focus on image registration as the foundation and explore methods to streamline 3D model construction, as I review in the next chapter.

2.2 Protein Binder Design

Next, we turn to the scientific foundations of protein binder design, a rapidly advancing field at the intersection of molecular biology and artificial intelligence that motivated the development of ProteinCraft. Proteins—vital building blocks of life—are chains of amino acids that fold into intricate three-dimensional shapes, and this structure underpins their diverse functions such as catalyzing reactions or mediating molecular recognition. Protein “binders” are specialized proteins engineered to bind specific targets with high affinity; the goal of binder design is to identify a sequence that folds stably and then binds to the target molecule via non-covalent interactions (such as hydrogen bonds, pi stacking, and electrostatic contacts). Historically, researchers have relied on labor-intensive, large-scale

experimental screening of carefully crafted libraries to discover such binders [20]. Recently, AI-driven pipelines have dramatically accelerated the design process: RFDiffusion [124] can generate *de novo* binder backbones for a given target surface, ProteinMPNN [22] fills in these backbones with plausible amino acid sequences, and AlphaFold2 initial guess (AF2ig) [11] rapidly evaluates whether the proposed binder indeed adopts its intended structure—assessing both stability and binding affinity. Despite these individual advances, however, current best practices often require the generation of thousands of initial candidates, only a small fraction of which pass *in-silico* filtering, which then are subject to *in-vitro* experimental validation, and the entire workflow remains a “black box.” We thus need more controllable (i.e., interactive and visual) methods to fully unlock the potential of computational binder design.

2.3 DNA Nanotechnology

We introduce DNA nanotechnology—especially DNA origami—as the domain underpinning SynopFrame’s simulation and visualization challenges. Here, DNA is used as a programmable construction material: designed sequences self-assemble via Watson–Crick base pairing and hydrogen bonding to form targeted 3D shapes at the nanoscale [31, 96]. Three main paradigms exist—*DNA origami*, *single-stranded tiles*, and *multi-stranded tiles* [121]. We focus on DNA origami because a long *scaffold* folded by many short *staples* yields the richest structural complexity and the most demanding visualization tasks. (Most techniques described here transfer to the other two paradigms.) A brief overview of the design and the simulation models we use appears in section B.2.

As illustrated in Figure 2.1, DNA uses four nucleotides (A, T, C, G). Synthetic strands with specified sequences and directionality are assembled so that complementary segments pair to form helices. DNA origami exploits this programmability: the scaffold provides continuity while staples bridge distant scaffold regions and introduce crossovers that define global geometry. These primitives pro-

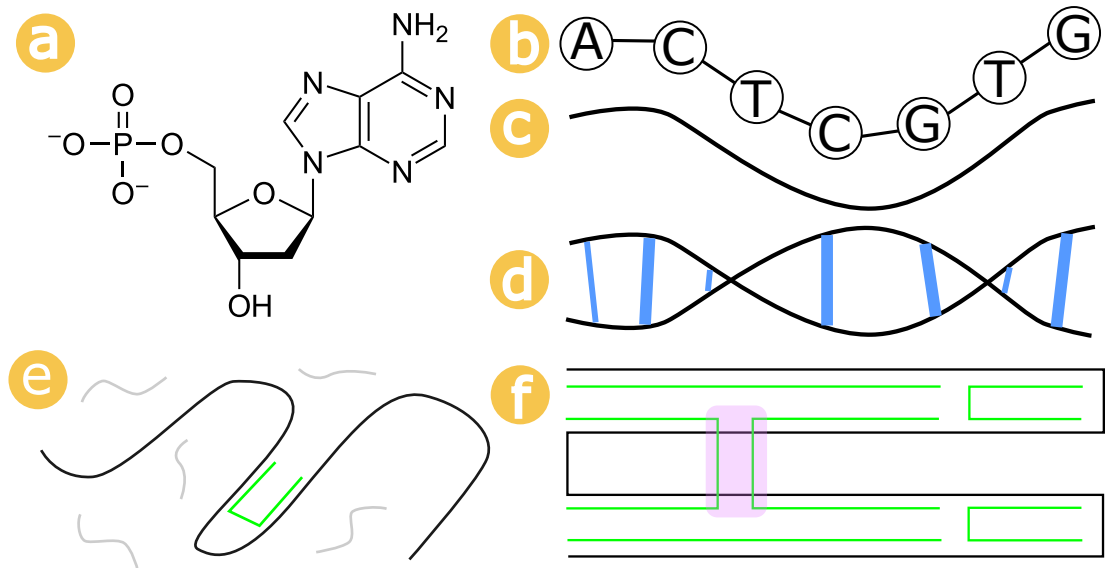



Figure 2.1: Conceptual build-up of DNA origami: (a) a nucleotide ([image](#) ); (b) covalently linked nucleotides form a strand; (c) polyline abstraction; (d) complementary strands hybridize into a double helix (A–T with two, C–G with three H-bonds); (e) a long scaffold (black) is folded by short staples (green); (f) the designed 3D shape emerges with *crossovers* connecting helices (purple). Staples are synthesized as linear strands and become “staples” upon hybridization.

duce large, multi-scale structures and dynamic behaviors—precisely the conditions that motivate SynopFrame’s multilevel abstractions and time-linked analysis.

Chapter 3

State of the Art

Macromolecular science has undergone a rapid transformation with advances in structural biology, artificial intelligence, and visualization. High-throughput experimental methods, large-scale computational modeling, and deep learning have enabled unprecedented insight into molecular structure and function. Yet, as data grows in scale and complexity, new bottlenecks have emerged—what I term the *non-optimizable gap*—where automation alone cannot fully resolve scientific challenges, and human expertise, intuition, and visualization become indispensable.

In this chapter, I survey the relevant state of the art across the domains that relate to this dissertation: macromolecular structure determination, scientific visual analytics, AI-driven protein design, and DNA nanotechnology. I place the focus on the interplay between automation and interactive analysis, the evolution of integrative tools, and the persistent challenges that motivate the research presented in this dissertation.

3.1 Structural Biology: From Experimental Determination to Integrative Modeling

Structural biology aims to elucidate the atomic arrangements of biomolecules, a foundation for understanding biological processes and enabling therapeutic innovation. Traditional approaches such as X-ray crystallography [14] and NMR spectroscopy [119] have yielded high-resolution models, but face limitations for large or heterogeneous assemblies [9]. The advent of cryo-EM has revolutionized the field, enabling imaging of biomolecular complexes in near-native states at high

and even atomic resolution [71, 8]. Single particle cryo-EM reconstructs 3D electron density maps from thousands of 2D projections, making possible the study of large and dynamic assemblies previously inaccessible to crystallography. A critical challenge following map reconstruction is the fitting of known or predicted atomic models into the density, achieving complete structural interpretation. To automate and improve the accuracy of model fitting in cryo-EM maps, researchers have drawn on advances in image registration, computer graphics, and optimization. Below, I review key methods from these areas and discuss how they inform our approach in DiffFit.

Image registration and geometric fitting

The fitting of 3D structures into captured or simulated volumes relates to the problem of image registration in image processing. It entails aligning two images, originating from the same or from different modalities, within a shared reference frame [42, 33]. This process involves feature extraction, determining transformations, and assessing accuracy through metrics. Scale-invariant features from images [59], for example, can facilitate matching across a diverse set of views, despite significant distortions or variations. This process involves detecting invariant keypoints using the difference-of-Gaussian function, determining locations and scales, assigning directions based on local gradients, and measuring gradients within selected scales around each keypoint. Extracted features are stored in a database, to make it possible for them to be matched with new images using fast nearest-neighbor algorithms, with applications including object recognition.

Among the many applications of the process, physicians rely on various imaging modalities to diagnose patients, each capturing images with differing orientation. Image registration addresses this variability by aligning images within a unified frame by optimizing parameters like orientation and translation. Medical image registration is an active research area which encompasses diverse methods, including techniques based on cross-correlation [26, 64] and mutual information

[82, 66, 109, 51]. Shang *et al.* [99], for example, introduced a method for medical image registration using principal component analysis (PCA) neural networks to extract feature images and compute rotation angles and translation parameters by aligning the first principal directions and centroids in a simple and efficient way. For complex spatial transformations, another recent approach [4] uses Kernel PCA and Teaching-Learning-based optimization (TLBO) for multi-modal image registration. In our case, similar to these methods, transformations and alignments have to be determined to fit the atomistic model into a volumetric map. We can thus also use optimization techniques in cryo-EM map fitting to refine the fit and optimize parameters such as orientation and translation—which we demonstrate in our work. The major difference to image registration is that, in our workflow, we fit two different data representations, where one is a sub-part of the whole that is potentially present at multiple locations in the target volume.

Model-to-data fitting, which is necessary for cryo-EM data, has also been investigated in depth in computer graphics and pattern recognition [85, 49], with applications in architectural geometry, virtual and augmented reality, robotics, and various other fields [30, 97, 54, 127]—in addition to structural biology. The key challenges in geometric fitting include accuracy, efficiency, robustness, and usability of the fitting module [97, 54]. Structural biology, in contrast, has special challenges such as noisy data, non-geometric shapes, and large data sizes so that geometric fitting methods are not directly applicable.

Yet our DiffFit algorithm still relates to techniques from computer graphics and pattern analysis. The differentiable compositing technique proposed by Reddy *et al.* [85], in particular, offers valuable insights into addressing fitting challenges as well as manipulating and understanding image patterns. With *differentiable compositing* we can handle patterns effectively, outperforming state-of-the-art alternatives in pattern manipulation [131, 98]. Reddy *et al.*’s method [85] discovers complex patterns by aligning elements with their own position and rotation, and facilitates refinement based on similarity to the target for precise adjustment. In

addition, Reddy *et al.* use a multi-resolution pyramid—relevant for handling the multi-resolution volumetric data in our domain. Their method [85], however, is restricted to certain pattern types, requires manual element marking, and may not always find the best solution, leading to orientation errors and missed elements. Nevertheless, we built our solution on top of their differentiable compositing.

Another approach, spline surface fitting [97], enhances the smoothness in aircraft engine geometry reconstruction by concurrently approximating point and normal data, ensuring boundary smoothness and optimal convergence, while exploring the effects of norm-like functions on error measurement. A further recently proposed adaptive spline surface fitting method [54], supported by empirical evidence, employs surface meshes for high-precision CAD applications. The reliance on control meshes of this approach, however, limits its applicability to irregular topologies and compromise the preservation of sharp features. All these methods have common objectives and tasks such as similarity measures, pattern matching, fitting, and geometric transformations; they thus can serve as a motivation and starting point toward our goals in structural biology. Structural biology data, however, often consists of large, complex structures without regular shapes such as CAD models or easy representations in geometric meshes with smooth surfaces so the aforementioned methods are not directly applicable to our data.

Fitting in structural biology

Existing fitting methods for structural biology can broadly be categorized into manual, semi-automated, and automated approaches, each with its own advantages and challenges when used for aligning molecular models with cryo-EM density maps. Manual or semi-automated methods naturally involve human intervention, yet they provide control and precision—which is particularly beneficial in the structural analysis of complex datasets or when specific adjustments are needed for accuracy. For example, *UCSF ChimeraX* [79], a popular tool for molecular manipulation and visualization, includes the *fitmap* technique [36]. It suggests

multiple possible placements of the atomistic model on the density map and then asks the user to make the final decision. The fitting process alternates between rigid-body rotation and gradient descent translation, maximizing the alignment between the atomic model and the density data by optimizing the sum of density values. Similarly, MarkovFit [3] is another technique that is often used for placing atomic-level protein structures within cryo-EM maps of moderate to low resolutions. It uses fast Fourier transform (FFT) for the conformational exploration and Markov random fields (MRF) for efficient representation of subunit interactions. Its use of Markov random fields also facilitates the probabilistic assessment of the fitted models. The complexity of modern structural datasets has also motivated researchers to explore *integrative modeling*, [84] combining multiple data types (e.g., cryo-EM, cross-linking, mass spectrometry) with expert-guided refinement. Nonetheless, all of these manual or semi-automatic approaches are time-consuming and require a significant level of expertise.

To tackle this challenge and to automate the fitting process, researchers have developed methods that rely on deep learning (DL) [125, 108, 123]. A^2 -Net by Xu *et al.* [125], for example, uses DL to accurately determine amino acids within a 3D cryo-EM density volume. It employs a sequence-guided Monte Carlo Tree Search (MCTS) to traverse candidate amino acids, considering the sequential nature of amino acids in a protein. The authors divide the problem of molecular structure determination into three subproblems: amino acid detection in the density volumes, assignment of atomic coordinates to determine the position of each amino acid, and main chain threading to resolve the sequential order of amino acids that form each protein chain. A remarkable speed improvement was also demonstrated by Xu *et al.*, being $100\times$ faster to find solutions at runtime than existing methods [32, 122], and achieving a high accuracy of 89.8%. In addition, they introduced the A^2 dataset with 250,000 amino acids in 1,713 cryo-EM density volumes, with a resolution of 3 \AA , pioneering automated molecular structure determination training benchmarks.

Another recent method by Mallet *et al.*, CrAI, uses machine learning (ML) to find antibodies in cryo-EM densities [65]. The authors formulate the objective as an object detection problem, using the structural properties of Fabs (Fragment antigen-binding) and VHHs (single-domain antibodies). Furthermore, DeepTracer [81] is a fully automated DL-based method designed to determine the all-atom structure of a protein complex using its high-resolution cryo-EM map and amino acid sequence. This method employs a customized deep convolutional neural network primarily for the precise prediction of protein structure, including the locations of amino acids, backbone, secondary structure positions, and amino acid types. The reprocessed cryo-EM maps are the input to the neural network, which transforms the output into a protein structure. Despite yielding accurate outcomes, the resulting atomistic structures may exhibit geometric issues, local fit-to-map discrepancies, misplaced side chains, or errors in tracing and/or connectivity. All DL-based techniques require a substantial amount of time for training (as opposed to their runtime performance) and rely on large training datasets of cryo-EM volumes and manually fitted sub-molecules—which is why we do not resort to DL approaches.

An alternative to DL is map-to-map alignment, which is used to accurately align 2D or 3D maps to facilitate comparison and analysis of spatial structures or features within the maps. CryoAlign [39] is a cryo-EM density map alignment method that achieves a fast, accurate, and robust comparison of two density maps based on local spatial feature descriptors. This approach involves sampling the density map to generate a point cloud representation and extracting key points by clustering based on local properties. CryoAlign then calculates local feature descriptors to capture structural characteristics, reducing the number of points considered and improving efficiency. By employing a mutual feature-matching strategy, CryoAlign establishes correspondences between keypoints in different maps and uses iterative refinement to enhance alignment. A combination of fast rotational matching search based on spherical harmonics and translational

scans [35] yields accurate fitting results in seconds or up to a few minutes. This ADP-EM approach is particularly reliable in fitting X-ray crystal structures to low-resolution density maps, with reduced docking times and while maintaining a thorough 6D exploration with fine rotational sampling steps to find valid docking solutions.

In our work, we design a differentiable optimization method for fitting atomistic structures into volumetric data, with the goal of precise fitting with fast-enough computation to be applicable in semi-automatic fitting in the standard tool ChimeraX. For this purpose, we make use of the PyTorch capabilities for GPU parallel computing, trilinear interpolation sampling in volumetric data, and auto differentiation.

3.2 AI-Driven Protein Design: Generative Models and Human-AI Teaming

Recent breakthroughs in deep learning have revolutionized protein structure prediction and design, making it possible to generate large libraries of candidate backbones, sequences, and predicted structures with unprecedented speed [47, 1, 7, 124, 22, 128, 11]. These automated pipelines, which combine backbone generation, sequence design, structure prediction, and property evaluation, are now standard. However, despite these advances, high-affinity or functional binders remain rare, and efficiently identifying promising candidates among thousands remains a major bottleneck. Next, I review related work spanning protein design visualization, graph-based analysis, human-AI teaming, and integrative visualization platforms that seek to address these emerging needs, but which still leave important gaps that this dissertation aims to fill.

Protein design visualization

High-fidelity 3D visualization of protein structures is key to effective structural biology analyses. Established tools such as ChimeraX [80] and PyMol [93] provide

users with high-quality rendering, real-time interaction, and integrated analytical algorithms, making them the domain’s go-to choices. InVADo [94] augments large-scale molecular docking workflows through an interactive visual analysis approach that integrates multiple 2D and 3D views for filtering and spatial clustering of docking results. A similar emphasis on user-centric design is apparent in visual support systems for loop grafting [74], which link 2D representations with 3D protein views to streamline engineering decisions. Falk et al. [30] proposed a unified framework that merges 2D heatmaps, 3D molecular visualizations, and interactive statistical views to enable comprehensive iterative assessment of cryo-EM models. Elfin UI [126] addresses larger protein architectures using modular building blocks for CAD-style design, and Atligator Web [57] addresses protein–peptide interactions by providing a web-based interface that simplifies motif extraction and design. Drawing on these advances, we integrated multiple 2D views with the ChimeraX 3D visualization to create an AI-driven protein design tool for domain experts.

Graph visualization

Representing complex residue-level interactions as networks or hypergraphs is a powerful way to reveal structural and functional patterns, as exemplified by RING [24]. Bipartite graphs [5] are particularly useful for clarifying relationships between distinct biological entities by restricting edges to occur only across two disjoint node sets, and they have been extensively applied—from ecological and biomolecular to epidemiological networks [78]. Beyond node-link representations, line-based multi-attribute visualizations such as slope graphs [110] and parallel coordinates [45] support comparative analysis of multiple attributes, with slope graphs emphasizing rank-based comparisons and parallel coordinates mapping actual attribute values onto continuous axes. Tools such as LineUp [37] build on these concepts to facilitate the interactive combination, refinement, and comparison of heterogeneous attributes for ranking items. Frameworks such as Tulip [6]

offer comprehensive support for various graph layouts and parallel coordinate views, providing a robust environment for both exploratory analysis and scalable rendering of large biological datasets. Accordingly, we implement several bipartite graphs and a parallel-coordinate views in Tulip for our own tool, ProteinCraft.

Human-AI teaming

Human-AI teaming approaches [72] stress the importance of closely coupling expert knowledge with advanced computational systems. Hong et al. [43], for instance, introduce a visualization platform enabling biologists to compare, validate, and refine predictions from multiple machine learning models in embryonic cell lineage tasks, illustrating how users with limited AI expertise can still effectively perform domain tasks through human-AI collaboration. In another example, Zhao et al. [129] describe a human-in-the-loop framework that combines active learning and visualization to identify critical “visual concepts” for post-hoc analysis and targeted refinement of complex neural networks. From the machine learning perspective, Mosqueira-Rey et al. [70] survey the spectrum of interactive paradigms, including active learning, machine teaching, and explainable AI—emphasizing the importance of who controls the learning process. In “AI-in-the-loop” [19], the focus shifts to emphasizing human agency and responsibility in biomedical analytics, reframing conventional “human-in-the-loop” systems. Moreover, Rogers et al. [86] underscore that what should be automated is the real essential question, rather than merely what can be automated, echoing prior observations by Schetinger et al. [91]. Building on these insights, with ProteinCraft, we adopt a human-AI teaming paradigm in which domain experts iteratively select and integrate promising generative AI outputs for subsequent prompting, ensuring expert oversight and steering throughout the modeling process.

Integrative Visualization Platforms

Modern *visual analytics* seeks to bridge the gap between computational analysis and interactive visualization, enabling scientists to interpret complex data through coordinated multiple views. Systems like Tulip [6] and Vitessce [48] exemplify this approach, supporting data filtering, ranking, and the linking of abstract and spatial representations. However, most visual analytics tools in molecular science remain specialized and non-extensible. Aforementioned widely used 3D structure viewers such as ChimeraX [80, 36], PyMOL [93], and VMD [44] provide powerful rendering and manipulation capabilities, but are not designed to handle the large, multivariate datasets produced by AI-driven protein design workflows—including predicted metrics, interaction networks, and 3D structures. Although RING [24] enables the computation and visualization of residue interactions, such tools remain fragmented and lack integrated analytics for comprehensive filtering, ranking, and iterative exploration of multiple protein structures. Consequently, expert-guided, human-in-the-loop analysis remains an unmet need for diagnosing failure modes and steering the design process—a persistent challenge I refer to as the *non-optimizable gap*. To address these challenges, I integrate Tulip and ChimeraX, combining their strengths to create a unified platform for visual analytics in AI-driven protein design.

3.3 DNA Nanotechnology and Molecular Dynamics Visualization

DNA nanotechnology leverages programmable DNA interactions to construct sophisticated 2D and 3D nanostructures [95, 96, 87]. Tools such as caDNAno [27], Adenita [23], and CATANA [55] streamline the design process, but typically yield static representations. To study the folding and dynamics of these structures, molecular dynamics simulation (MDS) systems such as oxDNA [104] and its associated viewer, oxView [83], are widely used. However, these tools struggle with visual clutter, disconnects between design and simulation, and lack of multi-

representational or temporal analysis. To mitigate these limitations, Miao *et al.* proposed abstraction spaces [68] to organize DNA representations across levels of granularity and layout, helping domain scientists to alleviate visual clutter, though focusing on static structures. However, the field still lacks robust, scalable visual analytics platforms that can provide multi-view, multi-scale, and time-synchronized exploration of dynamic and large-scale DNA nanotechnology data. Tools that enable users to trace assembly pathways, compare simulations, and systematically diagnose failures remain highly sought after.

Molecular dynamics visualization

To visualize the dynamic behavior of molecules, it is important to represent the trajectories resulting from the molecular dynamics simulation. Early work on MDS resulted in the VMD tool [44] that simulates and visualizes molecular dynamics for proteins and nucleic acids. Byška *et al.* [18] visualized protein tunnels by representing the path of each amino acid in the tunnel and aggregating the trajectories into profiles. Kolesár *et al.* [53] proposed a three-level system for illustrating the process of polymerization, coupling together an L-system with agent-based simulation and quantitative simulation techniques. Later, Kolesár *et al.* [52] proposed a way to rectify the simulated data to allow for comparative visualization of a cohort. A more general approach to particle-based spatiotemporal data visualization was proposed by Pálenik *et al.* [77], which enables rapid identification of patterns by simultaneously exploring temporal and spatial scales. VIA-MD [102] also focuses on large-scale MDS data visualization and exploration that links the dynamic 3D geometry to statistical analysis of the data to allow users to identify patterns. Recently, Ulbrich *et al.* [111] represented the MDS data as a node-based dataflow, sMolBoxes, to allow experts to analyze it while still being able to explore the 3D structure.

Many of these methods focus on aggregating the trajectory in some way, whereas for DNA-nano MDS it is essential to legibly reveal the simulating target

frame by frame. While event analysis that supports frame-by-frame analysis at various granularity levels has been realized in MD simulations in the past (e.g., [102, 111]), for the DNA-nano application we handle phenomena where pervasive small-scale events (H-bond formation) can happen everywhere in the simulation target and can lead to higher-scale events (forming and deforming of the local helix all the way up to the whole assembly)—while also supporting the traditional construction, analysis, and editing of the DNA-nano assemblies in ways that are familiar to the experts. They need to link the dynamics simulation with their envisioned 3D structure and the staple-based folding during construction. In a way our work is thus akin to other applications of MDS in specific domains [15].

DNA nanotechnology modeling and visualization

Several computer-aided design tools have been developed for DNA nanotechnology. Adenita [23], caDNAno [27], Vivern [56], CATANA [55], and oxView [83] are five examples, sampling the spectrum of available tools. Adenita lets the user design a DNA structure in multiple abstract levels in 3D in a user-friendly and intuitive manner. caDNAno uses parallel nucleic acid helix strands as placeholders and then lets users select active strands and edit the connections between them. The tool’s visual interface enforces all editing to be done in 2D, which facilitates easy interactions but hinders the mental understanding of the structure in 3D. Vivern is a VR application designed to enhance the design and analysis of DNA origami nanostructures, offering advanced visualization tools and demonstrating improved capabilities over traditional desktop applications. CATANA and OxView offer design functionality in a web browser. CATANA uses a “novel Unified Nanotechnology Format” and facilitates easy MDS export but does not support MDS trajectory visualization. OxView’s interface supports the visualization of dynamic simulations. It can render many nucleotides in 3D in a browser, but its lack of abstract views for different scales hinders users to identify and understand relevant simulation events buried in vast amounts of data, with erratic dynamic

behavior and visual occlusion. The handy abstractions used in the designing environment are fully disconnected from the trajectory representations. We aim to bridge this gap by leveraging the advantages of existing abstract views for dynamic scenarios.

Abstraction spaces in visualization

Previous research has shown the usefulness of organizing related visual representations as conceptual spaces, which Viola and Isenberg call *abstraction spaces* [116, 117]. For molecular visualization, Zwan *et al.* [113] and Lueks *et al.* [60] proposed a multidimensional space that organizes the visualization along axes such as *structure*, *illustrativeness*, and *spatial perception*. In contrast, Mohammed *et al.* [69] and Miao *et al.* [67] use their abstraction spaces as interactive panels that allow users to smoothly transition from one representation to another. While the former authors assign an axis for each structure of interest, the latter propose a more general approach that organizes representations along aspects of interest such as layout and scale. In addition to the mere organization of representations by means of abstraction spaces, the power of animated transitions between different representations—as made possible through the spaces—has been established by Heer and Robertson [40]. We build upon this general foundation by incorporating the MDS data into the concept of abstraction spaces. In particular, we generalize Miao *et al.*’s [67] space by incorporating the concept of *idiom*, which is essential for conceptualizing the design, as well as by designing an interaction framework that incorporates the dynamic character of DNA-nano structures by allowing experts to explore MDS data while they also study the spatial design.

3.4 Summary and Research Gaps

Despite remarkable progress in structural biology, visualization, and AI-driven protein design and DNA nanostructure design, persistent gaps remain:

- **Fragmented toolchains:** Researchers rely on specialized, disconnected

software for modeling, visualization, and analysis, limiting efficiency and integration.

- **Scalability challenges:** Most tools cannot handle the scale, heterogeneity, and high dimensionality of datasets produced by modern AI and simulation workflows.
- **Limited human–AI collaboration:** Key scientific decisions require human expertise and contextual judgment, yet current tools rarely support transparent, interactive workflows.

This dissertation addresses these challenges by developing a suite of visual analytics systems—*DiffFit*, *ProteinCraft*, and *SynopFrame*—that tightly integrate automated modeling, AI-driven inference, and interactive visualization. Together, with these systems, I demonstrate how integrative visual analytics can bridge the non-optimizable gap, enabling more effective human–machine collaboration and accelerating discovery in macromolecular science. In the following chapters, I introduce each of these systems in detail.

Chapter 4

DiffFit: Visually-Guided Differentiable Fitting of Molecule Structures to a Cryo-EM Map

Scientific discovery often advances when tools or ideas from one domain are reimagined for another. After a long and challenging period in my PhD—marked by repeated rejections of my first paper—I found myself leading a project on inverse procedural modeling at the mesoscale, with a focus on finding the modeling route given the final molecular architecture. During the exploring phase, I encountered a differentiable algorithm from the computer graphics community, originally developed for compositing bitmap images from elemental patches [85]. It was then that I recognized a unique opportunity: this graphics algorithm, devised for an entirely different context, could be adapted to address one of the central, unresolved challenges in structural biology—accurately fitting atomic models into cryo-EM volumes.

As defined in Chapter 1, cryo-EM fitting exemplifies the *non-optimizable gap*: the score landscape over the 6D pose space (three rotations + three translations) is rugged, with many local optima and narrow basins. Traditional pipelines therefore rely on experts to manually place subunits near a plausible pose before local optimizers can converge—creating a brittle, time-consuming bottleneck at the interface between automation and judgment. **DiffFit** targets precisely this slice of the gap by replacing manual coarse placement with differentiable, visually guided global-to-local optimization and robust initialization; the system then clusters and ranks candidate fits so experts can concentrate on verification and selection. In effect, **DiffFit** shifts human effort from low-level manipulation to high-level assessment and enables scalable assembly of large complexes.

With this cross-disciplinary insight, born from a confluence of persistence and serendipity, together with my collaborators, we developed DiffFit, leading to a dramatic improvement in the efficiency and accuracy of fitting protein structures into reconstructed cryo-EM maps, effectively eliminating the need for tedious manual alignment and opening the door to scalable, high-throughput structural modeling. I now present DiffFit mainly as it appears in the publication [61], with slight modifications to fit the dissertation.

4.1 Introduction

As humans we have been striving for centuries or even millennia to understand, as [Faust](#) stated, “*was die Welt im Innersten zusammenhält* [what binds the world, and guides its course]” [118]. Among other directions of scientific inquiry, this quest applies to the inner workings of the biological world, particularly to how biological processes at tiny scales work and how they keep us alive. In the field of structural biology, researchers have traditionally relied on techniques such as X-ray crystallography or nuclear magnetic resonance spectroscopy to understand the actual molecular composition of cells and organelles—yet with the limitation that these could only provide (still impressive and highly useful) estimates or manually constructed models of the structure of actual biological samples (e.g., [94, 112, 56, 103, 73, 28]). The recent cryo-EM approach [71], however, enables researchers to visualize biomolecules in *actual samples* at near-atomic resolution. In addition, over decades, the Protein Data Bank (PDB) initiative has collected thousands of molecular models of the building blocks of cells or organelles studied in structural biology. Researchers are thus on the brink of assembling the molecular composition of actual samples at the ground-truth level.

To achieve this type of assembly, researchers do not only need to interactively visualize molecular data, for which tools [101] exist, but also to faithfully place 3D models of known molecular building blocks, such as those from PDB data and the AlphaFold predicted library [47], into the captured cryo-EM datasets. Thus

far, the fitting process involves a substantial time commitment and numerous manual interventions by domain experts, rendering this process ineffective. The complexity and size of the involved molecules, combined with the variability and noise inherent in cryo-EM data, pose substantial obstacles. In contrast, a fully automatic process is also not ideal because the existence of local minima (wrong placement of compositing proteins) requires domain experts to verify each placement using their knowledge and experience. Fully automated methods are currently far from feasible. Instead, an optimal balance between user interaction and automation is required.

For this purpose, we developed DiffFit, an automated differentiable fitting algorithm coupled with visual inspection and decision-making, designed to optimize alignment between protein structures and experimental reconstructions of volumes (i.e., cryo-EM maps). Our technique works in one-to-one and many-to-one fitting scenarios, in which multiple protein subunit structures are precisely aligned with a single, large, experimentally reconstructed volume. The DiffFit method is iterative and gradually introduces the source protein structures into the target volumes to assemble the composition of the molecular subunits step by step. By employing advanced strategies such as volume filtering, multiresolution volumes, and negative space utilization we constructed a loss function to quantify the fitting accuracy during the iterations and for the final decision-making. This loss function helps us to iteratively reduce the differences between the two representations—volumetric and atomistic—until we achieve the desired fit. Our visually-guided fitting procedure eliminates the need for domain experts to manually place structures as they assemble the protein structures into the cryo-EM map. It thus significantly accelerates the process into a manageable interactive procedure, delivering precise results for visualizing and analyzing complex, real-world protein structures, ultimately facilitating large-scale structural modeling initiatives. In summary, we contribute:

- a differentiable fitting algorithm designed to fit multiple molecular subunits

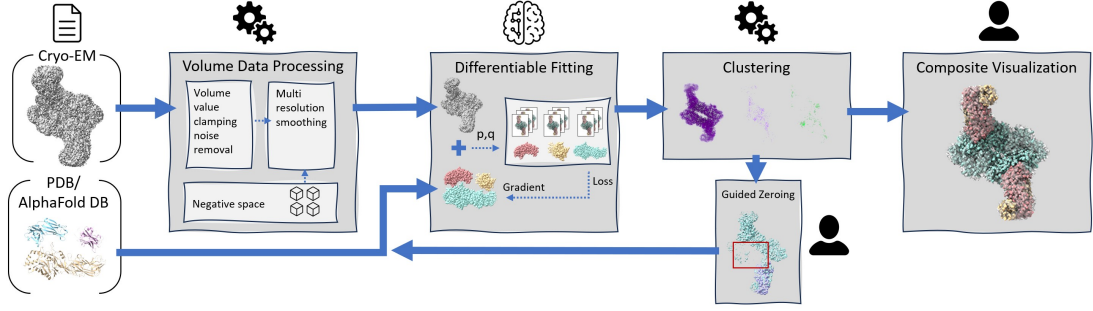


Figure 4.1: DiffFit workflow. The target cryo-EM volume and the structures to be fit on the very left serve as inputs, which are passed into the novel volume processing, followed by the differentiable fitting algorithm. The fitting results are then clustered and inspected by the expert. The expert may zero out voxels corresponding to the placed structures and feed the map back iteratively as input for a new fitting, round until the compositing is done.

to a single reconstructed cryo-EM volume;

- a human-in-the-loop strategy providing visual inspection and decision-making in an iterative structure assembly cycle;
- a novel loss function and data processing that calculates new updates in each iteration to expedite algorithm convergence and quantify the fitting accuracy; and
- three use-case scenarios of fitting one or multiple known subunits or identifying yet unknown subunits as part of the molecular assembly.

4.2 Method

We begin to describe our approach by explaining our process of differentiable structure fitting, before we show how it can be used for visually-guided fitting. After discussing these conceptual aspects, we also briefly discuss implementation details.

Differentiable structure fitting

Given a cryo-EM map, the domain practitioners—bioscientists—do not know the precise location and orientation parameters that govern where and how a protein’s sub-structures fit together. For some regions in the map, the bioscientists may

not even know which protein subunits are supposed to be present. Our goal is to develop a new approach that addresses both of these domain tasks. For certain protein subunits, bioscientist are highly confident about their presence in the map. In such cases, our technique will aid in the identification of their respective placement parameters. Second, for the regions with unknown protein subunits, the task of our technique is to identify potential protein subunit candidates from a large database that best fit the cryo-EM map region.

Inspiration and approach overview

We base our approach on the previously mentioned 2D differentiable compositing approach by Reddy *et al.* [85], which discovers pattern structure from wallpaper-like textures containing repetitive patterns made out of elementary patches. We first review their approach, before we describe how we build up our solution on top of their technique. In their case, given a 2D image, which is a composite of multiple small element patches, the task is to identify the number of occurrences of each patch and the placement parameters for each existing patch. The parameters include the type of patch out of several known patches as well as position, orientation, and depth. Their solution distributes tens to hundreds of patches in the image and uses the differentiable optimization methodology to translate and reorient patterns such that they correspond to the appearance of the patterns in the input wallpaper image. Each single instance E_i (out of total number of ω instances) of a pattern is stored in a layer J_i for each patch instance by sampling from that patch, with the translation, rotation, and patch-pattern type probability taken into account:

$$J_i(\mathbf{x}) = f_t(\mathbf{x}, E_i) = \sum_{j=1}^o \frac{e^{t_i^j}}{\sum_{k=1}^o e^{t_i^k}} h_j(R_{\theta_i}^{-1}(\mathbf{x} - \mathbf{c}_i)) \quad (4.1)$$

where $f_t(\mathbf{x}, E_i)$ is a differentiable function using the expected value over patch-pattern type probabilities stored in a tuple \mathbf{t} representing all o patch patterns; softmax $e^{t_i^j} / \sum_{k=1}^o e^{t_i^k}$ over type logits define the patch-pattern type probabilities;

h_j is the image patch sampler function; \mathbf{x} is the image location; \mathbf{c}_i is center location of patch element E_i ; and $R_{\theta_i}^{-1}$ is the inverse of a 2×2 matrix rotation with angle θ_i .

The solution image results from compositing of all instances together using f_c compositing function so that each known patch-pattern type is present multiple times with various positional parameters. Patches can overlap other patches, which leads to partial or full occlusion of a certain pattern. This is characterized by $v_i(\mathbf{x}), v \in \{0, 1\}$, which is the visibility of layer i at image location \mathbf{x} :

$$I(\mathbf{x}) = f_c(\{J_i(\mathbf{x})\}_i) = \sum_{i=0}^{\omega} J_i(\mathbf{x})v_i(\mathbf{x}) \quad (4.2)$$

This solution image is compared with the input image in an optimization, where the parameters of all patch instances are updated in every iteration. The solution image then becomes increasingly similar to the input image. Reddy *et al.* define the L^2 distance loss L_d for the optimizer as:

$$L_d(A, I) = \frac{1}{P} \sum_{p=1}^P \|A(\mathbf{x}_p) - I(\mathbf{x}_p)\|_2^2 \quad (4.3)$$

where the sum is over the number of all pixels P in the image. A is the input image and I is the composited solution image. The optimal elements \mathcal{E}^* are then found by minimizing loss L_d over the entire set of elements $\mathcal{E} = \{E_0, \dots, E_\omega\}$:

$$\mathcal{E}^* = \arg \min_{\mathcal{E}} L_d(A, f_c(\mathcal{E})). \quad (4.4)$$

Reddy *et al.*'s patch-pattern fitting problem is similar to ours in the sense that in both cases we are compositing element instances into a scene. But Reddy *et al.*'s differentiable compositing approach cannot be directly applied to the structural biology domain to solve the protein fitting problem for the following reasons:

1. the pattern image and element patches are defined in 2D with layers, while the cryo-EM map and protein subunits are defined in 3D;

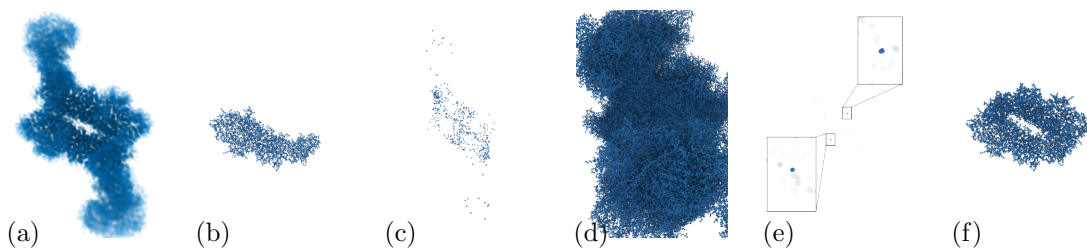


Figure 4.2: Clustering and filtering: (a) target volume, (b) atom coordinates of the source structure, (c) positions of 1000 fit results (the dots are clustered, hiding the number of results), (d) 1000 instances of the structure, (e) positions of 1000 fit results with a transparency level set based on an exponential scaling of the sum of sampled density metric (two clusters stand out, as in the zoom-in insets), and (f) instances of the structure at those two clusters.

2. the pattern image and element patches are of the same representation, i.e., 2D grid data, while the cryo-EM map and protein subunits are of different representations—one is a 3D volume while the other is a set of atom coordinates that can be regarded as a point cloud;
3. the instance patches in differentiable compositing are all of the same size, while the protein subunits differ in numbers of atoms;
4. differentiable compositing expects the patches to overlap, while protein subunits do not spatially overlap; and
5. forming 1000 layers of 2D images is possible to fit into the current graphics processing unit (GPU) memory while forming 1000 3D volumes is prohibitive with the currently available GPU memory.

Initially, we attempted to align our problem better with differentiable compositing by first *simulating* a cryo-EM map from the atomistic point cloud of the protein model and then fitting the simulated map to a target map. That way the representational discrepancy (see reason (2) above) is eliminated. That approach, however, was only successful for trivial cases, while for real-world scenarios it frequently fell into local minima. To illustrate this point, we provide several exemplary volume-only based molecular fitting videos for interested readers in our supplementary material at osf.io/5tx4q.

Driven by the successful fitting cases from many experiments, we gradually

built several novel strategies on top of the differentiable compositing that effectively address the problem of molecular structure fitting. Most notably, we address the substantially higher complexity of our scenario based on the knowledge, experience, and deep insight of the target audience: the bioscientists. Our solution is thus based on a human-in-the-loop strategy and we propose a fast and robust visual analytics approach, DiffFit, with two main steps: (1) an automated excessive molecular fitting and (2) a visual inspection and filtering of the fitted results by the bioscientists. Both consecutive steps are building blocks of a visual analytics feedback loop, in which multiple proteins are iteratively composited to fit the underlying cryo-EM volume.

We schematically present our DiffFit workflow in Figure 4.1. First, we seed an excess amount of all compositing molecules in the volume scene. If this simultaneous fitting all molecules exceeds the available GPU memory, we sort the molecules by atom count and partition them into batches. Then, we fit the batches of molecules within several iterations in descending atom-count order. This fitting relies on a novel loss function that calculates the average density value from the densities that we sample for each atom. The differentiable property of our fitting scenario allows us to optimize based on gradient-descent. In addition, we associate each fit with a numerical value that characterizes the fit quality. For this purpose, we create a simulated cryo-EM map of each fitted molecule and calculate the correlation of the simulated densities using the real cryo-EM map densities. Then, we collect the fitting results and cluster them based on positional and orientational parameters. For each cluster, we select one representative fit—the fit with the highest correlation value. Then, we sort the clusters by their representative correlations and interactively visualize them in ChimeraX to allow the bioscientists to inspect the solutions. Once they verify a given molecular placement, we disable molecule placements in the respective regions in the following iterations by setting the voxels covered by the molecular fit in the cryo-EM map volume to zero. We thus gradually erase the successful placements from the map,

forcing the following placements to search for a fit in non-zero volume locations. Once the feedback update in the map is completed, we perform the next fitting iteration with another molecular structure. We repeat this workflow pattern until the map has nearly all voxels zeroed out, and the entire sub-unit placement of the complex molecular structure is complete. Below we introduce the details of DiffFit, in the following order: (1) sample one coordinate, (2) fit one placement of one molecule, (3) fit multiple placements of one molecule, and (4) fit multiple placements of multiple molecules.

Sampling of one coordinate

Because our task is to determine the optimal alignment of an atomistic molecular structure to the reconstructed cryo-EM volume map, we determine the optimal fit characterized by two rigid-body transformation parameters: a translational offset \mathbf{p} and a rotation. We represent the rotation by a quaternion \mathbf{q} or its corresponding rotation matrix $M_{\mathbf{q}}$. The position \mathbf{x}_i corresponds to the center point of an atom i in the molecular subunit. To calculate the fit, we transform every atom position in one subunit according to the rotation and translational offset:

$$T(\mathbf{x}_i) = M_{\mathbf{q}} \cdot \mathbf{x}_i + \mathbf{p}.$$

We sample a density value D of the atom to be placed at position $T(\mathbf{x}_i)$ from a scalar volume V using trilinear interpolation as follows:

$$D(T(\mathbf{x}_i)) = S(T(\mathbf{x}_i), V).$$

Placement of one molecule (or one subunit)

We formulate an initial loss function L to determine the best \mathbf{p} and \mathbf{q} parameters. This function gives us the minimum negative average density per atom for a molecular subunit with N atoms that form the set \mathbf{X}_m of all atom center points

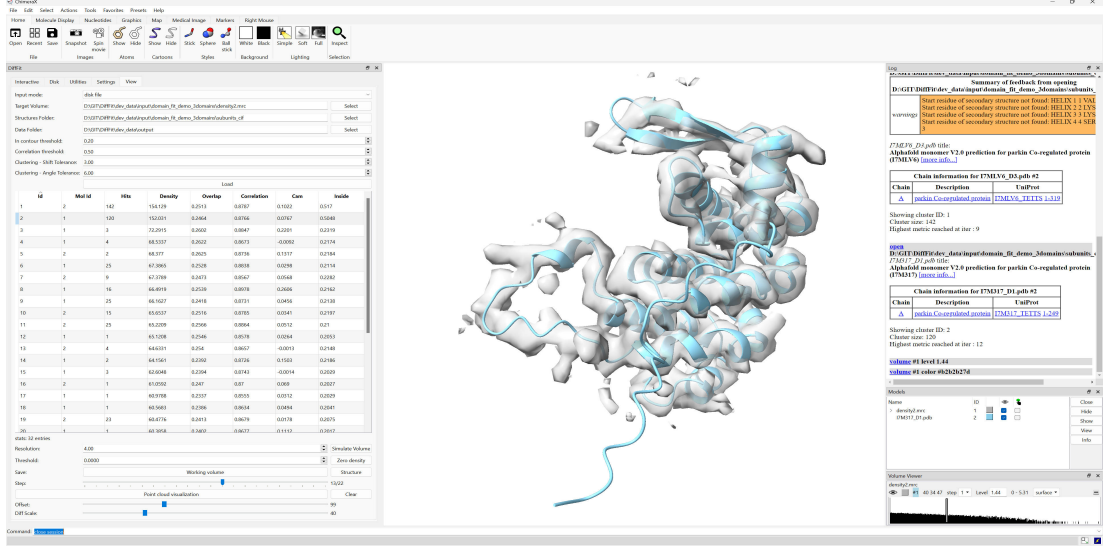


Figure 4.3: Visual browser based on ChimeraX. The target volume on the middle is overlaid with a fitted molecule corresponding to the selected fit result in the table on the left (clustered fits, each row is the representative placement with the highest correlation from that cluster). After inspection, users can save the placement and then select “Simulate volume” and “Zero density” to zero out the corresponding voxels from the target volume.

$T(\mathbf{x}_i) \in \mathbf{X}_m$ for a particular molecular subunit m :

$$L(\mathbf{p}, \mathbf{q}, \mathbf{X}_m, V) = - \left(\frac{1}{N} \sum_i^N D(T(\mathbf{x}_i)) \right) = - \frac{1}{N} \sum_i^N S(M_{\mathbf{q}} \cdot \mathbf{x}_i + \mathbf{p}, V). \quad (4.5)$$

We rely on the calculation of the gradient of the differentiable formulation and use it with the Adam optimizer [50] for the optimization. Although the Adam optimizer is known for robustness with respect to local minima, our initial loss function formulation frequently leads to a local minimum (i.e., a place that is not an optimal placement for the molecule in the map but from which the optimizer cannot find a better solution in the parameter-space neighborhood). Such local minima are a common and severe problem that also manifests in the functionality of the most commonly used tools for molecular subunit fitting (e.g., the fit-in-map feature in ChimeraX). We thus introduce several strategies to form a novel loss function, making DiffFit more robust.

The first strategy that we found to substantially contribute to a good fitting performance is **filtering the input cryo-EM map volume V** . For this purpose

we clamp the volume values based on a user-specified minimal threshold and a minimum size of connected voxels that form a cluster. We also detect all voxels with a density value less than a given threshold and set them to zero. The size of the connected voxel cluster after thresholding must be greater than the cluster size hyperparameter. Otherwise, we set all the voxels in that cluster to zero. This step leads to the filtered volume V_F and ensures that only relevant volume regions are considered for fitting, improving the focus and efficiency of the algorithm. Then, we normalize the filtered values to $[0, 1]$ —a typical practice in learning and optimization approaches—, which leads to a volume \hat{V}_F that turns out to be essential for controlling the magnitude of the calculations that lead to the loss function and hence the settings of the hyperparameters in the workflow.

To accommodate the inherent noise and variability in biological datasets, we apply a series of convolution iterations to the target volume, and capture each smoothing result as a separate volume. This iterative convolutional smoothing leads to an array of volumes, and we use each of these volumes in the fitting process. This **multi-resolution approach** enhances the robustness of the fitting process by mitigating the impact of noise and data irregularities. Empirically, we found that a 3-element array of increasingly smoothed volumes performs well, iteratively filtered with a Gaussian smoothing kernel. We expose the size of this array as a hyperparameter to allow users to control it. We experimented with Laplacian smoothing as well, which led to unsatisfactory performance. We denote the non-smoothed volume as $\hat{V}_F^{G_0}$ and express the recurrent formulation of the iterative convolution smoothing as:

$$\hat{V}_F^{G_n} = \hat{V}_F^{G_{n-1}} * G_n.$$

A third adaptation we apply to the initial fitting process is a stricter penalization of a mismatch. If an atom center is placed in the cryo-EM map volume but outside the extent of the molecular target structure (i.e., outside of the target *footprint*), the target density would normally be zero. To discourage such

misalignment even further, we assign these regions a **negative value**. After smoothing, for voxels with a density value of zero, we replace the zero with a negative value. We experimented with varying the negative values or creating a smooth gradient of negative values. We noticed that a constant value of -0.5 outside the molecular footprint in the map performs well. We expose this value as a tunable hyperparameter. We denote the resulting volume as $\hat{V}_{F-c}^{G_n}$, where $-c$ is the negative constant value. Finally, we update the loss function formulation with a volume smoothed after j iterations as:

$$L(\mathbf{p}, \mathbf{q}, \mathbf{X}_m, \hat{V}_{F-c}^{G_j}).$$

We weigh each fit with a multiresolution volume array element w_j for n resolutions, and sum up all the multiresolution components to form the final loss function for one \mathbf{p} and \mathbf{q} pair:

$$L_m([\mathbf{p}, \mathbf{q}]) = \sum_{j=1}^n w_j \cdot L(\mathbf{p}, \mathbf{q}, \mathbf{X}_m, \hat{V}_{F-c}^{G_j}). \quad (4.6)$$

To start the optimization, we need to initialize the position offset \mathbf{p} and the rotation quaternion \mathbf{q} . For orientations, we draw $N_{\mathbf{q}}$ i.i.d. samples *Haar-uniformly on* $\text{SO}(3)$ using the unit-quaternion method of Shoemake [100]. Concretely, for $u_1, u_2, u_3 \sim \mathcal{U}[0, 1]$,

$$\mathbf{q}(u_1, u_2, u_3)^\top = \left(\sqrt{1-u_1} \sin(2\pi u_2), \sqrt{1-u_1} \cos(2\pi u_2), \sqrt{u_1} \sin(2\pi u_3), \sqrt{u_1} \cos(2\pi u_3) \right). \quad (4.7)$$

See also our implementation.¹⁾ Instead of uniformly sampling positions from the volume bounding box (as in ChimeraX), we uniformly sample $N_{\mathbf{p}}$ positions from the positive voxels in the filtered and normalized volume \hat{V}_{F-c} . This **enveloped sampling based initialization** increases the success rate by a factor of two by searching from $N_{\mathbf{q}} \cdot N_{\mathbf{p}}$ initial placements, compared to the traditional initialization

¹[GitHub: DiffFit/src/DiffAtomComp.py](#)

in ChimeraX.

Fitting multiple placements of one molecule

To look for fits for multiple copies of a single molecule m , we then take advantage of GPU parallelization and optimize all $N_{\mathbf{q}} \cdot N_{\mathbf{p}}$ pairs of $[\mathbf{p}, \mathbf{q}]$ of the molecule with atoms \mathbf{X}_m altogether in one single loss function:

$$L_{par}(m) = \sum_{k=1}^{N_{\mathbf{q}} \cdot N_{\mathbf{p}}} L_m([\mathbf{p}_k, \mathbf{q}_k]). \quad (4.8)$$

Fitting multiple placements of multiple molecules

Finally, all subunit molecules have different numbers of atoms; it is thus not easy to parallelize the treatment of multiple molecules without overhead on the array padding of zeros. Usually, the $N_{\mathbf{q}} \cdot N_{\mathbf{p}}$ initial placements of \mathbf{X}_m atoms would result in a total number of sampling operations higher than the total number of GPU threads; therefore, we process different subunits molecules sequentially in a `for` loop and form an overall loss function for M molecules as follows:

$$L_{all} = \sum_{l=1}^M L_{par}(l). \quad (4.9)$$

Quantify the fit quality

By sampling in the simulated volume from the molecule, we can get a weight for each atom coordinate as $W(\mathbf{x}) = S(\mathbf{x}, V_{sim})$. Then, for all atoms in a molecule, we can form two vectors, a sampled density vector $\mathbf{D} = [D(\mathbf{x}_1), D(\mathbf{x}_2), \dots, D(\mathbf{x}_N)]$ from the target volume and a weight vector $\mathbf{W} = [W(\mathbf{x}_1), W(\mathbf{x}_2), \dots, W(\mathbf{x}_N)]$ from the simulated volume. Then, we can calculate three alignment metrics, the mean overlap μ , correlation ρ , and the correlation about the mean ρ_{μ} as:

$$\mu = \frac{\mathbf{D} \cdot \mathbf{W}}{N},$$

$$\rho = \frac{\mathbf{D} \cdot \mathbf{W}}{|\mathbf{D}| |\mathbf{W}|}, \text{ and}$$

$$\rho_\mu = \frac{(\mathbf{D} - D_\mu) \cdot (\mathbf{W} - W_\mu)}{|\mathbf{D} - D_\mu| |\mathbf{W} - W_\mu|},$$

where the subtraction operator represents subtracting the scalar average densities D_μ and W_μ from each component of the sampled density vectors. We use these quality metrics during the interactive assessment by the bioscientist in ChimeraX that we describe next.

Visually-guided fitting

A critical aspect of the post-processing of DiffFit involves the clustering and sorting of the fitting results to facilitate user-guided selection and refinement. After the optimization phase, the algorithm generates a vast array of potential fits, characterized by their translation and rotation parameters. To manage this abundance of data and facilitate efficient result exploration, we apply a clustering algorithm to group the fitting results based on their spatial and orientational similarity (Figure 4.2(c), (e)).

Each cluster represents a set of closely related fits, suggesting a consensus among them regarding the position and orientation of the fitted structure in the target volume. We sort these clusters based on a defined metric, such as the overall density overlap or correlation coefficient we just discussed, ensuring that the most promising fits are prioritized for user review. This hierarchical organization allows researchers to quickly identify the most accurate and relevant fitting results, streamlining the analysis process.

To further assist the experts in exploring the fitting results, we created an interactive visual browser as a comprehensive visualization tool to present the sorted clusters in a user-friendly format (Figure 4.3). In the browser we display key metrics for each cluster, including the average correlation coefficient, the

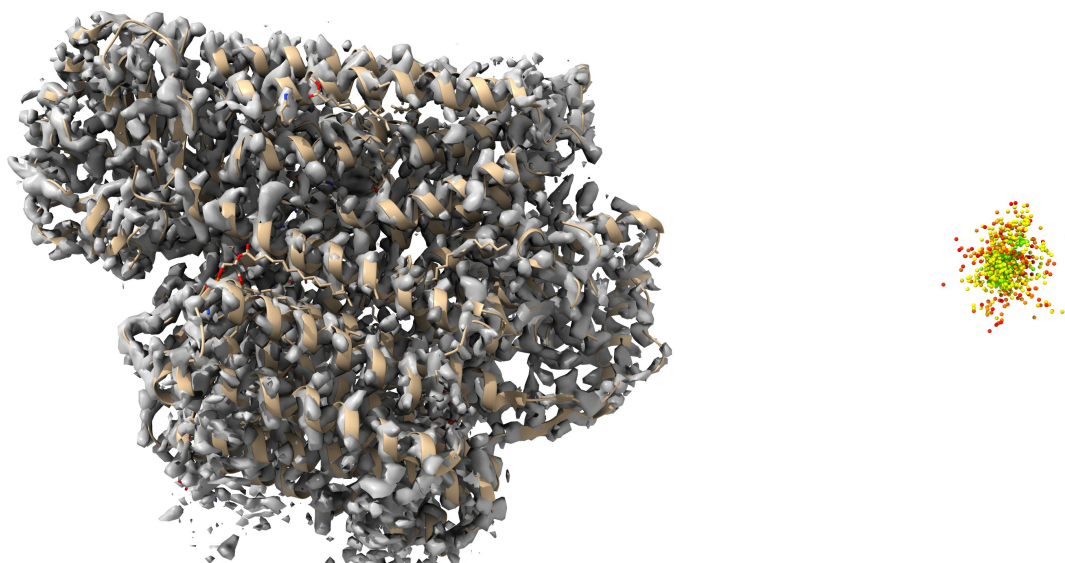


Figure 4.4: Interactive visual fit assessment tools we implemented in ChimeraX. Left: visualization of the best fit of the selected cluster of fits in the context of the cryo-EM volume map; right: abstracted summary of all clusters, in which each sphere represents the center of the molecule placement and the color represents the quality of the fit. Both views are interactive and move in sync, just being offset from each other in the screen’s x -direction (as they both represent the same spatial 3D space).

density overlap, and the consensus error measures, which provide a quick overview of the quality and relevance of each cluster. The browser also allows the biologists to select a cluster and visually inspect the fitting results within the 3D context of the target volume. This interactive exploration is crucial for assessing the fit quality in complex cryo-EM map regions, where subtle differences in position or orientation could substantially impact the biological interpretation of the results.

To help the experts to gain a spatial overview of the fitting result in the tabular data (Figure 4.3), we further provide means to visualize the clusters in the viewport (Figure 4.4). First, we display the representative molecule of each cluster of fits in the context of the cryo-EM volume map, as we show in Figure 4.4 on the left. The displayed fit can be controlled by selecting one from the list of fits. To also provide the experts with a visual summary of all fits, we also abstractly represent each fit cluster with the help of a sphere that we place at the center of the cluster’s representative molecule as it is transformed by the fit shift and rotation (Figure 4.4, right). We assign each sphere a color based on the clusters’

average density order and the experts can also use the spheres to select a different fit to be shown in the context of the cryo-EM volume map.

An innovative feature of our approach is the ability to refine the fitting process iteratively by selectively excluding already placed molecules densities. Once a bioscientist selects a cluster and verifies its fit (or multiple fits) as accurate, we can zero out the corresponding density in the target volume, thus effectively removing the respective volume region from further consideration in the following part of the fitting process. For this purpose we, first, use the fit structure to simulate a map; then we use the voxel positions from the original map to sample the density values from the simulated map. If the sampled density is higher than a user-specified threshold, we set the original voxel’s density to zero. By default we use a threshold of 0, which usually delivers good results. This step is crucial for complex volumes containing multiple closely situated structures, as it prevents the algorithm from repeatedly fitting structures to the same volume region and reduces false positives when fitting the remaining region.

By thus iteratively fitting and zeroing out densities, users can progressively shrink the target volume, isolating and identifying individual structures in dense or complex datasets. This iterative refinement ensures that the fitting process is not only guided by the algorithm’s optimization but also by the expert’s knowledge and visual assessment, ultimately leading to more accurate and biologically meaningful results.

Our resulting visually-guided fitting framework enhances the DiffFit algorithm by integrating clustering, sorting, and interactive exploration tools. These features enable users to efficiently filter through large datasets of fitting results, identify the most promising fits, and iteratively refine the fitting process based on a visual assessment. The combination of automated optimization with user-guided inspection and filtering addresses the challenge of accurately fitting molecular subunit structures within volumetric data.

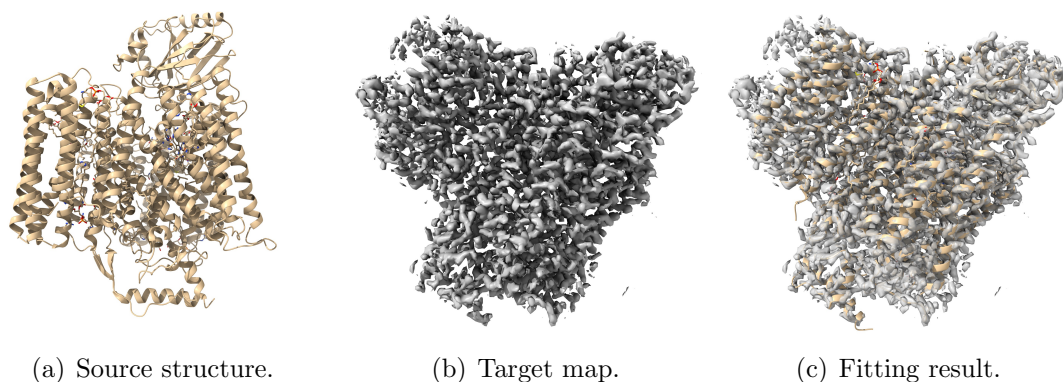


Figure 4.5: Fitting a single structure for [6WTI](#).

4.3 Implementation

We utilized `PyTorch` for the `DiffFit` algorithm implementation, capitalizing on its dynamic computational graph, automatic differentiation, and GPU acceleration to estimate positional offset and rotational quaternion parameters efficiently. Specifically, we employed `torch.nn.Conv3d` for Gaussian smoothing and `torch.nn.functional.grid_sample` for trilinear interpolation with padding mode set to “border.” This approach enables us to rapidly process large volumes and multiple structures. We leveraged tensor operations and the Adam optimization algorithm for accurate optimization results. We also integrated additional functions from `SciPy`, `Bio.PDB`, `mrcfile`, and `Numpy` libraries to enhance the algorithm’s functionality. In addition, we integrated `DiffFit` seamlessly into `ChimeraX` (Figure 4.3), based on its bundle development environment.

4.4 Use case scenarios

We designed `DiffFit` with its advanced fitting algorithms and integration with visualization tools to address a range of challenges in structural biology. To be able to illustrate their power, we now explore three key scenarios where `DiffFit` can be particularly effective, demonstrating its versatility and potential.

Scenario 1: Fit a single structure

The most straightforward application of DiffFit is the fitting of a single atomistic structure to a target volumetric map, as we illustrate in Figure 4.5. This scenario is common in cases in which an already-resolved protein structure or a predicted structure needs to be placed in a newly reconstructed volume captured via cryo-EM for further refinement.

In this scenario, DiffFit efficiently determines the optimal position and orientation of the protein (Figure 4.5(a)) in the volume (Figure 4.5(b)). The entire fitting process is automatic, removing the prerequisite of manually placing the structure at an approximate orientation close to the final optimal one. The interactive visually guided inspection process allows researchers to verify the fit (Figure 4.5(c)), who apply their expertise to ensure biological relevance and accuracy.

We use the dataset reported in the recent MarkovFit work [3] to benchmark DiffFit’s performance in this scenario, compare it with ChimeraX and MarkovFit, and report the results [successful hit rate, computation time, and root-mean-square deviation (RMSD)] in Table 4.1 (we provide a complete table that includes each structure’s EMD ID, the number of chains, the number of atoms, voxel size, and the used surface level threshold used in Table A.1 in section A.1). For each structure, we performed five experiment runs to obtain reliable results (we also attach each individual run’s metrics in our supplementary material). For each run, we fit $1000 \times$ to perform the search. For ChimeraX, we ran the command “`fit #1 in #2 search 1000`” to perform atom-to-map searching, which is about one-fold faster than the atom-simulated map-to-map fitting and delivers similar results. We regard the number of fits in the top-ranked cluster as the hit rate if the representative fit of that cluster is within 3 Å and 6 degrees (ChimeraX’s default threshold) from the ground truth. We do not repeat the MarkovFit computation as it takes an average of “7.7” plus “6.25” hours (as stated in prior work [3]) to finish a single run for each structure. Instead, we directly take the author-reported [3] RMSD. We take the “top-scored” model’s

RMSD (although it is often the same as that of the “best model by RMSD among top 10”) because, in practice, there is no ground truth to compare to in advance to get the best model. DiffFit significantly outperforms ChimeraX on the hit rate (on average, a $26.9\times$ gain) and the computation time ($26.0\times$ gain). The RMSD of DiffFit is also significantly better than MarkovFit but often slightly worse than ChimeraX’s. However, this can always be corrected (the last column in the table) by a single automatic fit using ChimeraX’s atom-to-map fitting, which results in a better RMSD on average. Compared with the overall averaged metrics, for high-resolution maps, DiffFit gets higher gains on hit rate; for medium-resolution maps DiffFit gets higher gains on the computation time. Of note is that, with expert knowledge, ChimeraX’s performance can be boosted. For example, after smoothing the map via a command similar to “`vol gaussian #2 sd 2`,” the hit rate goes up to 1 ([6WTI](#)), 67 ([7D8X](#)), and 12 ([6M5U](#)). We can also achieve a similar boosting with DiffFit. As this boosting highly depends on the user’s knowledge, we report only the simple version of the fitting where the user only needs to specify the surface-level threshold (which, in most cases, is ChimeraX’ built-in heuristic: the top 1% percentile of all the density values). We computed the reported performance metrics on a workstation that uses an Nvidia RTX 4090 GPU for DiffFit and a single thread on an AMD Ryzen Threadripper PRO 3995WX 2.70 GHz for ChimeraX (version 1.7.1 (2024-01-23)).

Scenario 2: Composite multiple structures

A more complex use case involves the fitting of multiple structures to a single, large, often complex, volumetric dataset, such as assembling a viral capsid from individual protein units or constructing a ribosomal complex from its constituent proteins and RNA molecules. DiffFit can handle such composite fitting tasks by iteratively optimizing the placement of each component, from largest to smallest, while considering the spatial relationships and interactions between them to prevent overlaps and ensure a coherent assembly.

Table 4.1: Performance results for fitting a single structure. Res stands for resolution, C stands for ChimeraX, D stands for DiffFit, M stands for MarkovFit [3], DC stands for DiffFit corrected by a single automatic ChimeraX fit; G stands for Gain and is D/C for Hit and C/D for Computing time (in seconds). High-avg stands for the averaged metrics for the high-resolution maps, and Med for medium maps, All for all maps.

PDB	Res	Hit rate			Computing time			RMSD (Å)			
		C	D	G	C	D	G	M	C	D	DC
6WTI	2.38	0.0	136.8	n/a	150.3	3.8	39.7	1.310	n/a	0.942	0.037
7D8X	2.60	0.0	202.0	n/a	196.0	5.2	37.6	1.960	n/a	0.984	0.014
7SP8	2.70	4.6	188	40.9	130.6	2.6	50.5	1.290	0.996	0.969	0.025
7STE	2.73	14.0	110.4	7.9	806.1	12.1	66.6	1.740	0.062	0.662	0.058
7JP0	3.20	5.4	191.8	35.5	250.7	6.7	37.2	2.540	0.017	0.922	0.015
7PM0	3.60	44.0	195.4	4.4	352.4	4.1	86.7	1.640	0.030	0.907	0.024
6M5U	3.80	0.0	105.0	n/a	162.2	4.1	39.2	2.360	n/a	0.912	0.018
6MEO	3.90	7.4	116.0	15.7	128.2	3.2	40.1	1.940	0.489	0.786	0.488
7MGE	3.94	4.8	123.6	25.8	337.6	4.3	78.1	1.870	0.017	0.819	0.017
High-avg		8.9	152.1	21.7	279.3	5.1	52.8	1.850	0.268	0.878	0.077
5NL2	6.60	1.8	163.2	90.7	94.6	2.0	48.0	2.440	0.093	1.124	0.056
7K2V	6.60	49.0	165.6	3.4	240.6	4.1	58.2	25.290	0.338	1.323	0.338
7CA5	7.60	55.8	72.4	1.3	322.6	2.9	110.0	3.290	2.042	1.207	2.042
5VH9	7.70	68.6	158.0	2.3	1147.8	14.1	81.3	0.960	0.085	0.991	0.085
6AR6	9.00	78.0	182.6	2.3	74.9	1.5	49.3	2.200	0.123	2.617	0.117
3J1Z	13.00	138.6	172.2	1.2	64.4	2.0	33.0	32.330	0.396	2.612	0.388
Med-avg		65.3	152.3	16.9	324.1	4.4	63.3	11.085	0.513	1.646	0.504
All-avg		31.5	152.2	19.8	297.3	4.9	57.0	5.544	0.366	1.185	0.248

In Figure 4.6 we show an example of compositing multiple structures into a cryo-EM volume map, for the PDB-protein [8SMK](#) [130]. In the first row we demonstrate how the middle and bottom parts are fitted in the first computation round, with the remaining top part of the protein being fitted in the second round of the interactive process.

The ability to *zero-out* densities, once a fit is interactively confirmed by the expert, allows DiffFit to fit multiple structures sequentially without interference from previously placed components, as we demonstrate in Figure 4.6. This iterative approach is particularly useful for densely packed molecular complexes, where individual components may be difficult to distinguish in the volumetric data. This

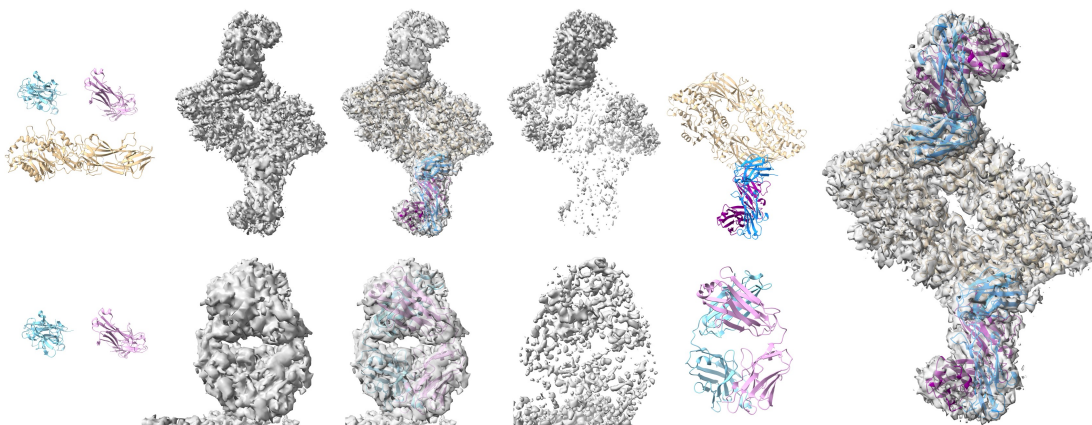


Figure 4.6: Compositing a protein (PDB 8SMK [130]) from its three unique chains. Top row from left to right: three input chains, input target volume, the best fits in the first fitting round, the remaining voxels after *zeroing-out*, and the fitted chains. Bottom row from left to right: two remaining input chains, remaining region of interest in the target volume from the first round, the best fits in the second round, the remaining voxels after *zeroing-out*, the fitted chains. Right: the final composited structure overlaid on the original target volume (RMSD: 0.138). The involved computation takes 10 seconds in total, and the human-in-the-loop interaction takes ≈ 3 minutes.

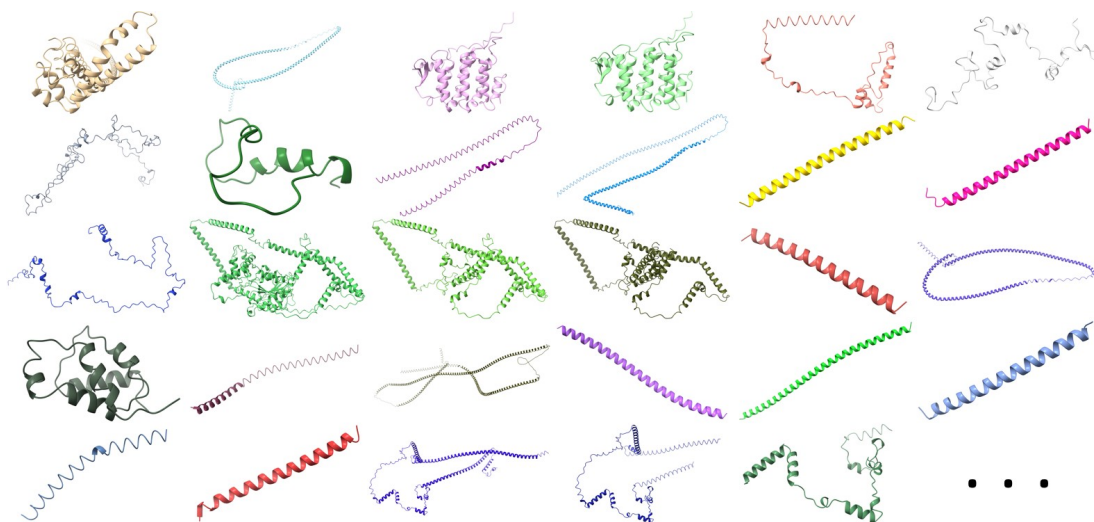
scenario is critical for understanding the functional context of proteins in larger biomolecular assemblies or cellular environments.

Scenario 3: Identify unknown densities

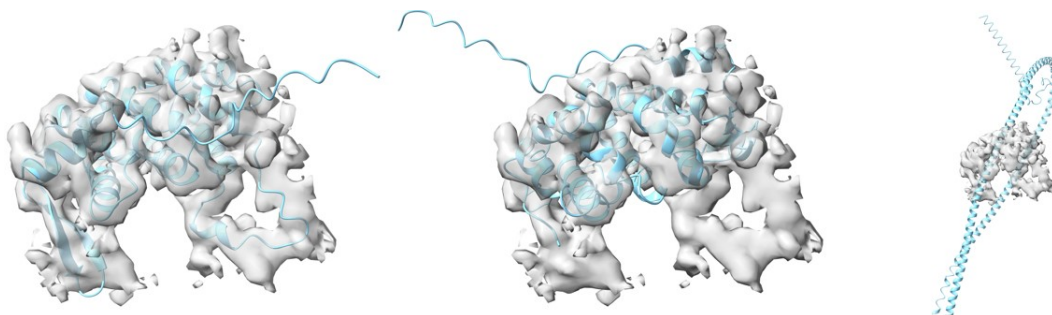
DiffFit also offers bioscientists the capability to identify and characterize unknown densities within volumetric datasets. In cases where a volume contains unassigned or ambiguous regions, possibly indicating the presence of previously unidentified molecules or molecular complexes, DiffFit can be used to screen a library of known structures and predicted structures for potential matches.

By fitting the structures from a library to the unidentified densities and evaluating the fit quality, researchers can hypothesize the identity of the unknown components. This scenario is invaluable for discovery-based research, where identifying novel components in complex molecular assemblies could lead to significant biological insights.

We performed a search with the demonstration dataset from a recent automated domain-level protein identification technique, DomainFit [34]. The task was to



(a) Library of structures to search against (subset).



(b) Unknown density identification: comparison of three potential fits which are overlaid on top of the target volumes.

Figure 4.7: Unknown density identification, where dozens to hundreds of molecular structures can be evaluated for potential fit.

Table 4.2: Performance results for identifying unknown structures. C stands for ChimeraX, D stands for DiffFit; Gain = D/C for Hit.

Structure	C Hit	D Hit	Gain
I7MLV6_D3	108	254	$2.4\times$
I7M317_D1	127	240	$1.9\times$

identify, from a library of 359 protein domains (Figure 4.7(a) shows a subset), which domains best fits into the given volume (Figure 4.7(b)). DomainFit identified two candidates (I7MLV6_D3, and I7M317_D1), after first fitting all domains using the `fitmap` command in ChimeraX and then performing statistical analyses to remove false positives. With our DiffFit, we identified these same two candidates with our fitting technique followed by a visual inspection, where the two candidates were the first to be inspected. We report the hit rates in Table 4.2, which shows a $\approx 2\times$ gain on the hit rate. In addition, DomainFit takes ≈ 12 hours to finish the task, while with DiffFit it takes ≈ 7 minutes—a $103\times$ gain in computation time.

4.5 Feedback

In addition to these quantitative metrics, qualitative feedback from users plays a crucial role in evaluating the practical utility and user experience of DiffFit. We thus solicited feedback from a diverse group of users, including PhD students, structural biologists, and computational scientists; through surveys, interviews, and hands-on testing sessions.

Specifically, we emailed cryo-EM practitioners at our university to test our tool. We performed three Zoom-based demo sessions and one in-person demo and feedback session with those people who expressed interest. The first session involved five people from a research group that studies structural biology and engineering (one senior PhD student specializing in structural and computational biology, one senior PhD student in experimental biology about protein structures, one senior PhD student in experiments elucidating the structure and function of proteins, one postdoc in experimental biology about protein structures involved

in cell signaling, one research scientist in using Cryo-EM and other techniques to study protein structures). The second session involved two people from a research group focusing on using cryo-EM to study the structural biology of human DNA replication and repair (an assistant professor and his postdoc). The third session involved one junior PhD student using cryo-EM to solve protein structures. The final in-person session involved one research scientist who provides cryo-EM service as a platform to the whole university and builds a bridge between the microscope and users at all levels. Apart from verbal feedback during and after these demo sessions, one participant in the first group, as well as the participant in the in-person session, also sent us written feedback, which we include in section A.2. In total, nine people provided qualitative feedback, which we report next.

Usability. The participants appreciated DiffFit’s intuitive interface and workflow (“quite intuitive and easy to use”) and the direct integration into ChimeraX with its known UI, which significantly lowers the barrier to entry, in particular, for new users, while still providing advanced features for experienced researchers. In particular the easy one-step fitting of a PDB file into a cryo-EM map was highlighted. The visually guided fitting process was appreciated as a powerful feature for refining fitting results based on expert judgment. Suggestions for further improvements of this interaction step and the UI in general were to display several visual clusters at a time, to visually compare the fit results, and to offer a tab-based method to check the various clusters. This approach would make it easier to observe how each cluster fits relative to the cryo-EM volume; thus, researchers would be better able to focus on a chain of interest. We have addressed this point already with our visual fit assessment tools (Figure 4.4 and section 4.2). One issue we noticed is that some experts commented on the computation times in the order of a few minutes, which points to the fact that they did not have a Compute Unified Device Architecture (CUDA)-compatible GPU at their disposal. As we demonstrated with our runtime analysis in section 4.4, with such affordable hardware, the computation is possible in well under a minute for many structures,

so cryo-EM labs can break free from extremely long computation times or the need for employing large (and expensive) computation resources.

Impact on Research. The participants reported that DiffFit has the potential to have a tangible impact on their research, enabling them to address complex fitting challenges that were previously out of reach. They thus get the ability to rapidly sample a large database of candidate structures for regions with unassigned protein density. The new ability to fit structures rapidly and accurately into volumetric data offers new directions of investigation and has the potential to accelerate discoveries in structural biology. One respondent stated that our automatic fitting and visual inspection approach “could be a key feature in ChimeraX that [could become] a standard in many pipelines” as well as “a key implementation [for] a standard modeling workflow.”

The respondents also made suggestions for future developments such as better structuring the handling of the associated files, further exploring the design space of the abstract cluster representations, a detailed protocol for downloading and installing the tool (already implemented in our GitHub repository), as well as adding workflows that would require automatically creating model subdivisions and performing fits on those before integrating the results into the reference volume. They also highlighted further potential uses, such as employing our approach to analyze density maps generated using X-ray crystallography.

4.6 Limitations and Future Work

Although our DiffFit technique demonstrates a substantial improvement over state-of-the-art techniques, it also has some limitations. One issue arises in Scenario 2, due to the gradual *zeroing-out* of the target map, as multiple structures are being composited onto a single map. In this case, the volume removal can potentially remove some voxels that are also part of adjacent interacting molecular subunits, especially in those cases when the chosen fit is not exactly perfect. Then, the subsequent subunit to be placed may have a higher difficulty of finding its correct

location because these few zeroed voxels penalize the correct position. We are currently investigating this limitation and plan to experiment with heuristics that remove only voxels that are fully covered by the molecular structure or slightly shrink (by using a higher user-specified threshold) the *to-be-zeroed-out* region before removing it from the map. Another related limitation is that the fitting process does not check for collisions in adjacent subunits, as previously investigated in visualization work (e.g., [92]). While the map shrinking approach should address this issue, subunits may still overlap in some cases. Resolving this issue is straightforward: we could remove those subunit poses that overlap with previously confirmed subunit placements or could minimally modify the pose to resolve the structural collisions.

Our technique is also not entirely parameter-free. The threshold value for removal of voxels with small density values, e.g., and the threshold value for the connected-voxel minimal cluster size have to be manually set. Manipulating these parameters requires prior domain experience with cryo-EM data from the bioscientists. We plan to automate the identification of these thresholds or, at least, define suitable default values and provide guidance on reasonable parameter ranges.

The visualization design in our current version of the tool also offers only the most essential visual encoding types, with a clear potential for further improvement. We plan to adapt comparative visualization techniques for intersecting surfaces and smart visibility techniques for combined rendering of the volumetric density representation with molecular surfaces or cartoon representations. Furthermore, the visualization design for analyzing the structural poses in a cluster suffers from high spatial occlusion among the subunits' various poses (see Figure 4.2(d)). This is a common domain problem and requires a dedicated research effort to combat the occlusion of this magnitude. Our tool can also be further extended in future work to, for instance, change the color encoding of the fitting results presentation to use one of the other alignment metrics (section 4.2) or to change the size of the

spheres in the results summary to make larger clusters appear more prominent.

Driven by the feedback we received, additional possible future research directions include extending DiffFit to deformable transformations, considering stoichiometry and symmetry, and speeding it up even further through an integration with CUDA.

4.7 Discussion

DiffFit provides a novel and efficient solution to atom-to-map fitting in structural biology, combining differentiable optimization with a human-in-the-loop visual analytics workflow within ChimeraX. We address these challenges with our differentiable fitting along with a set of essential strategies that make our algorithm robust for the domain problem. Our approach adds a human-in-the-loop visual analytics approach to the workflow and provides an open-source package designed to work as part of a standard software tool on the domain (ChimeraX). A key element of our approach is the change from pixel-to-pixel (or the equivalent, voxel-to-voxel) fitting to point-to-volume fitting, which enabled us to deal with the specific constraints of the fitting problem in structural biology. This observation also suggests a hypothesis for the originating graphics technique [85]: sparse, structure-aware sampling (atoms/landmarks) may suffice for effective optimization, implying that differentiable compositing could benefit from principled pixel/voxel sparsification schedules—an avenue for future work.

Our results depend on map quality (resolution, noise) and segmentation accuracy; ambiguous density can produce plausible but incorrect fits. Current experiments focus on rigid-body placement; flexible fitting and subunit rearrangements remain open. Multi-subunit fitting may encounter steric clashes that require additional constraints. Evaluation metrics (e.g., mean overlap, correlation, atom-wise weights) are informative but not perfect surrogates for biological plausibility. The expert feedback we collected was informal; a controlled study with pre-registered tasks, recruiting criteria, and measures (time, accuracy, con-

fidence) is needed to quantify usability and decision quality at scale. Finally, while GPU parallelism accelerates sampling, very large assemblies may still be compute-bound.

The quantitative performance metrics we reported collectively demonstrate DiffFit’s substantial improvement over traditional approaches, highlighting its potential to transform the field of structural biology. This potential is particularly large because the workflows of Assemblin, a protocol published in 2022 [84], and DomainFit [34], so far, rely on ChimeraX’s fitmap command as their first step. This stage can now be replaced by our more effective DiffFit, laying a new foundation for these and other workflows. By demonstrating the effectiveness of our technique across three scenarios, from fitting individual structures to assembling complex molecular architectures and identifying unknown components, we demonstrated that DiffFit overcomes the limitations of manually placing individual molecules before fitting, makes the compositing faster, more accurate, and intuitive, and opens the possibility of scanning the whole set of known and predicted molecules with the current computational resources. Ultimately, we thus escape the [Faustian bargain](#) [118] and instead are now free ourselves to explore the inner workings of the biological world.

Chapter 5

ProteinCraft: Integrative visualization of protein attributes and residue interactions in the AI era

DiffFit—and especially the success of its ChimeraX plugin—served as an unexpected catalyst for the next stage of my research journey. The impact of DiffFit in the structural biology community drew the attention of Prof. Le Song, a leader in AI-driven protein design, who identified the potential of similar visual analytics systems to transform the workflow for generative AI models in biology, especially those for protein modeling. Recognizing that visual interfaces were critical for navigating and interpreting the vast, complex datasets produced by AI in biology, Song approached our group to initiate a collaboration. After a personal trip of mine that included both tourism and a visit to Song’s institute, Mohamed bin Zayed University of Artificial Intelligence,¹ and further exchanges with my supervisor, we committed to this new direction together.

As defined in Chapter 1, AI-driven protein-binder design clearly manifests the *non-optimizable gap*: objectives are multiple and competing (affinity, stability, specificity, developability), the search space couples discrete sequence choices with continuous backbone and pose variables, and predictors return large, heterogeneous ensembles rather than a single optimum (e.g., conflicting scoring and contact signals). **ProteinCraft** targets this regime by coupling automated generation and prediction with coordinated 2D/3D visual analytics that let experts steer computation—triaging thousands of designs, aligning promising backbones to intended poses, applying local “jittering,” reseeding sequences, and iteratively validating with residue-contact and scoring evidence. In effect, the system moves

¹<https://mbzuai.ac.ae/>

algorithmic effort from the rough, multi-minimum landscape to the “sweet spots” around viable minima while reserving human attention for diagnosis, prioritization, and selection.

This initiative led directly to the conception of ProteinCraft, my most ambitious project to date. Drawing from the lessons and design philosophies that underpinned DiffFit, I developed ProteinCraft to integrate the scalable graph analytics of Tulip [6] with the molecular rendering capabilities of ChimeraX [80], providing researchers with an interactive platform to explore, filter, and analyze thousands of AI-generated protein binder designs. By empowering scientists to seamlessly bridge structural, sequence, and functional information, ProteinCraft illustrates how visualization becomes not merely a supporting tool, but an indispensable partner in accelerating discovery in the era of AI-driven protein design. I present ProteinCraft in its current form, based primarily on my in-preparation manuscript, with ongoing modifications and refinements to align with the evolving narrative of this dissertation.

5.1 Introduction

Recent breakthroughs in artificial intelligence have significantly advanced the prediction and design of protein structures, sequences, and their structural and functional attributes [1, 124]. Of particular interest is the potential to design protein binders, engineered proteins designed to selectively interact with specific target molecules, which hold immense promise for therapeutics and biotechnology [22, 11, 115]. Intuitive and interactive visual analysis of the extensive and complex spatial and scalar datasets generated in the involved workflows can enhance data filtering, facilitate hypothesis generation, deepen understanding, and boost workflow efficiency [48].

Despite advances in AI-driven protein design, researchers still rely on fragmented toolchains. Tools like ChimeraX [80] or PyMOL [93] excel at molecular rendering but lack integrated multivariate data visualization, whereas RING [24]

computes residue-level interactions for a single structure without linking to broader statistical analysis. The visualization community has developed graph-based techniques and coordinated multi-view layouts—such as Tulip [6]—but these approaches have not been systematically applied to protein engineering workflows. Consequently, identifying critical structural insights across hundreds of candidate designs remains cumbersome, underscoring the need for scalable, integrative platforms that bridge structure, interaction, and attribute data to enable comprehensive analyses.

We developed ProteinCraft (Figure 5.1) to overcome these limitations and facilitate an integrative visualization of multivariate protein attributes and structures, with a focus on residue interactions. ProteinCraft is a scalable interactive visualization system that combines Tulip’s powerful graph visualization capabilities with ChimeraX’s advanced structural viewer, and couples both with novel visual abstractions, supporting simultaneous exploration of up to millions of protein structures bearing billions of residue interactions and heterogeneous attributes. Specifically, it allows us to associate one line record of numerical and string attributes with multiple protein structure paths, and provides parallel coordinates, histograms, and scatter plots for attribute exploration with on-the-fly brushing and filtering. A distinguishing strength of ProteinCraft is the seamless synchronization between the abstracted information visualization and structural visualization, further enhanced by the novel interaction Tetris view and the residue binding bouquet view to pool the binding information across a batch of structures to quickly identify batch residue binding patterns.

Protein design workflows fall into two main categories: backbone-then-sequence [124, 22, 128] and backbone-sequence co-design [38, 76]. To accommodate all current and potential future paradigms, we pivot each design record to its final predicted structure (derived from the designed sequence) and link it to any intermediate models and attributes generated along the way. In this way, ProteinCraft is workflow-agnostic and can also be utilized to explore all deposited

protein structures. To pursue a more universal tool that is independent of specific post-design analysis workflows, we perform pre-processing steps including residue interaction identification, dimensionality reduction, and evaluation metrics calculation externally—a strategy that is similarly employed by other complex integrative visualization systems such as Vitessce [48].

Given the rapid development and growing adoption of AI-driven workflows and ProteinCraft’s capability to examine publicly available structures [12, 114], we designed the tool to serve a broad audience of structure data curators, consumers, and producers. To empower this audience to scrutinize—and ultimately refine—protein structures using integrative visualizations, we tackle five main challenges:

1. **Manage large, heterogeneous datasets:** Protein design projects can encompass tens of thousands of records, each comprising multiple structure models (e.g., backbone, sequence-populated, predicted). Individual structures may exhibit hundreds to thousands of residue interactions and an arbitrary set of tool-generated attributes; chain counts can vary between use scenarios. In addition, folder hierarchies are often arbitrary. We thus consolidate metadata in a single master Comma-separated values (CSV) file, where each row represents a design record identified by the predicted structure’s file path, along with paths to alternative models and associated attributes. This approach allows us to handle the large, heterogeneous datasets in a unified manner. With a column name agnostic table viewer and editor, users can easily rank, navigate, and filter the data.
2. **Support multi-level analysis:** Researchers require both high-throughput comparisons—ranking thousands of candidate designs by metrics such as binding affinity or predicted stability—and detailed, per-structure inspections of residue-level interaction networks. Yet integrating 3D coordinates, graph-based contacts, and numerical attributes into a unified interface is nontrivial. We leverage Tulip’s parallel coordinates, histograms, and scatter

plots for batch-level exploration. To track conserved residue bindings, we maintain a CSV file, whose rows correspond to target-chain residues and encode their interacting binder residues as [JavaScript Object Notation (JSON)], along with the associated structure path; we render this representation in our novel Tetris view. For fine-grained analysis we visualize residue interaction graphs in Tulip and synchronize bond-selection events with ChimeraX’s cartoon and atomic-style renderings.

3. **Mitigate visual clutter and occlusion:** Rendering thousands of designs in parallel coordinates or displaying dense all-atom structures and interaction graphs can overwhelm users with overlapping edges and crowded residues. To reduce noise, we leverage subgraph extraction in Tulip such that users can isolate and examine a subset of interest. For multi-chain complexes, we provide a bipartite [5] interaction view that separately shows interactions between two sets of chains (usually binder and target chains). At the ChimeraX side and for binder design scenarios, we pre-align all target structures to the same reference target. When applicable, we can apply the same strategy to other use cases on demand using ChimeraX’s `matchmaker` call. We further simplify residue interactions by drawing two pseudobonds per contact: a coarse backbone link for a low-density overview and a fine atom bond for detailed inspection on demand. Finally, users can toggle between representations, apply color highlighting, and hide flanking residues to focus on key regions.
4. **Overcome technical barriers:** Integrating advanced visualization components (parallel coordinates, scatterplots, histograms, interaction graphs) with high-fidelity molecular rendering ordinarily demands extensive engineering—extending ChimeraX with custom visualization modules is labor-intensive, while embedding a full molecular viewer inside a generic graph toolkit incurs API incompatibility and performance hurdles. We circumvent these challenges by coupling Tulip and ChimeraX via a lightweight

messaging layer: Tulip manages attribute-centric displays and graph analytics; ChimeraX handles structure rendering; Tulip sends selected bonds to ChimeraX via its `remote-control` mechanism; This division of labor reuses mature codebases, achieves high performance on both sides, and accelerates development.

5. **Guide users through low in silico success rates:** AI-driven binder design pipelines routinely generate thousands of candidates; rigorous in silico filtering is required to distill this pool into high-confidence designs for experimental validation, which usually leads to a few dozens of designs. To prevent users from being overwhelmed by this “lottery,” ProteinCraft facilitates an iterative design cycle: highly promising backbones and key interactions identified in one round inform sequence redesign in the next, fostering a human-AI teaming paradigm [72] that is more efficient than the traditional brute-force computing.

ProteinCraft facilitates the visual exploration of large-scale protein structure datasets—spanning AI-designed binders, AI-predicted structures, and experimentally solved PDB entries—within a unified, interactive environment. Through the demonstrations in Figure 5.2, we further (apart from the system) contribute: (1) a new filter (interchain predicted aligned error < 10 , binder-aligned binder RMSD $< 1 \text{ \AA}$, pLDDT > 90 , interchain H-bonds ≥ 3) that leads to increased in-vitro success rate of binder design in a public dataset containing half a million designs [20]; (2) an observation that high-affinity binders in the state-of-the-art designs [128] always bear an ample number of H-bonds in their predicted structures; (3) a workflow to iteratively perform backbone and sequence redesigning and to include the interchain residue interactions in binder design that dramatically increases the in-silico success rate; (4) an observation of residue binding bouquets that can lead to new computational design methodologies; (5) a workflow to examine multiple structures in detail in general (outside the protein design context).

The development of ProteinCraft has been propelled by a close collaboration

of a visualization group with AI-driven protein design teams. We began with an RFDiffusion-based binder design pilot study [124, 22, 11] and quickly abstracted the system to be workflow-agnostic to support additional generative protocols and general protein structure analysis. Future work includes: building a communication layer with AI models to facilitate interactive prompt authoring and result analysis to form a “Copilot” for protein engineering as well as lowering the granularity of the dataset records from structure level to bond level to facilitate more fine-grained analysis.

Decoupled from any specific design pipeline, ProteinCraft provides a workflow-agnostic platform for the integrative visualization of protein structures, residue interactions, and multivariate attributes. To the best of our knowledge, it is the first system to unify high-throughput statistical analysis with interactive 3D rendering and to explicitly correlate residue-level contacts with iterative designing and filtering strategies. We anticipate that ProteinCraft will empower the protein engineering community to design, refine, and analyze protein structures. In particular, similar to the atomic motifs in RFDiffusion2 [2], we foresee that the next generation of protein binder design models will leverage the batch-level residue binding bouquets as input to guide the iterative design and refinement of binder structures.

5.2 Method

System Architecture

We designed ProteinCraft as a modular, workflow-agnostic system for integrative visualization of protein structures, residue interactions, and multivariate attributes. To achieve this, we tightly couple two mature open-source platforms: Tulip [6], which we use for scalable graph and attribute visualization, and ChimeraX [80], which provides high-fidelity molecular structure rendering. We synchronize these environments through a lightweight, custom-developed messaging layer that we built specifically for real-time interaction and data exchange.

Tulip: Attribute and Interaction Visualization

We chose Tulip as our core environment for large-scale data exploration and graph-based analytics because it offers robust scalability, a flexible plugin architecture, and comprehensive support for interactive, multi-view visualization of complex networks. Within ProteinCraft, we leverage Tulip’s scalable architecture and plugin system to manage heterogeneous protein design datasets, visualize the predicted attributes and residue interactions as interactive graphs, and provide coordinated multi-view analytics. Specifically, we utilized and adapted the following features:

- **Attribute Exploration:** We treat the protein design workflow’s output data as a graph, consolidating all metadata and attributes in a single master CSV file where each row encodes a design record, including structure file paths and associated metrics. We utilize the column name-agnostic CSV importer, table viewer, and editor to enable users to easily navigate, rank, and filter large, heterogeneous datasets (C1). By consolidating data in this way, we can immediately pipe our records into Tulip’s native analytics and visualization features. We thus further take advantage of Tulip’s built-in parallel coordinate plots, scatter plots, and histograms to support high-

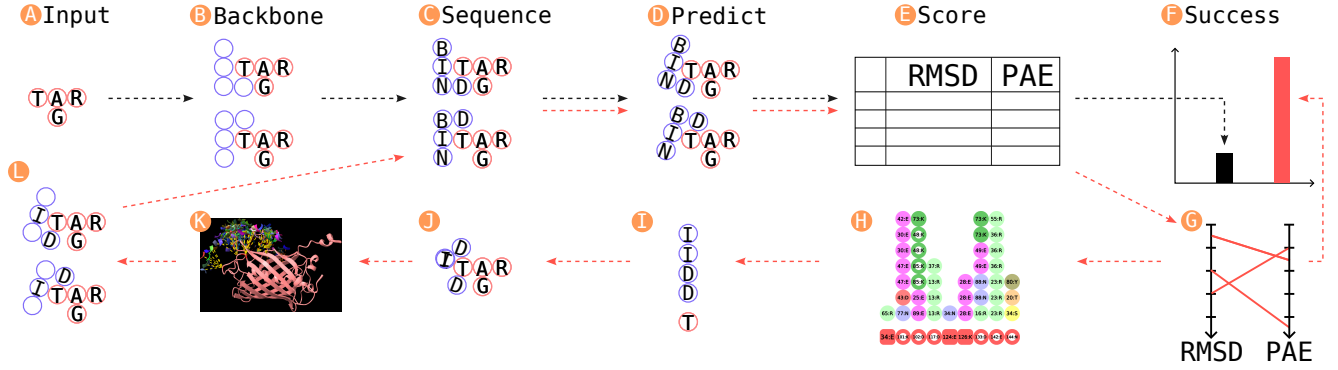


Figure 5.2: Use ProteinCraft in protein binder design workflow. Apart from **H** and **K**, all other sub-figures are skematic representations. **A.** Input target protein structure. **B.** Generated binder backbones from models like RFDiffusion. **C.** Populated sequences from models like ProteinMPNN. **D.** Predicted binder-target complex structures from models like AlphaFold2-initial guess. **E.** Ranking the binder candidates by predicted metrics. **F.** In-silico success rate by empirical filtering. **G.** Parallel coordinates view for predicted metrics ranking. **H.** Interaction tetris view for batch residue binding pattern identification. **I.** Schematic view to show the conserved bindings for residue "T". **J.** Schematic view to show the residue binding bouquet for residue "T". **K.** Binding bouquet view in practice. **L.** Feed the conserved bindings from the last round to the next round for sequence redesign.

throughput, multi-attribute exploration and candidate selection within this unified framework. Another advantage of our approach is the ability to create subgraphs directly in Tulip, enabling users to isolate and analyze specific subsets of data. This functionality helps mitigate visual clutter and occlusion, addressing the challenge of making sense of complex, overlapping relationships in large-scale datasets (C3).

- Dimensionality Reduction:** We use Uniform Manifold Approximation and Projection (UMAP) (with the option to switch to PCA or other dimensionality reduction methods) to project high-dimensional protein structures into two dimensions, allowing users to visually assess the diversity or clustering of designs. Specifically, we developed a Python script to extract alpha-carbon (CA) coordinates from the designed binder chain—while retaining the flexibility to extend this process to any chain in the structure. We construct fixed-length feature vectors via zero-padding and generate 2D embeddings using UMAP. We then append these 2D coordinates to the

master CSV file and set the layout properties of each record in Tulip to these values. As a result, node positions in Tulip’s node-link diagrams reflect the dimensionality-reduced layout, enabling intuitive and effective exploration of structural similarity among large sets of protein designs.

- Residue-Level Interaction Views:** We developed custom views—including the interaction Tetris view and bipartite interaction graphs—to summarize and allow users to examine residue-level contacts across large batches of protein structures. To systematically track conserved residue bindings, we generate a dedicated CSV file where each row represents a target-chain residue and encodes its interacting binder residues in JSON format, along with the relevant structure path. Our custom Python script parses interaction outputs (e.g., from RING [24]), aggregates binder–target residue contacts, and writes them in this tabular format. In Tulip, we use a specialized plugin to import this file and construct the Tetris view, where target residues are arranged as a single row at the bottom and their interacting residues are stacked vertically above them. This Tetris-style aggregation reveals conserved binding patterns across the dataset. For fine-grained analysis, we implemented a specialized Tulip plugin that constructs bipartite interaction graphs. This plugin automatically identifies all residues in the binder (chain A) and target (chain B) that participate in non-covalent interactions, and generates a subgraph containing just these nodes and their connecting edges. The layout arranges target residues along one axis and binder residues along a parallel axis, creating a bipartite visualization that makes it straightforward to inspect which residues interact across chains. The algorithm further optimizes the node arrangement to minimize edge crossings and total edge length, thus reducing visual clutter and improving interpretability. Node attributes such as residue type and secondary structure are encoded by color and shape, enabling efficient visualization and analysis of conserved binding patterns and residue-level interaction diversity,

and supporting intuitive, comparative exploration across large protein design datasets.

- **Coordinated Linked Views:** We support real-time selection and brushing across all views, enabling users to seamlessly link tabular, graphical, and structural representations. Users can customize their workspace by creating, arranging, pinning, linking, and deleting views, with up to six panels in a single workspace. Efficiency and scalability are further enhanced through a suite of utilities (Figure 5.1 I–L): users can drag and drop to reorder views, access a set of predefined algorithms for opening and synchronizing views and for analyzing data graphs, and navigate a hierarchical contents viewer for easy management and subgraph filtering of opened graphs. In addition, a built-in Python script editor and interpreter enables advanced data operations within the workspace, supporting complex and flexible exploration tasks.

ChimeraX: High-Fidelity Structure Rendering and Synchronized Exploration

ProteinCraft leverages ChimeraX as its core molecular structure viewer, supporting interactive inspection and advanced residue-level interaction visualization. Through a tightly integrated architecture, we bridge large-scale, attribute-centric analytics with detailed three-dimensional structural context, allowing users to move fluidly from high-level data filtering to fine-grained molecular structure examination. Our approach combines specialized data abstractions, a robust synchronization protocol, and feature-rich visualization controls.

Load RING data. To support advanced residue-level interaction analysis and visualization in ChimeraX, we leverage a robust data management strategy based on the RING [24] data structure. Each protein structure, when loaded, is associated with a RING instance that maintains comprehensive records of all

residue-residue contacts, their types (e.g., hydrogen bond, ionic, π - π stacking, van der Waals, disulfide, etc.), and their properties (distance, angle, involved atoms). Each contact is indexed by a **bond key**—a unique string constructed as `chain1:resnum1:atom1→chain2:resnum2:atom2|interaction_type`—which encodes both the identities of the atoms and residues involved and the type of interaction, e.g., `A:57:CA→B:99:CB|HBOND:MC_SC`. The RING object stores these bonds using their keys and manages associated pseudobond objects for efficient toggling and display within the ChimeraX environment. Structures are indexed by file path, enabling programmatic control and rapid retrieval of models and their associated interaction data. All bond types are color-coded and can be filtered, grouped, or toggled via both graphical and command-line interfaces, supporting both automation and interactive analysis.

For the visualization of these residue-level contacts, we implement two complementary drawing styles for pseudobonds. In **cartoon mode**, pseudobonds are drawn between backbone atoms (typically the $C\alpha$ atoms or principal backbone atoms) of the interacting residues, providing a simplified and uncluttered representation suitable for overview analysis. In **atom mode**, pseudobonds are rendered between the actual atoms involved in the interaction (e.g., sidechain or backbone atoms as specified in the contact), allowing for detailed, chemically precise inspection. Users can toggle between these modes, or cycle through them, using toolbar actions or commands, enabling flexible navigation between high-level and atomistic perspectives on the interaction network.

Binding Bouquet View. While the Interaction Tetris and bipartite/contact graph views facilitate large-scale contact analysis and are implemented in Tulip, we introduce the *Binding Bouquet View* natively in ChimeraX to summarize and visually encode the diversity and recurrence of binder residue contacts for each target residue in 3D. Upon loading a structure and its associated RING data and the user’s selection, the Binding Bouquet View displays all selected binder-side residues that interact with a specific target residue, presenting them spatially

as a “bouquet” around the target. This abstraction allows users to intuitively explore which positions are most consistently engaged across a batch, recognize binding hotspots, and relate these observations directly to the physical structure. Interactive controls in the ProteinCraft tool panel and toolbar allow users to adjust bouquet display, filter by interaction type, and quickly highlight or inspect specific motifs in the context of the target molecule. This targeted visualization, tightly coupled with the underlying RING data management, provides a powerful interface for binding motif discovery and for guiding iterative binder design directly within ChimeraX.

Synchronization Layer and Interactive Controls. To integrate data-driven analytics with molecular structure exploration, we implemented a lightweight, custom messaging protocol that transmits selected protein structures and bonds from Tulip to ChimeraX. This single-direction synchronization, realized via ChimeraX’s `remote-control` interface, enables real-time propagation of selections, ensuring all visual states remain consistent as the user explores. The protocol supports programmatic coordination of structural views, including control of multiple ChimeraX instances from a single Tulip session, facilitating collaborative or large-scale analyses.

Within ChimeraX, we provide extensive capabilities for interactive exploration and visualization. Users can load, superimpose, and manage predicted or experimental structures (PDB, mmCIF) with ChimeraX’s built-in functionalities. With our newly developed plugin, we enable users to visualize residue contacts as pseudobonds, supporting a variety of bond types, each with configurable colors and display modes. Users can toggle cartoon or atom representations, filter bonds by type or chain, and cycle visibility at both the individual and group level. The dedicated ProteinCraft tool panel organizes controls for bond types, model attributes, highlighting, and flanking regions into intuitive tabs, while toolbar actions provide rapid access to common workflows such as toggling bond types or applying highlights. Users can also highlight residues and local neighborhoods

with adjustable transparency and color blending, annotate special residues (e.g., mutations). All key actions—model loading, bond display, color and transparency adjustments, and chain category management—are accessible via both the GUI and a suite of custom ChimeraX commands, supporting advanced scripting and toolchain integration.

Through this unified, extensible architecture, ProteinCraft empowers researchers to seamlessly transition from large-scale data analytics in Tulip to detailed, context-rich 3D visualization and interpretation in ChimeraX. The system enables users to efficiently identify, inspect, and iterate on conserved residue-level interaction patterns and candidate designs, supporting discovery and hypothesis generation in modern protein engineering workflows.

Iterative Design and Feedback Workflow

A key innovation in ProteinCraft is its explicit support for iterative design cycles, leveraging feedback from structure prediction and interaction analysis to drive substantial improvements in binder design success rates. In practice, the landscape of backbone conformations in computational protein design is highly sensitive: a backbone that fails under conventional *in silico* filters (such as interchain PAE < 10) may be separated by only a few Angstroms from one that succeeds, indicating a steep and narrow energy funnel. Traditional approaches—random backbone sampling followed by brute-force sequence generation—frequently fail to capture these near-miss opportunities.

We overcome this limitation by enabling users to systematically propagate promising but imperfect backbone candidates through iterative rounds of redesign. ProteinCraft allows users to select candidates that narrowly miss strict design criteria (such as interchain PAE < 20), use predicted structures as new backbones, and launch additional rounds of sequence redesign. Through repeated prediction, analysis, and redesign, users can systematically improve the quality of binders and increase the pass rate for stringent *in silico* filters.

This iterative refinement is guided by interactive selection, multi-attribute filtering, and residue-level contact analysis. We find that generating and evaluating multiple sequences for each candidate backbone is essential to verify whether iterative improvement is possible, and that even small structural adjustments or alignment corrections can yield breakthroughs for otherwise intractable targets. Importantly, ProteinCraft decouples this feedback-driven process from any specific generative pipeline, supporting backbone-then-sequence, co-design, and emerging AI-driven protocols, as well as retrospective analysis of deposited PDB structures.

By providing an environment for tracking, selecting, and propagating intermediate backbones and key residue contacts across rounds, we enable researchers to systematically explore rugged energy landscapes, exploit near-miss opportunities, and achieve disruptive improvements in protein binder design.

Data Processing and Pre-Analysis

To ensure that ProteinCraft remains both workflow-agnostic and scalable, we perform all major data pre-processing steps outside the visualization environment. Residue contacts are identified using external tools such as RING [24], while dimensionality reduction (e.g., UMAP) and the calculation of evaluation metrics are carried out in batch scripts. All resulting metadata and attributes are consolidated into a master CSV file, which is then loaded into Tulip for analysis. This approach enables seamless handling of heterogeneous datasets, as the system is agnostic to column names and attribute types, ensuring flexibility and future extensibility.

By integrating scalable graph analytics with high-fidelity structure visualization and novel residue interaction abstractions, ProteinCraft enables users to explore, filter, and iteratively refine large protein design datasets. This tightly coupled system overcomes the limitations of brute-force workflows and accelerates the discovery and optimization of effective protein binders.

5.3 Implementation

We implemented ProteinCraft as a modular, high-performance visual analytics system, combining custom C++ plugins for Tulip with a Python extension for ChimeraX. Our architecture uses REST-based synchronization to enable seamless interactive analysis across both platforms.

Tulip Core and C++ Plugins. We extended Tulip with a suite of C++ plugins for project management, RING data import, and attribute-driven analytics. Our importers parse RING node and edge files, as well as tabular CSV data, to create detailed residue-level contact graphs, parallel coordinate views, Tetris interaction diagrams, and dimensionality-reduced layouts. We implemented dedicated plugins to generate subgraphs (such as bipartite or binder-target interaction networks), filter interactions, and apply custom layouts for clarity in complex graphs. We manage all project and structure state centrally, enabling linked selections and consistent view synchronization across all open graphs. We specify user parameters, attribute mappings, and color schemes via JSON configuration files for flexible extension to new data types.

ChimeraX Integration and Python Plugin. To provide interactive molecular structure visualization, we developed a Python plugin for ChimeraX. We use a REST API to receive real-time synchronization messages from Tulip, which encode selected structures, bond states, display modes (cartoon or atom), and highlighting commands. In ChimeraX, we provide an intuitive graphical user interface that lets users control residue-level bond display, toggle between cartoon and atom pseudobond rendering, filter and color bonds by interaction type, and highlight or flank specific residues. Our plugin supports rapid updates as the user explores data in Tulip, ensuring all selections and structure views stay synchronized. We manage binder and target chain categorization and provide toolbar shortcuts for common operations, streamlining iterative analysis.

Synchronization and Build. We implemented a lightweight REST-based protocol to transmit the current visualization state from Tulip to ChimeraX, ensuring robust, real-time coupling between data analytics and 3D molecular inspection. We compile all C++ modules as Tulip plugins using CMake, and maintain flexible configuration and extensibility through JSON-based project settings.

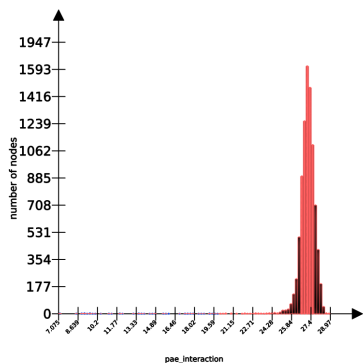
We have made all code and build instructions available at github.com/nanovis/ProteinCraft.

5.4 Use Case Scenarios

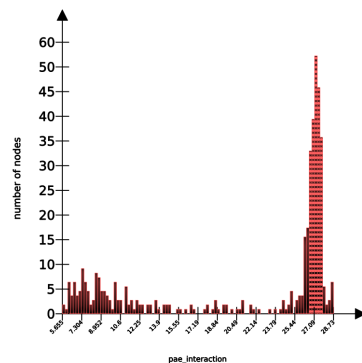
To illustrate the impact of our iterative feedback workflow, we present use case scenarios based on real-world applications of ProteinCraft in challenging binder design problems. Below, we detail two representative cases that highlight the efficiency and effectiveness of iterative, feedback-driven optimization compared to conventional brute-force approaches.

Case 1: SARS-CoV-2 Receptor Binding Domain—Iterative Rescue of Near-Miss Backbones

The design of binders targeting the SARS-CoV-2 Receptor Binding Domain exemplifies the challenges of protein engineering in a difficult regime. This target is recognized as more challenging than approximately 75% of representative *in silico* cases [128]. In our two-round experiment, an initial design campaign using standard RFdiffusion [124], ProteinMPNN [22], and AF2ig [11] workflows yielded only 18 successful designs (0.19%) out of 9,690 candidates (Figure 5.3(a)), as determined by the stringent filter of interchain PAE < 10 . By identifying and selecting the 61 backbones with interchain PAE < 20 , using their AF2ig-predicted structures as templates, and generating ten new sequences for each in a second round, we observed a dramatic increase: 92 designs out of 470 (19.6%) (Figure 5.3(b)) passed the strict PAE filter—a 103-fold gain in success rate. This



(a) Round 1: Initial design pool.



(b) Round 2: After iterative selection and redesign.

Figure 5.3: Distribution of inter chain PAE values for generated designs in ProteinCraft case study 1. **(a)** In the first round, nearly all designs cluster at PAE values higher than 24, indicating low-confidence binding in the initial pool. Only 18 out of 9,690 designs exhibit $\text{PAE} < 20$. **(b)** After iterative selection and redesign, the second round shows a broader distribution, with a clear increase in low-PAE designs. Specifically, 92 out of 470 designs achieve $\text{PAE} < 20$, reflecting the emergence of improved binders.

result highlights the power of leveraging intermediate “near-miss” backbones and feeding them back through iterative sequence redesign to systematically traverse challenging energy landscapes.

Case 2: Recovery from Hard Failures via Jittering and Alignment Correction

In other difficult scenarios, initial backbone sampling may yield no candidates that pass even a relaxed interchain $\text{PAE} < 20$ threshold. Here, ProteinCraft enables alternative strategies: we can select a backbone with strong intra-chain PAE (Figure 5.4(a), (d)), align it back to the intended design pose (Figure 5.4(b)), and perform local “jittering” to generate a large set of structurally similar variants. By generating multiple new sequences per backbone and iteratively re-applying structure prediction and contact analysis, we can eventually identify high-quality binders that overcome initial failures. For example, this approach reduced PAE from 26.341 (Figure 5.4(d)) to 7.994 (Figure 5.4(c)) in the next round, and further refining leads to as low as 4.4, with results independently verified by AlphaFold3 [1]

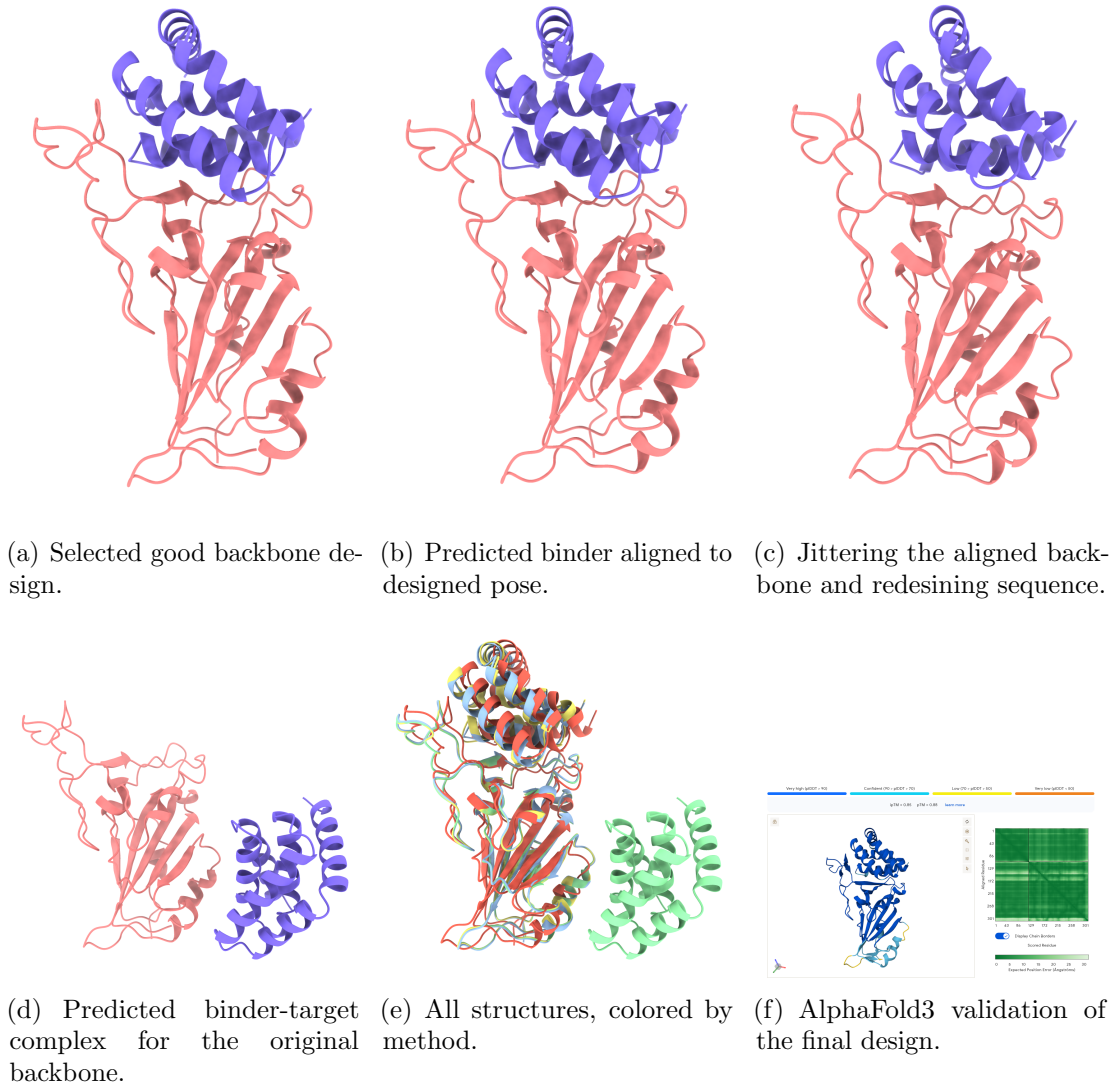


Figure 5.4: ProteinCraft case study 2: Iterative binder design and validation workflow.

(a) A promising backbone structure is selected from the initial design pool. (d) The structure of the binder-target complex is predicted for the original backbone with the sequence designed, allowing assessment of the initial binding interface. Inter-chain PAE = 26.341; intra-chain PAE = 3.143. (b) The predicted binder is aligned to the designed pose to enable sequence redesign. (c) The backbone's binder interface is locally perturbed ("jittered") and the sequence is redesigned to improve binding potential. Inter-chain PAE = 7.994. (e) All structures are shown together, colored by design method. Cyan for the original backbone, green for the predicted binder pose, yellow for the aligned binder pose, red for the jittered and sequence redesigned binder. (f) The final binder-target complex is validated using AlphaFold3, with model confidence scores and PAE scores visualized to confirm the design quality.

predictions (Figure 5.4(f)). Such recovery from hard cases demonstrates the flexibility of feedback-guided, interactive design workflows.

This strategy demonstrates that iterative backbone and sequence refinement—guided by intermediate structure predictions and residue-level contact analysis—can transform the design process from a “lottery” of random computation into a targeted, data-driven workflow.

5.5 Discussion

ProteinCraft represents a significant advance in the landscape of computational protein design, demonstrating how tightly integrated visual analytics, interactive feedback, and scalable data management can transform both everyday workflows and the boundaries of what is possible in the AI era. By bridging the gap between large-scale attribute exploration, fine-grained residue interaction analysis, and high-fidelity molecular visualization, we have enabled researchers to move beyond the limitations of fragmented toolchains and brute-force design pipelines.

Our workflow-agnostic architecture, built upon Tulip and ChimeraX, empowers users to interrogate, refine, and optimize large, heterogeneous protein datasets in ways not previously feasible. Iterative design and feedback—supported by synchronization, custom visual abstractions, and extensible plugin infrastructure—allow researchers to systematically convert near-miss candidates into high-quality binders, dramatically improving *in silico* success rates even for the most challenging targets. The examples presented here highlight not just performance gains, but also a fundamental shift in the design process: from random search to a targeted, data-driven, and human-guided exploration of the vast sequence-structure-function space.

Our case studies reflect specific targets and expert workflows; results may vary with target class and predictor choice. Metrics such as PAE/pLDDT are proxies and do not guarantee binding; experimental validation remains essential. Interactive steering can introduce user bias; provenance capture and replayable

pipelines mitigate but do not eliminate this risk. Compute budgets constrain iteration depth and ensemble size. A larger, controlled user study (tasks, baselines, cross-lab replication) is needed to quantify gains in time-to-solution and downstream success.

As AI-driven modeling continues to evolve, we foresee ProteinCraft serving as a blueprint for integrative visual analytics systems in computational biology and beyond. Future work will expand the platform’s capabilities, enabling even deeper coupling with generative models, finer-grained residue-level analytics, and new modes of human–AI collaboration. Ultimately, ProteinCraft’s contribution is not just technical, but conceptual: it affirms that in the era of big data and powerful prediction, visualization and interactive feedback remain essential to scientific discovery and innovation.

Chapter 6

SynopFrame: Multiscale time-dependent visual abstraction framework for analyzing DNA nanotechnology simulations

While I present SynopFrame here as the final system in my dissertation, it is actually the project that spanned the entirety of my PhD journey. My initial goal was to build a comprehensive framework capable of visualizing the rich abstraction space of DNA nanostructures—an ambition that demanded new strategies for organizing, interpreting, and interacting with large-scale, complex simulation data. Along the way, my work on DiffFit and ProteinCraft helped further validate the design philosophies that underpin SynopFrame. DiffFit, with its focused workflow and rudimentary table viewer, demonstrated how integrating automation and visual inspection could streamline the analysis of fitting results. ProteinCraft, in contrast, marked a significant leap in complexity: it integrated diverse representations and multiple coordinated views, enabling interactive exploration of massive AI-generated protein design datasets.

As defined in Chapter 1, the DNA-nanotechnology problems addressed here locate at the heavy human intervention end of the automation spectrum and lie in the *non-optimizable gap*: the task lacks a single target or scalar objective (e.g., *why did a design fail to assemble?* or *when/how does a conformation switch occur?*), the evidence is heterogeneous and time-dependent, and “correctness” depends on expert interpretation rather than a unique optimum. **SynopFrame** targets this regime by structuring simulation analysis into a *visualization space* that coordinates structural abstractions on granularity, visual idiom, and layout aspects, enabling linked navigation, overview, and detailed analysis. The system thus provides solutions for the open-ended questions for hypothesis formation and

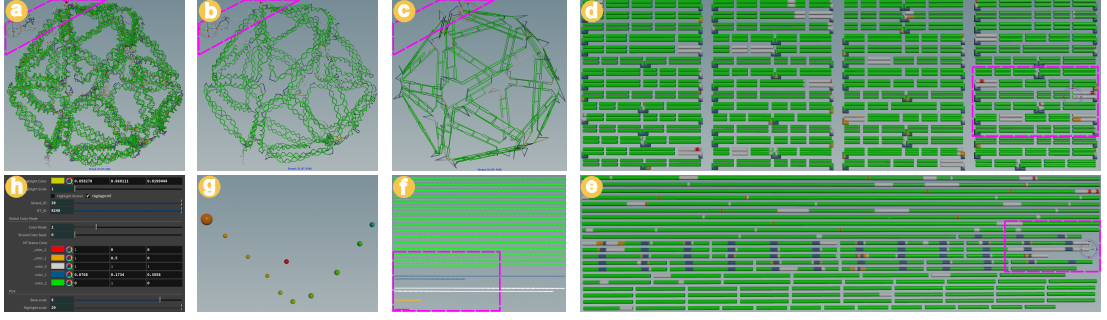


Figure 6.1: SynopFrame dashboard shows an icosahedron design (6540NTs) in various different representations (arranged in a clockwise fashion): (a) All-NT, (b) Snake, (c) Schematic3D, (d) Schematic2D, (e) Heatbar, (f) Progress bar, (g) Principal component analysis plot, and (h) Control panel.

validation—localizing failure modes, tracing transition pathways, and explaining outcomes at scale.

With SynopFrame, these principles are fully leveraged to address the unique challenges of DNA nanotechnology. The system I present in this chapter brings together detailed and abstract visualizations in a flexible, multiscale environment—empowering domain experts to navigate, interpret, and compare the dynamic evolution of large DNA assemblies over time. In particular, I organized all the related views in a unified visualization space, enabling seamless navigation and comparison across different levels of abstraction and representation. In this way, SynopFrame does not merely extend prior work, but encapsulates the ongoing evolution and maturation of my research perspective across my doctoral studies. I now present SynopFrame mainly as it appears in the original publication, with slight modifications to fit the narrative of this dissertation.

6.1 Introduction

Recent years have seen a series of breakthroughs in the DNA nanotechnology (DNA-nano) domain [95, 96], with a technique known as *DNA origami* [87] bringing dramatic success on various designs and use cases. A hallmark of DNA origami designs is their structure, constructed from one long scaffold strand that is folded by many short staple strands. The complexity of these designs,

which is often characterized by the structure’s huge size, the DNA strands’ complicated routing, and the high-frequency motion in their dynamics can lead to difficulties for experts in designing and analyzing them. This challenge becomes even larger when scientists rely on MDS to examine the behavior of DNA-nano structures at nanoscale resolution, where standard visual inspection methods can be overwhelmed by the dynamic complexity and large data size.

DNA-nano design and simulation tools, e.g., caDNAno [27], Adenita [23], CATANA [55], oxView [83], and oxDNA [104], provide semi-automatic workflows to lower these difficulties. Domain experts can author designs in high-level geometries (lines, squares, honeycomb), generate the DNA sequences for each strand, and then make modifications if necessary, followed by feeding the designs to MDS tools for further analysis.

Yet, experts still have to examine the structure carefully to envision the resulting design and to mentally connect the designed shape, the strands, and the sequences with the findings from the simulations. Visualizing the design can help experts inspect its structure directly. These designs, however, typically contain tens of thousands ofNTs on hundreds of strands. Some of the staple strands are “high-degree” strands that pair with several parts of the scaffold strand to form complicated folding patterns. Conventional visualizations of all theNTs in a design in turn generally produce cluttered and occluded views. In the past, Miao *et al.* [68] had solved some of these problems with abstract views, yet only for static structures rather than for dynamic simulations.

In our work we address the problem of efficiently interpreting and visually analyzing large-scale MDS trajectories for DNA-nano designs. We propose SynopFrame, a multi-viewport, multiscale, multi-dimensional, time-dependent, and comprehensive visual abstraction framework that aims to help experts identify and interpret the dynamic evolution of their DNA-nano structures. Given an MDS trajectory of a DNA-nano design, our solution offers interactive, synchronized viewports for both detailed and abstract representations as we show in Figure 6.1.

This approach enables users to: (1) navigate and compare the overall structural progress in relation to the designed shape, (2) identify problem regions via color-coded H-bond status, and (3) focus on specific areas for deeper inspection of local pairing events.

Overall, our contributions are threefold:

- we introduce an abstraction space that extends existing representations for DNA-nano structures and bridges design and MDS analysis;
- we provide a new way to categorize and encode H-bond status, enabling the quick identification of design flaws and conformational changes; and
- we develop a synchronized multi-view environment that links different abstraction levels (from detailed 3D shapes to schematic progress bars), helping experts effectively explore and interpret dynamic simulation data.

Our user feedback indicates that these contributions can substantially aid in explaining and troubleshooting unsuccessful self-assembly in DNA-nano designs.

6.2 Motivation, approach, and prerequisites

As we showed, one major shortcoming of the state of the art of visualizing DNA-nano designs is the lack of representation of the dynamic properties. We thus teamed up with experts who design DNA-nanotechnology structures as a part of their scientific work. Our primary collaborator is a domain expert in DNA-nano design and wet-lab experiments (a co-author of this paper), who has been leading a team that collectively designs components of nano-robots that would be able to destroy cancer cells or neutralize pathogens as a part of next-generation health treatments. To gain understanding of the domain workflows, we met several times a month over the period of six months. We were also in contact with several other experts, including developers of the oxDNA simulation package and researchers, who all are familiar with DNA origami experiments.

Goals

Through our discussions with these experts, we established a set of high-level goals. The primary objective for DNA-nano designers is to understand the dynamic behavior of their nanoscale structures—specifically, to predict whether a design will self-assemble correctly or exhibit a certain behavior in a wet-lab experiment. MD simulations are a key computational tool for this purpose, allowing scientists to test designs cost-effectively before committing to expensive and time-consuming lab work. The analysis of the resulting large and complex trajectory data, however, is a major bottleneck. Currently, experts analyze MDS properties by compiling statistics such as energy and H-bond occupancy, but these approaches often lose structural information and provide little insight into the structure’s dynamic behavior. Structural information is currently conveyed by a fast-forward animation or selection of representative images capturing simulation emergence. Both these methods are of presentational character and cannot be used as analytical tools that would lead to structure-related insight. Therefore, our central goal is to facilitate the efficient visual analysis of DNA-nanotechnology MDS trajectories to help experts interpret simulation outcomes and gain actionable insights for improving their designs.

Task analysis

To translate our high-level goal into concrete requirements, we worked with our collaborators to identify key analysis scenarios and derive a set of user tasks that a visualization tool must support. First, we found that beginners and lay audiences often view dynamic processes using schematic diagrams to comprehend the process as a whole. Second, experts often work on abstracted spatial views during the designing phase to keep the cognitive load low. However, the DNA simulation is usually performed at a more detailed granularity, such as theNT and atomistic levels. So the MDS results visualization needs to bridge the gap between the abstracted spatial views and the detailed views. In a third key scenario,

the process of structural assembly and disassembly as generated by the DNA simulation models is also of great importance to domain scientists. Of particular interest here is the information about the order of association and dis-association on two conceptual levels of structural detail: on the nucleotide level and on the level of a DNA strand. Finally, experts sometimes do not understand why their designed 3D structure does not self-assemble in laboratory experiments, even with MDS results at hand. So they are looking for ways to examine these structures. From these scenarios, we distilled the following essential user tasks:

T1: Assess the overall quality of the simulation—seeing the current structural assembly progress in comparison to the fully assembled state can help one decide whether more simulation time or more detail is needed.

T2: Identify and interpret patterns throughout the simulation—the high-level observation of an MDS run in the schematic view allows the experts to pinpoint potential problems of a simulation run, interesting simulation periods, and the change of H-bond status over time.

T3: Examine H-bond pairing events for specific, interesting periods—seeing the order and the exact position of individual NT and strand pairing can lead to insights and thus the identification of problems with the design. Thus the proposed solution should allow users to focus on specific periods from the whole simulation and to analyze them in detail, frame by frame.

T4: Inspect how one structural conformation converts to another—the aforementioned tasks should easily interplay with each other to enable the experts to comprehensively and seamlessly understand the overall biological dynamics at both abstract and detailed levels.

T5: Determine why some structures do not form—one of the major challenges in DNA-nano is the low yield in wet-lab assembly experiments, and sometimes (e. g., our case study in section 6.4) the structure does not form at all. It is thus important to study the dynamic simulation to understand the reasons for the low yield.

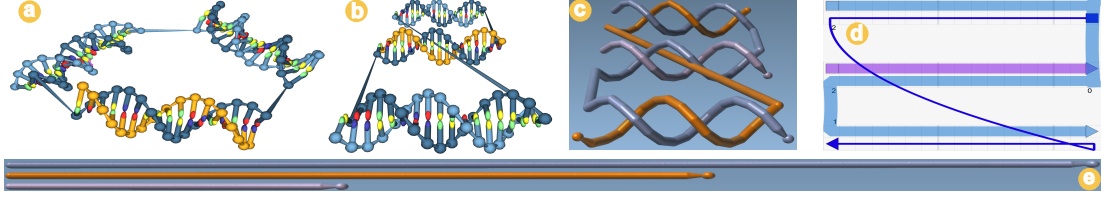


Figure 6.2: Inspirations for fundamental representations from existing tools, all for the same triangle structure: (a) **All-NT**—3D spatial representation from oxView withNT detail, in its relaxed configuration; (b) All-NT in the triangle’s designed configuration; (c) **Snake**—Miao *et al.*’s [68] “Single Strands” 3D spatial representation that only shows each DNA strand as a backbone; (d) **Schematic2D**—parallel straight lines showing the pairing between the scaffold and staples as well as the “flying lines” (lines that diagonally cross the structures) showing the linking between different parts of the staples; (e) **Heatbar**—each DNA strand straightened and sorted by length, as done by Miao *et al.* [68].

T6: Present real data and avoid artifacts such as those caused by structural averaging—the averaged structure of a trajectory often deviates from any specific frame and bears artifacts that might lead to false insights [83]. Such problems are even less visible in more abstracted statistics. It is thus essential to show both the aggregated and the non-aggregated real data from the simulator.

These tasks can be coupled with conventional analysis approaches such as energy, distance, or angle graphs. Such analysis is essential yet beyond the scope of this article: it does not directly deal with structural abstraction, on which we focus here.

Challenge analysis

To effectively support the identified tasks, we first had to understand the inherent challenges posed by the data itself. We found that issues arise due to the specific properties of the MDS trajectories, which prevent existing tools from producing visualizations that are effective for drawing actionable conclusions.

Specifically, we inspected a range of various MDS trajectories in oxView. We visualized small and large systems, with short and long simulation durations, for assembly/disassembly and stability simulations. We found that, for extremely small systems (less than 50NT) and extremely short durations (less than 100

frames), oxView works well. For other configurations we applied the widely recommended structural alignment as well as several smoothing methods but realized that several challenges exist that prevent oxView and similar tools from producing effective visualizations that show both the spatial structure and the dynamic behavior to come to actionable conclusions:

- C1: High frequency, large structure, many frames.** The high motion frequency is due to eachNT’s position changing in all consecutive frames. When this characteristic meets a large structure with tens of thousands ofNTs and tens of thousands of frames or more, analysts face a huge amount of information for each step. No matter how slowly the trajectory is being presented, the resulting visualization always vastly exceeds an analyst’s ability to comprehend and make use of it as long as the actual position of eachNT is shown. And even if one can invest the time to digest the changes between two frames, this is an extremely inefficient way of studying the trajectory because MDS often exhibits long periods devoid of any important events.
- C2: Clutter and occlusion.** Large structures include manyNTs, making it difficult to distinguish different strands and causing occlusion in 3D structures. This situation results in an ineffective visualization that can only reveal the surfaceNTs in the foreground.
- C3: Periodic bounding box.** OxDNA MDS uses periodic bounding boxes (a technique involving a simulated unit cell that is regularly repeated throughout the space, allowing a finite system to be artificially modeled as infinite but seemingly splitting the molecules over the boundaries [16]) to emulate a boundless environment. It breaks up strands at the borders of the box, however, and thus prevents analysts from understanding the structure correctly.
- C4: Lack of well-defined formats for analysis.** Most observables from conventional analysis, in particular H-bond occupancy, are only generated on-demand by improvised scripts and lack well-defined formats. Such an approach thus prevents the experts from performing a reliable, standardized,

and reproducible analysis.

C5: Conceptual views lost in dynamics. When a disassembly or stability simulation starts, the structure changes immediately and diverges from its “canonical” shape, making it difficult to perceive for analysts. As a consequence, all the design phase helpers that were created to assist users in understanding the topology of the structure are lost in the now-changed structure in the trajectory.

Based on this analysis of goals, tasks, and challenges, we concluded that an effective solution should not rely solely on aggregating data into synthetic statistics, which often obscures critical structural information. Instead, our approach centers on displaying both structural and dynamic information from an MDS trajectory concurrently across multiple, synchronized abstract views. By coupling these views with visual highlighting, we can provide the derived information, usually communicated with statistics plots, directly in the structural context. We believe that seeing both types of information together enables domain experts to find actionable improvements for their designs. In the following section, we describe the design of our framework built on these prerequisites.

6.3 Design of the SynopFrame

To help users with the mentioned tasks and challenges, we developed our SynopFrame approach. Below, we first explain our efforts in exploring the entire possible visualization space in the context of DNA-nano design that ultimately led to a number of design decisions for our framework. Next, we describe—in an implementation-agnostic way—a sequence of the transformations that realize the representations in use and address the challenges we just described in section 6.2. We also report how we arrange and connect the various representations together, how we color code theNTs by their H-bond status, and how we add the highlighter that links all parts. To ensure reproducibility and extensibility, we discuss in B.4 the Houdini-specific implementation details and in B.5 the transitions between

different representations.

SynopSpace: The visualization space

To discuss the entire space of possible data mappings we begin by analyzing well-established visual representations. One of these is the depiction of positions and orientations of eachNT output by the simulator (*All-NT* as in Figure 6.1a or Figure 6.2a).¹ In the simple example in Figure 6.2a we show a 3D triangle structure in the spatially final, “relaxed” configuration, while Figure 6.2b shows the same structure but in its “designed” configuration (converted from its caDNAno design format to oxDNA format). In this mapping, eachNT is drawn as three parts, a sphere at the *backbone* site, an ellipsoid at the *base* site, and a cylinder that connects the sphere and the ellipsoid. The backbone spheres of theNTs from the same strand are then connected by cones. This representation is used in both oxView [83] and Adenita [23] and shows whether an H-bond is formed—through an examination of the distance between a pair ofNTs, and their respective types. Structures were traditionally designed in such a configuration because it allowed the experts to focus on the matching between scaffold and staples. Yet, for such a design theNT detail shown in Figure 6.2b is not needed and may even be detrimental for some tasks. So the “Single Strands” representation from Miao *et al.*’s work [68] can be used to reduce the detail, while still showing the actual spatial positions of each strand in the dynamic context (*Snake*—once the strands start moving, they look very much like snakes in this representation—as in Figure 6.1b and Figure 6.2c). A related representation as it is used in caDNAno [27] further simplifies the double helices to parallel straight line segments shown in a 2D abstract representation (*Schematic2D*—as in Figure 6.2d). It no longer shows any 3D spatial information but visualizes the pairing between the scaffold and staples as well as the linking between different staple segments. Finally, we may want to compare and see information about the scaffold and the staples by arranging them

¹In the brackets we give our abbreviations for each representation.

independently next to each other, sorted by length (*Heatbar*—as in Figure 6.2e; by Miao *et al.* [68]).

What is less obvious about these representations is that they can be thought of in terms of three orthogonal aspects or axes. The first of these axes is **Granularity**, which describes the primary intact physical *individual* that we are dealing with. For example, in All-NT (Figure 6.2a–b), eachNT is depicted individually as sphere-cylinder-ellipsoid so that the primary element is the *NT*. In Snake and Heatbar (Figure 6.2c and e), in contrast, all theNTs on the same strand are merged into a line entity and the physical individual now corresponds to the *Strand*. In Schematic2D (Figure 6.2d), then, each pair of the parallel straight lines (each individual in focus) represents a continuous double *Helix*. The three levels on the granularity axis we can extract from the discussed representations are thus *NT*, *Helix*, and *Strand*, with a decreasing amount of granularity. In particular, we consider *Helix* to have a higher amount of granularity (finer) than *Strand* because a strand usually spans multiple helices. Even though we did not encounter more granularity levels in the above-discussed representations, we can still reason that there should be another one, *Assembly*, which treats the whole design as an intact individual. This *Assembly* naturally is a coarser level than *Strand*. Similarly, we can see that *Atom* is another level with a higher granularity than *NT*. We show the *Granularity* axis as the horizontal blue coordinate direction in our abstraction space in Figure 6.3. In the figure, we label the levels at the bottom to avoid clutter and occlusion. We also do not explicitly indicate *Atom* as a dedicated level because the DNA-nano domain rarely simulates the dynamics of assemblies at that granularity.

Next to the granularity of the model, another way of looking at the abstraction of the depiction is to use different graphical primitives or elements. For example, in Heatbar and Schematic2D, (with the exception of the flying linkage lines in the latter) straight lines are used, to which we refer as *Bars*. If we allow the bars to bend and follow flexible paths, then we call the primitives *Snakes*. Finally, the

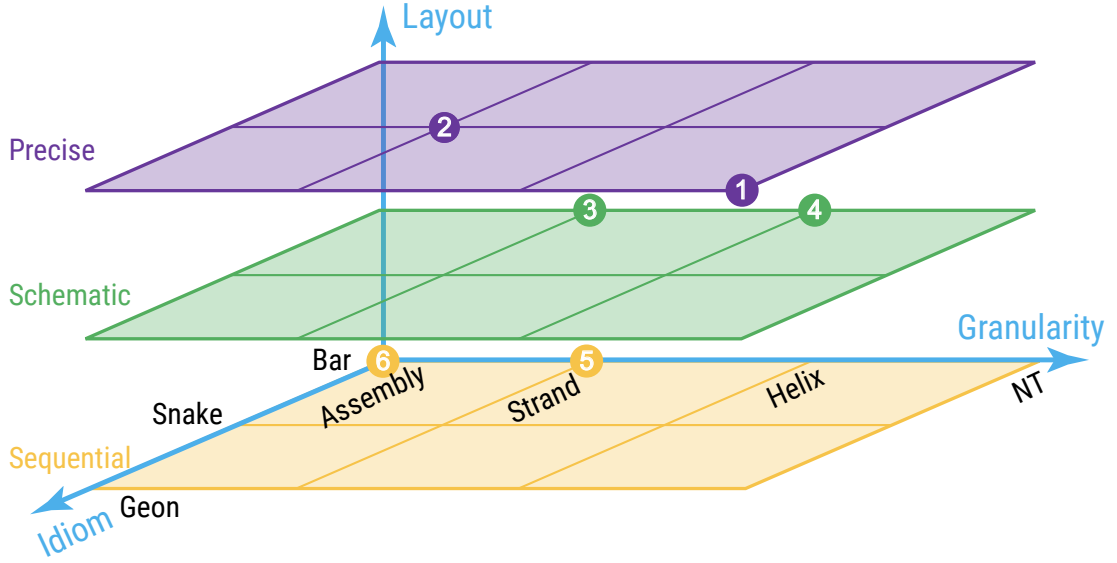


Figure 6.3: Schematic view of the **SynopSpace** with its three axes *granularity*, *idiom*, and *layout*. Among the $4 \times 3 \times 3 = 36$ possible **SynopPoints** we highlighted those we discuss here with dots and numbers 1–6 in the order of (a)–(f) of Figure 6.1 and Figure 6.4. We discuss the others at the end of 6.3.

primitives used in the All-NT representations are spheres, ellipsoids, cylinders, and cones. These simple geometric forms are examples of what neuroscientist Irving Biederman called *Geons* [13]—geometrical ions, which is “a modest set of generalized-cone components.” We call the resulting abstraction direction the visual **Idiom** axis, in which the shape complexity that an idiom is capable of representing grows from *Bars* to *Snakes*, to *Geons*. Beyond the *Geons*, in fact, we could argue that another level that is able to show even more complex structures is *Surfaces* that are commonly used, for instance, in protein rendering [90]. We show the idiom axis in blue in Figure 6.3 and label the levels at the upper-left of the figure. We do not explicitly indicate *Surface* in Figure 6.3 because it is not commonly used in the DNA-nano domain.

The third axis in our space is **Layout**: the arrangement of the visual idiom of a granularity in the scene. In All-NT, eachNT’s sphere, cylinder, and ellipsoid are placed at their *precise* 3D positions as calculated based on the output by the simulator. In Snake, even though the details of theNTs are abstracted out, the snake still passes the *precise* center of mass (CMS) positions of eachNT. In Schematic2D, each helix is no longer twisted but straight. So the bars no longer

represent the precise positions. As the semantics of the helix and the relative positions between the helices are maintained, we call it a *schematic* layout. In Heatbar, the NTs are *sequentially* arranged along a bar and the bars are again *sequentially* arranged in screen space. So the three levels of the *Layout*, with increasing spatial faithfulness to the simulation, are *Sequential*, *Schematic*, and *Precise*, which we show as the vertical blue axis in Figure 6.3 and label the levels at the right of each layout plane, with the corresponding color.

The three axes are independent from each other, so we can sort them in increasing amounts of detail and arrange them in an abstraction space [116, 117] we call SynopSpace (Figure 6.3)—owing to its capability of showing various levels of synopsis of an MDS trajectory—and points within it SynopPoints. As discussed by Viola et al. [117, 116] and demonstrated in various examples in the past (section 3.3), such abstraction spaces allow us to understand aspects of existing visual mappings of our data. For example, the existing *All-NT* representation is at point *(Precise, NT, Geon)* within SynopSpace (i. e., point (1) in Figure 6.3), so we can understand that it is far from the origin of the space and thus the information density it encodes is high and it is likely useful for tasks that require detailed analysis.

But the established representations do not cover all points within our space, and we can also use it to discover alternative representations that may be useful for particular tasks. So let us examine some positions that are not yet covered. A possible representation at the origin of the space at *(Sequential, Assembly, Bar)*, for instance, triggers us to think about how a bar could be used to show the entire structure assembly in a sequential manner. This thought brings to mind the metaphor of a progress bar (such as for showing the progress of copying a file). We could use a progress bar representation, e. g., to show the number of H-bonds that are formed in the entire structure as it dynamically assembles from the scaffold and staples (T1; we discuss this representation further in section 6.3). Another example of reasoning in the space allows us to address the dilemma that,

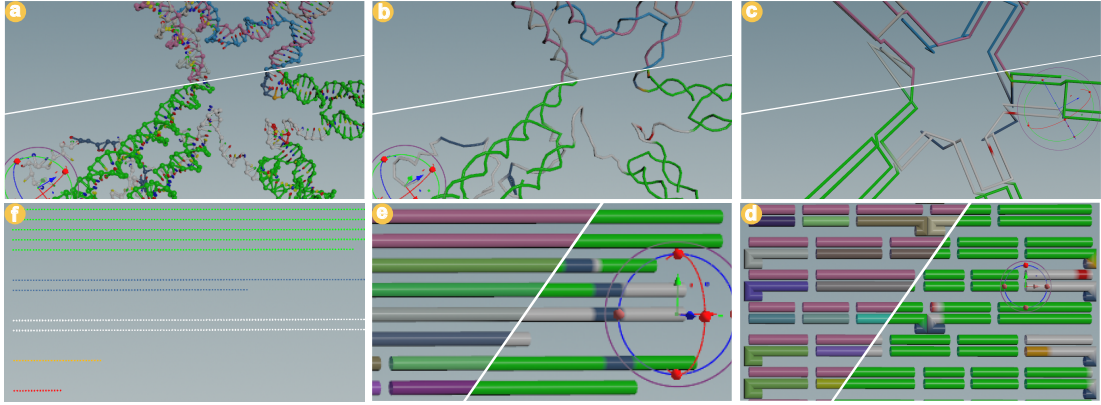


Figure 6.4: Zoomed-in views (arranged in a clockwise fashion) for the purple dashed regions in the icosahedron design from Figure 6.1. Apart from (f), we show all views in two color schemes—strand identity (by color) in the upper-left and MD progress (five colors) in the bottom-right. In the latter, the not-formed H-bonds are gray and the singletonNTs (i. e., designed not to be in an H-bond) are dark blue. (a) displays theNT details, which shows theNT types of the not-formed H-bonds. (b) removes the backbone and the base details and only shows the strand. (c) shows the designed geometry. Even though the highlighter-indicatedNT hangs in the air in (a) and (b), in (c) it becomes clear where thisNT should be attached if assembled correctly. (d) shows that thisNT is at the end of the helix to which it belongs, while (e) shows it is at the end of a staple strand. (f) shows the progress bar.

when using a caDNAno representation (Figure 6.2d), the detail on helices (T3 and T4) and the information about linkage between segments on the strands (T5) come at the cost of a lot of occlusion and clutter (C2). To improve the situation we examine the caDNAno representation closely in SynopSpace. First, it is easy to tell that its layout is a *schematic*. Second, it mostly uses straight *bars* even though there are also some curves that connect segments across bars, yet those are used to indicate the connections rather than to encode actualNTs. But, third, what is its granularity? The straight bars represent the helices well, while linking curves provide information about the strands. So the granularity is coarser than the helix level but finer than the strand level. So, if we were to assign a point for the caDNAno representation, it would be in-between point (3) and point (4) in Figure 6.3. Based on this classification we can now try to address the mentioned issues. For example, we can derive two separate representations, one solely dedicated to helices and the other solely to strands. Creating the first is easy, we can remove the linkages between the segments from the caDNAno

representation and thus move it to point (4). The latter is more difficult. We need to remove the linkages while keeping the strand intact. Our solution is to shorten those linkages and place the helices in 3D rather than 2D (as if the curved links are shrunk to drag the helices on both ends together to fold them). We thus changed the original caDNAno representation to now be located at point (3). With the new mappings, which we name *schematic2D* and *schematic3D* and discuss further in section 6.3, we demonstrated that we can use SynopSpace as a mental tool to design proper mappings to tackle the various challenges and fulfill the tasks. Later in section 6.3 we also show that SynopSpace helps us to arrange the different representations in the linked views.

We do not describe all possible SynopPoints, rather focus on few that we found useful in specific situations. For example, (*Precise, Assembly, Bar*) may appear bizarre yet if we need to show multiple different designs in an MDS system then it may be useful to place a *Bar* for each design (*Assembly*) at its *Precise* location. Similarly, the point (*Schematic, Assembly, Bar*) makes sense if we place the *Bars* at predefined locations. A similar scenario can also make use of another set of SynopPoints, (**, Assembly, Geon*) by using simple geometries to abstract the whole assembly rather than justNT and then showing them either *Sequentially*, *Schematically*, or *Precisely*. We can also potentially extend the space, for instance, by allowing the granularity to have one more level, *Atom*, to allow us to expand our work to all-atom simulators. So the use of new SynopPoints depends on the given application and whether it needs respective representations or not. Once a scenario expands or changes, SynopSpace can be used as a mental tool for designing the proper representations.

Realizing each representation

Having established the SynopSpace abstraction and identified the six key representations within it, we summarize in Table 6.1 their SynopPoint index, coordinate tuple, example figures, and the specific tasks and challenges that it addresses,

Table 6.1: Summary of the six key representations in SynopSpace, with their names, SynopPoints, coordinates in SynopSpace, examples, tasks and/or challenges, and description. They are described in detail in B.1.

Name	Point	Coordinate	Examples	Tasks/challenges	Description
All-NT	1	(Precise, NT, Geon)	Figure 6.1(a), 6.4(a)	T3-T4, T6	Displays every nucleotide at precise (T6) position and orientation, enabling detailed inspection of individual H-bond interactions (T3) and subtle conformational shifts (T4)
Snake	2	(Precise, Strand, Snake)	Figure 6.1(b), 6.4(b)	T2-T4, C1-C2	Shows each strand as a smooth 3D curve without nucleotide detail (C1), reducing clutter and occlusion (C2) to contextualize strand-pairing events (T3) in their broader spatial context (T2, T4).
Schematic3D	3	(Schematic, Strand, Bar)	Figure 6.1(c), 6.4(c)	T2-T4, C1-C3, C5	Presents the idealized pre-simulation geometry as straight, parallel bars for each strand (C1), minimizing clutter (C2), avoiding bounding-box artifacts (C3), and leveraging users' mental models (C5), while the user is observing the dynamic H-bond events encoded in color overlaid on the static geometry (T2-T4).
Schematic2D	4	(Schematic, Helix, Bar)	Figure 6.1(d), 6.4(d)	T2-T3, C2	Lays out double helices in a simplified 2D schematic without crossover links, eliminating occlusion (C2) so users can monitor helix pairing and unpairing events (T2-T3) at various zoom levels.
Heatbar	5	(Sequential, Strand, Bar)	Figure 6.1(e), 6.4(e)	T1, C2	Arranges each strand as a colored bar chart, removing 3D occlusion (C2) to provide a quick overview (T1) of strand-level dynamics and highlight strands with unusual behavior.
Progress bar	6	(Sequential, Assembly, Bar)	Figure 6.1(f), 6.4(f)	T1, C1	Abstracts per-nucleotide H-bond status into a linear progress-bar view over time, condensing high-frequency (C1) changes into an overview (T1) that quickly reveals key simulation phases.

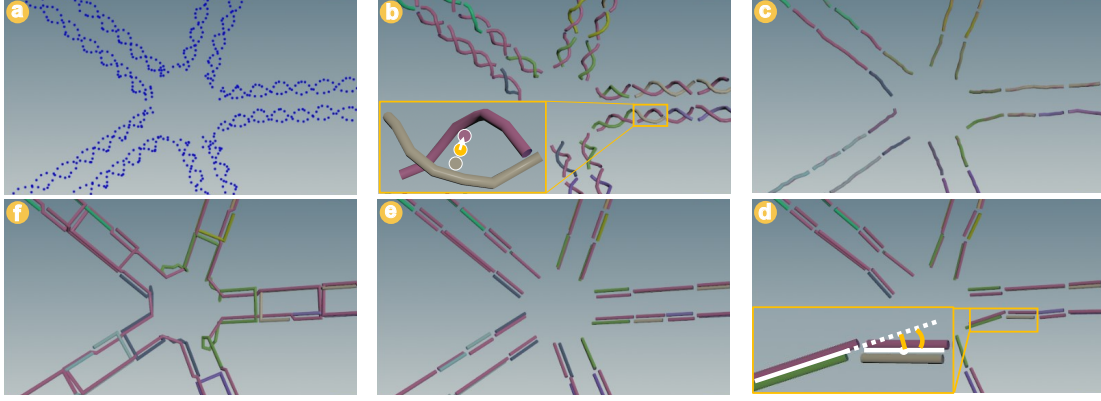


Figure 6.5: Transformations in the Schematic3D algorithm (arranged in a clockwise fashion): (a) Input CMS positions. (b) Connected polylines (color showing strand identity). Notice the broken double helices (result of **Step 1**/Algorithm 1). The enlarged inset shows the attributes CMS (purple dot), CMS of its pair (brown dot), CMS of the whole double helix (orange dot), and direction vector from the double helix’s CMS to the purple polyline’s own CMS (white arrow). (c) Smoothed double helices (one line per double helix)—result of **Step 3**. (d) Straightened and shifted double helices (two lines per double helix)—result of **Step 4**. The enlarged inset shows the distance (straight orange line) and angle (orange curve) that we threshold. (e) Straightened double helices group (those with the same *schematic2D_row* value)—result of **Step 5**. (f) Connected strands—result of **Step 6**.

along with a concise description (for more details see B.1).

Among the six SynopPoints, Schematic3D is key to bridging the 3D and 2D representations. To realize it, we use the topology and CMS positions (Figure 6.5a) of allNTs of one frame (usually the designed configuration) as input, followed by a series of transformations: **Step 1**: We construct polyline primitives differently than for *Snake*, so that only theNTs that form a continuous double helix end up in the same polyline. The singletonNTs that do not form H-bonds are not in any polyline. We use Algorithm 1 (B.10) to exhaustively check the conditions that could potentially break a continuous helix and create the primitives for the continuous ones. We show the result in Figure 6.5b. **Step 2**: We prepare the required attributes for each polyline for downstream transformations. We run this algorithm for each polyline primitive using the firstNT to find its pair and then using the pair to find the pair’s polyline ID and save it as an attribute. We also save the following attributes for each polyline: CMS, CMS of its pair, CMS of the whole double helix, and the direction vector from the double helix’s CMS to its

own CMS, see Figure 6.5b for illustrations. **Step 3:** We use a smoothed line to represent each helix (Figure 6.5c). We move eachNT to the center of itself and its pair. The two strands of each helix thus overlap with each other. **Step 4:** We straighten the helix and shift the two polylines in each helix at a user-controllable distance and evenly space theNTs on each polyline at a user-controllable distance (Figure 6.5d). For this purpose, we first perform a linear regression against all the vertices in each polyline. We then use Algorithm 2 (B.10) to shift the two lines in each helix and distribute theNTs on them. Now, we can use the same preprocessing and rendering technique as those in the *Snake* representation to create the final visual representation. Sometimes (e.g., when a relaxed configuration rather than the designed one is used), however, there are *multiple* double helices that are supposed to be along a straight line but are tilted against each other. We thus use the following steps to refine this issue and to prepare the intermediate data for *Schematic2D*. **Step 5:** We create a new attribute (*schematic2D_row*) for each polyline, and shift those double helices with the same *schematic2D_row* so that they locate precisely along one straight line (Algorithm 3, B.10). It works on two user-specified threshold values: one for distance and the other for angle. Those double helices within a certain distance and within a certain tilt against each other will have the same *schematic2D_row* value. For each *schematic2D_row*, if there are multiple double helices, we then extract each helix' CMS into an array and perform a linear regression so that we shift the CMS of all double helices bearing the same *schematic2D_row* to sit on exactly the same straight line (Figure 6.5e). **Step 6:** We delete all polyline primitives and construct the *Snake*. Since allNTs are now at straightened positions our final visual representation consists of straight bars (Figure 6.5f) instead of curved snakes.

The dashboard: Linked views, highlighter, H-bond status

Each of the six representations reveals the structure at an important level of abstraction, solves particular challenges, and fits certain tasks. To address challenges

and tasks we thus link all six via the time axis (Figure 6.1 and the supplemental video) such that, once the time slider moves or is moved, all six change together. In addition, we can optionally link *All-NT* and *Snake* in a structural way, i. e., once the camera parameters change in one (rotate or zoom) the other will follow. We purposefully do not structurally link *Schematic3D* in the same way because, when the structure is no longer fully assembled, the similarity between *Schematic3D* and *Snake* no longer holds. We further support the linking with a synchronized highlighter, similar to the linking and brushing applied by Becker *et al.* [10]. Once, in selection mode, the user clicks on a specificNT in any representation, we highlight the sameNT in all rest except in the *Progress bar* as the latter does not convey any structural information. The strand, a highlightedNT belongs to, can optionally be highlighted as well with an increased radius of the circle that we use to sweep along the strand’s polyline.

On top of the structural views, MDS analysts often check additional abstract representations. A frequently used one is a PCA projection of the entire trajectory data, where each simulation frame becomes a data point in a 3D space with similar simulation frames being mapped to nearby spatial points. This view can be regarded as a projected conformational space plot. We also generate (similar to Poppleton et al. [83]) and link this PCA plot (Figure 6.1g) to further assist the user with navigating the linked views. We link it interactively so that, once the time slider moves, we show the dot for the new frame’s configuration with a bigger radius. Alternatively, once the user clicks a certain dot in the PCA plot, we move the time slider (and hence all the linked views) to that frame. We order all seven views clock-wise according to the abstraction level, with *All-NT* first and *PCA* last. The projected conformational space plot essentially shows a vector with three scalars for each frame. In a similar way, SynopFrame can also be coupled with the scalar versus time plots or the scalar versus another scalar plots showing the statistical metrics along the temporal aspect (more detail in B.9).

In addition to providing experts with this DNA-nano data dashboard, we also

introduce a new way of categorizing H-bond statuses for better comprehensibility. OxDNA’s built-in analysis only detects all H-bonds in each frame and then outputs the twoNT IDs for each H-bond, regardless of whether that H-bond is in the designed configuration (C4). This reporting does not facilitate an effective analysis: mispairing ofNTs may happen and mixing the designed H-bonds with non-designed ones hides the accurate H-bond counts, so potentially important mispairing events may go unnoticed. We thus categorize eachNT in each frame into one of five groups (which forms a new format, which we detail in B.7), indexed from -2 to 2 : (-2) designed to be in an H-bond, but wrongly formed; (-1) designed to be a singletonNT but an H-bond is formed; (0) designed to be in an H-bond but not yet formed; (1) designed to be a singletonNT and is single; and (2) designed to be in an H-bond and correctly formed. We then assign a color to each category (by default green to 2 and red to -2) and use it to color all representations, except for *PCA* which we color by time, by the entire configuration’s energy, or other properties. Except for this coloring according to H-bond statuses, we also allow users to color the six representations according to the strand ID to reveal the routing of the staples. Users can also use a custom scalar value for eachNT in each frame and use it instead for the color encoding such as the H-bond distance, the forces thatNT is bearing, theNT’s speed, etc. In the default color scheme for H-bond status we use red for bonds that are designed to be in an H-bond but are wrongly formed; correcting such cases requires first breaking the bond and then forming the correct one. We use orange when a nucleotide is designed to be a singleton but an H-bond is formed; this case is less severe than the former as it only requires breaking the bond. Gray indicates a designed H-bond that is not yet formed, also an uncritical status. Blue shows a correctly single-nucleotide designed to be a singleton, signaling a correct status without need for change. We use green for correctly formed H-bonds as designed, signaling success. For the projected conformation space plot (*PCA* plot), we use a gradient from red to green based on the timeframe, as simulations typically start with fewer correctly formed H-bonds

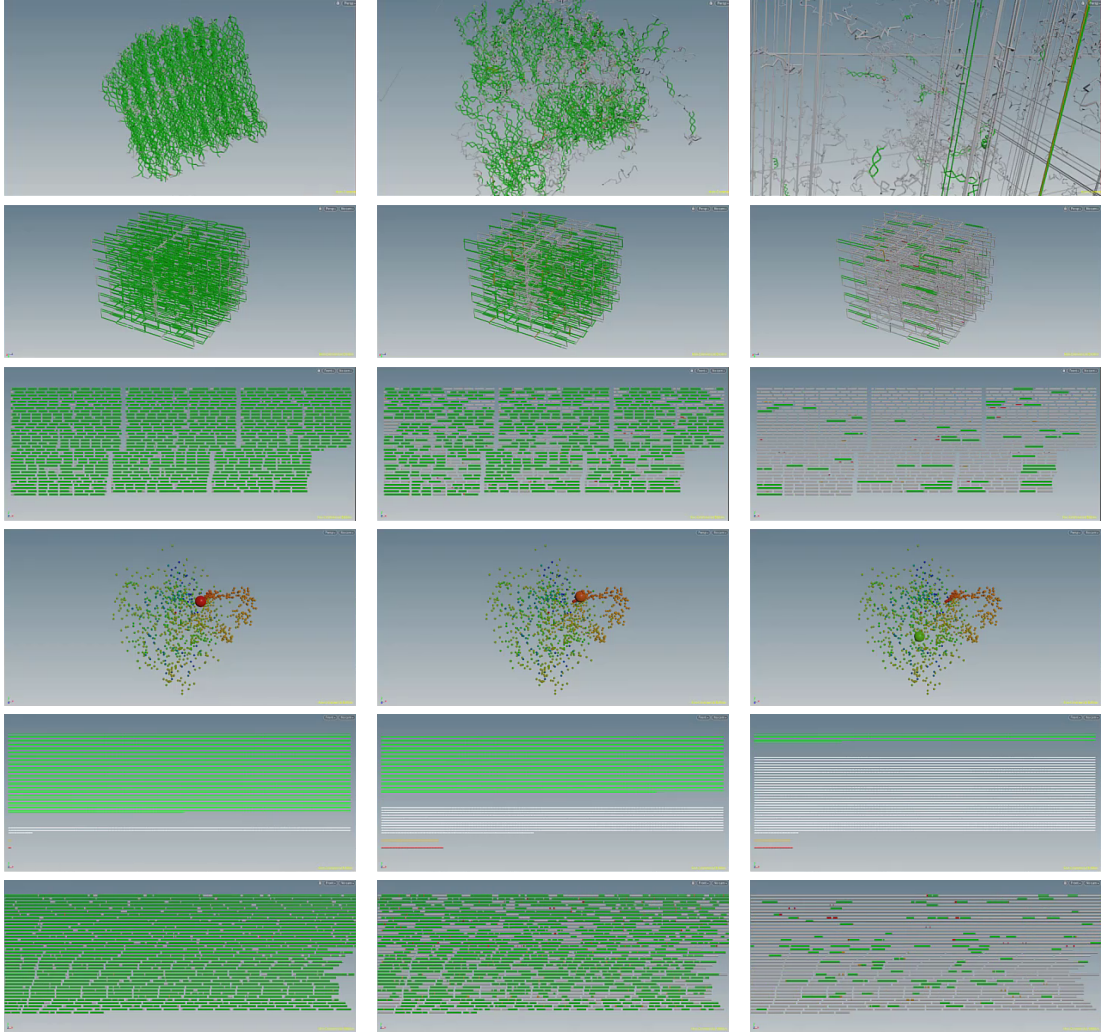


Figure 6.6: Three frames from an animation (from left to right) that showcase the break-up of a structure. From the top: snake, schematic3D, schematic2D, PCA plot, progress bar, and heatbar.

and progress towards more correctly formed H-bonds over time. We represent the identity of the strands by random colors, due to the large number of staple strands often present, which would otherwise lack meaningful differentiation. For the base ellipsoids, our color scheme follows oxView: blue for A, red for T, green for C, and yellow for G. We also allow users to fully control all colormap assignments and adjust as needed.

Dynamic data analysis

Previous approaches to coping with the different DNA-nano representations by relying on abstraction spaces [67, 68] focused on showing different visual represen-

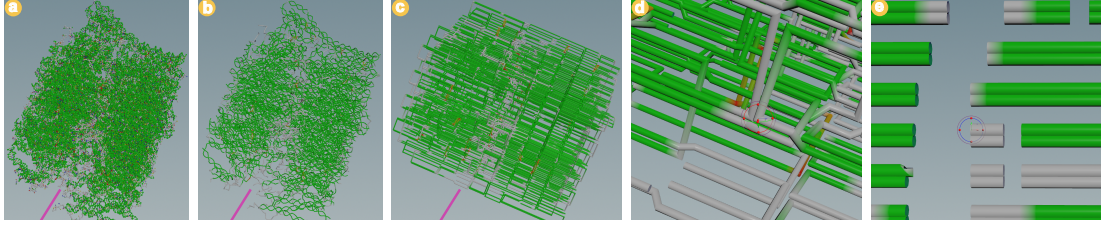


Figure 6.7: Cube case study. All-NT (a), Snake (b), and Schematic3D (c) show that the cube’s middle disassembles first, as if the cube is cut by a knife (see purple line). A synchronized highlighter attached in Schematic3D (d) highlights the cause: the short double helix visible in Schematic2D (e).

tations of static data. In contrast to this past work, our approach of embedding our SynopSpace abstraction space into our SynopFrame dynamic framework—for the first time—allows us to not only traverse different representations of the structures at some point in time but to actually use MDS to simulate the *dynamic* behavior of the structures as they assemble or not (T2–T5). We provide such dynamic analysis largely not in the form of summary views of dynamic behavior (the PCA plot is an exception and we describe another in B.9) but instead as actual animations (see the accompanying videos) in which the experts can observe whether and how a structure forms or not. Summarizing the dynamic behavior of the designs into a single and dedicated ‘dynamics overview’ representation is, in fact, not needed—our H-bond status visualization together with SynopFrame’s playback possibility allows experts to quickly identify interesting time periods (as we demonstrate in Figure 6.6).

The means to show dynamic changes we studied in this paper vary between a progress bar and a full 3D representation with a changing H-bond status. The progress bar is a well known visual encoding of an emergent process completion, which completely abstracts from structure and as such, animates throughout the simulation time how many desired H-bonds have been created and how many are not established. In contrast, on the other end of the spectrum is the animated structure depicted with its highest level of detail, where the bonding is conveyed spatially as well as through color mapping. Our design showcases some of many possible abstractions of animated process visualization that abstract from

particular structural detail. Abstracting granularity allows us, in the absence of fine detail and its motion, to abstract these detailed motions, thus lower motion frequencies become visually more prominent. This way our representation allows viewers to follow the animation at faster simulation playback. The Snake representation used for this purpose can be further abstracted into a static 3D structure, where the dynamics remains in the color-coded animation of the bonding state of nucleotides. To facilitate a clear view of every such nucleic-acid substructure, we further abstracted the 3D representation into 2D or even 1D layouts where all details can be observed and additional details, such as strand length, become visually promoted. Finally, due to the linear nature, strands can be hierarchically grouped and merged together. On each level, the animated color-coding conveys the degree of bonding. Finally, at the most abstract level, all 1D strands are grouped into a single linear structure where still the order of the NT sequence is preserved. One final representation that abstracts from this order leads to the animated progress bar. All these visual encodings represent animated visualizations of an emergent process in time. Either only photometric visual channels, i.e., color mapping, are animated, or gradually geometric spatial animation adds the structural detail. Such detail can be varied throughout the simulation, or even within the simulation, different parts of DNA sequence could, in principle, be encoded by varying level of procedural detail. Our work thus explores the visual abstraction continuum where structural analysis of animated DNA is encoded by animated visual metaphors, which can be, if needed, further complemented by visualizations where the time is encoded in a different way than through animation.

In the following section we showcase an example application case and demonstrate the benefit of the analysis of dynamic DNA-nano data with SynopFrame, specifically the identification of a problematic design issue for a given DNA-nano design.

6.4 DNA-nano MDS exploratory analysis case study

To better illustrate how DNA-nano experts can make use of SynopFrame, we now describe a case study for performing an exploratory analysis for MDS trajectories. This realistic example showcases the overall process that experts can use to understand why it cannot assemble in wet lab experiments (T1, T5).

A cube structure with 16,128NT (Figure 6.7; one scaffold, 238 staples) was designed in caDNAno by a domain expert (a co-author of this paper) and his group. After the in-silico design, the structure appeared to be a robust design that will also self-assemble throughout the experiment. The research team had tried to assemble it in many wet lab experiments, but all attempts had failed, even though an experienced postdoctoral researcher had spent three months and ample resources on the project. We thus performed molecular dynamics simulations to examine the stability of the structure at various temperatures. While animating through the MDS with SynopFrame and looking at its various views, together with the expert we then noticed an interesting phenomenon that occurred at 78°C.

In the *Schematic3D* view, in which—with the animation—the expert could quickly identify (via fast-dragging the handle on the animation’s playbar with the mouse) the time period in which the structure was attempting to assemble, we can immediately perceive a prominent pattern of the H-bond statuses at the beginning of the simulation (Figure 6.7c). The pattern shows that the middle of the cube disassembles first, much like being cut by a knife right in the middle Figure 6.7a–c. By attaching a highlighter to the disassembledNT (see our supplemental video), we can then easily identify the cause of such a pattern from the combination of *Schematic3D* and *Schematic2D*: there are many short double helices (3NTs) aligned at that knife-cutting plane Figure 6.7d–e. This observation can further be mapped back to the original caDNAno design view (supplemental video).

When observing this behavior of the simulation, the expert commented that it “*is very helpful in understanding why the structures did not form. In caDNAno, these mistakes (the short double helices) are not easily spottable.*” In our video

interview with him (see our evaluation below in section 6.5) he also mentioned that *“it could have saved three months of working time and resources of a postdoc experienced in wet labs but who has limited know-how in biophysics. This tool will give lab practitioners who do not have enough biophysics knowledge to understand the details of an MDS and its analysis a faster way to digest what is happening in the simulation and to remove bad designs. And even people who use traditional statistics such as root-mean-square deviation to analyze an MDS trajectory will face problems in finding insights for the case of large structures with long simulation durations. This tool comes right in place for these cases to help the analyst dive into the details.”* We later learned from our collaborating domain expert that the cube’s main designer was surprised by the “knife-cutting” pattern because he had not realized this problem when using caDNAno.

An actionable insight from this analysis for the experts is thus that designers need to fix those short helices to prevent the respective parts from disassembling, which our collaborator then incorporated into his future DNA-nano designs. In addition, the caDNAno developers could improve their software by highlighting short strands to easily solve cases like ours. Ultimately, this aspect of the reported case study demonstrates that our H-bond color scheme allows experts to observe dynamic properties of the dataset using “normal” play-back animations of the MDS data, simply by seeing the characteristics of the H-bond status in different parts of a design. This visual representation of the dynamic characteristics relies entirely on the interactive play-back animations of the normally static views that we described (e.g., Figure 6.6)—it does not require any dedicated visual summary of the dynamic characteristics of the MDS data to be effective.

We describe another case study of a smaller structure in B.8, where we focus on how it converts from one configuration to another (T4) as well as on its potential design flaws.

6.5 Further feedback

We gathered feedback from six expert oxDNA users and developers through a questionnaire. One of them is our previously mentioned close collaborator, who was our main contact in our user-centered design process and who is also a co-author of this paper. We communicated with him via e-mail and video meetings and he had access to a local SynopFrame installation for independent exploration. We interviewed him for qualitative feedback, and he also filled in the questionnaire with Likert-scale questions. In addition, we contacted eleven active oxDNA users who had raised issues in oxDNA’s GitHub repository in the past year and who revealed e-mail addresses on their GitHub profiles. We also personally approached two additional DNA-nano experts. Out of these, the mentioned 5 additional experts answered our questionnaire. Due to their time constraints we did not expect them to install and learn our new interface from the ground up, but instead we provided them with a video description of our system (similar to our supplemental video) and asked them to fill in the same questionnaire as our close collaborator, and we received one anonymous and four signed responses (details about their backgrounds in B.11).

In Figure 6.8 we report the questionnaire responses (see the full questions and the detailed answers in B.11) from all contacted oxDNA users, based on a 5-point Likert scale with 1 meaning *strongly disagree/very useless/very ineffective* and 5 meaning *strongly agree/very useful/very effective*. Generally, the experts found the linked views, the connection with the PCA, the H-bond coloring, and theNT highlighter to be effective for analyzing DNA-nano MDS trajectories. The transitions between different representations, however, play a less important role in the case studies, and may thus have received lower ratings from the experts. In addition, we purposefully separated the transitions from the animation of the molecular dynamics because, otherwise, viewers may confuse both types of animation. The linked views and the highlighter, in contrast, already seem to be sufficient for the experts to understand the relationships between the

representations, so the effectiveness of the transitions may be shadowed by the other features in the tested scenario. The 2-rating for “Understand” comes from an oxView developer who prefers much more the conventional statistics-based analysis approach and who tried our tool without training, he may have missed already implemented functionality. We thus conclude that our tool requires more training to be fully effective.

The concerns that the evaluators raised are twofold. First, the accessibility of the tool is reduced if it resides in Houdini and requires careful preparation of the input data in specific formats. So, after our current proof-of-concept, the domain users are eager to get access to a more accessible tool with the functionalities of SynopFrame, e.g., as a web-based tool. Second, even though experts appreciated the structural abstraction, they still would like to see an integration of more conventional statistics. We aim to realize both goals together in our future development.

6.6 Limitations and future development

While multiple views at different scales and abstractions are necessary, showing all views at once indeed can overwhelm a viewer. A future direction is to adaptively show the appropriate abstraction level based on the MDS phenomena. The implementation in Houdini is more of a prototype for proof-of-concept, which is fast to develop, but installing the whole Houdini software takes unnecessary disk usage, and the huge amount of widgets in Houdini are distractors for the user. A dedicated development with technologies such as WebGPU could be followed after the proof-of-concept to solve these problems. We also think that it would dramatically increase the efficiency of the whole workflow if it was possible to directly modify the design upon identifying the problematic regions, followed by feeding the updated design back to the oxDNA simulation. Features for the comparative analysis of multiple trajectories could also be helpful. In addition, the ability to locally and schematically animate the simplified static geometries

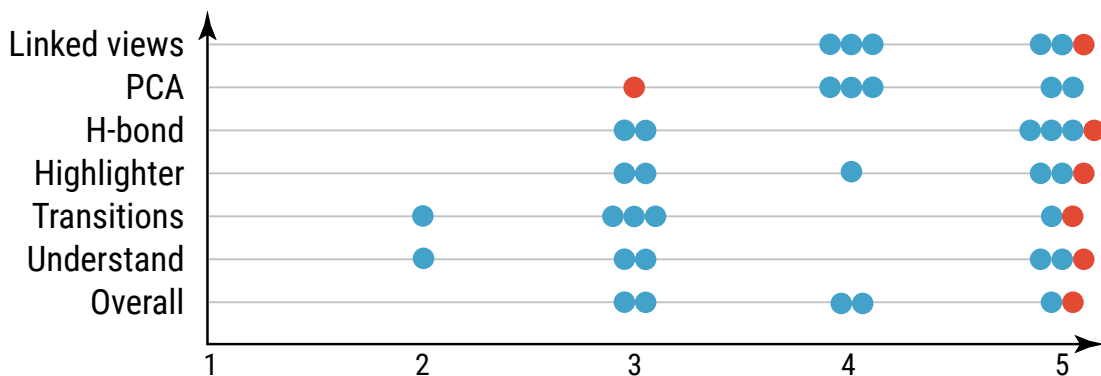


Figure 6.8: User feedback for the effectiveness of **SynopFrame**. Our collaborator’s response is colored in red, while the external evaluators’ are colored in blue. We provide the full questions (*y*-axis) in B.11.

to convey certain dynamic properties could further unleash the power of the schematic representations we developed. It is also worth mentioning that our whole design study could be expanded to other domains—e.g., in protein MDS, one could change the granularity axis of our SynopSpace to fit the hierarchical multiscale nature of protein structure and solve similar challenges there.

6.7 Discussion

The extensive research on DNA-nano structures for various applications has led to the use of MDS as a cost-effective method for identifying poor designs and understanding the dynamic nature of these structures. Nonetheless, analyzing and comprehending an MDS trajectory continues to present significant challenges. We systematically analyzed the tasks, challenges, and established representations from the domain, and proposed a visualization abstraction space as a foundation, based on which we developed a proof-of-concept visual analysis tool that enables experts to see the connections between the different representations and to better explore and understand MDS trajectories, i.e., the dynamic aspects of DNA origami assemblies. While our tool is not yet integrated into the toolchain of the experts, we still demonstrate that the approach works in principle and with some more development an integration is possible.

In a broader sense, as MDS systems increase in size and approach the mesoscale,

such as the whole cell simulation mentioned by Stevens *et al.* [106], a critical need arises for the abstraction of the structure for visualizing the resulting trajectory. With the novel approach we developed, SynopFrame, domain experts are now able to identify design flaws in a DNA cube design, such as the one that troubled our collaborators for months and cost them vast amounts of experiment resources. Users can also identify the transition period for the conformational change in an RNA tile design to understand how theNTs' H-bonds change during that period. SynopFrame goes beyond Miao *et al.*'s work [67] by extending the abstraction space to the *idiom* axis, encoding dynamic spatial data from the simulation in the more detailed views, connecting the static design and its dynamics through the novel Schematic3D view to lower the cognitive load, as well as taking advantage of color-coding the H-bond status in the more abstract views to allow experts to identify problems with their designs through traditional playback or the use of a time slider (for a detailed comparison see B.12). As such our approach is best suited for the post-design phase, while Miao *et al.*'s work supports experts in navigating the different design spaces that are important at design time. The organization of various data mappings/representations in a holistic space, reasoning within the space, and linking 3D and abstract views in our work collectively enable a new paradigm of MDS trajectory analysis. Rather than drawing insights only from statistics compiled from the trajectory, this new approach allows experts to gain insights by directly visualizing the trajectory. We thus extend the previous design-only solution to a comprehensive MDS analysis scenario. This analysis is based on all available information in the data, avoiding any bias toward certain statistical approaches that compress information in certain aspects. In this context, SynopFrame facilitates a shift in thinking about the analysis of MDS and the design of new visualization techniques in the emerging mesoscale era.

Chapter 7

Conclusions and Outlook

In the final chapter, I offer a mapping of my projects to the non-optimizable gap, followed by a synthesis of the key insights, challenges, and conceptual advances that emerged throughout my doctoral research. I reflect on the lessons learned from integrating visualization and computation, consider the broader impact of these approaches, and outline potential directions for future work.

Projects mapped to the non-optimizable gap at a glance.

- **DiffFit:** robust pose sampling, negative-space-aware loss, and GPU-based differentiable search with clustering/ranking \rightarrow removes manual coarse placement (*narrow basins & rough paths*) and shifts expert effort to selecting among high-quality candidates (*multiple good minima*).
- **ProteinCraft:** coordinated multivariate/3D views (attributes, residue-residue interactions), guided realignment and local jittering, and iterative sequence redesign with contact/PAE evidence \rightarrow steers computation through rough landscapes toward promising basins when no single global objective exists.
- **SynopFrame:** a *visualization space* of synchronized, gradually abstracted views across granularity, idiom, and layout axes for time-dependent simulations \rightarrow supports open-ended reasoning (diagnosing assembly failures, tracing conformational switching) without a single scalar objective.

7.1 Reflections and Lessons Learned: Discover the Non-optimizable Gap Through Visualization

As my research progressed, the interplay between visualization, computation, and automation became a recurring theme. The journey of ProteinCraft, in particular, crystallized this lesson. Through direct experimentation with large-scale AI-driven protein binder design workflows, I observed that iterative, visually guided refinement can yield dramatic improvements over traditional “brute force” approaches. For instance, in designing binders to the SARS-CoV-2 receptor binding domain—a moderately challenging target [128]—I observed a more than 100-fold increase in downstream success rate by strategically selecting and evolving promising backbones. Specifically, by first filtering for designs with interchain PAE < 20 , then iteratively generating and testing sequences using these “near-miss” backbones, I was able to boost the pass rate from just 0.19% to nearly 20% in subsequent rounds. This process uncovered backbones with progressively better PAE values, eventually converging on designs that were independently verified by AlphaFold3 [1], and could be further optimized with each iteration.

This broader realization, however, was first sparked by a very specific question that arose during my attempt to reproduce a snake toxin binder design [115]. While working with a set of backbone structures that appeared—by all visual and structural metrics—to be very close to the published, experimentally validated backbone, I found that almost none of my reproduced designs could pass the established computational filters. Curiously, when I used the exact published backbone, the pass rate was high. This puzzling observation prompted deeper investigation: why could structures so close in geometry lead to such dramatically different design outcomes?

Visualization proved essential in formulating and pursuing this question. Only by overlaying and visually comparing the backbones could I confirm that my candidates were indeed extremely similar to the published reference. This question led me to reconsider and modify the workflow: if the final predicted structure

from AF2ig [11] looked correct, why not feed it back for sequence redesign? When I adopted this approach, I discovered a dramatic increase in the downstream pass rate. Here, visualization was not just a tool for validation, but a catalyst for insight and innovation, directly inspiring a more successful iterative design strategy.

What became increasingly evident is that even slight perturbations of near-successful backbones often resulted in dramatic drops in performance, indicating that many designs classified as failures may in fact be very close to success, but remain overlooked by purely automated filters. This insight echoes lessons learned with DiffFit [61]: in both cases, user interaction and visualization enable researchers to use a kind of “informed navigation” through a complex search space, guiding computational resources toward the most promising regions.

Critically, these findings highlight a deeper conceptual takeaway: visualization and automation must be co-designed, each component compensating for the other’s limitations. In DiffFit, we saw how visual inspection and manual adjustment could be replaced or augmented by GPU-accelerated optimization. Yet in the iterative protein binder workflow, it is precisely the visualization layer that allows experts to identify and “rescue” sub-optimal but promising candidates, focusing automation where it is most effective. This conceptual lesson leads to a practical answer for an often-posed question in visualization research: What should we visualize, and what should we automate? The answer, I found, is dynamic and context-dependent—but in the era of large datasets and differentiable models, visualization is most effective where the computational gradient vanishes: at those non-differentiable decision points, where algorithmic optimization cannot proceed further, but human intuition and exploratory analysis can open new paths.

7.2 Generalization and Human-in-the-Loop Design: The Broader Impact of DiffFit

Another significant insight that emerged from this dissertation is the versatility of the DiffFit framework beyond its initial application in macromolecular structure fitting. At its core, DiffFit leverages differentiable optimization and modular loss design to facilitate automated alignment and registration between complex data objects. This architecture is not restricted to biomolecular structures; rather, it readily extends to a broad class of registration challenges, such as point cloud alignment in computer vision, multi-modal data integration, and other scientific or engineering domains where matching disparate representations is required [42, 33, 4].

A key feature of DiffFit is its use of massive parallelism: by launching a large number of initializations on the GPU, the system explores multiple starting points simultaneously, thereby increasing the likelihood of finding a robust and globally optimal solution. After parallel optimization, the best result is automatically selected according to the predefined loss function or other final evaluation metrics. This approach not only accelerates the registration process but also improves reliability, mitigating the risk of suboptimal solutions that can arise from poor initialization. The essential requirement for applying DiffFit to new problems is the construction of a differentiable computational pipeline and a thoughtfully crafted loss function that accurately reflects the domain-specific objectives. With these components in place, the same automated and scalable optimization approach can be used to solve a variety of alignment and registration tasks, making DiffFit a flexible template for many cross-disciplinary challenges.

Crucially, even as DiffFit shifts much of the registration workload onto automated optimization, it retains an essential role for human expertise through a final visual check. This design ensures that, regardless of the problem domain or the sophistication of the underlying algorithms, expert users remain empowered to validate, interpret, and refine the results. In doing so, DiffFit exemplifies

the broader philosophy of this dissertation: that the most effective systems are those that balance computational power with human intuition, allowing each to operate where it is most effective. This principle of human-AI teaming design not only safeguards the integrity of the results but also fosters greater trust and understanding, qualities that are critical as automated systems become more prevalent in scientific research [72].

7.3 Differentiability in Visualization: The Perspective of SynopSpace

A unifying thread across my work is the relationship between what is “differentiable”—what can be directly optimized by algorithms—and what is not, requiring human insight. Traditionally, differentiability has drawn a line between automated and manual steps in computational science. However, through the development of frameworks like SynopFrame and its SynopSpace abstraction, I began to consider a new perspective: can visualization itself become a differentiable space?

SynopSpace, by organizing molecular data and simulation trajectories into a continuous, multi-dimensional abstraction space, hints at this possibility. Here, users fluidly traverse levels of detail and representation, dynamically linking structural, temporal, and abstracted views. Looking ahead, I envision visualization spaces where not only humans, but algorithms (AI agents), can “see” and act—optimizing visual representations, recommending informative views, or even learning from user interactions in real time. In such a co-differentiable environment, the boundary between computational optimization and human exploration becomes increasingly seamless, enabling a continuous loop between visual reasoning and algorithmic search.

This perspective opens an exciting frontier. In this view, visualization is not merely about exposing data to human users, but also about creating a space accessible to machines, enabling both differentiable optimization and human exploration to interact and co-evolve within the same environment. This outlook resonates

with emerging protocols at the AI agent frontier,¹ where models are given access to shared contextual spaces and interfaces, allowing them to interpret, reason, and act alongside human collaborators. As visualization environments become increasingly expressive and structured, they can serve not only as interactive and interpretive tools for people but also as differentiable landscapes for autonomous agents, supporting joint workflows in which both humans and machines leverage a common visual context, thereby enabling new modes of scientific discovery.

7.4 Toward Standardization and Benchmarking in Visualization Research

Another key reflection is the transformative role that standardization and benchmarking have played in other data-driven disciplines—most notably, computer vision. The explosive progress in computer vision over the past decade was not just a result of better hardware or algorithms, but of widely adopted datasets, clear evaluation metrics, and structured challenges (such as the ImageNet competition) that accelerated community-driven innovation [25].

By contrast, visualization research, especially in scientific and interactive analytics, remains largely fragmented. Evaluations are often qualitative or context-specific, making it difficult to compare methods or track field-wide progress. To bring visualization research to the next level, I believe our community needs to learn from the successes of computer vision by:

- Curating open, representative datasets that reflect both real-world and broadly applicable visualization tasks;
- Defining clear, multi-faceted evaluation metrics—capturing not only performance, but insight generation, hypothesis refinement, and even adaptability to user intent;

¹anthropic.com/news/model-context-protocol

- Establishing reproducible benchmarks and shared challenges that can drive healthy competition and convergence toward best practices.

Such an ecosystem will not only accelerate research and adoption, but will also make it possible to objectively measure the progress of visualization as a scientific discipline, further amplifying its impact on data-driven fields like computational biology.

In my own research, I have made standardization and benchmarking central to the design and dissemination of my systems. DiffFit introduced a fully reproducible pipeline for fitting atomic structures to cryo-EM maps, with public release of all test data, parameter settings, detailed logs, and comprehensive documentation. Notably, DiffFit received a Graphics Reproducibility Stamp², recognizing its commitment to transparency and reproducibility. Its direct integration with ChimeraX further enables easy adoption and independent reproduction of results within a widely used molecular visualization platform. By leveraging established EMDB datasets and providing automated metric calculations, DiffFit supports rigorous, side-by-side comparison of fitting algorithms.

With ProteinCraft, I took a complementary approach by enabling systematic comparison against previously reported case studies in AI-driven protein design and interaction analysis. ProteinCraft’s support for standardized data formats and published benchmarks makes it possible to directly replicate and extend existing visual analytics workflows, facilitating objective assessment of visualization effectiveness across diverse protein design scenarios.

Across DiffFit, ProteinCraft, and SynopFrame, all code, datasets, and documentation have been released as open source. This commitment not only promotes transparency and reproducibility, but also lowers the barrier for other researchers to adopt, evaluate, and extend these tools—contributing to a more standardized and benchmarked landscape for scientific visualization.

²replicabilitystamp.org/index.html#https-github-com-nanovis-diff-fit

7.5 Summary

In this dissertation, I have demonstrated that integrative visual analytics is a key enabler in bridging the “non-optimizable gap” in modern macromolecular science. By building systems that combine large-scale, automated computation with interactive, expert-driven visualization, I have shown that it is possible to dramatically improve both the efficiency and the scientific value of complex workflows.

- **DiffFit** automates the critical optimization steps in structure fitting, reducing manual overhead but preserving the expert’s ability to validate and adjust as needed. It removes manual coarse placement (*narrow basins & rough paths*) and shifts expert effort to selecting among high-quality candidates (*multiple good minima*)
- **ProteinCraft** unlocks scalable, multi-modal exploration of protein structures and interactions, empowering users to not just filter large datasets but actively guide the design process, resulting in concrete improvements in in-silico success rates for challenging targets. It steers computation through rough landscapes toward promising basins when no single global objective exists.
- **SynopFrame** facilitates multi-scale analysis of DNA nanotechnology simulations, revealing design flaws and dynamic phenomena that were previously inaccessible to domain experts. supports open-ended reasoning (diagnosing assembly failures, tracing conformational switching) without a single scalar objective.

Together, these systems demonstrate a simple principle: *automate where “gradients” exist; visualize where they vanish*, converting computational scale into scientific understanding. Across these efforts, the common thread is clear: human intuition and computational power are most impactful when brought together through thoughtfully designed visualization systems.

7.6 Conclusion and Outlook

As the biological sciences move further into the “AI era,” the challenge is no longer just generating massive amounts of data or deploying ever-larger computational models—it is in making sense of this information, and in transforming potential into actual scientific discovery. The case studies and results that I presented here show that visual analytics is not an optional add-on, but an essential interface for navigating complex, high-dimensional design spaces. By combining the strengths of automation (speed, scale, reproducibility) with the strengths of human reasoning (interpretation, adaptation, creative insight), we can traverse scientific landscapes that would otherwise remain inaccessible.

Looking forward, I envision systems where visualization and automation are ever more deeply intertwined: with AI models that learn from user-driven visual exploration, and visualization tools that adapt dynamically to guide automated search. Lowering the granularity of analysis, incorporating real-time feedback, and broadening these approaches to new domains—all present exciting opportunities for future research. Above all, my work supports a central idea: In the age of big data and AI, visualization remains the indispensable bridge between algorithmic progress and scientific understanding. By designing systems that empower human expertise at the points where computation cannot reach, we can close the non-optimizable gap and further enable meaningful discovery in molecular science.

REFERENCES

- [1] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C.-C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Žemgulytė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Žídek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis, and J. M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. doi: [10/gttgxm](https://doi.org/10/gttgxm)
- [2] W. Ahern, J. Yim, D. Tischer, S. Salike, S. M. Woodbury, D. Kim, I. Kalvet, Y. Kipnis, B. Coventry, H. R. Altae-Tran, M. Bauer, R. Barzilay, T. S. Jaakkola, R. Krishna, and D. Baker. Atom level enzyme active site scaffolding using rfdiffusion2. *bioRxiv*, preprint, 2025. doi: [10/g9vc9p](https://doi.org/10/g9vc9p)
- [3] E. Alnabati, J. Esquivel-Rodriguez, G. Terashi, and D. Kihara. MarkovFit: Structure fitting for protein complexes in electron microscopy maps using Markov random field. *Front Mol Biosci*, 9, article no. 935411, 13 pages, 2022. doi: [10/gtmxvj](https://doi.org/10/gtmxvj)
- [4] P. Arora, R. Mehta, and R. Ahuja. An integration of meta-heuristic approach utilizing kernel principal component analysis for multimodal medical image registration. *Cluster Comput*, 27:6223–6246, 24 pages, 2024. doi: [10/gtmxvd](https://doi.org/10/gtmxvd)
- [5] A. S. Asratian, T. M. Denley, and R. Häggkvist. *Bipartite Graphs and their Applications*. Cambridge University Press, UK, 1998. doi: [10/bsfd59](https://doi.org/10/bsfd59)
- [6] D. Auber, D. Archambault, R. Bourqui, M. Delest, J. Dubois, A. Lambert, P. Mary, M. Mathiaut, G. Melançon, B. Pinaud, B. Renoust, and J. Vallet. Tulip 5. In R. Alhajj and J. Rokne, eds., *Encyclopedia of Social Network Analysis and Mining*, pp. 3185–3212. Springer, New York, 2018. doi: [10/g89wsr](https://doi.org/10/g89wsr)
- [7] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, et al. Accurate

- prediction of protein structures and interactions using a three-track neural network. *Sci*, 373(6557):871–876, 2021. doi: [10/gk7nhq](https://doi.org/10/gk7nhq)
- [8] X.-c. Bai, G. McMullan, and S. H. W. Scheres. How cryo-EM is revolutionizing structural biology. *Trends Biochem Sci*, 40(1):49–57, 2015. doi: [10/f6wq7v](https://doi.org/10/f6wq7v)
- [9] A. Banari, A. K. Samanta, A. Munke, T. Laugks, S. Bajt, K. Grünewald, T. C. Marlovits, J. Küpper, F. R. Maia, H. N. Chapman, D. Oberthür, and C. Seuring. Advancing time-resolved structural biology: Latest strategies in cryo-em and x-ray crystallography. *Nature Methods*, 22(7):1420–1435, 2025. doi: [10/g9vc9q](https://doi.org/10/g9vc9q)
- [10] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987. doi: [10/bdqp5z](https://doi.org/10/bdqp5z)
- [11] N. R. Bennett, B. Coventry, I. Goreschnik, B. Huang, A. Allen, D. Vafeados, Y. P. Peng, J. Dauparas, M. Baek, L. Stewart, F. DiMaio, S. De Munck, S. N. Savvides, and D. Baker. Improving de novo protein binder design with deep learning. *Nat Commun*, 14(1), article no. 2625, 9 pages, 2023. doi: [10/gss9xm](https://doi.org/10/gss9xm)
- [12] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res*, 28(1):235–242, 2000. doi: [10/c7g](https://doi.org/10/c7g)
- [13] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychol Rev*, 94(2):115–147, 1987. doi: [10/c9dqh2](https://doi.org/10/c9dqh2)
- [14] W. H. Bragg and W. L. Bragg. The reflection of x-rays by crystals. *Proc R Soc London. Series A, Containing Papers a Math Physical Character*, 88(605):428–438, 1913. doi: [10/c3sfjz](https://doi.org/10/c3sfjz)
- [15] M. Brossier, R. Skånberg, L. Besançon, M. Linares, T. Isenberg, A. Ynnerman, and A. Bock. Moliverse: Contextually embedding the microcosm into the universe. *Comput Graph*, 112:22–30, 2023. doi: [10/gr7bbz](https://doi.org/10/gr7bbz)
- [16] B. M. H. Bruininks, T. A. Wassenaar, and I. Vattulainen. Unbreaking assemblies in molecular simulations with periodic boundaries. *J Chem Inf Model*, 63(11):3448–3452, 2023. doi: [10/mt3x](https://doi.org/10/mt3x)
- [17] C. Bustamante, J. F. Marko, E. D. Siggia, and S. Smith. Entropic elasticity of λ -phage DNA. *Sci*, 265(5178):1599–1600, 1994. doi: [10/dh8fcj](https://doi.org/10/dh8fcj)

- [18] J. Byška, M. Le Muzic, M. E. Gröller, I. Viola, and B. Kozlíková. Animo-AminoMiner: Exploration of protein tunnels and their properties in molecular dynamics. *IEEE Trans Vis Comput Graph*, 22(1):747–756, 2016. doi: [10/gtsjbd](https://doi.org/10/gtsjbd)
- [19] K. Bühler, T. Höllt, T. Schulz, and P.-P. Vázquez. AI-in-the-loop: The future of biomedical visual analytics applications in the era of AI. arXiv preprint 2412.15876, 2024. doi: [10/g89wss](https://doi.org/10/g89wss)
- [20] L. Cao, B. Coventry, I. Goreschnik, B. Huang, W. Sheffler, J. S. Park, K. M. Jude, I. Marković, R. U. Kadam, K. H. G. Verschueren, K. Verstraete, S. T. R. Walsh, N. Bennett, A. Phal, A. Yang, L. Kozodoy, M. DeWitt, L. Picton, L. Miller, E.-M. Strauch, N. D. DeBouver, A. Pires, A. K. Bera, S. Halabiya, B. Hammerson, W. Yang, S. Bernard, L. Stewart, I. A. Wilson, H. Ruohola-Baker, J. Schlessinger, S. Lee, S. N. Savvides, K. C. Garcia, and D. Baker. Design of protein-binding proteins from the target structure alone. *Nature*, 605(7910):551–560, 2022. doi: [10/gptzv3](https://doi.org/10/gptzv3)
- [21] W. Chen, X. Wang, and Y. Wang. FFF: Fragment-guided flexible fitting for building complete protein structures. In *Proc. CVPR*, pp. 19776–19785. IEEE Computer Society, Los Alamitos, 2023. doi: [10/gtmxvf](https://doi.org/10/gtmxvf)
- [22] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning-based protein sequence design using ProteinMPNN. *Sci*, 378(6615):49–56, 2022. doi: [10/gqtj2d](https://doi.org/10/gqtj2d)
- [23] E. de Llano, H. Miao, Y. Ahmadi, A. J. Wilson, M. Beeby, I. Viola, and I. Barišić. Adenita: Interactive 3D modelling and visualization of DNA nanostructures. *Nucleic Acids Res*, 48(15):8269–8275, 2020. doi: [10/gmt49h](https://doi.org/10/gmt49h)
- [24] A. Del Conte, G. F. Camagni, D. Clementel, G. Minervini, A. M. Monzon, C. Ferrari, D. Piovesan, and S. E. Tosatto. RING 4.0: Faster residue interaction networks with novel interaction types across over 35,000 different chemical structures. *Nucleic Acids Res*, 52(W1):W306–W312, 2024. doi: [10/g89wsp](https://doi.org/10/g89wsp)
- [25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009. doi: [10/cvc7xp](https://doi.org/10/cvc7xp)

- [26] J. P. Didon and F. Langevin. Registration of MR images: From 2D to 3D, using a projection based cross correlation method. In *Proc. EMBC*, vol. 1, pp. 489–490. IEEE, Piscataway, 1995. doi: [10/bgp7mt](#)
- [27] S. M. Douglas, A. H. Marblestone, S. Teerapittayanon, A. Vazquez, G. M. Church, and W. M. Shih. Rapid prototyping of 3D DNA-origami shapes with caDNAno. *Nucleic Acids Res*, 37(15):5001–5006, 2009. doi: [10/cb9dg7](#)
- [28] D. Duran, P. Hermosilla, T. Ropinski, B. Kozlíková, Á. Vinacua, and P.-P. Vázquez. Visualization of large molecular trajectories. *IEEE Trans Vis Comput Graph*, 25(1):987–996, 2019. doi: [10/gjbdnk](#)
- [29] M. C. Engel, D. M. Smith, M. A. Jobst, M. Sajfutdinow, T. Liedl, F. Romano, L. Rovigatti, A. A. Louis, and J. P. K. Doye. Force-induced unravelling of DNA origami. *ACS Nano*, 12(7):6734–6747, 2018. doi: [10/gdqrsr](#)
- [30] M. Falk, V. Tobiasson, A. Bock, C. Hansen, and A. Ynnerman. A visual environment for data driven protein modeling and validation. *IEEE Trans Vis Comput Graph*, 30(8):5063–5073, 2024. doi: [10/gsc3c9](#)
- [31] C. Fonseca Guerra, F. M. Bickelhaupt, J. G. Snijders, and E. J. Baerends. Hydrogen bonding in DNA base pairs: Reconciliation of theory and experiment. *J Am Chem Soc*, 122(17):4117–4128, 2000. doi: [10/bm2qkg](#)
- [32] B. Frenz, A. C. Walls, E. H. Egelman, D. Veisler, and F. DiMaio. RosettaES: A sampling strategy enabling automated interpretation of difficult cryo-EM maps. *Nat. Methods*, 14(8):797–800, 2017. doi: [10/gf5cmq](#)
- [33] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang. Deep learning in medical image registration: A review. *Phys Med Biol*, 65(20), article no. 20TR01, 27 pages, 2020. doi: [10/ghn45h](#)
- [34] J. Gao, M. Tong, C. Lee, J. Gaertig, T. Legal, and K. H. Bui. DomainFit: Identification of protein domains in cryo-EM maps at intermediate resolution using AlphaFold2-predicted models. bioRxiv preprint 2023.11.28.569001, 2023. doi: [10/gs63f2](#)
- [35] J. I. Garzón, J. Kovacs, R. Abagyan, and P. Chacón. ADP-EM: Fast exhaustive multi-resolution docking for high-throughput coverage. *Bioinf*, 23(4):427–433, 2006. doi: [10/c9xkmg](#)
- [36] T. D. Goddard, C. C. Huang, and T. E. Ferrin. Visualizing density maps with UCSF Chimera. *J Struct Biol*, 157(1):281–287, 2007. Software tools for macromolecular microscopy. doi: [10/ft849s](#)

- [37] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: Visual analysis of multi-attribute rankings. *IEEE Trans Vis Comput Graph*, 19(12):2277–2286, 2013. doi: [10/f3sxh9](#)
- [38] T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, R. Badkundri, I. Shafkat, J. Gong, A. Derry, R. S. Molina, N. Thomas, Y. A. Khan, C. Mishra, C. Kim, L. J. Bartie, M. Nemeth, P. D. Hsu, T. Sercu, S. Candido, and A. Rives. Simulating 500 million years of evolution with a language model. *Sci*, 387(6736):850–858, 2025. doi: [10/g82p7n](#)
- [39] B. He, F. Zhang, C. Feng, J. Yang, X. Gao, and R. Han. Accurate global and local 3D alignment of cryo-EM density maps using local spatial structural features. *Nat Commun*, 15, article no. 1593, 15 pages, 2024. doi: [10/gtmxvg](#)
- [40] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Trans Vis Comput Graph*, 13(6):1240–1247, 2007. doi: [10/dvqddt](#)
- [41] M. A. Herzik, M. Wu, and G. C. Lander. High-resolution structure determination of sub-100 kDa complexes using conventional cryo-EM. *Nat Commun*, 10, article no. 1032, 9 pages, 2019. doi: [10/gg7zp4](#)
- [42] D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes. Medical image registration. *Phys Med Biol*, 46(3):R1–R45, 2001. doi: [10/fgbhgj](#)
- [43] J. Hong, R. Maciejewski, A. Trubuil, and T. Isenberg. Visualizing and comparing machine learning predictions to improve human-AI teaming on the example of cell lineage. *IEEE Trans Vis Comput Graph*, 30(4):1956–1969, 2024. doi: [10/kt3r](#)
- [44] W. Humphrey, A. Dalke, and K. Schulten. VMD: Visual molecular dynamics. 14(1):33–38, 1996. doi: [10/b3tgfk](#)
- [45] A. Inselberg. The plane with parallel coordinates. *Vis Comput*, 1(2):69–91, 1985. doi: [10/fdpvz2](#)
- [46] D. Jia, A. Irger, L. Besancon, O. Strnad, D. Luo, J. Bjorklund, A. Kouyoumdjian, A. Ynnerman, and I. Viola. VOICE: Visual Oracle for Interaction, Conversation, and Explanation . *IEEE Trans Vis Comput Graph*. To appear. doi: [10/g9r73s](#)
- [47] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Židek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes,

- S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021. doi: [10/gk7nfp](https://doi.org/10/gk7nfp)
- [48] M. S. Keller, I. Gold, C. McCallum, T. Manz, P. V. Kharchenko, and N. Gehlenborg. Vitessce: Integrative visualization of multimodal and spatially resolved single-cell data. *Nature Methods*, 22(1):63–67, 2025. doi: [10/g9g8k3](https://doi.org/10/g9g8k3)
- [49] M. Kesäniemi and K. Virtanen. Direct least square fitting of hyperellipsoids. *IEEE Trans Pattern Anal Mach Intell*, 40(1):63–76, 2018. doi: [10/gcqrjrb](https://doi.org/10/gcqrjrb)
- [50] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint 1412.6980, 2017. doi: [10/hnkr](https://doi.org/10/hnkr)
- [51] S. Klein, J. P. W. Pluim, M. Staring, and M. A. Viergever. Adaptive stochastic gradient descent optimisation for image registration. *Int J Comput Vis*, 81(3):227–239, 2009. doi: [10/dghjcp](https://doi.org/10/dghjcp)
- [52] I. Kolesár, S. Bruckner, I. Viola, and H. Hauser. A fractional Cartesian composition model for semi-spatial comparative visualization design. *IEEE Trans Vis Comput Graph*, 23(1):851–860, 2017. doi: [10/f92dz8](https://doi.org/10/f92dz8)
- [53] I. Kolesár, J. Parulek, I. Viola, S. Bruckner, A.-K. Stavrum, and H. Hauser. Interactively illustrating polymerization using three-level model fusion. *BMC Bioinf*, 15, article no. 345:16, 345:16 pages, 2014. doi: [10/f6tw85](https://doi.org/10/f6tw85)
- [54] Y.-B. Kou, Y.-F. Feng, L.-Y. Shen, X. Li, and C.-M. Yuan. Adaptive spline surface fitting with arbitrary topological control mesh. *IEEE Trans Vis Comput Graph*, 30:12, 13 pages, 2024. To appear. doi: [10/gtmxw4](https://doi.org/10/gtmxw4)
- [55] D. Kut’ák, L. Melo, F. Schroeder, Z. Jelic-Matošević, N. Mutter, B. Bertosa, and I. Barišić. CATANA: An online modelling environment for proteins and nucleic acid nanostructures. *Nucleic Acids Res*, 50(W1):W152–W158, 2022. doi: [10/gs5tck](https://doi.org/10/gs5tck)
- [56] D. Kut’ák, M. N. Selzer, J. Byška, M. L. Ganuza, I. Barišić, B. Kozlíková, and H. Miao. Vivern—A virtual environment for multiscale visualization and modeling of DNA nanostructures. *IEEE Trans Vis Comput Graph*, 28(12):4825–4838, 2022. doi: [10/mpwt](https://doi.org/10/mpwt)

- [57] J. P. Kynast and B. Höcker. Atligator Web: A graphical user interface for analysis and design of protein–peptide interactions. *BioDes Res*, 5, article no. 0011, 6 pages, 2023. doi: [10/g89wsn](https://doi.org/10/g89wsn)
- [58] K. Lasker, O. Dror, M. Shatsky, R. Nussinov, and H. J. Wolfson. EMatch: Discovery of high resolution structural homologues of protein domains in intermediate resolution cryo-EM maps. *IEEE/ACM Trans Comput Biol Bioinf*, 4(1):28–39, 2007. doi: [10/ccbr4m](https://doi.org/10/ccbr4m)
- [59] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int J Comput Vision*, 60(2):91–110, 2004. doi: [10/bqrmisp](https://doi.org/10/bqrmisp)
- [60] W. Lueks, I. Viola, M. van der Zwan, H. Bekker, and T. Isenberg. Spatially continuous change of abstraction in molecular visualization. In *IEEE BioVis Abstracts*, 2011. url: hal.science/hal-00781520.
- [61] D. Luo, Z. Alsuwaykit, D. Khan, O. Strnad, T. Isenberg, and I. Viola. Diffit: Visually-guided differentiable fitting of molecule structures to a cryo-em map. *IEEE Trans Vis Comput Graph*, 31(1):558–568, 2025. doi: [10/njn2](https://doi.org/10/njn2)
- [62] P. K. Maiti, T. A. Pascal, N. Vaidehi, J. Heo, and W. A. Goddard III. Atomic-level simulations of Seeman DNA nanostructures: The paranemic crossover in salt solution. *Biophys J*, 90(5):1463–1479, 2006. doi: [10/fm5cb7](https://doi.org/10/fm5cb7)
- [63] S. Malhotra, S. Träger, M. Dal Peraro, and M. Topf. Modelling structures in cryo-EM maps. *Curr Opin Struct Biol*, 58:105–114, 2019. doi: [10/gtmxvc](https://doi.org/10/gtmxvc)
- [64] M. Malinsky, R. Peter, E. Hodneland, A. J. Lundervold, A. Lundervold, and J. Jan. Registration of FA and T1-weighted MRI data of healthy human brain based on template matching and normalized cross-correlation. *J Digit Imaging*, 26(4):774–785, 2013. doi: [10/gtmxw5](https://doi.org/10/gtmxw5)
- [65] V. Mallet, C. Rapisarda, H. Minoux, and M. Ovsjanikov. Finding antibodies in cryo-EM densities with CrAI. bioRxiv preprint 2023.09.27.559736, 2023. doi: [10/gtn9kj](https://doi.org/10/gtn9kj)
- [66] D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellen, and W. Eubank. PET-CT image registration in the chest using free-form deformations. *IEEE Trans Med Imaging*, 22(1):120–128, 2003. doi: [10/cs4pch](https://doi.org/10/cs4pch)
- [67] H. Miao, E. De Llano, T. Isenberg, M. E. Gröller, I. Barišić, and I. Viola. DimSUM: Dimension and scale unifying map for visual abstraction of DNA origami structures. *Comput Graph Forum*, 37(3):403–413, 2018. doi: [10/gdw4j6](https://doi.org/10/gdw4j6)

- [68] H. Miao, E. De Llano, J. Sorger, Y. Ahmadi, T. Kekic, T. Isenberg, M. E. Gröller, I. Barišić, and I. Viola. Multiscale visualization and scale-adaptive modification of DNA nanostructures. *IEEE Trans Vis Comput Graph*, 24(1):1014–1024, 2018. doi: [10/gcqbttq](https://doi.org/10/gcqbttq)
- [69] H. Mohammed, A. K. Al-Awami, J. Beyer, C. Cali, P. Magistretti, H. Pfister, and M. Hadwiger. Abstractocyte: A visual tool for exploring nanoscale astroglial cells. *IEEE Trans Vis Comput Graph*, 24(1):853–861, 2018. doi: [10/gcqqk6d](https://doi.org/10/gcqqk6d)
- [70] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal. Human-in-the-loop machine learning: A state of the art. *Artif Intell Rev*, 56(4):3005–3054, 2023. doi: [10/gscetfh](https://doi.org/10/gscetfh)
- [71] T. Nakane, A. Kotecha, A. Sente, G. J. McMullan, S. Masiulis, P. M. G. E. Brown, I. Grigoras, L. Malinauskaite, T. Malinauskas, J. Miehl, L. Yu, D. Karia, E. V. Pechnikova, E. de Jong, J. Keizer, M. Bischoff, J. McCormack, P. Tiemeijer, S. W. Hardwick, D. Y. Chirgadze, G. N. Murshudov, A. R. Aricescu, and S. H. W. Scheres. Single-particle cryo-EM at atomic resolution. *Nature*, 587(7832):152–156, 2020. doi: [10/gjwcfp](https://doi.org/10/gjwcfp)
- [72] National Academies of Sciences, Engineering, and Medicine. *Human-AI Teaming: State-of-the-Art and Research Needs*. The National Academies Press, Washington, DC, 2022. doi: [10/pdnm](https://doi.org/10/pdnm)
- [73] N. Nguyen, O. Strnad, T. Klein, D. Luo, R. Alharbi, P. Wonka, M. Maritan, P. Mindek, L. Autin, D. S. Goodsell, and I. Viola. Modeling in the time of COVID-19: Statistical and rule-based mesoscale models. *IEEE Trans Vis Comput Graph*, 27(2):722–732, 2021. doi: [10/k8sh](https://doi.org/10/k8sh)
- [74] F. Opaleny, P. Ulbrich, J. Planas-Iglesias, J. Byska, J. Stourac, D. Bednar, K. Furmanova, and B. Kozlikova. Visual Support for the Loop Grafting Workflow on Proteins . *IEEE Transactions on Visualization & Computer Graphics*, 31(01):580–590, Jan. 2025. doi: [10/g9vdcf](https://doi.org/10/g9vdcf)
- [75] T. E. Ouldridge, A. A. Louis, and J. P. K. Doye. Structural, mechanical, and thermodynamic properties of a coarse-grained DNA model. *J Chem Phys*, 134(8), article no. 085101:22, 085101:22 pages, 2011. doi: [10/cbrjcm](https://doi.org/10/cbrjcm)
- [76] M. Pacesa, L. Nickel, J. Schmidt, E. Pyatova, C. Schellhaas, L. Kissling, A. Alcaraz-Serna, Y. Cho, K. H. Ghamary, L. Vinué, B. J. Yachnin, A. M. Wollacott, S. Buckley, S. Georgeon, C. A. Goverde, G. N. Hatzopoulos, P. Gönczy, Y. D. Muller, G. Schwank, S. Ovchinnikov, and B. E. Correia.

BindCraft: One-shot design of functional protein binders. bioRxiv preprint 2024.09.30.615802, 2024. doi: [10/nmfm](https://doi.org/10/nmfm)

- [77] J. Páleník, J. Byška, S. Bruckner, and H. Hauser. Scale-space splatting: Reforming spacetime for cross-scale exploration of integral measures in molecular dynamics. *IEEE Trans Vis Comput Graph*, 26(1):643–653, 2020. doi: [10/kvh8](https://doi.org/10/kvh8)
- [78] G. A. Pavlopoulos, P. I. Kontou, A. Pavlopoulou, C. Bouyioukos, E. Markou, and P. G. Bagos. Bipartite graphs in systems biology and medicine: A survey of methods and applications. *GigaSci*, 7(4), article no. giy014, 31 pages, 2018. doi: [10/g89wsq](https://doi.org/10/g89wsq)
- [79] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. UCSF Chimera—A visualization system for exploratory research and analysis. *J Comput Chem*, 25(13):1605–1612, 2004. doi: [10/b4bq4c](https://doi.org/10/b4bq4c)
- [80] E. F. Pettersen, T. D. Goddard, C. C. Huang, E. C. Meng, G. S. Couch, T. I. Croll, J. H. Morris, and T. E. Ferrin. UCSF ChimeraX: Structure visualization for researchers, educators, and developers. *Protein Sci*, 30(1):70–82, 2021. doi: [10/ghr6mn](https://doi.org/10/ghr6mn)
- [81] J. Pfab, P. Minh Nhut, and D. Si. DeepTracer for fast de novo cryo-EM protein structure modeling and special studies on CoV-related complexes. *PNAS*, 118(2), article no. e2017525118, 12 pages, 2021. doi: [10/gjnkwt](https://doi.org/10/gjnkwt)
- [82] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: A survey. *IEEE Trans. Med. Imaging*, 22(8):986–1004, 2003. doi: [10/czrq8h](https://doi.org/10/czrq8h)
- [83] E. Poppleton, J. Bohlin, M. Matthies, S. Sharma, F. Zhang, and P. Šulc. Design, optimization and analysis of large DNA and RNA nanostructures through interactive visualization, editing and molecular simulation. *Nucleic Acids Res*, 48(12):e72:1–e72:12, 2020. doi: [10/gs5xc8](https://doi.org/10/gs5xc8)
- [84] V. Rantos, K. Karius, and J. Kosinski. Integrative structural modeling of macromolecular complexes using Assemblin. *Nat Protoc*, 17:152–176, 2022. doi: [10/gq586q](https://doi.org/10/gq586q)
- [85] P. Reddy, P. Guerrero, M. Fisher, W. Li, and N. J. Mitra. Discovering pattern structure using differentiable compositing. *ACM Trans Graph*, 39(6), article no. 262, 15 pages, 2020. doi: [10/gtmxvb](https://doi.org/10/gtmxvb)

- [86] J. Rogers, M. Anastacio, J. Bernard, M. Chakhchoukh, R. Faust, A. Kerren, S. Koch, L. Kotthoff, C. Turkey, and E. Wall. Visualization and automation in data science: Exploring the paradox of humans-in-the-loop. In *Proc. VDS*, pp. 1–5. IEEE Computer Society, Los Alamitos, 2024. doi: [10/g858cm](https://doi.org/10/g858cm)
- [87] P. W. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006. doi: [10/dzh8m2](https://doi.org/10/dzh8m2)
- [88] L. Rovigatti, F. Romano, E. Poppleton, M. Matthies, and P. Šulc. Documentation – oxDNA. lorenzo.rovigatti.github.io/oxDNA/, 2022. Accessed in Apr. 2024.
- [89] L. Rovigatti, P. Šulc, I. Z. Reguly, and F. Romano. A comparison between parallelization approaches in molecular dynamics simulations on GPUs. *J Comput Chem*, 36(1):1–8, 2015. doi: [10/b5k3](https://doi.org/10/b5k3)
- [90] L. Sael and D. Kihara. Protein surface representation and comparison: New approaches in structural proteomics. In *Biological Data Mining*, chap. 5, pp. 109–130. Chapman and Hall/CRC, New York, 2009. doi: [10/c83vqf](https://doi.org/10/c83vqf)
- [91] V. Schetinger, S. Di Bartolomeo, M. El-Assady, A. McNutt, M. Miller, J. P. A. Passos, and J. L. Adams. Doom or deliciousness: Challenges and opportunities for visualization in the age of generative models. *Comput Graph Forum*, 42(3):423–435, 2023. doi: [10/pdf9](https://doi.org/10/pdf9)
- [92] J. Schmidt, R. Preiner, T. Auzinger, M. Wimmer, M. E. Gröller, and S. Bruckner. YMCA – Your mesh comparison application. In *Proc. VAST*, pp. 153–162. IEEE Computer Society, Los Alamitos, 2014. doi: [10/f3sttr](https://doi.org/10/f3sttr)
- [93] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. Online: pymol.org/support.html, November 2015.
- [94] M. Schäfer, N. Brich, J. Byška, S. M. Marques, D. Bednář, P. Thiel, B. Kozlíková, and M. Krone. InVADo: Interactive visual analysis of molecular docking data. *IEEE Trans Vis Comput Graph*, 30(4):1984–1997, 2024. doi: [10/gtnn2n](https://doi.org/10/gtnn2n)
- [95] N. C. Seeman. Nucleic acid junctions and lattices. *J Theor Biol*, 99(2):237–247, 1982. doi: [10/fbkm7q](https://doi.org/10/fbkm7q)
- [96] N. C. Seeman and H. F. Sleiman. DNA nanotechnology. *Nat Rev Mater*, 3(1), article no. 17068:23, 17068:23 pages, 2017. doi: [10/gghfhj](https://doi.org/10/gghfhj)

- [97] A. Seiler, D. Großmann, and B. Jüttler. Spline surface fitting using normal data and norm-like functions. *Comput Aided Geom Des*, 64:37–49, 2018. doi: [10/gd5gt4](https://doi.org/10/gd5gt4)
- [98] T. R. Shaham, T. Dekel, and T. Michaeli. SinGAN: Learning a generative model from a single natural image. In *Proc. ICCV*, pp. 4569–4579. IEEE Computer Society, Los Alamitos, 2019. doi: [10/gg8fc3](https://doi.org/10/gg8fc3)
- [99] L. Shang, J. Lv, and Z. Yi. Rigid medical image registration using PCA neural network. *NeuroComput*, 69(13–15):1717–1722, 2006. doi: [10/fsrrb2](https://doi.org/10/fsrrb2)
- [100] K. Shoemake. Uniform random rotations. In *Graphics Gems III (IBM Version)*, pp. 124–132. Elsevier, 1992.
- [101] R. Skånberg, I. Hotz, A. Ynnerman, and M. Linares. VIAMD: A software for visual interactive analysis of molecular dynamics. *J Chem Inf Model*, 63(23):7382–7391, 2023. doi: [10/gs7tzh](https://doi.org/10/gs7tzh)
- [102] R. Skånberg, M. Linares, C. König, P. Norman, D. Jönsson, I. Hotz, and A. Ynnerman. VIA-MD: Visual interactive analysis of molecular dynamics. In *Proc. MolVA*, pp. 19–27. The Eurographics Association, Goslar, 2018. doi: [10/gf9rvn](https://doi.org/10/gf9rvn)
- [103] R. Skånberg, P.-P. Vázquez, V. Guallar, and T. Ropinski. Real-time molecular visualization supporting diffuse interreflections and ambient occlusion. *IEEE Trans Vis Comput Graph*, 22(1):718–727, 2016. doi: [10/mpwv](https://doi.org/10/mpwv)
- [104] B. E. Snodin, F. Randisi, M. Mosayebi, P. Šulc, J. S. Schreck, F. Romano, T. E. Ouldridge, R. Tsukanov, E. Nir, A. A. Louis, and J. P. K. Doye. Introducing improved structural properties and salt dependence into a coarse-grained model of DNA. *J Chem Phys*, 142(23), article no. 234901:12, 234901:12 pages, 2015. doi: [10/f7sbb2](https://doi.org/10/f7sbb2)
- [105] B. Sommer, D. Inoue, and M. Baaden. Design X Bioinformatics: A community-driven initiative to connect bioinformatics and design. *J Integr Bioinf*, 19(2), article no. 20220037:8, 20220037:8 pages, 2022. doi: [10/gtsjbc](https://doi.org/10/gtsjbc)
- [106] J. A. Stevens, F. Grünewald, P. van Tilburg, M. König, B. R. Gilbert, T. A. Brier, Z. R. Thornburg, Z. Luthey-Schulten, and S. J. Marrink. Molecular dynamics simulation of an entire cell. *Front Chem*, 11, article no. 1106495:9, 1106495:9 pages, 2023. doi: [10/grqd9f](https://doi.org/10/grqd9f)
- [107] P. Šulc, F. Romano, T. E. Ouldridge, L. Rovigatti, J. P. Doye, and A. A. Louis. Sequence-dependent thermodynamics of a coarse-grained DNA model.

- J Chem Phys*, 137(13), article no. 135101:14, 135101:14 pages, 2012. doi: [10/gkqqbf](https://doi.org/10/gkqqbf)
- [108] T. Terwilliger, P. Adams, P. Afonine, and O. Sobolev. A fully automatic method yielding initial models from high-resolution cryo-electron microscopy maps. *Nat Methods*, 15(11):905–908, 2018. doi: [10/gfmb8d](https://doi.org/10/gfmb8d)
 - [109] P. Thevenaz and M. Unser. Optimization of mutual information for multiresolution image registration. *IEEE Trans Image Process*, 9(12):2083–2099, 2000. doi: [10/fs4j7h](https://doi.org/10/fs4j7h)
 - [110] E. R. Tufte and P. R. Graves-Morris. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, 1983.
 - [111] P. Ulbrich, M. Waldner, K. Furmanová, S. M. Marques, D. Bednář, B. Kozlíková, and J. Byška. sMolBoxes: Dataflow model for molecular dynamics exploration. *IEEE Trans Vis Comput Graph*, 29(1):581–590, 2023. doi: [10/kvjv](https://doi.org/10/kvjv)
 - [112] P. Ulbrich, M. Waldner, K. Furmanová, S. M. Marques, D. Bednář, B. Kozlíková, and J. Byška. sMolBoxes: Dataflow model for molecular dynamics exploration. *IEEE Trans Vis Comput Graph*, 29(1):581–590, 2023. doi: [10/kvjv](https://doi.org/10/kvjv)
 - [113] M. van der Zwan, W. Lueks, H. Bekker, and T. Isenberg. Illustrative molecular visualization with continuous abstraction. *Comput Graph Forum*, 30(3):683–690, 2011. doi: [10/c893rz](https://doi.org/10/c893rz)
 - [114] M. Varadi, D. Bertoni, P. Magana, U. Paramval, I. Pidruchna, M. Radhakrishnan, M. Tsenkov, S. Nair, M. Mirdita, J. Yeo, O. Kovalevskiy, K. Tunyasuvunakool, A. Laydon, A. Žídek, H. Tomlinson, D. Hariharan, J. Abrahamson, T. Green, J. Jumper, E. Birney, M. Steinegger, D. Hassabis, and S. Velankar. Alphafold protein structure database in 2024: Providing structure coverage for over 214 million protein sequences. *Nucleic Acids Res*, 52(D1):D368–D375, 2023. doi: [10/gtkw5z](https://doi.org/10/gtkw5z)
 - [115] S. Vázquez Torres, M. Benard Valle, S. P. Mackessy, S. K. Menzies, N. R. Casewell, S. Ahmadi, N. J. Burlet, E. Muratspahić, I. Sappington, M. D. Overath, E. Rivera-de Torre, J. Ledergerber, A. H. Laustsen, K. Boddum, A. K. Bera, A. Kang, E. Brackenbrough, I. A. Cardoso, E. P. Crittenden, R. J. Edge, J. Decarreau, R. J. Ragotte, A. S. Pillai, M. Abedi, H. L. Han, S. R. Gerben, A. Murray, R. Skotheim, L. Stuart, L. Stewart, T. J. A. Fryer,

- T. P. Jenkins, and D. Baker. De novo designed proteins neutralize lethal snake venom toxins. *Nature*, 639(8053):225–231, 2025. doi: [10/g8z5cw](https://doi.org/10/g8z5cw)
- [116] I. Viola, M. Chen, and T. Isenberg. Visual abstraction. In *Foundations of Data Visualization*, chap. 2, pp. 15–37. Springer, Cham, 2020. doi: [10/gk874c](https://doi.org/10/gk874c)
- [117] I. Viola and T. Isenberg. Pondering the concept of abstraction in (illustrative) visualization. *IEEE Trans Vis Comput Graph*, 24(9):2573–2588, 2018. doi: [10/gd3k7m](https://doi.org/10/gd3k7m)
- [118] J. W. von Goethe. *Faust. Eine Tragödie [Faust: A Tragedy]*. J. G. Cotta, Tübingen, 1808. Online: [gutenberg.org/ebooks/21000](https://www.gutenberg.org/ebooks/21000); English translation: [gutenberg.org/ebooks/3023](https://www.gutenberg.org/ebooks/3023).
- [119] G. Wagner, W. Braun, T. F. Havel, T. Schaumann, N. Gö, and K. Wüthrich. Protein structures in solution by nuclear magnetic resonance and distance geometry: The polypeptide fold of the basic pancreatic trypsin inhibitor determined using two different algorithms, disgeo and disman. *J Mol biol*, 196(3):611–639, 1987. doi: [10/d85vgd](https://doi.org/10/d85vgd)
- [120] T. Walton, M. Gui, S. Velkova, M. R. Fassad, R. A. Hirst, E. Haarman, C. O’Callaghan, M. Bottier, T. Burgoyne, H. M. Mitchison, and A. Brown. Axonemal structures reveal mechanoregulatory and disease mechanisms. *Nature*, 618(7965):625–633, 2023. doi: [10/gscb95](https://doi.org/10/gscb95)
- [121] P. Wang, G. Chatterjee, H. Yan, T. H. LaBean, A. J. Turberfield, C. E. Castro, G. Seelig, and Y. Ke. Practical aspects of structural and dynamic DNA nanotechnology. *MRS Bull*, 42(12):889–896, 2017. doi: [10/gcqfzw](https://doi.org/10/gcqfzw)
- [122] R. Y.-R. Wang, M. Kudryashev, X. Li, E. H. Egelman, M. Basler, Y. Cheng, D. Baker, and F. DiMaio. De novo protein structure determination from near-atomic-resolution cryo-EM maps. *Nat Methods*, 12(4):335–338, 2015. doi: [10/gf5cn8](https://doi.org/10/gf5cn8)
- [123] X. Wang, E. Alnabati, T. W. Aderinwale, S. R. M. V. Subramaniya, G. Terashi, and D. Kihara. Detecting protein and DNA/RNA structures in cryo-EM maps of intermediate resolution using deep learning. *Nat Commun*, 12, article no. 2302, 9 pages, 2020. doi: [10/gmz55b](https://doi.org/10/gmz55b)
- [124] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh,

- I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi: [10/gsgbqt](https://doi.org/10/gsgbqt)
- [125] K. Xu, Z. Wang, J. Shi, H. Li, and Q. C. Zhang. A2-Net: Molecular structure estimation from cryo-EM density volumes. *Proc AAAI Conf Artif Intell*, 33(1):1230–1237, 2019. doi: [10/ghkjcf](https://doi.org/10/ghkjcf)
- [126] C.-T. Yeh, L. Obendorf, and F. Parmeggiani. Elfin UI: A graphical interface for protein design with modular building blocks. *Front Bioeng Biotechnol*, 8, article no. 568318, 11 pages, 2020. doi: [10/g89wsm](https://doi.org/10/g89wsm)
- [127] I.-C. Yeh, C.-H. Lin, O. Sorkine, and T.-Y. Lee. Template-based 3D model fitting using dual-domain relaxation. *IEEE Trans Vis Comput Graph*, 17(8):1178–1190, 2011. doi: [10/dkbjmd](https://doi.org/10/dkbjmd)
- [128] V. Zambaldi, D. La, A. E. Chu, H. Patani, A. E. Danson, T. O. C. Kwan, T. Frerix, R. G. Schneider, D. Saxton, A. Thillaisundaram, Z. Wu, I. Moraes, O. Lange, E. Papa, G. Stanton, V. Martin, S. Singh, L. H. Wong, R. Bates, S. A. Kohl, J. Abramson, A. W. Senior, Y. Alguel, M. Y. Wu, I. M. Aspalter, K. Bentley, D. L. V. Bauer, P. Cherepanov, D. Hassabis, P. Kohli, R. Fergus, and J. Wang. De novo design of high-affinity protein binders with AlphaProteo. arXiv preprint 2409.08022, 2024. doi: [10/g89wst](https://doi.org/10/g89wst)
- [129] Z. Zhao, P. Xu, C. Scheidegger, and L. Ren. Human-in-the-loop extraction of interpretable concepts in deep learning models. *IEEE Trans Vis Comput Graph*, 28(1):780–790, 2022. doi: [10/gnhb8s](https://doi.org/10/gnhb8s)
- [130] X. Zhou, S. Kong, A. Maker, S. G. Remesh, K. K. Leung, K. A. Verba, and J. A. Wells. Antibody discovery identifies regulatory mechanisms of protein arginine deiminase 4. *Nat Chem Biol*, 20(6):742–750, 2024. doi: [10/mpwf](https://doi.org/10/mpwf)
- [131] Y. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Trans Graph*, 37(4), article no. 49, 13 pages, 2018. doi: [10/gd52tv](https://doi.org/10/gd52tv)

APPENDICES

In these appendices, I provide additional explanations, tables, plots, and charts that offer further detail and context beyond what is presented in the main chapters. These supplementary materials are included to enhance the clarity, reproducibility, and completeness of the research, allowing interested readers to explore the data and analyses more deeply.

Appendix A

Appendix for DiffFit

A.1 Detailed benchmark results for use case scenario 1— Fit a single structure

In Table A.1 we provide a detailed benchmark table for the first use case.

A.2 Details on the user feedback sessions

One participant in the first feedback group sent us written feedback in addition to the comments during the Zoom session, which we attach in anonymized form at the very end of this appendix. In addition, the expert who participated in the in-person session also sent additional feedback by e-mail, which we also include in anonymized form at the end of the appendix.

Appendix B

Appendix for SynopFrame

B.1 Detailed description of SynopPoints

We describe each of the implemented SynopPoints in more detail and discuss how they help us to address the tasks and challenges we outlined before. We use an icosahedron nanostructure (6,540 NTs) as an example DNA-nano design and, for each representation, describe its rendering method and the algorithms used to preprocess the input data when necessary.

All-NT, SP1 (*Precise*, *NT*, *Geon*), (for each representation, we mention its name, SynopPoint index, and its coordinate in SynopSpace) shown in Figure 6.1a and 6.4a, is the traditionally used representation in most DNA-nano-related tools, and is the only one in oxView [83]. It shows the precise positions and orientations (T6) of all NTs output by the simulator and is ideal for scrutinizing an H-bond pairing event in its local environment. Hence it helps with identifying interesting local events and understanding conformation changes (T3, T4). The output data from oxDNA are the center of mass (CMS) position and two orientation vectors for each NT. Following the oxDNA2 model’s geometry [88] (see also B.3), we calculate the required positions and orientations for the *backbone*, the *base*, the *backbone-backbone connector* and the *backbone-base connector*. We then instantiate the backbone repulsion sites with spheres, base stacking sites with ellipsoids, backbone connector sites with truncated cones, and base connector sites with cylinders. To be consistent with the domain we use the same color scheme as in oxView for the base ellipsoids, i.e., blue for *A*, red for *T*, green for *C*, yellow for *G*. We leave the backbone spheres and connectors to color-encode

other properties (section 6.3).

Snake, SP2 (*Precise, Strand, Snake*), shown in Figure 6.1b and 6.4b, in its dynamic context is inspired by less formal representations used on various occasions such as on-demand drawings, gestures in a conversation, or educational animations (DNA origami folding animation, see youtu.be/p4C_aFlyhfI). To create it we remove the detail of the base and backbone of an NT and only show the position of each strand. As such, this representation reduces, to some extent, the high frequencies (C1) as well as clutter and occlusion (C2). It is ideal for examining a strand pairing event in a slightly wider local environment than that of a single H-bond, it thus helps experts to identify events and understand conformation changes (T3, T4) at a coarser level as well as with understanding those events in a larger context (T2). With the CMS data of each NT and the design's topology, we construct a polyline primitive by connecting it from the 3' end NT toward the 5' end for each strand. Translational sweeping of a circle along the curve then forms a snake-like geometry.

Schematic3D, SP3 (*Schematic, Strand, Bar*), shown in Figure 6.1c and 6.4c, is a caDNAno-like representation that we created by reasoning with the help of SynopSpace. It can be thought of as the 3D version of caDNAno's representation which, instead of using the precise positions of each frame's configuration, relies on a single frame. In practice, we observe that the designed configuration—before any relaxation or simulation (C5)—has the optimal geometry for a user to comprehend the structure, for several reasons: First, it is designed by people who often also analyze its MDS. Second, it is usually the configuration that people mentally construct. And, third, the double helices are still straight, which helps the experts to comprehend the structure. Even though sometimes elongated backbones exist, they help users understand the relationship between the surrounding structures by placing the paired NTs together to keep the clutter low, rather than causing any illusion or confusion. After simplifying double helices into two parallel lines, this representation further decreases the clutter and occlusion (C2), which is

most beneficial in large structures (C1). As the geometry is now static, there is no more periodic bounding box issue (C3). After color coding the properties, e. g., H-bond statuses onto the geometry, it also helps the user to understand the simulation schematically (T2) at various levels depending on the zoom level. Furthermore, it serves as a bridge to connect other representations to tackle more tasks and challenges, e. g., the user might observe in the to-be-described *Schematic2D* representation that a helix is unpairing (T3), and then move to *Schematic3D* to understand which strand this helix belongs to, the routing (the crossovers involved) of that strand, as well as the H-bond statuses of the whole strand (T2). Then they may move to *Snake* to examine this strand's precise positions and see how it interacts with other strands in the spatial context (T4). We describe the implementation details in section 6.3.

Schematic2D, SP4 (*Schematic, Helix, Bar*), shown in Figure 6.1d and 6.4d, is adapted from the caDNAno-like representation for which we removed all the linkages that show the crossover between continuous double helices that can cause a great deal of occlusion. *Schematic2D* is thus capable of showing all the double helices (T3) without occlusion (C2). It is ideal to monitor the pairing and/or unpairing of any number of double helices depending on the zoom level (T2). We use the *schematic2D_row* value from before, but still need *schematic2D_col* to place each segment of a helix and each NT on a segment. For the calculation of *schematic2D_col* for each NT we need to take into account the directionality of the segment: the offset from the 3' end of the segment. We then arrange each helix according to its row and length as well as arranging each NT. We also arrange the singleton NTs next to their closest NT's helices. We use the same rendering method as for *Snake* and expose parameters to allow the user to control the width, height, and row and column spacing of the arrangement of the helices.

Heatbar, SP5 (*Sequential, Strand, Bar*), shown in Figure 6.1e and 6.4e, is adapted from Adenita [23] where it is used to convey the length of all strands. Each strand is shown as a continuous vertical straight line and the strands are

horizontally laid out, resulting in a representation that resembles bar plots. As the scaffold is usually more than ten times longer than the staples, however, in our scenario this representation wastes a lot of screen space and it becomes difficult to observe color changes on the short staples. To address these issues we first define how many NTs each row can harbor, split the scaffold into multiple consecutive rows, and then place the staples horizontally with spacing in-between. We allow users to control the maximum NTs per row, the height of the bar, the width of each NT, and the horizontal and vertical spacing with sliders. This representation is ideal for glancing over the dynamics simulation to understand the overall process (T1) at the strand granularity and identify issues of certain strands if there are any. As the strands are laid out in 2D there is also no occlusion (C2). We realize this view by assigning a row and column index to each NT according to the length of the strand it belongs to and the offset to the 3' end on that strand. We use the same rendering method as for *Snake*.

Progress bar, SP6 (*Sequential, Assembly, Bar*), shown in Figure 6.1f and 6.4f, is the linear layout of the NTs based on their H-bond statuses. It is ideal to understand the overall process of H-bond changes (T1) and let the user decide whether a certain simulation period should be further examined. No matter how frequently H-bonds change, how large a structure is, or how many frames a simulation has (C1), at this abstract level the user can quickly perceive any changes and make decisions accordingly. We implemented this view by again assigning a row and column index to each NT, but here only according to the NT's H-bond status. The layout can also be adapted by the user to the viewport size and aspect ratio. We achieve the final visual representation by rendering impostors to form a circle at the given positions. Thus, when zooming out, the representation looks like a progress bar and, when zooming in, each NT is observable as an individual circle. We also do not use numbers to record the H-bond status because changing text does not communicate the dynamic nature of the simulation efficiently. Instead, we use color coding that we explain in

section 6.3.

B.2 DNA-nano design simulation

After a DNA-nano structure is designed, usually before performing the wet lab experiments, domain experts subject it to simulation to test its *dynamic* properties. There are several DNA simulators, ranging from a focus on the atom level [62] to the polymer level [17]. We rely on a commonly used one, oxDNA [75, 89, 104, 107]. OxDNA runs at the NT level and captures the movement of each nucleotide, the binding of Watson-Crick base pairs, and “zipping” events (binding of multiple consecutive NTs) between complementary strands. To prepare a newly designed structures for the actual dynamics simulations, we first need to run relaxation simulations that prepare the structure in a proper configuration that can further be simulated via Monte Carlo or molecular dynamics methods. We focus on the molecular dynamics simulation that, with a given initial configuration, generates the configuration in consecutive time frames via integration on small time steps, according to a selected biophysics model. The resulting configurations can be sampled at a user-specified time interval to form an MD trajectory and then stored. In the case of DNA-nano, a typical trajectory has thousands to millions of frames and each frame has tens of thousands of NTs, which easily leads to gigabytes to terabytes of data.

B.3 OxDNA2’s model geometry

With the center of mass (CMS) position vector (\mathbf{r}) and two orientation vectors ($\mathbf{a}_1, \mathbf{a}_2$) for each NT, we calculate the required positions (\mathbf{P}) and normals (\mathbf{N})

with the following equations. The *end3* in the equation refers to the 3' end.

$$P_{backbone} = r - 0.34 * a1 + 0.3408 * a2$$

$$P_{base} = r + 0.34 * a1$$

$$P_{backbone_connector} = (P_{backbone_end3} + P_{backbone_end5})/2$$

$$N_{backbone_connector} = \text{normalize}(P_{backbone_end3} - P_{backbone_end5})$$

$$P_{base_connector} = (P_{base} + P_{backbone})/2$$

$$N_{base_connector} = (P_{base} - P_{backbone})/2$$

B.4 Houdini-specific implementation

There are significant benefits to using Houdini as our prototyping environment. First, the *Apprentice* version is free and available for all three major platforms (Windows, Mac OS, and Linux). Second, from the user's perspective, after our application is developed, a typical user needs just around a 15-minute onboarding tutorial and a one-page cheat sheet to be comfortable navigating inside the application and then focusing on the analysis of their MDS trajectory. Third and most important, it is extremely friendly from the developer's perspective. Multiple viewports, which are required by the linked views, can be created with a few mouse clicks, followed by specifying the geometry to be rendered. Another valuable feature is Houdini's node-based workflow.¹ Each node harbors a tabular data structure for the points, vertices, primitives, and detail it contains so the developer can assign attributes to them and leverage the handy and fast attribute fetching function via its high-performance expression language VEX.² Each node is also a small program that performs certain transformations on the data it receives, much like the concept of a shader. The program can be written in VEX, Python, or C++ via Houdini Developer Kit (HDK).³ In the case of VEX, the program can be chosen to execute just once, or in parallel for each point/vertex/primitive.

¹sidefx.com/tutorials/intro-to-houdinis-node-based-workflow

²sidefx.com/docs/houdini/vex

³sidefx.com/docs/hdk

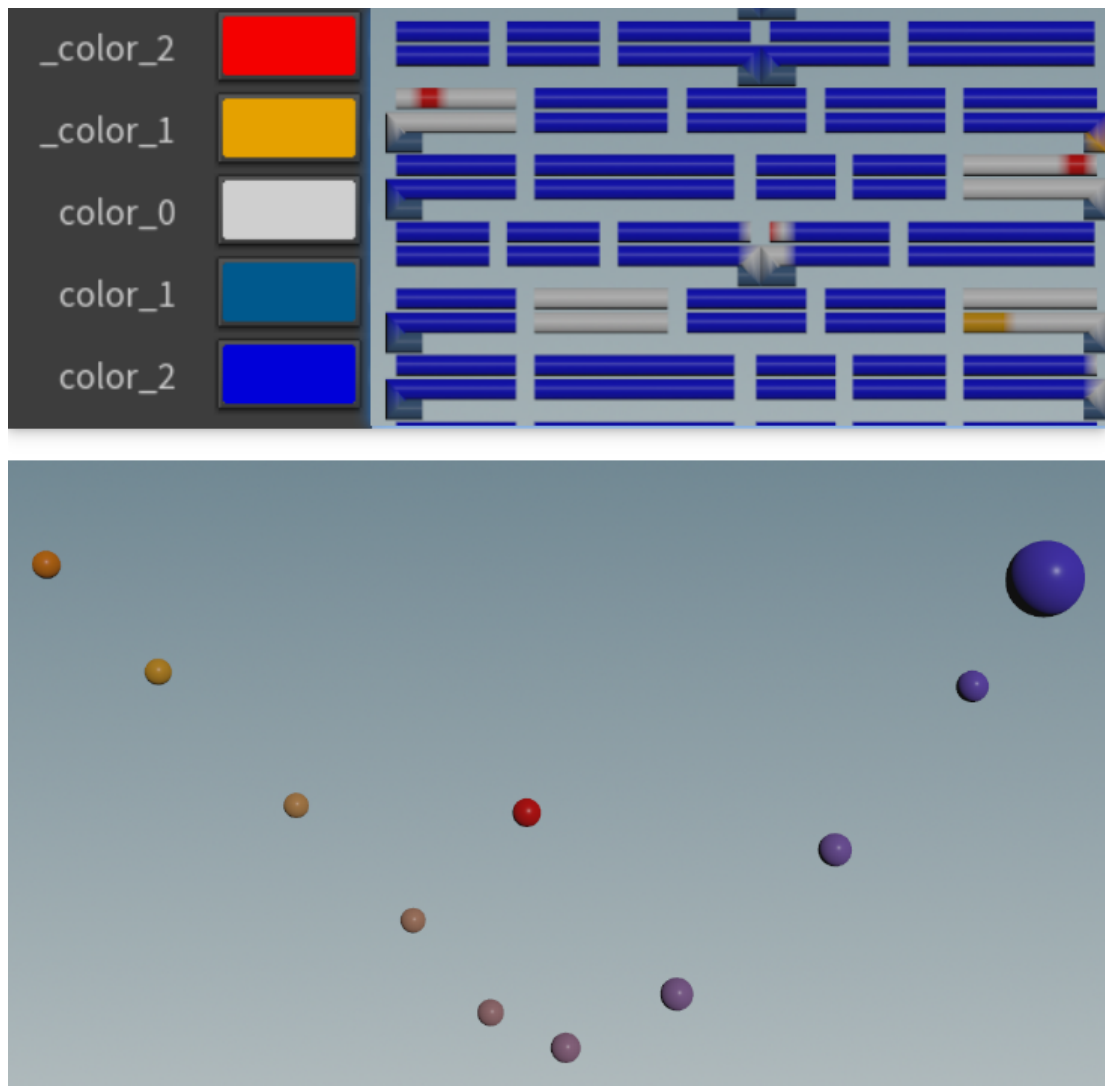


Figure B.1: Example for the use of a different color map for people with color deficiencies. The top panel shows the new zoomed-in area of Figure 6.1d; the bottom panel shows the new Figure 6.1g.

Another feature we have used is the data cache node⁴ that can cache the processed data into binary for fast loading the next time.

We use C++ via HDK to parse the large trajectory data as well as the newly defined H-bond data to gain the highest performance; we use VEX to perform all the geometry transformations and have leveraged the parallel processing to the best we can; we use Python for various other small tasks as well as the Python Viewer State⁵ to achieve most of the user interactions.

We can also adjust the specific coloring of, in particular, the H-bonds to avoid

⁴sidefx.com/docs/houdini/nodes/sop/cache.html

⁵sidefx.com/docs/houdini/hom/python_states.html

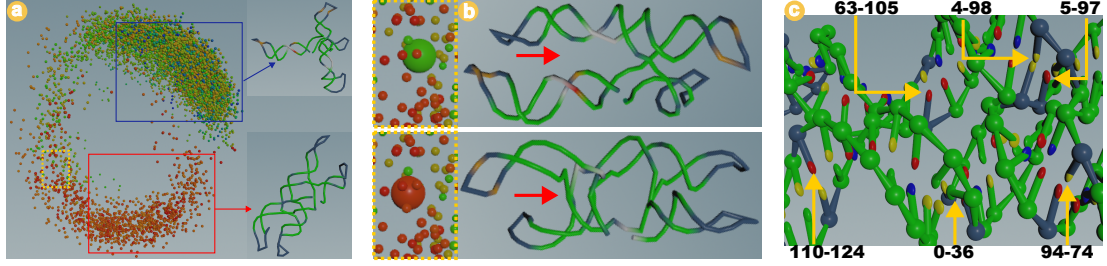


Figure B.2: Case study for an RNA tile design. (a) PCA plot shows two groups of configurations. The yellow dashed rectangle is further zoomed in (b). (b) Two different configurations are located closely in the PCA plot. The difference is indicated by the red arrows. (c) The non-Watson-Crick base pairs in the design are easily identified as they are colored dark blue (indicated by the yellow arrows).

red-green color contrasts for those people with color deficiencies. Figure B.1 shows an example with a different color map that is safe for people with color deficiencies.

B.5 Transitions

Even though all the representations are linked together, it is still important to visualize the transitions (see the supplementary video) between some of them. The users can view the transitions at the beginning of the analysis to understand what each representation is conveying for a specific DNA-nano design and how they are related to each other spatially. We implement these consecutive transitions: *Snake* \triangleright *Schematic3D* \triangleright *Schematic2D* \triangleright *Heatbar*. As all of them are transformed from the CMS of all NTs, we interpolate the CMS values between them to achieve the transition. For the case of large structures, the transitions from *Schematic3D* to the more abstracted ones usually give illegible results if all strands transit together. So, we applied staged transitions by moving one strand after another.

B.6 SynopFrame performance

We record the performance-related number on a Windows 10 desktop (Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz (2 processors), 256GB RAM) with an Nvidia RTX 3090 graphics card. To load and cache the cube structure (in the first case study) with 16,128 NTs, 1,000 frames, 3,076 MB data, SynopFrame takes 63

seconds. After caching into Houdini native binary data, the file size reduces to 494 MB, corresponding to 494 KB per frame. Reloading from the cache takes 6.2 seconds. The frame rate to show all seven views together is 1.65 fps; to show All-NT only is 6.27 fps; to show Schematic3D only is 24.97 fps; to show Progress bar only is 19.75 fps. As analysts very often stop at certain frames and scrutinize the structure, such frame rate is still considered to be interactive.

B.7 The SynopSpace.hb format

For a whole trajectory with many frames, we record the H-bond status code (mentioned in section 6.3) of each NT in each frame as follows. We first record the frame number and then sequentially record the status code for each NT in a new line, with `StatusCode` $\in [-2, 2]$ as an integer to classify the H-bond pairing status. Below we first give a *generic format description* (`*.synospace.hb`)⁶ and then a specific *example* for its use.

```
t = <FrameNumber>

<StatusCode> [Pair ID if StatusCode == 2] # for NT0
<StatusCode> [Pair ID if StatusCode == 2] # for NT1
<StatusCode> [Pair ID if StatusCode == 2] # for NT2
...

t = <FrameNumber>
...
```

An example that shows in Frame number 1000, the first 5 NTs' `StatusCode`, with the 5th's pair (NT ID 7923) recorded:

```
t = 1000
-2
-1
0
```

⁶This format alone triggered a discussion among domain users and has since been integrated by some users into their own analysis approaches: github.com/lorenzo-rovigatti/oxDNA/issues/45

```

1
2 7923
...
t = 2000
...
```

B.8 Case Study 2: An RNA tile design

As oxDNA can simulate RNA designs, we also performed a case study for an RNA tile design to demonstrate that SynopFrame can also work with RNA designs. In an experiment with an RNA tile with 132 NT (one strand) from [83] two different conformations were known to occur. From a simulation at 45 °C, the two groups of conformations are well manifested in the *PCA* plot Figure B.2a. Further examination reveals that H-bond statuses are not always correlated with *PCA* dimension reduction results. For example, frame 259's conformation and frame 7765's are very close to each other in the *PCA* plot. But their H-bond statuses show a big difference in one critical crossover Figure B.2b. This observation raises a caveat that although in general, *PCA* followed by clustering performs well in categorizing the conformations, it cannot reliably capture the subtle H-bond changes that will cause disproportionate effects on the conformation.

The transition period between the RNA tile's two conformations can be easily identified, see the supplementary video. The biggest difference between the two conformations is the loss of the two crossovers at around 1/3 of the structure. So we can attach a highlighter to this helix and then focus on *Schematic2D*, which is much easier to absorb the H-bond status changes. But the *Schematic2D* view first shows the opening of another helix from frame 601 to 607. So it forces the analyst to go back to the 3D structure (*Schematic3D*, *Snake*, *All-NT*) and think about the relation between these two helices. Only after this helix opened, the initially attended helix is then opened at frame 1181 to 1184. The analyst may then proceed to crop out these frames to perform further-detailed analysis, such

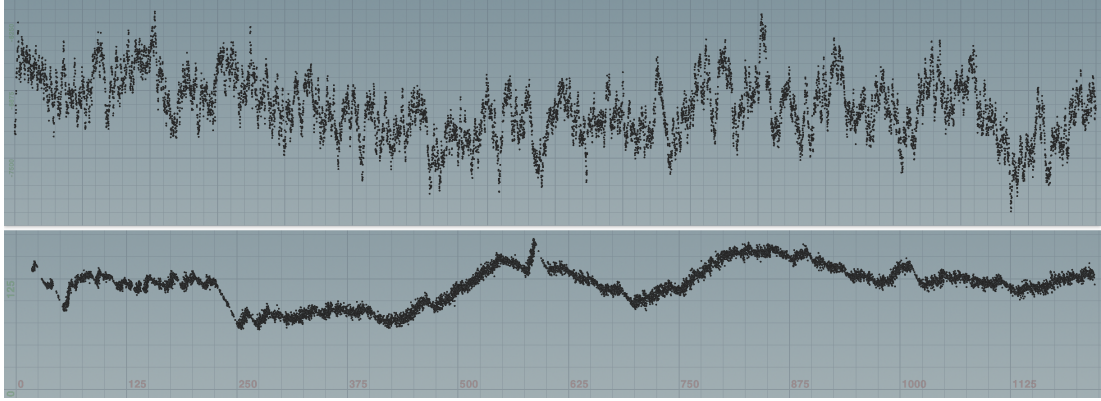


Figure B.3: Statistical scalar plots. The top panel shows an energy versus time plot. The bottom panel shows a force-extension curve.

as distance, angle, and energy distribution.

Looking at the bizarre *Schematic2D* view, one might wonder what those NTs are that are not forming helices. A close examination from the linked view with help of the highlighter reveals that apart from the non-pairing loop on the side of the structure, there are six base pairs that are within the H-bond distance threshold, but their sequences do not obey the Watson–Crick rule Figure B.2c. The designer might then proceed with changing some of the sequences on those base pairs and see how the design behaves.

B.9 Statistical scalar plots

Although SynopFrame mainly explores the abstract views to visualize the MDS trajectory frame-by-frame, it can also be coupled with traditional data analysis plots focusing on the aggregated statistical metrics along the temporal aspect, i. e., generate a scalar to represent each frame and then plot that scalar. Figure B.3 shows two such plots, with the top panel showing the energy versus time and the bottom showing the “force-extension curve” [29] previously used in the domain. The force-extension curve encodes each frame by two scalars, one for the force between two NTs, showing on the Y axis, and the other for the extension, or the distance, between these two NTs.

B.10 Algorithms

Here we detail the algorithms used to perform the transformations needed for the *Schematic3D* representation (section 6.3). With the CMS of all NTs and the topology of the design as input, Algorithm 1 exhaustively checks the conditions that could potentially break a continuous helix and create the primitives for the continuous ones. With the CMS data of a straightened polyline, its pair, and the whole double helix it belongs, Algorithm 2 shifts the two polylines in each helix for a user-defined distance and then evenly space the NTs on each polyline. With the user-specified distance and angle thresholds, Algorithm 3 assigns a *schematic2D_row* value for each polyline so that those with the same value can be then aligned along one straight line.

B.11 User feedback details

As explained in the main paper, we received feedback from our co-author collaborator, from one anonymous respondent, and four signed responses. These latter four were a senior researcher in computational chemistry who focuses on molecular dynamics and bioinformatics, a PhD student in the OxDNA developers group, a PhD student who focuses on wet-lab DNA-nano experiments, and a professional bioinformatician who specializes in wet-lab experiments for DNA origami. Please note that, for completeness, we also provide the video (`Video_for_User_Feedback.mp4`) that we sent to the invited experts as an explanation of our approach, which served as the basis of the following questionnaire, as additional material.

At the end of the appendix we provide the filled-in questionnaire responses from all participants that provided us with feedback. Note that we also provide the video we provided to participants as the basis of their evaluation as additional material, in addition to the actual paper video itself. In total we received 1 anonymous response and 5 signed responses (the latter including our closely collaborating expert who is a co-author). We provide these answers as screenshots

from the online questionnaire tool to show both the stimuli and the specific questions we asked about it, as well as the answer possibilities and the detailed comments that all our participants provided.

We also note that SynopFrame visuals won 2nd place in the Design X Bioinformatics [105] Student Competition. Committee members from the Scripps Research Institute (USA), STUDIO ABOVE&BELOW (UK), the Royal College of Art (UK), and the Tokyo Institute of Technology (Japan) praised our “*highly useful and computationally interesting system to display DNA data in multiple ways.*”

B.12 Comparison to Miao et al.’s DNA origami abstraction space DimSUM

To better illustrate our conceptual extension of past work, in particular the work by Miao et al. [67, 68], we list the fundamental differences between their and our abstraction spaces in Table B.1 (we do not include Miao et al.’s [68] first approach because it only focused on a single visual representations sequence).

Table A.1: Performance results for fitting a single structure. S stands for subunits, A stands for atoms, Res stands for resolution (in Å), Vs stands for voxel size, L stands for surface level threshold, C stands for ChimeraX, D stands for DiffFit, M stands for MarkovFit [3], DC stands for DiffFit corrected by a single automatic ChimeraX fit; G stands for Gain and is D/C for Hit and C/D for Computing time (in seconds).

PDB	EMDB	#S	#A	Res	Vs	L	Hit rate			Computing time			RMSD (Å)			
							C	D	G	C	D	G	M	C	D	DC
6WTI	21897	4	9,980	2.38	1.08	0.7660	0.0	136.8	n/a	150.3	3.8	39.7	1.310	n/a	0.942	0.037
7D8X	30614	4	10,928	2.60	1.08	0.0229	0.0	202.0	n/a	196.0	5.2	37.6	1.960	n/a	0.984	0.014
7SP8	25368	3	6,090	2.70	1.08	5.5755	4.6	188	40.9	130.6	2.6	50.5	1.290	0.996	0.969	0.025
7STE	25426	5	14,249	2.73	0.83	0.0963	14.0	110.4	7.9	806.1	12.1	66.6	1.740	0.062	0.662	0.058
7JPO	22417	5	16,087	3.20	1.07	0.0240	5.4	191.8	35.5	250.7	6.7	37.2	2.540	0.017	0.922	0.015
7PMO	13508	3	10,169	3.60	1.10	0.0068	44.0	195.4	4.4	352.4	4.1	86.7	1.640	0.030	0.907	0.024
6M5U	30093	3	10,549	3.80	1.06	0.0350	0.0	105.0	n/a	162.2	4.1	39.2	2.360	n/a	0.912	0.018
6ME0	9108	3	7,465	3.90	1.06	0.0500	7.4	116.0	15.7	128.2	3.2	40.1	1.940	0.489	0.786	0.488
7MGE	23827	4	9,010	3.94	0.94	0.2550	4.8	123.6	25.8	337.6	4.3	78.1	1.870	0.017	0.819	0.017
High-avg		3.78	10,503	3.21	1.03	n/a	8.9	152.1	21.7	279.3	5.1	52.8	1.850	0.268	0.878	0.077
5NL2	3658	2	4,312	6.60	1.35	0.0297	1.8	163.2	90.7	94.6	2.0	48.0	2.440	0.093	1.124	0.056
7K2V	22647	2	5,717	6.60	1.05	0.0050	49.0	165.6	3.4	240.6	4.1	58.2	25.290	0.338	1.323	0.338
7CA5	30324	2	6,484	7.60	1.06	0.0100	55.8	72.4	1.3	322.6	2.9	110.0	3.290	2.042	1.207	2.042
5VH9	8673	2	22,042	7.70	1.20	0.0074	68.6	158.0	2.3	1147.8	14.1	81.3	0.960	0.085	0.991	0.085
6AR6	8898	2	2,395	9.00	3.00	0.0739	78.0	182.6	2.3	74.9	1.5	49.3	2.200	0.123	2.617	0.117
3J1Z	5450	2	4,586	13.00	2.74	3.8989	138.6	172.2	1.2	64.4	2.0	33.0	32.330	0.396	2.612	0.388
Med-avg		2.00	7,589	8.42	1.73	n/a	65.3	152.3	16.9	324.1	4.4	63.3	11.085	0.513	1.646	0.504
All-avg		3.01	9,337	5.29	1.31	n/a	31.5	152.2	19.8	297.3	4.9	57.0	5.544	0.366	1.185	0.248

Algorithm 1: Create polylines for continuous double helices

Data: CMS of all NTs, topology of the design
Result: Polyline primitives for continuous double helices

```

1 totN = total number of NTs
2 N = 0
3 primList = [N]
4 while N < totN do
5   end5 = the 5' end ID of N
6   if end5 == -1 then
7     CreatePolyline(primList, N)
8   else
9     pair = the pair ID of N
10    strand = the strand ID of N
11    if pair == -1 then
12      CreatePolyline(primList, N)
13    else
14      end5Pair = the pair ID of end5
15      if end5Pair == -1 then
16        CreatePolyline(primList, N)
17      else
18        pairStrand = the strand ID of pair
19        end5PairStrand = the strand ID of end5Pair
20        if pairStrand != end5PairStrand then
21          CreatePolyline(primList, N)
22        else
23          pairEnd3 = the 3' end ID of pair
24          if pairEnd3 != end5Pair then
25            CreatePolyline(primList, N)
26          else
27            push(primList, end5)
28          end
29        end
30      end
31    end
32  end
33 end
34 Function CreatePolyline(primList, N):
35   /* When this function is called, it means the continuity of the
36   helix is broken. */
37   Add a polyline primitive that has all the NTs in the primList
38   primList = [+ + N]
39 return None

```

Algorithm 2: Shift the two polylines in each helix and evenly space NTs on each polyline

Data: User define: *halfPairDistance*, *spacing*
Result: Two parallel polylines with NTs evenly distributed for each continuous double helix

```

/* This algorithm is supposed to be run in parallel for each polyline
*/
1 dir = the white arrow vector in Figure 6.5b
2 primDisplacement = halfPairDistance * normalize(dir)
3 unitBackboneDir = spacing * (the direction along the polyline)
4 pts[] = all the points (NTs) ID on this polyline
5 avgPtnum = average(pts)
   /* The IDs on the same polyline are guaranteed to be consecutive so
   that we can take the average value */
6 for ptnum in pts[] do
7   | offset = ptnum - avgPtnum
8   | newP = polylineCMS + unitBackboneDir * offset + primDisplacement
9   | set the position for ptnum as newP
10 end

```

Algorithm 3: Assign *schematic2D_row* for each polyline)

Data: User define: distanceThreshold, angleThreshold (Figure 6.5d)

Result: *schematic2D_row* assigned for each polyline

```

1 Function PushPrimToDict(prim, primPair, nextRow, rowDict, key):
2   helixInfo = a dictionary to hold the prim's CMS and direction
3   helixArray[] = fetch from rowDict by key or initialize an empty array
4   push(helixArray, helixInfo)
5   dictKey = key if key exists, otherwise nextRow
6   rowDict[dictKey] = helixArray
7   set schematic2D_row for prim and primPair as int(dictKey)
8 return None
9 numprim = total number of polyline primitives
10 schematic2D_row = {}; nextRow = 0
11 for prim = 0; prim < numprim; prim ++ do
12   primPair = the pair ID of prim
13   if prim < primPair then
14     if len(schematic2D_row) > 0 then
15       minDist = 100.0, minAngle = 90.0
16       overlapKey = " - 1"
17       rowKeys[] = keys(schematic2D_row)
18       for key in rowKeys do
19         dict helixArray[] = schematic2D_row[key]
20         Loop through the CMS and direction of each helix in helixArray
21         Calculate the distance and angle value as shown in Figure 6.5d
22         if distance < minDist then
23           minDist = distance
24           minAngle = angle
25           overlapKey = key
26         end
27       end
28       if minDist < distanceThreshold AND minAngle < angleThreshold then
29         /* This means the current helix is overlapping with a row
30         group, so we push it using the overlapKey */
31         PushPrimToDict(prim, primPair,
32           -1, schematic2D_row, overlapKey)
33       else
34         /* This means the current helix is not overlapping with
35         any row group, so we push it using nextRow++ as a new
36         key */
37         PushPrimToDict(prim, primPair,
38           nextRow ++, schematic2D_row, "")
39       end
40     else
41       /* Same as the last push */
42       PushPrimToDict(prim, primPair,
43         nextRow ++, schematic2D_row, "")
44     end
45   end
46 end

```

Table B.1: Comparison between SynopSpace and Miao et al.’s [67] DNA origami abstraction space.

<i>Criterion</i>	Miao et al.’s DimSUM [67]	Our own work
<i>focus of the work</i>	seamless animation between 1D, 2D, and 3D layouts and multiple 3D semantic representations for different levels of detail	comprehensive MDS analysis scenario to identify issues for the wet-lab assembly of previously designed DNA-nano structures
<i>target application</i>	DNA-nano design phase	DNA-nano post-design phase
<i>represent. of layout</i>	1D, 2D, 3D layouts; i. e., a geometric view of possible layouts	sequential, schematic, precise; i. e., a domain-centered view of layouts
<i>represent. of scale</i>	10 named scales , organized based on a perceived level of “concreteness” or visual abstraction	two separate and independent components for a greater flexibility: <ul style="list-style-type: none"> • the four levels of <i>granularity</i>: nucleotide, helix, strand, assembly; also organized based on a perceived level of “concreteness” or visual abstraction • the three–four levels of <i>idiom</i>: bar, snake, geon, and surface; to characterize different visual encodings of a given data component
<i>unique views</i>	<ul style="list-style-type: none"> • DimSUM abstraction view 	<ul style="list-style-type: none"> • Schematic3D (Figure 6.1c, Figure 6.4c) • progress bar (Figure 6.1f, Figure 6.4f) • PCA plot (Figure 6.1g)
<i>represent. of dynamic data character</i>	n/a (only one data view is shown at any given time)	MDS-based animation of assembly that relies on a visual encoding of the H-bond status , observable in multiple points of the abstraction space that are shown in parallel



Deng Luo <xxxxxxx@kaust.edu.sa>

Fwd: testing new molecular fitting technique

XXXXXXXXXX <xxxxxxxxxxx@xxxxxxxx.xx>

Mon, Apr 1, 2024 at 11:38 AM

To: Roden Deng Luo <xxxxxxxx@kaust.edu.sa>

Cc: XXXXXXXXXXXX <xxxxxxxxxxx@xxxxxxxx.xx>

Hi Roden,

Thanks for sending over the MS and the GitHub, and for your demonstration this morning. The demonstration helped see the viability of your program, compared to the quick video I saw last week.

Do you think the workflow of automatic fitting + visual inspection + selection to remove regions in the target volume is the desired form of working? Or do you envision another workflow scenario?

The workflow that you mention through this tool, with an automatic fitting followed by visual inspection, then conversion of the model to a map for deletion of regions in the target volume to find fits to regions that were assigned low-classification scores and hard to find in the viewer is unique and could actually be quite useful.

I think the visual inspection step may be a bit of a bottleneck and potentially including some way of visualizing clusters at a time may be beneficial for rapid assessment of different fits. Perhaps having an alternate view than the interactive table, such as a heatmap (may not be feasible/more confusing though). You could also incorporate separate tables for each of the chains, with separate tabs for each of the tables. This would make it easier to see how each chain fits relatively and would enable researchers to focus on a particular chain of interest, once they've found the best fit for the other chains.

Even without the final step of removing regions from the target volume, I think just the automatic fitting and visual inspection, if implemented in a very user-friendly way, could be a key feature in ChimeraX that becomes a standard in many pipelines.

Additional workflows could be as we discussed, taking a model and automatically creating subdivisions (at the level of domains or secondary structures) and then fitting those automatically to the reference volume, though I understand that is out of the scope of this initial submission.

Whether this tool, DiffFit, could be useful in your work, or is it potentially changing your workflow? If yes, is the change incremental or dramatic? Or is the tool irrelevant/useless/conceptually wrong?

I think DiffFit could become a key implementation into a standard modeling workflow, greatly helping with the initial steps of downloading/opening additional models and having them rapidly fit the target volume to facilitate structural interpretation and comparison. Then, taking it deeper, DiffFit could also provide a way to find fits to more tricky regions of the map through the easy fit-and-subtract feature that you outline.

Additionally, for the regions with unassigned protein density, the ability to rapidly sample a large database of candidate structures could be very valuable, given the massive speed improvements compared to the current SOTA. Combined with the ability to sample sub-regions of the proteins (automatically) given a database of structures (AlphaFold DB), I think it could be quite powerful.

With the speed increases that you mention, I would say that the change relative to the current SOTA is dramatic, and is not irrelevant/useless or conceptually wrong.

Other things you would like to say

Overall I think the DiffFit program appears quite intuitive and easy to use. Perhaps some optimization could be made in the selection of the files/reliance on the subfolders to make it more user-friendly, but that is minor. Additionally, ensuring that the program could run within a reasonable amount of time on a Mac laptop would also be great.

I also don't know how computationally intensive the program is to run, and if it's affected by having many different models/maps open within a ChimeraX session. Often I have quite a few maps/models open, and it would be nice to be

able to run the program from maps/models that are already loaded within the ChimeraX environment, rather than opening a new session each time, though that's just a quality of life improvement.

Also, ChimeraX has a toolshed in which users can install programs directly through the ChimeraX interface. That would enable wider use and ease of access for researchers. [See ISOLDE](#)

Also, while the Cryo-EM field is quite exciting/trendy, your workflow may also work for density maps generated through X-ray crystallography, so that could be a point to consider to increase the breadth of relevance to include more than Cryo-EM.

~

Hope that helps! Would be interested to try it out and hear more about the developments.

Best,

XXXXXXXXXX

[Quoted text hidden]



Deng Luo <xxxxxxxx@kaust.edu.sa>

DiffFit User Feedback

XXXXXXXXXX <xxxxxxxxxx@xxxxxxxxxxxx>
To: Deng Luo <xxxxxxxx@kaust.edu.sa>

Tue, Jul 2, 2024 at 9:31 AM

Hi Roden,

Thanks for showing the demo to me yesterday. It was really more impressive than just watching a video. Here I put all my comments below.

Do you think the workflow of automatic fitting + visual inspection + selection to remove regions in the target volume is the desired form of working? Or do you envision another workflow scenario?

In cryoEM, the structural analysis is a very critical step. For many new users who are not familiar with their own structures, this workflow in DiffFit do really save them a lot of time and efforts to do the structural analysis. We can easily fit the pdb file into the EM map without any extra step and it is integrated into ChimeraX, which is already a very common tool in cryoEM filed, which makes it easier to use.

For the results shown in ChimeraX, I think now the UI is not the best, you are still working on it to make it more simple and user-frendly. Now after the fitting, it will show automaticly the highest score results. You shold consider if you have multiple chains or domains, they can be independent components and fit into the map independently. I think this part needs to be further optimised.

Whether this tool, DiffFit, could be useful in your work, or is it potentially changing your workflow? If yes, is the change incremental or dramatic? Or is the tool irrelevant/useless/conceptually wrong?

Definitely it will be useful for all structural biologist. User can just open the cryoEM map and pdb file, simply click a button (or few buttons), the fitted result will appear in few minutes, which would save a lot of time to roughly align the map and pdb file as a starting point to use fit in map command in Chimera. The function of selection to remove regions is also helpful because we have some cases where you have a quite large map and contains lots of different proteins, you are interested to see whether there are some extra densities, this function will make the task much easier.

Other things you would like to say

I believe DiffFit would really benefit structral biologist for the anlysis process. And of course there are still some more work needs to be done to further optimise. The first thing is the UI integrated in Chimera X should be more user- friendly. It will be better if you can prepare a detailed protocol for how to download, install and test if DiffFit works well. Then also you should make sure that the fitting process does not request intensive computational resources. We do not want to have a high performance workstation for just the visilization step.

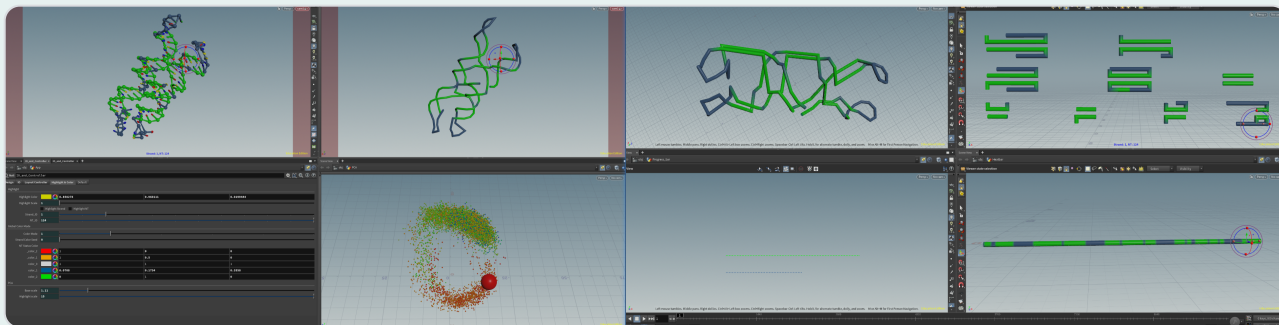
Last, for the function to assign model to extradensity, it would be really helpful. But now this part is not fully implemented, we have to run many steps before going to DiffFit. I would suggest you to furthe optimise this function.

As my role, I am staff seientist in EM group, mainly in responsible for cryoEM. I work as a bridge between the microscope and the users.

Best regards,

XXXXXXXXXX

[Quoted text hidden]



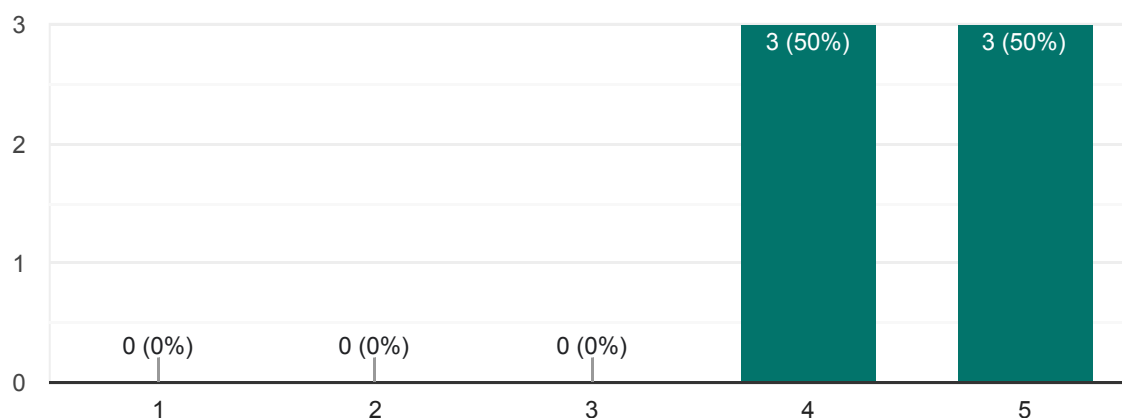
SynopFrame Feedback - 7 Questions (5 min maximum)

6 responses

Feature 1: the linked 3D and 2D views. Useful?

 Copy

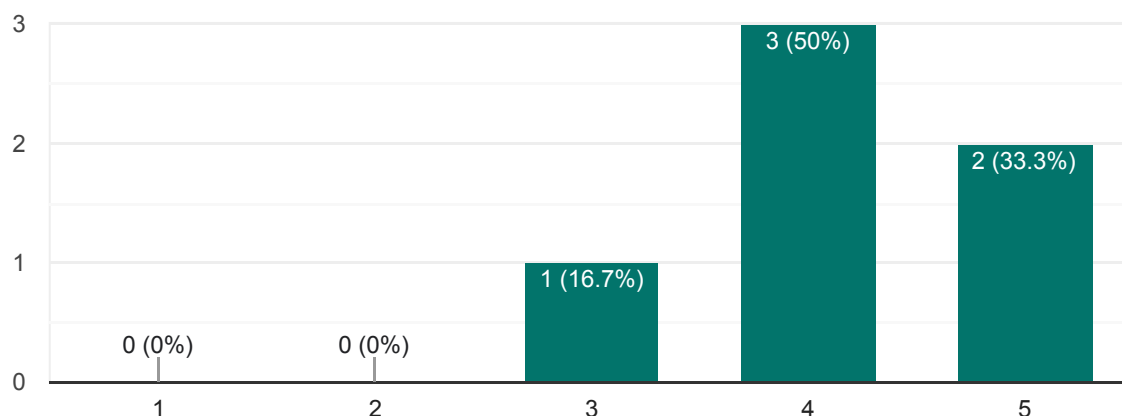
6 responses



Feature 2: the connection between PCA plot and structural representations. Useful?

 Copy

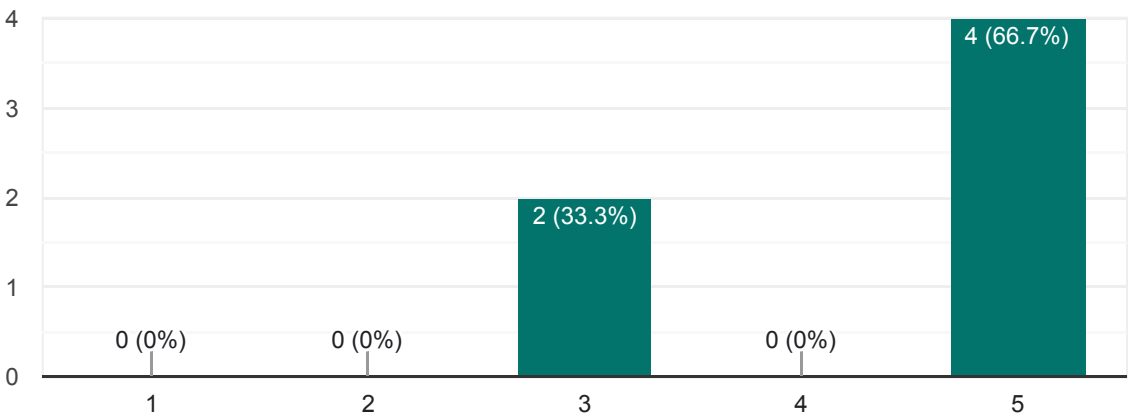
6 responses



Feature 3: the H-bond status coloring. Useful?

 Copy

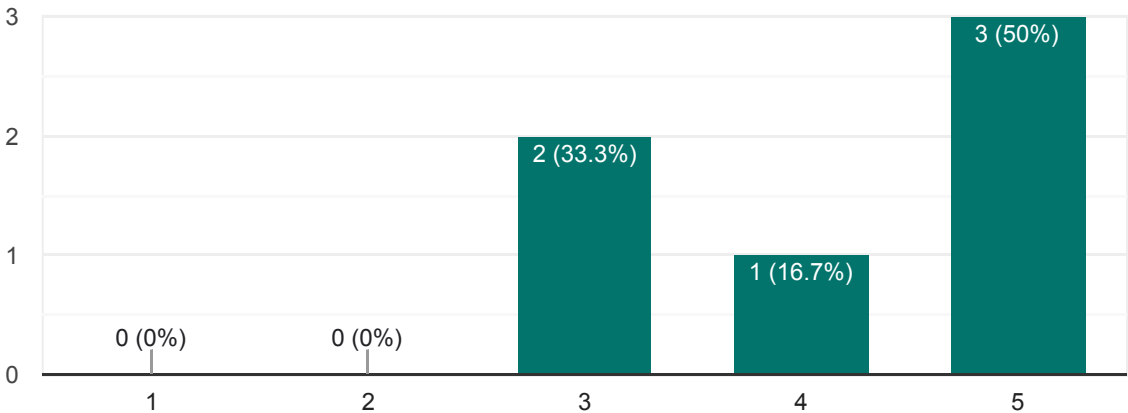
6 responses



Feature 4: the synchronized highlighter across views. Useful?

 Copy

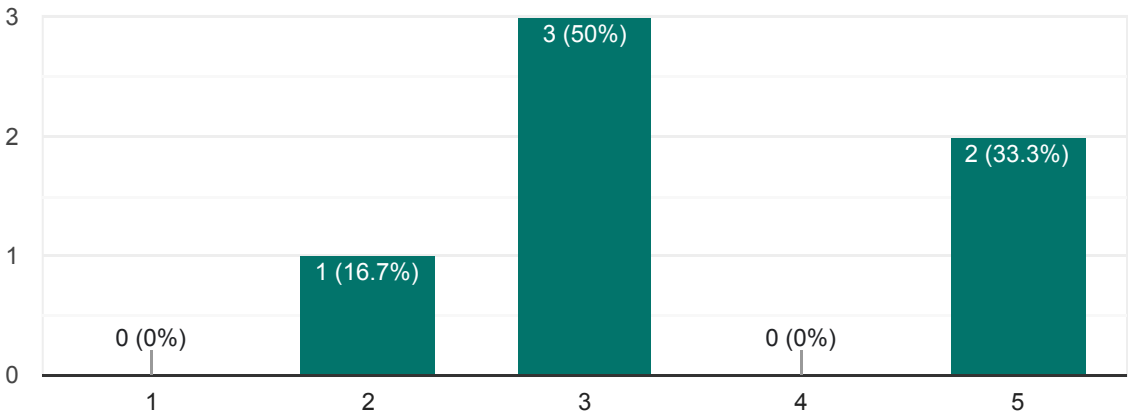
6 responses



Feature 5: the transitions between different views. Useful?

 Copy

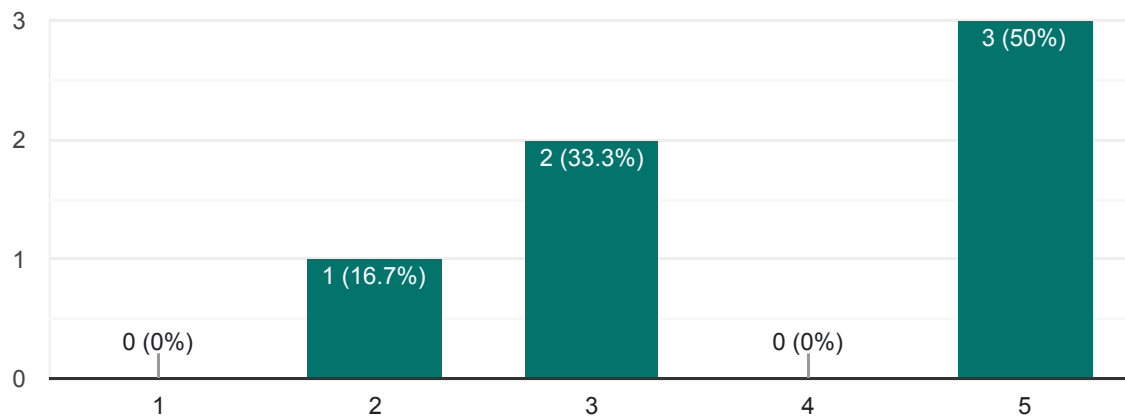
6 responses



Rating 1: SynopFrame will help you understand, communicate, and improve your designs



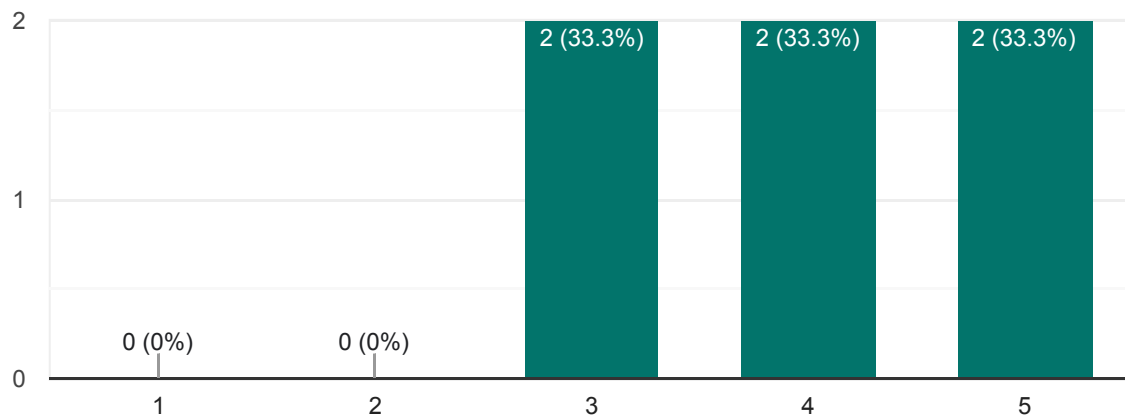
6 responses



Rating 2: overall, how would you rate SynopFrame?



6 responses



General feedback

5 responses

Very effective is to be determined after using the tool.

SynopFrame explores some interesting aspects of oxDNA trajectory visualizations. However it's really impractical to download a huge commercial tool, which is tailored towards video effects production just to have a look at a simulation.

practical

The video is not easy to understand out of context and without additional explanations.

I appreciate that SynopFrame could help me with my designs



Suggestions for improvement

4 responses

- Most parameters which need to be calculated are project specific so adding an interface to easily add new order parameter visualizations might help.
- Along these lines most of the order parameters are 1D
So having 2D / 3D plots of them might show relationships. (similar to the PCA view) but have 3 / 4 (if you count the energy) parameters of the users choice plotted out.
- Exploring the PCA view i was expecting interactive clicking through the coordinates, but somehow this was not implemented , browsing through the PCA space using the trajectory slider makes little sense.

As a user new to the software, a user-friendly interface or tool to import data into the program instead of requiring users to run a few oat analysis would be helpful. The tool could allow users to select and upload their dataset (just oxdna.dat and oxdna.top), and then automatically generate the necessary input files and folder structure required by SynopFrame. This would make it easier for users to get started with the program and increase its accessibility.

Describe the features before demonstrating them. For instance I haven't understood the hydrogen-bond color coding feature and couldn't follow in the video what I was supposed to see.

I think making the packaging/setup for SynopFrame easier might help make it more widely usable - since grad students are mostly the ones using this tool, saving time for them would always be appreciated. I think also being able to easily choose the sequences that require redesign and export them would also help.

Name

4 responses

anonymized

anonymized

anonymized

anonymized



Email

4 responses

anonymized

anonymized

anonymized

anonymized

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms



