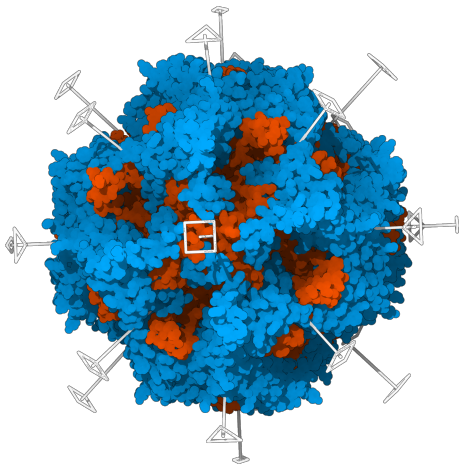
New Challenges in Structural Bioinformatics: When Physics Meets Big Data

Sergei Grudinin

HABILITATION THESIS

HABILITATION À DIRIGER DES RECHERCHES

SERGEI GRUDININ



New Challenges in Structural Bioinformatics: When Physics Meets Big Data

Computer Science and Mathematics Doctoral School (MSTII)
Université Grenoble Alpes

Jury	
Dr. Jessica Andreani	Examinateur
Prof. Jean-Christophe Gelly	Rapporteur
Prof. Mikael Lund	Rapporteur
Prof. Anne Milet	Examinateur
Prof. Michael Nilges	Rapporteur
Prof. Édouard Oudet	Examinateur

Defended on January 23, 2024 in Grenoble, France

Sergei Grudinin: *Habilitation Thesis,* New Challenges in Structural Bioinformatics: When Physics Meets Big Data, © Defended on January 23, 2024 in Grenoble, France

To my beloved family and friends

CONTENTS

FOREWORD

In the last couple of years, we witnessed a revolution in the field of protein structure prediction, with protein models reaching unprecedented levels of near-experimental accuracy [37, 109]. The 50-year-old problem of determining how a single protein folds in three dimensions (3D) seems to be resolved. However, this is not the end of structural bioinformatics and structural biology, it is rather the very beginning of our understanding of how to solve very difficult or even previously thought infeasible biological problems using big data and machine learning (ML) techniques [131]. For example, the next big breakthroughs in the field will be most likely linked to predicting structures of large protein assemblies and their interactions in the living cell, predicting structures and dynamics of macromolecules at physiological conditions, and virtual design of novel proteins and drug-like molecules.

I was lucky to start working in the field of structural bioinformatics during my Ph.D. project supervised by Valentin Gordeliy, Georg Bueldt, and Artur Baumgaertner. I have been at the frontier of big-data discoveries for some time already and witnessed the revolution of protein structure prediction happening right now. This manuscript covers some of the developments I have been working on together with my colleagues and students since my Ph.D. thesis.

I started my independent research using classical techniques fully based on physics and geometry and gradually switched to data-driven approaches. I am very grateful to Gerhard Gompper from Forschungszentrum Jülich, who allowed me to freely work on my project during my first post-doctoral contract. Later, my studies were very much influenced by two researchers I was lucky to work with. The first was my postdoctoral adviser Stephan Redon, a group leader from Inria Grenoble, who later left academia to lead his startup. He shared with me lots of his crazy ideas (and his office!) about transferring developments and algorithms from computer graphics to the modeling of biological objects. It was also thanks to him that I finally learned algorithms from computer science that I had never formally studied, and how to create and manage big software projects in C++. The second one was Dave Ritchie, a group leader from Inria Nancy, whose passion dragged me into the study of polynomial expansions for the description of 3D shapes and interactions between them.

I should add that physics-based approaches, for example, molecular dynamics simulations of proteins and ML would rarely meet together even twenty years ago. In my case, the first critical turn towards ML and the discovery of its power happened around 2010, when a Master's student from MIPT Moscow, Georgy Derevyanko, fulfilling his Master's project in my lab, forced me, despite my initial skepticism, trying the formulation similar to support vector machines (SVM) for the problem of classification of native protein-protein interfaces. Thanks to him, I became confident that convex optimization techniques can be very powerful and helpful in many problems related to structural bioinformatics and can be used together or even instead of more classical approaches. Again, I was lucky to be surrounded by true mathematicians Anatolii Juditsky, Roland Hildenbrandt, and Jerome Malick, who supported me in multiple ways and helped me to find proper optimization formulations for some of my problems.

I shall also confess that my work would not be possible without the help and ideas of my students and colleagues. Chapter 2 heavily relies on the initial formulation of describing molecular shapes with orthogonal polynomials by Dave Ritchie. We extended it for the Hermite polynomials with Georgy Derevyanko and roto-translational correlations with Dmitrii Zhemchuzhnikov. Developments in Chapter 3 would not be possible

without Guillaume Pagès, who worked on describing protein assemblies during his Ph.D. thesis. Guillaume has also created most beautiful illustrations in this thesis. The part on symmetrical protein docking was developed together with Dave Ritchie, who also created the corresponding software package. Chapter 4 uses a part of Alexandre Hoffmann's Ph.D. thesis that we later extended with my colleague Elodie Laine. Chapter 5 presents some developments of my Master's students Maria Garkavenko and Loic Broyer. Finally, Chapter 6 includes contributions from Georgy Derevyanko, who proposed the convex formulation for ML-based potentials and a 3D convolutional network for the assessment of 3D protein models, Maria Kadukova and Georgy Cheremovskiy, who worked on ML for protein-ligand interactions, Petr Popov and Emilie Neveu, who applied ML for protein-protein interactions, Guillaume Pagès and Benoit Charmettant, who extended the 3D CNN framework and Ilia Igashov, Kliment Olechnovic, Nikita Pavlichenko, and Dmitrii Zhemchuzhnikov who worked on different aspects of the graph, and more generally, geometric learning.

1 PUZZLES OF STRUCTURAL BIOLOGY

Structural biology hides many puzzles that seem very difficult to solve by using pure statistical physics approaches. For example, the question of how a protein sequence adopts its 3D shape arose already more than a half-century ago. In the early 1970s, Christian B. Anfinsen, the Nobel Prize Laureate, postulated that at least for a small globular protein at its physiological conditions, its native structure is fully determined by the protein's amino acid sequence [11]. Around the same time, in 1969, Cyrus Levinthal conducted the famous thought experiment, now known as the *Levinthal paradox* [138]. He noted that because of the very big number of degrees of freedom in a polypeptide chain, a protein molecule has an astronomical number of possible conformations. Indeed, according to his estimates, a polypeptide composed of 100 residues will have 99 peptide bonds, and therefore 198 ϕ and ψ dihedral angles (in more realistic folding experiments we have to necessarily include other degrees of freedom!). Assuming an sp3-hybridization of the C_α atom, each of these dihedral angles can be around one of three stable conformations, and thus, the protein's polypeptide chain may adopt as many as $3^{198} \approx 10^{95}$ different states.

Therefore, if a protein were to attain its correctly folded configuration by sequentially sampling all the possible conformations, even at a speed of 1 femtosecond per conformation (the fastest molecular vibration time), it would require a time longer than 10^{80} seconds, much more than the age of the universe (about 4.3610^{17} seconds). The "paradox" consists of the fact that most small proteins fold spontaneously on a microsecond-millisecond time scale. These findings motivated the research community to study computational approaches for protein folding and structure prediction and later led to the establishment of the blind CASP (Critical Assessment of protein Structure Prediction) challenge and a practical solution to the protein structure prediction problem using deep-learning techniques [109].

1.1 EXPERIMENTAL TECHNIQUES AND NEW CHALLENGES

Our knowledge about macromolecular structures and interactions has been mostly gained thanks to numerous experiments in structural biology based on a range of experimental methods. These include X-ray crystallography, Nuclear Magnetic Resonance (NMR), cryo-electron microscopy (cryo-EM), and more. X-ray crystallography is a relatively old and well-established technique [59]. A vast majority of macromolecular structures stored in the Protein Data Bank (PDB) [24] are currently solved with it. However, the structure of molecular crystals may bias our understanding of protein functions and interactions under physiological conditions. Indeed, high-resolution diffraction images can only be obtained from very-well ordered particles in a crystal lattice with minimal motion, typically at low temperatures, and with possibly artificial crystal contacts.

Sometimes what we see in a crystal is different from what we would see in a solution. This has been demonstrated, e.g., in the CASP13 small-angle X-ray scattering (SAXS) data-assisted subchallenge [101]. There, over half of the protein targets (7 out of 12) were found to be in a different architectural conformation than that found in the crystal. Indeed, the solution SAXS profiles did not correspond to those computed from the crystal structures. This discrepancy may suggest non-physiological protein conformations for some of the crystallographic protein structures. SAXS is a very promising

technique in this respect, as it allows rapidly verifying global protein shape in solution under physiological conditions [68].

NMR is another technique that allows studying proteins in solution [273]. Indeed, NMR can be applied to proteins under physiological conditions, it can record high-resolution signals, and the proteins can even be flexible or disordered. The downside of this technique, however, is that it can only study rather short, single-domain proteins.

Larger molecules, including macromolecular complexes without crystal packing, can be studied with cryo-EM [45]. This technique, however, contrary to NMR, is currently only applicable to larger particles, typically, macromolecular complexes. It also requires state-of-the-art and rather expensive hardware. Very recently, cryo-EM has advanced to also reconstruct continuous structural heterogeneity [210, 281].

Most often proteins do not act alone and perform their function via interactions with other molecules. Sometimes they form stable complexes, which can be homo- or heteromeric. They can even be organized in higher-order assemblies. And very often these assemblies follow strict symmetrical principles. Thus, our understanding of these principles will help us to also understand the physics of life and hopefully will pave the way to designing new macromolecular machines [78].

Protein structures under physiological conditions are neither rigid - indeed, proteins often perform their function by changing conformational states, or regulating the amplitude of fluctuations upon binding. Describing and predicting their internal motions will be the next frontier of structural biology and bioinformatics. However, we still have very little high-quality experimental data on protein structural heterogeneity to reliably train transferrable deep-learning models. Therefore, physics-based models and priors for machine-learning models will still be widely used in the future for modeling protein flexibility.

The thesis below describes some of my studies about protein interactions and flexibility. Chapter 2 gives some mathematical preliminaries. Chapter 3 describes several new methods to analyze and predict symmetrical protein assemblies. Chapter 4 introduces our approach to modeling nonlinear protein motions. Chapter 5 describes our procedure to compute small-angle scattering profiles and accordingly optimize molecular shapes. Chapter 6 lists our developments with machine and more recently deep learning. Finally, Chapter 7 summarizes my outlook on the future of structural bioinformatics.

2 POLYNOMIAL EXPANSIONS

This Chapter gives some mathematical preliminaries used later in the Manuscript.

2.1 SPHERICAL HARMONICS AND SPHERICAL BESSEL TRANSFORM

Spherical harmonics are complex functions defined on the surface of a unit sphere that constitute a complete set of orthonormal functions and, thus, an orthonormal basis. They are generally defined as

$$Y_l^m(\Omega) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{im\psi}, \tag{1}$$

where $0 \leq \theta \leq \pi$ is the azimuthal angle of point Ω, $0 \leq \psi \leq 2\pi$ is its polar angle, and $P_l^m(\cos\theta)$ are associated Legendre polynomials with indices l and m referred to as the degree and the order, respectively. As mentioned above, by definition these functions are orthonormal,

$$\int_{4\pi} Y_l^m(\Omega)\overline{Y_{l'}^{m'}}(\Omega)d\Omega = \delta_{ll'}\delta_{mm'}, \tag{2}$$

where the second function $\overline{Y_{l'}^{m'}}(\Omega)$ is complex-conjugated, and δ_{ij} is the Kronecker delta. Any square-integrable function $f(\vec{r}) : \mathcal{R}^3 \to \mathcal{C}$ can be expanded in spherical harmonics as

$$f(\vec{r}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{+l} f_l^m(r) Y_l^m(\Omega_r). \tag{3}$$

Reversely, the expansion coefficients $f_l^m(r)$ can be determined as

$$f_l^m(r) = \int_{4\pi} f(\vec{r})\overline{Y_l^m(\Omega_r)}d\Omega_r. \tag{4}$$

Spherical Bessel transform (SBT, sometimes referred to as spherical Hankel transform) of order l computes Fourier coefficients of spherically symmetric functions in 3D,

$$F_l(\rho) = \mathrm{SBT}_l(f(r)) = \int f(r) j_l(\rho r) r^2 dr, \tag{5}$$

where ρ is the reciprocal radius, and $j_l(r)$ are spherical Bessel functions of order l. The inverse transform has the following form,

$$f(r) = \mathrm{SBT}_l^{-1}(F_l(\rho)) = \frac{2}{\pi} \int F_l(\rho) j_l(\rho r) \rho^2 d\rho. \tag{6}$$

Spherical Bessel functions of the same order are orthogonal with respect to the argument,

$$\int_0^{\infty} j_l(\rho_1 r) j_l(\rho_2 r) r^2 dr = \frac{\pi}{2\rho_1\rho_2} \delta(\rho_1 - \rho_2). \tag{7}$$

2.2 PLANE WAVE EXPANSION

The plane wave expansion is the decomposition of a plane wave into a linear combination of spherical waves. It is very useful when changing the basis from Cartesian to spherical coordinates, or when decoupling a function with respect to its arguments. According to the spherical harmonic addition theorem, a plane wave can be expressed through spherical harmonics and spherical Bessel functions,

$$e^{i\vec{\rho}\vec{r}} = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^{l} i^l j_l(\rho r) Y_l^m(\Omega_\rho) \overline{Y_l^m(\Omega_r)}. \tag{8}$$

2.3 3D FOURIER TRANSFORMS IN SPHERICAL COORDINATES

The 3D Fourier transform of a function $f(\vec{r})$ is defined as

$$F(\vec{\rho}) = \int_{\mathbb{R}^3} f(\vec{r}) \exp^{i\vec{r}\vec{\rho}} d\vec{r}. \tag{9}$$

The spherical harmonics expansion of this transform has the following form,

$$F_l^m(\rho) = \int_{4\pi} F(\vec{\rho}) Y_l^m(\Omega_\rho) d\Omega_\rho. \tag{10}$$

The Fourier spherical harmonics expansion coefficients relate to the real-space spherical harmonics coefficients through SBT,

$$F_l^m(\rho) = 4\pi(-i)^l \mathrm{SBT}_l(f_l^m(r)). \tag{11}$$

This equation can be also rewritten as follows,

$$F_l^m(\rho) = 4\pi(-i)^l \int_{\mathbb{R}^3} f(\vec{r}) j_l(r\rho) \overline{Y_l^m(\Omega_r)} d\vec{r}. \tag{12}$$

2.4 TRANSLATION OPERATOR

Let us translate a 3D function $f(\vec{r})$ along the z-axis by an amount Δ. The expansion coefficients of a translated function will be

$$[F_z]_l^m(\rho) = \int F(\vec{\rho}) e^{-i\Delta\vec{\rho}.\vec{e_z}} \overline{Y_l^m(\Omega)} d\Omega. \tag{13}$$

Using the plane-wave expansion and triple spherical harmonics integrals defined through Slater coefficients $c^{l_2}(l, m, l_1, m_1)$,

$$c^{l_2}(l, m, l_1, m_1) = \int_{4\pi} \overline{Y_l^m(\Omega)} Y_{l_1}^{m_1}(\Omega) Y_{l_2}^{m-m_1}(\Omega) d\Omega, \tag{14}$$

we obtain

$$\begin{aligned}
[F_z]_l^m(\rho) &= \sum_{p=0}^{\infty} \sum_{l'=\max(|l-p|,|m|)}^{l+p} i^p j_p(\rho\Delta) F_{l'}^m(\rho) 4\pi \sqrt{\frac{2p+1}{4\pi}} \int_{4\pi} Y_l^m(\Omega) Y_p^0(\Omega) \overline{Y_{l'}^m(\Omega)} d\Omega \\
&= \sum_{p=0}^{\infty} \sum_{l'=\max(|l-p|,|m|)}^{l+p} i^p j_p(\rho\Delta) F_{l'}^m(\rho) 4\pi \sqrt{\frac{2p+1}{4\pi}} c^p(l', m, l, m).
\end{aligned} \tag{15}$$

Changing the summation order and introducing the maximum expansion order L, we arrive at

$$[F_z]_l^m(\rho) = \sum_{l'=|m|}^{L} T_{l,l'}^m(\rho,\Delta) F_{l'}^m(\rho), \tag{16}$$

where

$$T_{l,l'}^m(\rho,\Delta) = \sum_p i^p j_p(\rho\Delta) 4\pi \sqrt{\frac{2p+1}{4\pi}} c^p(l',m,l,m). \tag{17}$$

2.5 ROTATION OF SPHERICAL HARMONICS AND WIGNER D-MATRICES

The Wigner D-matrices $\mathbf{D}_l$ are the irreducible representations of SO(3) that can be applied to spherical harmonics functions to express the rotated functions with tensor operations on the original ones,

$$Y_l^m(\Lambda\Omega) = \sum_{m'=-l}^{l} D_{mm'}^l(\Lambda) Y_l^{m'}(\Omega), \tag{18}$$

where $\Lambda \in \mathrm{SO}(3)$.

2.6 ORTHOGONALITY OF WIGNER D-MATRICES

Let us consider the *normalized* orthogonality relation of the Wigner rotation matrices $D_{mk'}^j$

$$\left\langle D_{m'k'}^{j'}(\alpha,\beta,\gamma)^* D_{mk}^j(\alpha,\beta,\gamma) \right\rangle_{\Lambda\equiv\{\alpha,\beta,\gamma\}} \equiv \frac{\int_0^{2\pi} d\alpha \int_0^{\pi} d\beta \sin\beta \int_0^{2\pi} d\gamma \; D_{m'k'}^{j'}(\alpha,\beta,\gamma)^* D_{mk}^j(\alpha,\beta,\gamma)}{\int_0^{2\pi} d\alpha \int_0^{\pi} d\beta \sin\beta \int_0^{2\pi} d\gamma} \tag{19}$$

$$= \frac{\frac{8\pi^2}{2j+1}\delta_{m'm}\delta_{k'k}\delta_{j'j}}{8\pi^2} = \frac{1}{2j+1}\delta_{m'm}\delta_{k'k}\delta_{j'j}. \tag{20}$$

Thus, the following angular average will simplify according to

$$\left\langle \left[\sum_{m_1=-l}^{l} \sum_{l_1=|m_1|}^{L} \sum_{m_2=-l_1}^{l_1} D_{mm_1}^l(\Lambda_2) T_{l,l_1}^{m_1}(q\Delta) D_{m_1m_2}^{l_1}(\Lambda_1) O_{l_1m_2}(q) \right] \tag{21}$$

$$\left[\sum_{m_1'=-l}^{l} \sum_{l_1'=|m_1'|}^{L} \sum_{m_2'=-l_1'}^{l_1'} D_{mm_1'}^{l*}(\Lambda_2) T_{l,l_1'}^{m_1'*}(q\Delta) D_{m_1'm_2'}^{l_1'*}(\Lambda_1) O_{l_1'm_2'}^*(q) \right] \Bigg\rangle_{\Lambda_1\Lambda_2} \tag{22}$$

$$= \frac{1}{1+2l} \sum_{m_1=-l}^{l} \sum_{l_1=|m_1|}^{L} \sum_{m_2=-l_1}^{l_1} \frac{1}{1+2l_1} |T_{l,l_1}^{m_1}(q\Delta)|^2 |O_{l_1m_2}(q)|^2. \tag{23}$$

2.7 ORTHOGONALITY OF SMALL-D WIGNER MATRICES

Similarly, we can also consider the *normalized* orthogonality relation for the real-valued small-d Wigner rotation matrices $d_{mk'}^j$

$$\left\langle d_{mk}^{j'}(\beta) d_{mk}^j(\beta) \right\rangle_\beta \equiv \frac{\int_0^{\pi} d\beta \sin\beta \; d_{mk}^{j'}(\beta) d_{mk}^j(\beta)}{\int_0^{\pi} d\beta \sin\beta} = \frac{\frac{2}{2j+1}\delta_{j'j}}{2} = \frac{1}{2j+1}\delta_{j'j}. \tag{24}$$

Similarly,

$$\left\langle d^l_{00}(\beta) \right\rangle_\beta \equiv \frac{\int_0^\pi d\beta \sin\beta d^l_{00}(\beta)}{\int_0^\pi d\beta \sin\beta} = \frac{2\delta_l}{2} = \delta_l. \tag{25}$$

2.8 LINEAR WIGNER AVERAGES

First of all, we consider singular-term averages of the form

$$\left\langle D^j_{mk}(\alpha,\beta,\gamma) \right\rangle_{\Lambda\equiv\{\alpha,\beta,\gamma\}} \equiv \frac{\int_0^{2\pi} d\alpha \int_0^\pi d\beta \sin\beta \int_0^{2\pi} d\gamma\ D^j_{mk}(\alpha,\beta,\gamma)}{\int_0^{2\pi} d\alpha \int_0^\pi d\beta \sin\beta \int_0^{2\pi} d\gamma} = \frac{8\pi^2\delta_m\delta_k\delta_j}{8\pi^2} = \delta_m\delta_k\delta_j. \tag{26}$$

Cross-terms of the form

$$\frac{1}{4\pi}\sum_{l=0}^{L}\sum_{m=-l}^{l}\left\langle \left[\sum_{m_1=-l}^{l}\sum_{l_1=|m_1|}^{L}\sum_{m_2=-l_1}^{l_1} D^l_{mm_1}(\Lambda_2)T^{m_1}_{l,l_1}(q\Delta)D^{l_1}_{m_1m_2}(\Lambda_1)O_{l_1m_2}(q)\right] O^*_{lm}(q)\right\rangle_{\Lambda_1\Lambda_2\Delta} \tag{27}$$

$$= \frac{1}{4\pi}|O_{00}(q)|^2\langle T^0_{0,0}(q\Delta)\rangle_\Delta. \tag{28}$$

thus reduce to a single value. Slater integrals of zero angular oder equal to

$$c^p(0,0,0,0) = \frac{\delta_p}{2\sqrt{\pi}}. \tag{29}$$

Therefore, zero-order translation matrix elements are

$$T^0_{0,0}(\rho\Delta) = j_0(\rho\Delta). \tag{30}$$

And radial averages for a given *fractal dimension* d will be

$$\langle j_0(\rho\Delta)\rangle_{\Delta,d} = \frac{\int_{r_{\min}}^{r_{\max}} j_0(\rho r) r^{d-1} dr}{\int_{r_{\min}}^{r_{\max}} r^{d-1} dr}, \tag{31}$$

thus

$$\langle j_0(\rho\Delta)\rangle_{\Delta,d=1} = \frac{\mathrm{Si}(\rho r_{\max}) - \mathrm{Si}(\rho r_{\min})}{\rho(r_{\max} - r_{\min})} \tag{32}$$

$$\langle j_0(\rho\Delta)\rangle_{\Delta,d=2} = \frac{2[\cos(\rho r_{\min}) - \cos(\rho r_{\max})]}{\rho^2(r^2_{\max} - r^2_{\min})} \tag{33}$$

$$\langle j_0(\rho\Delta)\rangle_{\Delta,d=3} = \frac{3[\sin(\rho r_{\max}) - \rho r_{\max}\cos(\rho r_{\max}) - \sin(\rho r_{\min}) + \rho r_{\min}\cos(\rho r_{\min})]}{\rho^3(r^3_{\max} - r^3_{\min})} \tag{34}$$

More general cross-terms will be also reduced to a single value,

$$\frac{1}{4\pi}\sum_{l=0}^{L}\sum_{m=-l}^{l}\left\langle \left[\sum_{m_1=-l}^{l}\sum_{l_1=|m_1|}^{L}\sum_{m_2=-l_1}^{l_1} D^l_{mm_1}(\Lambda_2)T^{m_1}_{l,l_1}(q\Delta)D^{l_1}_{m_1m_2}(\Lambda_1)O_{l_1m_2}(q)\right] \tag{35}$$

$$\left[\sum_{m'_1=-l}^{l}\sum_{l'_1=|m'_1|}^{L}\sum_{m'_2=-l'_1}^{l'_1} D^{l*}_{mm'_1}(\Lambda'_2)T^{m'_1*}_{l,l'_1}(q\Delta')D^{l'_1*}_{m'_1m'_2}(\Lambda'_1)O^*_{l'_1m'_2}(q)\right]\right\rangle_{\Lambda_1\Lambda_2\Delta\Lambda'_1\Lambda'_2\Delta'} \tag{36}$$

$$= \frac{1}{4\pi}|O_{00}(q)|^2\langle T^0_{0,0}(q\Delta)\rangle_\Delta\langle T^0_{0,0}(q\Delta')\rangle_{\Delta'} = \frac{1}{4\pi}|O_{00}(q)|^2\langle T^0_{0,0}(q\Delta)\rangle^2_\Delta. \tag{37}$$

2.9 HERMITE FUNCTIONS

There are multiple ways to analytically rotate and translate functions defined in 3D. Another way of doing this will be 3D Hermite decomposition. Orthogonal Hermite function of order n is defined as,

$$\psi_n(x;\lambda) = \frac{\sqrt{\lambda}}{\sqrt{2^n n!\sqrt{\pi}}} \exp(-\frac{\lambda^2 x^2}{2}) H_n(\lambda x), \tag{38}$$

where $H_n(x)$ are the Hermite polynomials of order n and λ is the scaling parameter. These functions form an orthonormal basis set in $L^2(\mathbb{R})$. The 3D orthogonal Hermite functions can be composed as follows,

$$\psi_{n,l,m}(x,y,z;\lambda) = \psi_n(x;\lambda)\psi_l(y;\lambda)\psi_m(z;\lambda). \tag{39}$$

This composition forms an orthonormal basis set in $L^2(\mathbb{R}^3)$. A 3D function $f(x,y,z)$ represented as a band-limited expansion in this basis reads

$$f(x,y,z) = \sum_{i=0}^{N} \sum_{j=0}^{N-i} \sum_{k=0}^{N-i-j} \hat{f}_{i,j,k} \psi_{i,j,k}(x,y,z;\lambda). \tag{40}$$

2.10 LAPLACIAN FILTER IN THE HERMITE BASIS

The Cartesian expression of the polynomial basis may have multiple advantages over the spherical expression. For example, in the Hermite basis, the Laplacian filter has a particularly simple form. Using the well-known recurrence relation for the derivatives of the Hermite functions, we can easily derive the following relation for the second derivative of a 1D basis function:

$$\frac{d^2}{dx^2}\psi_n(x;\lambda) = \frac{\lambda^2}{2}\left(\sqrt{n(n-1)}\psi_{n-2}(x;\lambda) + (2n+1)\psi_n(x;\lambda) + \sqrt{(n+1)(n+2)}\psi_{n+2}(x;\lambda)\right). \tag{41}$$

A similar relationship holds for the coefficients of the decomposition,

$$\hat{h}''_n = \frac{\lambda^2}{2}\left(\sqrt{n(n-1)}\hat{h}_{n-2} + (2n+1)\hat{h}_n + \sqrt{(n+2)(n+1)}\hat{h}_{n+2}\right), \tag{42}$$

where $\hat{h}_n$ and $\hat{h''}_n$ are the n-th order decomposition coefficients of the original basis and its Laplacian representation, respectively. For $n < 0$ and $n > N$ we let $\hat{h}_n = 0$ and $\hat{h''}_n = 0$. Due to the properties of the Laplace operator and the 3D Hermite decomposition, the contribution of the derivatives along each axis are additive. The derivation of the formula for the 3D decomposition derivative is straightforward and we omit it for brevity.

2.11 ROTATION OF THE HERMITE DECOMPOSITION

Following Park et al. [188], who presented a method to perform an in-plane rotation of a 2D orthogonal Hermite decomposition, we extended id it for the 3D case [54]. Let us first consider the decomposition of a 2D function into a 2D orthogonal Hermite function basis,

$$f(x,y) = \sum_{n=0}^{N} \sum_{m=0}^{N-m} \hat{f}_{n,m} \psi_n(x;\lambda)\psi_m(y;\lambda). \tag{43}$$

The decomposition of a function $f^{\theta}(x,y)$ rotated clock-wise by an angle θ reads

$$f^{\theta}(x,y) = \sum_{m=0}^{N}\sum_{k=0}^{m}(\sum_{n=0}^{m}\hat{f}_{n,m-n}S_{k,n}^{m})\psi_k(x;\lambda)\psi_{m-k}(y;\lambda), \tag{44}$$

where coefficients $S_{k,n}^{m}$ are computed using the following recurrent formulas [188],

$$\begin{aligned}
S_{q,n}^{m+1} &= \sqrt{\frac{n}{m-q+1}}\sin(\theta)S_{q,n-1}^{m} + \sqrt{\frac{m-n+1}{m-q+1}}\cos(\theta)S_{q,n}^{m} \\
S_{q,0}^{m+1} &= \sqrt{\frac{m+1}{m-q+1}}\cos(\theta)S_{q,0}^{m}. \\
S_{m+1,n}^{m+1} &= \sqrt{\frac{n}{m+1}}\cos(\theta)S_{m,n-1}^{m} - \sqrt{\frac{m-n+1}{m+1}}\sin(\theta)S_{m,n}^{m} \\
S_{m+1,0}^{m+1} &= -\sin(\theta)S_{m,0}^{m}
\end{aligned}$$

The key idea that allows to generalize these formulas to a 3D decomposition is that we can factorize a rotation in 3D space into 3 independent in-plane rotations around three different axes, and then rotate each 2D decomposition using eq. 44. Let us consider the following 3D decomposition,

$$f(x,y,z) = \sum_{n=0}^{N}\psi_n(x;\lambda)\sum_{m=0}^{N-n}\sum_{l=0}^{N-m-n}\hat{f}_{n,m,l}\psi_m(y;\lambda)\psi_l(z;\lambda). \tag{45}$$

If we rotate this decomposition about x axis, this rotation will be equivalent to N rotations of different 2D decompositions in the yz-plane,

$$f_n(y,z) = \sum_{m=0}^{N-n}\sum_{l=0}^{N-m-n}\hat{f}_{n,m,l}\psi_m(y;\lambda)\psi_l(z;\lambda). \tag{46}$$

This observation means that in order to perform such rotation, we need to recompute rank-3 tensor of coefficients $\hat{f}_{n,m,l}$ slice by slice N times using eq. 44. Each rotation of the coefficients in one plane corresponds to a multiplication of these coefficients with a rotation matrix. Therefore, a 3D rotation defined with three Euler angles is equivalent to three sequential rotations of coefficients in three planes.

2.12 TRANSITION FROM THE HERMITE TO THE FOURIER BASIS

In order to perform a fast convolution of two 3D signals, we proposed to convert the decomposition coefficients from the Hermite basis into the Fourier basis [54]. This allows using the fast convolution algorithm based on the Fourier convolution theorem. Consider the decomposition of a function $f(\mathbf{r})$ in the 3D Hermite basis with the decomposition coefficients $\hat{f}_{i,j,k}$ (eq. 40). Orthogonal Hermite functions are the eigenfunctions of the continuous Fourier transform,

$$\int \psi_n(x;\lambda)e^{-2\pi i\omega x}\,dx = (-i)^n\psi_n(\omega;\frac{2\pi}{\lambda}) \equiv \tilde{\psi}_n(\omega;\lambda), \tag{47}$$

where ω is the frequency in the reciprocal space. In order to compute Fourier coefficients of $f(\mathbf{r})$ up to order M, we first compute the Fourier transforms of the basis functions $\psi_i(x;\lambda)$, $\psi_j(y;\lambda)$, and $\psi_k(z;\lambda)$ using eq. 47. After, we substitute these coefficients into eq. 40 and obtain the following expression for $\tilde{f}_{l,m,n}$, the Fourier coefficients of $f(\mathbf{r})$,

$$\tilde{f}_{l,m,n} = \frac{1}{L_xL_yL_z}\sum_{i=0}^{N}\sum_{j=0}^{N-i}\sum_{k=0}^{N-i-j}\hat{f}_{i,j,k}\tilde{\psi}_i(\frac{l}{L_x};\lambda)\tilde{\psi}_j(\frac{m}{L_y};\lambda)\tilde{\psi}_k(\frac{n}{L_z};\lambda). \tag{48}$$

These values can be computed in $O(M^3\cdot N + M^2\cdot N^2 + M\cdot N^3)$ steps [54].

3

SYMMETRICAL ASSEMBLIES AND ANALYSIS OF PROTEIN COMPLEXES

Large macromolecular machines are very often symmetric. This symmetry is not occasional but is linked with their structure, function, and evolution. I have been interested in multiple aspects of *macromolecular symmetry* – how symmetrical complexes can be efficiently analyzed, what is the physics of the formation of the assemblies, and how one can efficiently predict molecular complexes under symmetry constraints. Our approach is generally based on the analysis of the correspondence between molecular subunits $A(\vec{r})$ related by spatial transformations, which also include point-group symmetry operators $\hat{R}_y(\omega)$. This correspondence can be formally expressed by the following equation,

$$\hat{T}_z(D)\hat{R}(\alpha,\beta,\gamma)A(\vec{r}) \longleftrightarrow \hat{R}_y(\omega)\hat{T}_z(D)\hat{R}(\alpha,\beta,\gamma)A(\vec{r}), \tag{49}$$

where $\hat{T}_z(D)$ and $\hat{R}(\alpha,\beta,\gamma)$ are the rigid-body translation and rotation operators, correspondingly, applied to the assembly subunits. The correspondence sign $\longleftrightarrow$ can minimize the geometrical mismatch, maximize the shape complementarity, etc., depending on a particular application.

Using this formalism, we proposed an interactive modeling tool under symmetry constraints [85], studied efficient methods to compute interactions between rigid objects [13], and then developed a very efficient tool for the analytical analysis of symmetries and pseudo-symmetries in molecular complexes [183, 185, 186]. The latter method has already been transferred to the PDBe web-based resource and was also used in the assessment of multimeric submissions for the international protein structure prediction challenge CASP starting from Round 13. On our side, it allowed us to study the spatial organization of large molecular complexes on the PDB-wise level.

We have also used this formalism for symmetry-assisted protein docking under analytical symmetry constraints of any point-group symmetry [217]. We demonstrated that the above rigid-body correspondence equation could be expressed in the Fourier space in spherical coordinates using the Fourier correlation theorem,

$$S(\omega; D; \alpha, \beta, \gamma) = \sum_{nlmp} e^{-i(p-m)\alpha} d^{(l)}_{mp}(\omega) A'_{nlp} A''^{*}_{nlm}, \tag{50}$$

with $d^{(l)}_{mp}(\omega)$ being the small Wigner rotation matrix, and A'_{nlp} and A''_{nlm} precomputed expansion coefficient of a shape $A(\vec{r})$. This allowed us to accelerate the exhaustive search of positioning subunits in a symmetric assembly using the fast Fourier transform. Currently, I am extending this formalism for modeling protein assemblies in crowded cell environments [260].

I have been studying how *3D molecular shapes can be compared* efficiently. This led me to discoveries using the *quaternion-based arithmetic*, and, more generally, *geometric algebra*. For example, with my student Petr Popov, we demonstrated that the 2-norm in the Cartesian space between conformations of a rigid molecule could be seen as a quadratic form of the following shape,

$$RMSD^2(\omega, \vec{n}, \vec{t}) = \sin\frac{\omega}{2}\,\frac{4}{N}\vec{n}^T\mathbf{I}\vec{n} + \vec{t}^2, \tag{51}$$

where a molecule with N atoms and inertia tensor $\mathbf{I}$ is rotated by an angle ω about a unit axis $\vec{n}$ and translated by a vector $\vec{t}$ [204]. This allowed us to construct very efficient algo-

rithms for *constant-time* comparison of molecular shapes. We then demonstrated multiple applications and extensions of this equation for docking of trimeric assemblies [206], comparison of flexible shapes [177], comparison of symmetrical assemblies [183, 186], equidistant rigid-body assembly [203] and more. Currently I am reusing this equation for modeling protein assemblies in crowded cell environments. The proof-of-concept study has just been published with my US colleagues Ilya Vakser and Eric Deeds [260].

Another way to compare 3D shapes would be to use their representation using compact *polynomial expansions*. This interest led me to the development of several shape-matching algorithms using novel polynomial forms, i.e., using *orthogonal Hermite polynomials* [54], *Spherical Harmonics* [176, 217], and *classical Fourier functions* extended to higher-order correlations [95]. Currently, we are developing even more efficient ways to encode molecular shapes in 3D [278].

3.1 INTRODUCTION

Symmetrical protein complexes are very common in nature [186], and many of these are deposited to the Protein Data Bank (PDB) [219]. Indeed, it appears that symmetrical assemblies have many advantages compared to individual proteins [142, 143] and thus many of these have been selected during evolution. Thus, there is a considerable interest in studying the structures and mechanisms of formation of symmetric assemblies [4, 10, 141, 154, 217, 229]. In particular, it has been demonstrated that molecular symmetries are important for evolution [143, 230], stability [29], and folding and function [77]. As function of proteins is very often determined by their structure, it appears that complex function requires complex structures [142, 143]. High-order symmetries are thus essential to build large and complex protein assemblies. In particular, dihedral and cubic groups are overrepresented among large protein assemblies with some specific structural functions, for example those of viral capsids. Also, high-order symmetry drastically reduces the complexity of *de novo* design of self-assembling nanomaterials [20, 97, 122, 123].

Although many symmetrical complexes have been solved by X-ray crystallography and cryo-electron microscopy, this can often be a difficult and time-consuming process, and it would be useful to be able to generate high quality candidate complex structures for use as templates in molecular replacement (MR) techniques [174, 221], to provide angular parameters for locked MR search functions [257], or to dock high resolution structural models into low resolution cryo-EM density maps [220], for example. From a protein design point of view, it would also be very useful to be able to predict computationally whether or not a given monomer might self-assemble into a symmetrical structure [98].

Also, the growing amount of data from constantly solved structures of macromolecules together with even bigger amount of data obtained with protein structure prediction methods and molecular dynamics simulations require fast and robust computational tools for the processing of these data. For example, some tools have been developed to detect and assess internal cyclic symmetries, based either on protein sequence [171], structure [42, 173], or both [232]. All these have a common idea of comparing a protein structure with a rotated version of itself. Another set of methods for the continuous chirality and symmetry analysis has been developed by David Avnir and colleagues [60, 196, 197] and also by Michel Petitjean [191], however these do not seem computationally suitable for processing large amounts of macromolecular data, specifically those from PDB. On the other hand, determining a symmetry group of a molecular assembly, finding its axes of symmetry, and assessing the quality of this symmetry are the essential steps in analysis of structural molecular data. For example, a basic analysis method has been proposed by Emmanuel Levy [142], but this is not fully satisfying due to its

limited precision imposed by a set of discretely chosen axes with about 6 degrees of angular step, which results in total of about 600 axes. Also, this method is significantly more time consuming compared to the one presented below.

Regarding symmetry-assisted docking, in the last few years, several *ab initio* protein-protein docking programs such as MolFit [22], ClusPro [48], M-Zdock [193], and Symm-Dock [223] have been adapted to apply various geometric filtering constraints to extract approximately symmetrical pair-wise docking orientations. Symmetry-constraint protocols may be applied to refine the coordinates of a given symmetric structure using RosettaDock [9]. The Haddock docking engine allows up to six distance restraints to be defined when refining oligomeric complexes with certain cyclic or dihedral symmetries [116]. However, to our knowledge, we developed the first *ab initio* docking algorithm which can automatically generate perfectly symmetrical protein complexes for arbitrary point group symmetry types.

In order to build symmetrical protein complexes, it is necessary to locate a certain number of protein monomers in orientations that satisfy the symmetry elements of a given point group. Here, we are mainly concerned with cyclic (C_n) and dihedral (D_n) point groups, but we have also generalized our methods for building complexes with tetrahedral (T), octahedral (O), and icosahedral (I).

To assess the quality of symmetry for molecular assemblies, a cyclic symmetry measure is necessary, as the cyclic axes constitute the basic bricks from which one can reconstruct high-order symmetry groups. However, considering each symmetry axis separately would result in a globally incorrect assessment, as there are strict geometrical constraints between different axes of symmetry in high-order symmetry groups. This also motivated us to develop a symmetry detection method for cyclic groups and extend it to dihedral and cubic groups. Indeed, the need for such symmetry detection method exists, as some approximate methods, i.e. those from BioJava [209], are massively used to display the symmetry axes on the PDB website [219].

Inspired by the quaternion arithmetic applied to the best superposition of a set of points [55, 96, 121] together with our recent developments [177, 204], this Chapter also proposes a new symmetry measure and an analytical method to find the best symmetry axes of a symmetrical assembly possessing multiple symmetry axes. The method guaranties that the detected axes are consistent with the symmetry constraints. Our method produces results with a machine precision, its cost function is solely based on 3D Euclidean geometry, and most of the operations are performed analytically. This makes it extremely fast and particularly suitable for exhaustive analysis of PDB data. Below we provide details about the high-order symmetry measure and the computation of the symmetry axes for an assembly possessing any point symmetry group. The method first perceives the topology between different chains, and is able to deal with complex subunits that are composed of multiple chains. Then it iteratively solves a constrained quadratic optimization problem using a set of analytical solutions.

3.2 MATERIALS AND METHODS

3.2.1 *Notations*

Below, for 3D rotations and translations, we will be generally dealing with 3×3 matrices and 3-vectors. Therefore, for linear algebra operations we will stick to the following notation. Bold upper case letters (i.e. $\mathbf{A}$) will denote matrices, bold lower case letters (i.e. $\mathbf{b}$) will denote vectors, and normal weight lower case letters (i.e. c) will denote scalars. For trigonometric operations and illustrations we will also use an arrow notation for 3-vectors, such as $\vec{v}$. A rotation by an angle α about an axis $\vec{v}$ will be noted $R(\alpha, \vec{v})$.

3.2.2 *Quaternion arithmetic*

It is very convenient to express three-dimensional rotations using quaternion arithmetic. Thus, we will give a brief summary of it here. More informations on quaternions can be found elsewhere [204], for example. We consider a quaternion Q as a combination of a scalar s with a 3-component vector $\mathbf{q} = \{q_x, q_y, q_z\}^T$, $Q = [s, \mathbf{q}]$. Quaternion algebra defines multiplication, division, inversion and norm, among other operations. The product of two quaternions $Q_1 = [s_1, \mathbf{q}_1]$ and $Q_2 = [s_2, \mathbf{q}_2]$ is a quaternion and can be expressed through a combination of scalar and vector products,

$$\begin{aligned} Q_1 \cdot Q_2 &= [s_1, \mathbf{q}_1] \cdot [s_2, \mathbf{q}_2] \\ &= [s_1 s_2 - (\mathbf{q}_1 \cdot \mathbf{q}_2), s_1 \mathbf{q}_2 + s_2 \mathbf{q}_1 + (\mathbf{q}_1 \times \mathbf{q}_2)] \,. \end{aligned} \tag{52}$$

The squared norm of a quaternion Q is given as $|Q|^2 = s^2 + \mathbf{q} \cdot \mathbf{q}$, and a unit quaternion is a quaternion with its norm equal to 1. Finally, a unit quaternion $\hat{Q}$ corresponding to a rotation by an angle α around a unit axis $\mathbf{v}$ is given as $\hat{Q} = [\cos\frac{\alpha}{2}, \mathbf{v}\sin\frac{\alpha}{2}]$, and its inverse is $\hat{Q}^{-1} = [\cos\frac{\alpha}{2}, -\mathbf{v}\sin\frac{\alpha}{2}]$.

3.2.3 *Shape Matching Master Equation*

In order to develop the equations necessary for a docking search, or for the analysis of protein assemblies, it is useful to introduce a "matching operator", $\longleftrightarrow$, such that the notation

$$A(\underline{r}) \longleftrightarrow B(\underline{r}) \tag{53}$$

is taken to mean a *geometrical* or *functional correspondence* (e.g. docking, steric interaction, or match of potential energy fields) between proteins (or, more generally, shapes) A and B.

In the rigid-body shape-matching problem we let the expression

$$A(\underline{r}) \longleftrightarrow \hat{T}(x, y, z)\hat{R}(\alpha, \beta, \gamma)B(\underline{r}) \tag{54}$$

represent a general interaction between protein A and a rotated and translated version of protein B.

It is worth noting that the symbol $\longleftrightarrow$ can be treated like an equality in the sense that applying an inverse translation to each side of Equation 54

$$\hat{T}(x, y, z)^{-1}A(\underline{r}) \longleftrightarrow \hat{R}(\alpha, \beta, \gamma)B(\underline{r}) \tag{55}$$

represents exactly the same relative orientation of the two protein monomers as in the previous expression.

3.2.4 *Root mean square deviation*

The root mean square deviation (RMSD) is one of the most widely used similarity criteria in structural biology and bioinformatics. It can also be seen as a 2-norm between points in a N-dimensional space (see below) and be applied to the comparison of rigid bodies. We will stick to this measure for multiple reasons, e.g., it is very powerful, easy to understand and also because it can be computed very efficiently. For our particular needs we will use the definition of RMSD between two ordered sets of points, where each point has an equal contribution to the overall RMSD loss. More precisely, given a set of N points $A = \{\mathbf{a}_i\}_N$ and $B = \{\mathbf{b}_i\}_N$, the RMSD between them is defined as

$$\mathrm{RMSD}(A, B)^2 = \frac{1}{N} \sum_{1 \le i \le N} |\mathbf{a}_i - \mathbf{b}_i|^2 \,. \tag{56}$$

3.2.5 *RMSD Master Equation*

Let us formally define the problem of the best superposition of two rigid bodies (e.g. molecules). Suppose that the operator associated with a rotation about axis $\vec{v}$ by an angle α may be labelled $\hat{R}(\alpha, \vec{v})$. Let us also suppose that the operator associated with a translation by a vector $\vec{u}$ is labelled $\hat{T}(\vec{u})$. We should mention that we have borrowed the presented formalism from the molecular docking methods [217], where it appears very useful.

Let $\mathbf{u}$ be a translation vector and $\hat{Q} \equiv [s, \mathbf{q}]$ a rotation quaternion corresponding to the operators $\hat{T}(\vec{u})$, and $\hat{R}(\alpha, \vec{v})$, respectively. We apply these to an assembly A composed of N_s subunits with N_a atoms at positions $A = \{\mathbf{a}_{i,j}\}_{N_s,N_a}$ with $\mathbf{a}_{i,j} = \{x_{i,j}, y_{i,j}, z_{i,j}\}^T$, and compare the result with the positions of a molecule B with the same number of subunits and atoms at positions $B = \{\mathbf{b}_{i,j}\}_{N_s,N_a}$ with $\mathbf{b}_{i,j} = \{x'_{i,j}, y'_{i,j}, z'_{i,j}\}^T$. Using a similar reasoning to what we presented previously [204], the RMSD between new positions of A and B in the reference frame bound to the center of mass (COM) of A is given as

$$\mathrm{RMSD}^2(\hat{T}(\vec{u})\hat{R}(\alpha, \vec{v})A, B) = \frac{4}{N}\mathbf{q}^T\mathbf{I}'\mathbf{q} + 4s\mathbf{q}^T\mathbf{x}_\perp + \mathbf{u}^2 + 2\mathbf{u}^T\mathbf{x}_m + x_s. \tag{57}$$

Here, the modified inertia tensor $\mathbf{I}'$ is given as

$$\mathbf{I}' = \begin{pmatrix} \sum(y_{i,j}y'_{i,j} + z_{i,j}z'_{i,j}) & -\sum(x'_{i,j}y_{i,j} + x_{i,j}y'_{i,j})/2 & -\sum(x'_{i,j}z_{i,j} + x_{i,j}z'_{i,j})/2 \\ -\sum(x_{i,j}y'_{i,j} + x'_{i,j}y_{i,j})/2 & \sum(x_{i,j}x'_{i,j} + z_{i,j}z'_{i,j}) & -\sum(y'_{i,j}z_{i,j} + y_{i,j}z'_{i,j})/2 \\ -\sum(x_{i,j}z'_{i,j} + x'_{i,j}z_{i,j})/2 & -\sum(y_{i,j}z'_{i,j} + y'_{i,j}z_{i,j})/2 & \sum(x_{i,j}x'_{i,j} + y_{i,j}y'_{i,j}) \end{pmatrix}. \tag{58}$$

The vectors $\mathbf{x}_\perp$, $\mathbf{x}_m$, and the scalar x_s are

$$\begin{aligned} \mathbf{x}_\perp &= \sum_{i,j} \mathbf{b}_{i,j} \times \mathbf{a}_{i,j}/N \\ \mathbf{x}_m &= -\sum_{i,j} \mathbf{b}_{i,j}/N \\ x_s &= \sum_{i,j} (\mathbf{a}_{i,j} - \mathbf{b}_{i,j})^2/N. \end{aligned} \tag{59}$$

Below, we will analytically determine axes that correspond to the chosen C_n symmetries by minimizing eq. 57 with proper constraints.

We should specifically mention that if the coordinates of A and B are only different by a permutation of their indexes, as it happens in many practical cases of symmetry detection described below, then the vector $\mathbf{x}_m$ becomes zero. This uncouples the RMSD master equation with respect to the translation and rotation and greatly simplifies many corresponding equations. More precisely, minimization of RMSD with respect to $\mathbf{u}$ in this case gives a trivial solution $\mathbf{u} = 0$.

We shall also mention that if A and B conformations are equivalent in eq. 57, then we obtain a simplified RMSD equation,

$$\mathrm{RMSD}^2(\hat{T}(\vec{u})\hat{R}(\alpha, \vec{v})A, B) = \frac{4}{N}\mathbf{q}^T\mathbf{I}'\mathbf{q} + \mathbf{u}^2 = \frac{4}{N}\sin^2\frac{\alpha}{2}\mathbf{v}^T\mathbf{I}'\mathbf{v} + \mathbf{u}^2. \tag{60}$$

If the center of mass $\mathbf{c}$ of the A subunit is nonzero, we will have an additional $+2\mathbf{u}^T(\mathbf{R} - \mathbf{E}_3)\mathbf{c}$ term in this equation, where $\mathbf{R}$ is a rotation matrix corresponding to the rotation quaternion $\hat{Q}$, and $\mathbf{E_3}$ is a 3×3 identity matrix.

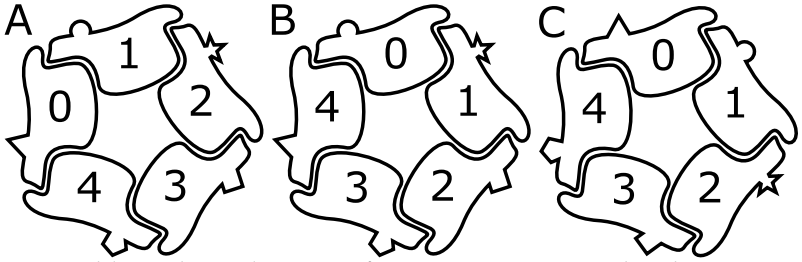


Figure 1: **A**: An assembly with an almost perfect C_5 symmetry. Each subunit is represented with an outline whose shapes are slightly different from each other. **B**: The 1-permuted version of this assembly, the shapes are the same as in A but the labelling is different. **C**: The rotated version of the assembly A by an angle $2\pi/5$.

3.2.6 *Working with molecular assemblies*

As we work with assemblies composed of macromolecules such as proteins, it is convenient to introduce an intermediate level of structural hierarchy between the *complete* assembly and its N atoms. Let us consider a molecular assembly as a list of N_s subunits, each containing N_a atoms such that $N = N_a N_s$. The RMSD between two assemblies is then

$$\mathrm{RMSD}(A,B)^2 = \frac{1}{N} \sum_{0 \le i < N_s} \sum_{0 \le j < N_a} |\mathbf{a}_{i,j} - \mathbf{b}_{i,j}|^2 \,. \tag{61}$$

We can assume that every subunit has the same number of *reference* points. Technically, we achieve it by performing a multiple sequence alignment of the subunits and keeping only the aligned parts for the subsequent analysis. More precisely, the reference points are located at the positions of the aligned C_α atoms. This makes our method robust against various inconsistencies in the input data.

It will be convenient to assume that the subunits in the assembly are labelled with integers modulo of n, i.e. i and $i+n$ refer to the same subunit. Let us also assume that the labelling is *sequential*, meaning that the subunit i is located *between* the subunits $i-1$ and $i+1$. Finally, let us define a *k-permuted* version A^k of the assembly A by

$$\mathbf{a}^k_{i,j} = \mathbf{a}_{i+k,j} \tag{62}$$

Note that according to this definition, A is equal to its *o-permuted* version, and a *k-permuted* assembly matches itself rotated by $2k\pi/n$. If the subunits are not labelled sequentially, finding the permutation between the subunits, that is associated with every rotation operator, is not straightforward. Our initial approach consisted in projecting the centers of mass of the different subunits on the plane orthogonal to the principal eigenvector of the inertia matrix of the assembly, and then reordering the subunits according to this projection. During the second part of this work [183], we developed a much more general and robust method that automatically determines the permutations between the subunits for each rotation operator in a certain symmetry group including cyclic, dihedral and cubic cases.

3.2.7 *Complete C_n assembly*

Let us first assume that we have as input a *complete* cyclic assembly, for which we want to assess the quality of the cyclic symmetry. A cyclic symmetry group of order n can be uniquely described with its symmetry axis $\vec{v}$, the position of this axis, and its order n. As it is explained above, the translational part of the RMSD master equation 57 in this case is equal to zero, because the two sets of points are permutations of each other. The angles of the rotation operators are constrained to be $\{k\omega\}_{0 \le k < n}$ with $\omega = 2\pi/n$. To

determine the *quality* of a rotation axis $\vec{v}$, we compute the RMSD between the assembly rotated by an angle of $k\omega$ (see Fig. 1C) and a *k-permuted* version of the original assembly (see Fig. 1B), as it is shown in Figure 1. This RMSD will thus be our *symmetry measure*.

The quaternion representation of the k^{th} C_n symmetry operator is given as

$$\hat{Q}^k \equiv [s, \mathbf{q}] = [\cos\frac{k\omega}{2}, \sin\frac{k\omega}{2}\mathbf{v}], \tag{63}$$

with $0 \leq k < n$. According to the RMSD master equation, with $B = A^k$ and $\mathbf{u} = 0$, we obtain

$$\text{RMSD}^2(\hat{R}(k\omega, \vec{v})A, A^k) = \frac{4}{N}\mathbf{q}^T\mathbf{I}'_k\mathbf{q} + 4s\mathbf{q}^T\mathbf{x}_{k\perp} + x_{ks}. \tag{64}$$

Here

$$\mathbf{I}'_k = \begin{pmatrix} \sum(y_{i,j}y_{k+i,j} + z_{i,j}z_{k+i,j}) & -\sum(x_{k+i,j}y_{i,j} + x_{i,j}y_{k+i,j})/2 & -\sum(x_{k+i,j}z_{i,j} + x_{i,j}z_{k+i,j})/2 \\ -\sum(x_{i,j}y_{k+i,j} + x_{k+i,j}y_{i,j})/2 & \sum(x_{i,j}x_{k+i,j} + z_{i,j}z_{k+i,j}) & -\sum(y_{k+i,j}z_{i,j} + y_{i,j}z_{k+i,j})/2 \\ -\sum(x_{i,j}z_{k+i,j} + x_{k+i,j}z_{i,j})/2 & -\sum(y_{i,j}z_{k+i,j} + y_{k+i,j}z_{i,j})/2 & \sum(x_{i,j}x_{k+i,j} + y_{i,j}y_{k+i,j}) \end{pmatrix}, \tag{65}$$

and

$$\begin{aligned} \mathbf{x}_{k\perp} &= \sum_{i,j} \mathbf{a}_{k+i,j} \times \mathbf{a}_{i,j}/N \\ x_{ks} &= \sum_{i,j} (\mathbf{a}_{i,j} - \mathbf{a}_{k+i,j})^2/N. \end{aligned} \tag{66}$$

Finding the best rotation axis reduces to the following optimization problem,

$$\begin{aligned} \min_{\mathbf{v}} \quad & \text{RMSD}^2(\mathbf{v}) = \mathbf{v}^T\mathbf{A}_k\mathbf{v} + \mathbf{d}_k^T\mathbf{v} + f_k \\ \text{s.t.} \quad & \mathbf{v}^T\mathbf{v} = 1, \end{aligned} \tag{67}$$

where

$$\begin{aligned} \mathbf{A}_k &= \frac{4}{N}\sin^2\frac{k\omega}{2}\mathbf{I}'_k \\ \mathbf{d}_k &= 2\sin(k\omega)\mathbf{x}_{k\perp} \\ f_k &= x_{ks}. \end{aligned} \tag{68}$$

Equations 67-68 formulate a minimization problem to find an axis corresponding to a particular rotation operator with a fixed rotation angle. However, our goal is to determine the axis that is the best for all the rotation operators. We can thus sum up the above expressions for every k, as the axis $\vec{v}$, which we are seeking for, is *the same* for all the rotation operators. Finally, finding the best axis of symmetry for a C_n group is equivalent to solving the following *trust-region subproblem*,

$$\begin{aligned} \min_{\mathbf{v}} \quad & \mathbf{v}^T\sum_{k=1}^{k<n}\mathbf{A}_k\mathbf{v} + \sum_{k=1}^{k<n}\mathbf{d}_k^T\mathbf{v} + \sum_{k=1}^{k<n} f_k \\ \text{s.t.} \quad & \mathbf{v}^T\mathbf{v} = 1. \end{aligned} \tag{69}$$

This is a well-studied optimization problem. It can be efficiently solved with a number of different methods. In our case, the dimensionality of the problem is very low and thus we have chosen the solver based on the Sorensen method [235], which typically converges to machine precision in 3 - 10 iterations in our case. Equation 69 constitutes the first principal result of this work. We should note that in a particular C_2 case, the $\mathbf{d}_k^T$ coefficients vanish and the solution of the problem 69 reduces to the smallest eigenvector of matrix $\sum_{k=1}^{k<n}\mathbf{A}_k$.

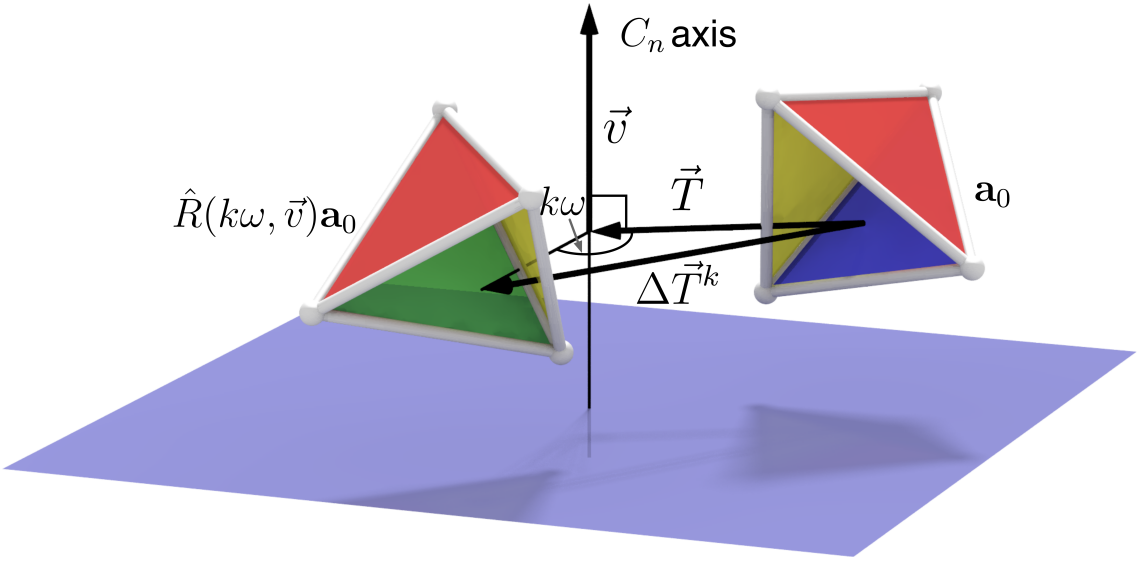


Figure 2: Illustration of the rotation of a subunit $\mathbf{a}_0$. The original and rotated versions of $\mathbf{a}_0$ are represented as tetrahedrons having four differently colored faces (red, green, blue and yellow). For the clarity of the representation, the green face was removed from $\mathbf{a}_0$ and the blue face was removed from $\hat{R}(k\omega, \vec{v})\mathbf{a}_0$. The $\vec{T}$ vector connects the COM of $\mathbf{a}_0$ with the symmetry axis. The $\Delta\vec{T}$ vector connects the COM of $\mathbf{a}_0$ with the COM of $\hat{R}(k\omega, \vec{v})\mathbf{a}_0$.

3.2.8 *C_n assembly with missing subunits*

Some examples of molecular assemblies with presumably cyclic symmetry are not complete and have missing subunits. This automatically raises two questions: what should be the order of the complete assembly and how to reconstruct it? The ability to find the rotation operator that produces the smallest RMSD between the present subunits with a constrained angle answers these two questions. To determine the best order of the cyclic symmetry, we can simply exhaustively test all the different possible orders by changing the constraint on the angle of the rotation operator, as it is given by equation 63, and then solving the RMSD master equation 57. Once this step is done, we obtain the order and the axis of symmetry, which makes the reconstruction of the complete assembly trivial. However, in this case, we need to solve the full version of the RMSD master equation, since the translational component of RMSD is not null.

To determine the axis of the rotation operator, similarly to the case with the complete assembly considered above, we will compare the rotated version of the *partial* assembly with its permuted version. We should mention that in the case of partial assembly we assume the sequential order of the input subunits. If it is not the case, the order has to be specified manually, since the performance of the automatic procedure for the order perception is largely affected by the missing subunits. Let us assume that the subunit $\mathbf{a}_0 = \{x_{0,j}, y_{0,j}, z_{0,j}\}^T_{(1\leq j\leq N_a)}$ is present. Let us label the vector that connects the COM of the $\mathbf{a}_0$ subunit with the symmetry axis $\vec{v}$, and which is perpendicular to it, as $\vec{T}$. Following Figure 2, the translation vector $\Delta\vec{T}^k$ that connects the COM of $\mathbf{a}_0$ with the COM of $\hat{R}(k\omega, \vec{v})\mathbf{a}_0$ is

$$\Delta\mathbf{T}^k = (1 - \cos(k\omega))\,\mathbf{T} - \sin(k\omega)\mathbf{v} \times \mathbf{T}. \tag{70}$$

The squared 2-norm of this vector is given as

$$\left(\Delta\mathbf{T}^k\right)^2 = 4\sin^2\frac{k\omega}{2}\mathbf{T}^2. \tag{71}$$

Now we are ready to substitute the rotation quaternion from equation 63 and the obtained translation vector into the RMSD master equations 57. The RMSD is now a func-

tion of $\mathbf{T}$ and $\mathbf{v}$ vectors. Keeping the quaternion representation from equation 63, we obtain

$$\begin{aligned}\mathrm{RMSD}_k^2 =& \frac{4}{N}\mathbf{q}^T\mathbf{I}'\mathbf{q} + 4s\mathbf{q}^T\mathbf{x}_\perp + 4\sin^2\frac{k\omega}{2}\mathbf{T}^2 \\ &+ \left((1-\cos(k\omega))\,\mathbf{T} - \sin(k\omega)\mathbf{v}\times\mathbf{T}\right)^T\mathbf{x}_m + x_s,\end{aligned} \tag{72}$$

which reduces to the following optimization problem,

$$\begin{aligned}\min_{\mathbf{v},\mathbf{T}} \quad \mathrm{RMSD}_k^2(\mathbf{v},\mathbf{T}) =& \quad \mathbf{v}^T\mathbf{A}_k\mathbf{v} + b_k\mathbf{T}^2 + \mathbf{v}^T\mathbf{C}_k\mathbf{T} \\ &+ \mathbf{d}_k^T\mathbf{v} + \mathbf{e}_k^T\mathbf{T} + f_k \\ \text{s.t.} \quad &\begin{cases}\mathbf{v}^T\mathbf{v} = 1 \\ \mathbf{v}^T\mathbf{T} = 0\end{cases}.\end{aligned} \tag{73}$$

Here, the coefficients are given as

$$\begin{aligned}\mathbf{A}_k &= \frac{4}{N}\sin^2(\frac{k\omega}{2})\mathbf{I}' \\ b_k &= 4\sin^2(\frac{k\omega}{2}) \\ \mathbf{C}_k &= 2\sin(k\omega)\begin{pmatrix} 0 & \mathbf{x}_{m3} & -\mathbf{x}_{m2} \\ -\mathbf{x}_{m3} & 0 & \mathbf{x}_{m1} \\ \mathbf{x}_{m2} & -\mathbf{x}_{m1} & 0\end{pmatrix} \\ \mathbf{d}_k &= 2\sin(k\omega)\mathbf{x}_\perp \\ \mathbf{e}_k &= -4\sin^2(\frac{k\omega}{2})\mathbf{x}_m \\ f_k &= x_s,\end{aligned} \tag{74}$$

which follows from the substitution of eqs. 63 and 70 into the RMSD master equation 57. In the above equation the definitions of matrix $\mathbf{I}'$, and vectors $\mathbf{x}_\perp$ and $\mathbf{x}_m$ are taken from equations 58 and 59 with the substitutions of $\mathbf{a} = \mathbf{a}_0$ and $\mathbf{b} = \mathbf{a}_\mathbf{k}$. At this point, vectors $\mathbf{v}$ and $\mathbf{T}$ are defined independently from the index k, thus we can sum up equation 73 for all k corresponding to the present subunits, and provide the global coefficients that will define the overall symmetry measure $\mathrm{RMSD}^2(\mathbf{v},\mathbf{T}) = \sum_k \mathrm{RMSD}_k^2(\mathbf{v},\mathbf{T})$ as

$$\begin{aligned}\mathbf{A} &= \sum_k \mathbf{A}_k \\ b &= \sum_k b_k \\ \mathbf{C} &= \sum_k \mathbf{C}_k \\ \mathbf{d} &= \sum_k \mathbf{d}_k \\ \mathbf{e} &= \sum_k \mathbf{e}_k \\ f &= \sum_k x.\end{aligned} \tag{75}$$

Using the *Lagrangian formalism*, we can introduce two Lagrange multipliers λ_1 and λ_2 with the Lagrangian function $L(\mathbf{v},\mathbf{T},\lambda_1,\lambda_2)$ that incorporates two equality constraints from eq. (73) as

$$\begin{aligned}L(\mathbf{v},\mathbf{T},\lambda_1,\lambda_2) =& \mathbf{v}^T\mathbf{A}\mathbf{v} + b\mathbf{T}^T\mathbf{T} + \mathbf{v}^T\mathbf{C}\mathbf{T} \\ &+ \mathbf{d}^T\mathbf{v} + \mathbf{e}^T\mathbf{T} + f + \lambda_1(\mathbf{v}^T\mathbf{v} - 1) + \lambda_2\mathbf{v}^T\mathbf{T}.\end{aligned} \tag{76}$$

Here, matrix $\mathbf{A}$ is symmetric and positive definite, while matrix $\mathbf{C}$ is skew-symmetric. Setting the gradient $L_{\mathbf{T}}$ to zero gives

$$\begin{aligned}&(\mathbf{C}^T + \lambda_2 \mathbf{E}_3)\mathbf{v} + \mathbf{e} + 2b\mathbf{T} = 0\\ &\text{s.t.} \quad \begin{cases}\mathbf{v}^T\mathbf{v} = 1\\ \mathbf{v}^T\mathbf{T} = 0\end{cases},\end{aligned} \tag{77}$$

where $\mathbf{E}_3$ is a 3×3 identity matrix. Left-multiplying the first equation by $\mathbf{v}^T$, we obtain

$$\lambda_2 + \mathbf{e}^T\mathbf{v} = 0. \tag{78}$$

Therefore, we can determine the first unknown vector $\mathbf{T}$ as

$$\mathbf{T} = -\frac{1}{2b}\left(\mathbf{e} + \mathbf{C}^T\mathbf{v} - (\mathbf{e}^T\mathbf{v})\mathbf{v}\right). \tag{79}$$

Now, substituting it to the minimization function $\text{RMSD}^2(\mathbf{v}, \mathbf{T})$, we obtain

$$\begin{aligned}\text{RMSD}^2(\mathbf{v}, \mathbf{T}) =& \mathbf{v}^T\mathbf{A}\mathbf{v} + \mathbf{d}^T\mathbf{v} + f\\ &+ \frac{1}{4b}(-\mathbf{e}^2 + 2\mathbf{e}^T\mathbf{C}\mathbf{v} - \mathbf{v}^T\mathbf{C}\mathbf{C}^T\mathbf{v} + \mathbf{v}^T\mathbf{e}\mathbf{e}^T\mathbf{v}).\end{aligned} \tag{80}$$

As a result, our initial optimization problem 73 reduces to the following form,

$$\begin{aligned}&\min_{\mathbf{v}} \mathbf{v}^T\mathbf{X}\mathbf{v} + \mathbf{y}^T\mathbf{v} + z\\ &\text{s.t.} \quad \mathbf{v}^T\mathbf{v} = 1\end{aligned}, \tag{81}$$

where the coefficients $\mathbf{X}$, $\mathbf{y}$, and z are given as

$$\begin{aligned}\mathbf{X} &= \mathbf{A} + \frac{1}{4b}(-\mathbf{C}\mathbf{C}^T + \mathbf{e}\mathbf{e}^T)\\ \mathbf{y} &= \frac{1}{2b}\mathbf{C}^T\mathbf{e} + \mathbf{d}\\ z &= -\frac{1}{4b}\mathbf{e}^T\mathbf{e} + f.\end{aligned} \tag{82}$$

This is once again the previously introduced *trust-region subproblem*. Equations 81-82 constitute the second principal result of this work.

3.2.9 *Detection of helical symmetry parameters*

Let us parametrize a rigid body by its reference frame (for now let us assume the origin of the reference is aligned with the COM of the rigid body) and its inertia tensor $\mathbf{I}$ in the COM frame. Let us assume that the rigid body moves along a screw (a spiral or a helix) along the unit normal vector $\vec{n}$, such that the shortest distance from the COM of the rigid body to the screw axis is r. Generally, we suppose that the screw axis does not intersect COM of the rigid body. Let also p be the pitch of the screw, i.e. the height of one complete screw (helix) turn, measured parallel to the axis of the screw $\vec{n}$. Now, we are ready to write the expression of the RMSD^2 between rigid body's positions as a function of the rotation angle about the screw α,

$$\begin{aligned}\text{RMSD}_k^2 =& \frac{4}{N}\mathbf{q}^T\mathbf{I}'\mathbf{q} + 4s\mathbf{q}^T\mathbf{x}_\perp + 4\sin^2\frac{\omega_k}{2}\mathbf{T}_0^2 + \frac{p^2\omega_k^2}{(2\pi)^2}\\ &+ 2\left((1-\cos\omega_k)\,\mathbf{T}_0 - (\sin\omega_k)\mathbf{v}\times\mathbf{T}_0 + \frac{p\omega_k}{2\pi}\mathbf{v}\right)^T\mathbf{x}_m + x_s.\end{aligned} \tag{83}$$

Let us first compute the optimal pitch p^* by setting the derivative of $\sum_k \mathrm{RMSD}_k^2$ with respect to p to zero,

$$p^* \sum_k \omega_k^2 = -\pi \mathbf{v}^T \sum_k \omega_k \mathbf{x}_m^k. \tag{84}$$

This leads to

$$\frac{p^{*2}}{(2\pi)^2} \sum_k \omega_k^2 = \frac{1}{4} \frac{\mathbf{v}^T \left(\sum_k \omega_k \mathbf{x}_m^k\right) \left(\sum_k \omega_k \mathbf{x}_m^k\right)^T \mathbf{v}}{\sum_k \omega_k^2}, \tag{85}$$

and

$$\frac{p^*}{2\pi} \mathbf{v}^T \sum_k \omega_k \mathbf{x}_m^k = -\frac{1}{2} \frac{\mathbf{v}^T \left(\sum_k \omega_k \mathbf{x}_m^k\right) \left(\sum_k \omega_k \mathbf{x}_m^k\right)^T \mathbf{v}}{\sum_k \omega_k^2}. \tag{86}$$

Let us also express $\vec{T}_0$ (that connect the COM of $\mathbf{a}_0$ (or, more generally, the i-th subunit), $\vec{m}^i$ with the rotation axis $\vec{v}$) in the global reference system, so we can solve different subunits with the same global vector $\vec{T}$, which connects the origin of the global coordinate system with $\vec{v}$ and is perpendicular to it. It is easy to show that

$$\vec{T}_0 = -\vec{m}^i + \vec{T} + (\vec{v} \cdot \vec{m}^i)\vec{v}. \tag{87}$$

This gives

$$\mathbf{T}_0^2 = \mathbf{m}^{i2} + \mathbf{T}^2 - \mathbf{v}^T \mathbf{m}^i \mathbf{m}^{iT} \mathbf{v} - 2\mathbf{m}^i \mathbf{T}, \tag{88}$$

and

$$\begin{aligned} &2\left(\left(1-\cos\omega_k\right)\mathbf{T}_0 - (\sin\omega_k)\mathbf{v}\times\mathbf{T}_0\right)^T \mathbf{x}_m = \\ &2\left(1-\cos\omega_k\right)\left(-\mathbf{m}^{iT}\mathbf{x}_m\right) + 2\mathbf{x}_m^T\mathbf{T} + \mathbf{v}^T 2\mathbf{m}^i\mathbf{x}_m^T\mathbf{v} \\ &+2\sin\omega_k(\mathbf{v}\times\mathbf{m}^i)^T\mathbf{x}_m - 2\sin\omega_k(\mathbf{v}\times\mathbf{T})^T\mathbf{x}_m. \end{aligned} \tag{89}$$

This allows to reduce the original least-square problem to the following optimization form, provided that optimization in ω is performed at a separate step,

$$\begin{aligned} \min_{\mathbf{v},\mathbf{T}} \quad & \mathrm{RMSD}_{ij}^2(\mathbf{v},\mathbf{T}) = \mathbf{v}^T\mathbf{A}_{ij}\mathbf{v} + b_{ij}\mathbf{T}^2 + \mathbf{v}^T\mathbf{C}_{ij}\mathbf{T} \\ & \qquad\qquad + \mathbf{d}_{ij}^T\mathbf{v} + \mathbf{e}_{ij}^T\mathbf{T} + f_{ij} \\ \text{s.t.} \quad & \begin{cases} \mathbf{v}^T\mathbf{v} = 1 \\ \mathbf{v}^T\mathbf{T} = 0 \end{cases} . \end{aligned} \tag{90}$$

Here, the coefficients at fixed ω_k and p are given as

$$\begin{aligned} \mathbf{A}_{ij} &= \frac{4}{N}\sin^2(\frac{\omega_{ij}}{2})\mathbf{I}' + 2\mathbf{m}^i\mathbf{x}_m^T - 4\sin^2(\frac{\omega_{ij}}{2})\mathbf{m}^i\mathbf{m}^{iT} \\ b_{ij} &= 4\sin^2(\frac{\omega_{ij}}{2}) \\ \mathbf{C}_{ij} &= 2\sin\omega_{ij}\begin{pmatrix} 0 & \mathbf{x}_{m3} & -\mathbf{x}_{m2} \\ -\mathbf{x}_{m3} & 0 & \mathbf{x}_{m1} \\ \mathbf{x}_{m2} & -\mathbf{x}_{m1} & 0 \end{pmatrix} \\ \mathbf{d}_{ij} &= 2(\sin\omega_{ij})\mathbf{x}_\perp + \frac{p\omega_{ij}}{2\pi}\mathbf{x}_m + 2\sin\omega_{ij}\mathbf{m}^i\times\mathbf{x}_m \\ \mathbf{e}_{ij} &= -4\sin^2(\frac{\omega_{ij}}{2})\mathbf{x}_m - 8\sin^2(\frac{\omega_{ij}}{2})\mathbf{m}^i + 2\mathbf{x}_m \\ f_{ij} &= x_s + \frac{p^2\omega_{ij}^2}{(2\pi)^2} + 4\sin^2(\frac{\omega_{ij}}{2})\mathbf{m}^{i2} + 2\left(1-\cos\omega_{ij}\right)\left(\mathbf{m}^{iT}\mathbf{x}_m\right). \end{aligned} \tag{91}$$

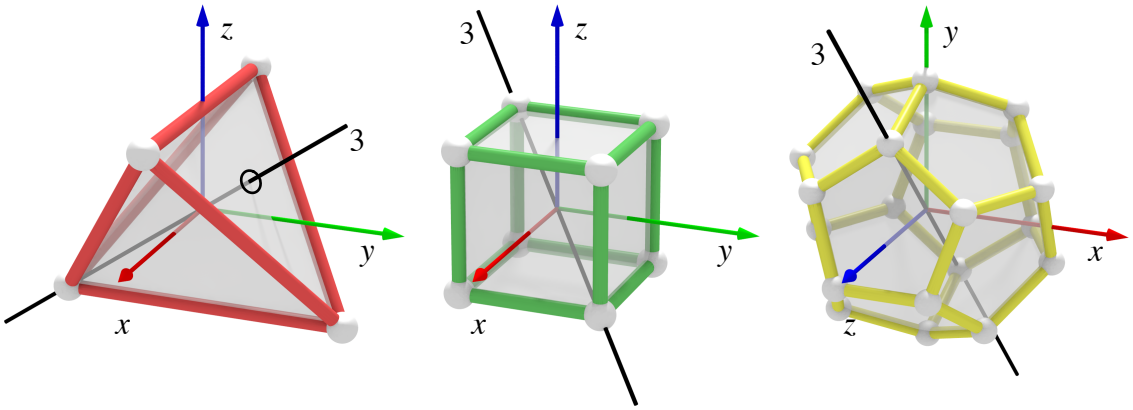


Figure 3: Computational orientations for tetrahedral (T), octahedral (O) and icosahedral (I) complexes. For each symmetry type, a solid black line shows one of the 3-fold axes. Candidate symmetrical complexes may be created by placing a C_3 trimer at each vertex (white sphere) of the desired symmetry type.

At this point, vectors $\mathbf{v}$ and $\mathbf{T}$ are defined independently from the indices ij, thus we can sum up equation 73 for all ij, and provide the global coefficients that will define the overall geometric loss. If we also provide p at its optimal value p^*, then the coefficients will be given as

$$
\begin{aligned}
\mathbf{A} &= \sum_{ij}\left[\frac{4}{N}\sin^2(\frac{\omega_{ij}}{2})\mathbf{I}'_{ij} + 2\mathbf{m}^i\mathbf{x}_m^{ijT} - 4\sin^2(\frac{\omega_{ij}}{2})\mathbf{m}^i\mathbf{m}^{iT}\right] - \frac{1}{4}\frac{\left(\sum_{ij}\omega_{ij}\mathbf{x}_m^{ij}\right)\left(\sum_{ij}\omega_{ij}\mathbf{x}_m^{ij}\right)^T}{\sum_{ij}\omega_{ij}^2} \\
b &= \sum_{ij} 4\sin^2(\frac{\omega_{ij}}{2}) \\
\mathbf{C} &= \sum_{ij} 2\sin\omega_{ij}\begin{pmatrix} 0 & \mathbf{x}_{m3}^{ij} & -\mathbf{x}_{m2}^{ij} \\ -\mathbf{x}_{m3}^{ij} & 0 & \mathbf{x}_{m1}^{ij} \\ \mathbf{x}_{m2}^{ij} & -\mathbf{x}_{m1}^{ij} & 0 \end{pmatrix} \\
\mathbf{d} &= \sum_{ij}\left[2\sin\omega_{ij}\mathbf{x}_{\perp}^{ij} + 2\sin\omega_{ij}\mathbf{m}^i \times \mathbf{x}_m^{ij}\right] \\
\mathbf{e} &= \sum_{ij}\left[-4\sin^2(\frac{\omega_{ij}}{2})\mathbf{x}_m^{ij} - 8\sin^2(\frac{\omega_{ij}}{2})\mathbf{m}^i + 2\mathbf{x}_m^{ij}\right] \\
f &= \sum_{ij}\left[x_s^{ij} + 4\sin^2(\frac{\omega_{ij}}{2})\mathbf{m}^{i2} + 2\left(1-\cos\omega_{ij}\right)\left(\mathbf{m}^{iT}\mathbf{x}_m^{ij}\right)\right].
\end{aligned}
\tag{92}
$$

These equations extend the cyclic case for the helical symmetry with the additional search variables.

3.2.10 *Higher-order symmetry groups*

Here we will use the same notations as in our previous Section on cyclic symmetries [186]. From now on, for simplicity, we will only write equations for the cubic group. Figure 3 gives a schematic illustration of these groups. Indeed, the equations for the dihedral group are obtained by substituting the index n for the index 3. Our goal is to minimize the loss function for each element g of the chosen symmetry group. The contribution to the loss function is the RMSD between $r_g(A)$ and $A_g = \{\mathbf{a}_{\sigma_g(i),j}\}$. According

to the RMSD master equation 57 with $B = A_g$, we can say that A and B have the same COM, so the translational part of RMSD becomes null and we obtain

$$\text{RMSD}^2(r_g(A), A_g) = \frac{4}{N}\mathbf{q}^T\mathbf{I}_g\mathbf{q} + 4s\mathbf{q}^T\mathbf{x}_{g\perp} + x_{gs}, \tag{93}$$

where

$$\mathbf{I}_g = \begin{pmatrix} \sum(y_{i,j}y_{\sigma_g(i),j} + z_{i,j}z_{\sigma_g(i),j}) & -\sum(x_{\sigma_g(i),j}y_{i,j} + x_{i,j}y_{\sigma_g(i),j})/2 & -\sum(x_{\sigma_g(i),j}z_{i,j} + x_{i,j}z_{\sigma_g(i),j})/2 \\ -\sum(x_{i,j}y_{\sigma_g(i),j} + x_{\sigma_g(i),j}y_{i,j})/2 & \sum(x_{i,j}x_{\sigma_g(i),j} + z_{i,j}z_{\sigma_g(i),j}) & -\sum(y_{\sigma_g(i),j}z_{i,j} + y_{i,j}z_{\sigma_g(i),j})/2 \\ -\sum(x_{i,j}z_{\sigma_g(i),j} + x_{\sigma_g(i),j}z_{i,j})/2 & -\sum(y_{i,j}z_{\sigma_g(i),j} + y_{\sigma_g(i),j}z_{i,j})/2 & \sum(x_{i,j}x_{\sigma_g(i),j} + y_{i,j}y_{\sigma_g(i),j}) \end{pmatrix}, \tag{94}$$

and

$$\begin{aligned} \mathbf{x}_{g\perp} &= \sum_{i,j} \mathbf{a}_{\sigma_g(i),j} \times \mathbf{a}_{i,j}/N \\ x_{gs} &= \sum_{i,j} (\mathbf{a}_{i,j} - \mathbf{a}_{\sigma_g(i),j})^2/N. \end{aligned} \tag{95}$$

Our aim will be to minimize the sum of squared RMSDs over all elements g of the group Γ. Let us first assume that we know the value of one of the two axes $\mathbf{v}_3$ or $\mathbf{v}_2$, for example, $\mathbf{v}_3$. In practice, we first compute $\mathbf{v}_3$ axis as a cyclic axis using the method from the cyclic Section [186], then we alternate the computations of $\mathbf{v}_2$ and $\mathbf{v}_3$ considering the other axis as known. This method converges to machine precision in about 10 iterations. Thanks to the RMSD master equation, we can write the loss function as a function of the axis $\mathbf{v}_2$ as follows,

$$\begin{aligned} &\sum_{g\in\Gamma} \text{RMSD}_g^2(\mathbf{v}_2) = \\ &\mathbf{v}_2^T \left(\sum_{g\in\Gamma} b_g^2 \frac{4}{N}\mathbf{I}_g + 2\sum_{g\in\Gamma} b_g c_g \frac{4}{N}\mathbf{I}_g[\mathbf{v}_3]_\times + \sum_{g\in\Gamma} c_g^2 [\mathbf{v}_3]_\times{}^T \frac{4}{N}\mathbf{I}_g[\mathbf{v}_3]_\times \right) \mathbf{v}_2 \\ &+ \left(2\sum_{g\in\Gamma} a_g b_g \mathbf{v}_3^T \frac{4}{N}\mathbf{I}_g + 2\sum_{g\in\Gamma} a_g c_g \mathbf{v}_3^T \frac{4}{N}\mathbf{I}_g + 4\sum_{g\in\Gamma} s_g b_g \mathbf{x}_{g\perp}^T \right) \mathbf{v}_2 \\ &+ \sum_{g\in\Gamma} a_g^2 \mathbf{v}_3^T \frac{4}{N}\mathbf{I}_g\mathbf{v}_3 + 4\sum_{g\in\Gamma} s_g b_g \mathbf{x}_{g\perp}^T\mathbf{v}_3 + \sum_{g\in\Gamma} x_{gs}. \end{aligned} \tag{96}$$

We can rewrite this equation as the following minimization problem with respect to $\mathbf{v}_2$,

$$\begin{aligned} \arg\min_{\mathbf{v}_2} \quad & \mathbf{v}_2^T\mathbf{A}\mathbf{v}_2 + \mathbf{b}^T\mathbf{v}_2 + c \\ \text{s.t.} \quad & \begin{cases} \mathbf{v}_2^T\mathbf{v}_2 = 1 \\ \mathbf{v}_3^T\mathbf{v}_2 = \alpha. \end{cases} \end{aligned} \tag{97}$$

The two constraints come from the unit norm of the rotation axes and the geometry of the generator axes. The above equations have the following coefficients,

$$\begin{aligned} \mathbf{A} &= \sum_{g\in\Gamma} b_g^2 \frac{4}{N}\mathbf{I}_g + 2\sum_{g\in\Gamma} b_g c_g \frac{4}{N}\mathbf{I}_g[\mathbf{v}_3]_\times + \sum_{g\in\Gamma} c_g^2 [\mathbf{v}_3]_\times{}^T \frac{4}{N}\mathbf{I}_g[\mathbf{v}_3]_\times \\ \mathbf{b}^T &= 2\sum_{g\in\Gamma} a_g b_g \mathbf{v}_3^T \frac{4}{N}\mathbf{I}_g + 2\sum_{g\in\Gamma} a_g c_g \mathbf{v}_3^T \frac{4}{N}\mathbf{I}_g + 4\sum_{g\in\Gamma} s_g b_g \mathbf{x}_{g\perp}^T \\ c &= \sum_{g\in\Gamma} a_g^2 \mathbf{v}_3^T \frac{4}{N}\mathbf{I}_g\mathbf{v}_3 + 4\sum_{g\in\Gamma} s_g b_g \mathbf{x}_{g\perp}^T\mathbf{v}_3 + \sum_{g\in\Gamma} x_{gs}. \end{aligned} \tag{98}$$

Similar equations can be written for the optimization of the loss function with respect to $\mathbf{v}_3$.

3.2.11 *2D trust-region optimization problem*

The optimization problem (97) can be efficiently solved by reducing it to the standard form of the *trust-region subproblem*. However, in our particular case, we can use one of the constraints in eq. (97) to project the optimization problem to a two-dimensional subspace. This allows us to solve it analytically, as we explain below.

First of all, it is convenient to chose an orthonormal basis $(\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_3)$ and rewrite the vector $\mathbf{v}_2$ in this basis as

$$\mathbf{v}_2 = \alpha\mathbf{v}_3 + x\mathbf{v}_x + y\mathbf{v}_y. \tag{99}$$

Then, the optimization problem (97) reduces to

$$\begin{aligned} \arg\min_{x,y} \quad & x^2\left(\mathbf{v}_x^T\mathbf{A}\mathbf{v}_x\right) + 2xy\left(\mathbf{v}_x^T\mathbf{A}\mathbf{v}_y\right) + y^2\left(\mathbf{v}_y^T\mathbf{A}\mathbf{v}_y\right) \\ & + x\left(2\alpha\mathbf{v}_x^T\mathbf{A}\mathbf{v}_3 + \mathbf{b}^T\mathbf{v}_x\right) + y\left(2\alpha\mathbf{v}_y^T\mathbf{A}\mathbf{v}_3 + \mathbf{b}^T\mathbf{v}_y\right) \\ & + \alpha^2\mathbf{v}_3^T\mathbf{A}\mathbf{v}_3 + \mathbf{b}^T\mathbf{v}_3 + c \\ \text{s.t.} \quad & x^2 + y^2 = 1 - \alpha^2. \end{aligned} \tag{100}$$

To solve it, we find stationary points of the corresponding Lagrangian $L(x, y, \lambda)$,

$$L(x, y, \lambda) = kx^2 + 2lxy + my^2 + 2px + 2qy + \lambda(x^2 + y^2 - 1 + \alpha^2), \tag{101}$$

with the following coefficients

$$\begin{aligned} k &= \mathbf{v}_x^T\mathbf{A}\mathbf{v}_x \\ l &= \mathbf{v}_x^T\mathbf{A}\mathbf{v}_y \\ m &= \mathbf{v}_y^T\mathbf{A}\mathbf{v}_y \\ p &= \alpha\mathbf{v}_x^T\mathbf{A}\mathbf{v}_3 + \frac{1}{2}\mathbf{b}^T\mathbf{v}_x \\ q &= \alpha\mathbf{v}_y^T\mathbf{A}\mathbf{v}_3 + \frac{1}{2}\mathbf{b}^T\mathbf{v}_y. \end{aligned} \tag{102}$$

Assigning the partial derivatives of the Lagrangian to zeros, we arrive to the following system of equations,

$$\begin{cases} kx + ly + p + \lambda x = 0 \\ lx + my + q + \lambda y = 0 \\ x^2 + y^2 = 1 - \alpha^2. \end{cases} \tag{103}$$

After eliminating λ we obtain

$$\begin{cases} lx^2 + (m - k)xy - ly^2 + qx - py = 0 \\ x^2 + y^2 = 1 - \alpha^2. \end{cases} \tag{104}$$

Finally, we exclude the last equation by changing the variables and introducing the new optimization variable t,

$$x = \frac{2t\sqrt{1-\alpha^2}}{1+t^2}; \quad y = \frac{(1-t^2)\sqrt{1-\alpha^2}}{1+t^2}. \tag{105}$$

Then, making the change of variables and multiplying the first equation by non-zero $(1 + t^2)^2$ we obtain,

$$\begin{aligned} & \left(-l(1-\alpha^2) + pt\sqrt{1-\alpha^2}\right)t^4 + 2\left((1-\alpha^2)(k-m) + t\sqrt{1-\alpha^2}q\right)t^3 \\ & + 6(1-\alpha^2)lt^2 + 2\left(1-\alpha^2)(-k+m) + \sqrt{1-\alpha^2}q\right)t - (1-\alpha^2)l - p = 0. \end{aligned} \tag{106}$$

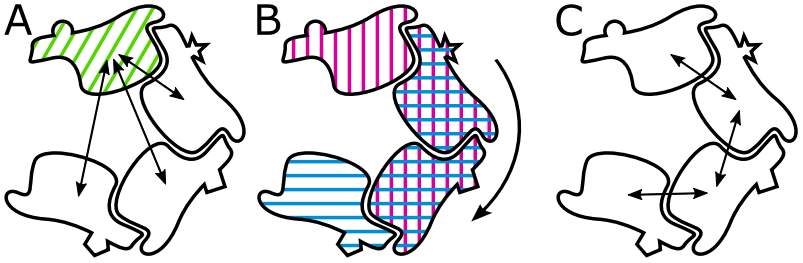


Figure 4: Assembly with C_5 symmetry and a missing subunit. **A**: The arrows show the comparisons made using the subunit with diagonal lines as the master subunit $\mathbf{a}_0$. **B**: With the same assembly, one rotation operator has been chosen, the part with vertical lines represents the *virtual reference* subunit and the part with horizontal lines is the *virtual target* subunit (they overlap). **C**: The arrows shows the comparisons resulting from the subunits' definition made in B.

This is our final fourth-order algebraic equation, whose roots can be found analytically [175]. After finding all of its roots, we discard the complex ones, then compute the corresponding values of x and y, substitute them in the original quadratic function (100) and choose the pair of x and y that gives the smallest value. We also additionally test the case of $y = -\sqrt{1-\alpha^2}$ and $x = 0$ that has been excluded during the change of variables in eq. (105).

3.2.12 *Choice of Symmetry Measure*

While the symmetry measure for the complete cyclic assembly is trivial and unique, there are multiple choices of this for partial assemblies. Indeed, in the later case the determined symmetry axis depends on the choice of the *master* subunit $\mathbf{a}_0$ and also on the performed comparisons. Figure 4A shows the simplest choice of the symmetry measure, where the *master* subunit is progressively superposed with every other subunit, while the other ones are only superposed with $\mathbf{a}_0$. The symmetry measure then reports the mean RMSD corresponding to the symmetry-constrained superposition of the *master* subunit with the rest of the assembly. Ideally, we would like to compare every subunit to every other subunit. However, this type of comparison makes the RMSD master equation 57 intractable using the presented techniques.

Therefore, orthogonally to the first approach, we can also choose a symmetry rotation operator and compare all the subunits that are superposed by this operator, as it is shown in Figures 4B-C. This can be seen as a redefinition of subunits by grouping all the matching subunits into new larger *virtual* subunits. More precisely, we can introduce a *virtual reference* subunit composed of all subunits that will be matched with other subunits by this operator. We can also introduce a *virtual target* subunit composed of all the subunits to which the *virtual reference* subunit matches. These *virtual* subunits are automatically perceived to contain the maximum number of individual subunits. We then compare these two *virtual* subunits, as it is shown in Figure 4B. This way, we uniquely define the symmetry measure for one rotation operator. This will report the mean RMSD corresponding to the subunits superposed by this operator.

The released version of our method implements the rotation operator approach, as it is shown in Figures 4B-C. Once the cyclic group to be tested is specified by the user, the software automatically tests each rotation operator of this group, and provides the best rotation axis and the resulting RMSD. We should specifically mention that in most of the practical cases we have assemblies with only two subunits. In this case, there is only one rotation operator that superposes the present subunits and the comparisons presented in Figure 4A and 4C will be equivalent to each other. In the examples we have encountered, the different results coming from the choice of different rotation operators

are very close to each other, and in the case where the symmetry is perfect, any chosen method will provide exactly the same result.

3.2.13 *Docking Master Equation*

Let us now move to the docking problem. Let us assume the functions $A(\underline{r})$ and $B(\underline{r})$ represent 3D shape-density functions of the two proteins, while $\underline{r}$ represents a spherical coordinate in 3D space, $\underline{r} = (r, \theta, \phi) \equiv (x, y, z)$. Without loss of generality, these can be geometrical 3D shapes, or, more generally, interaction potential fields.

Following the original Hex docking algorithm, $A(\underline{r})$ and $B(\underline{r})$ consist of linear combinations of 3D interior and surface skin density functions [215]. Thus, a 3D overlap integral of the form

$$S = K \int A(\underline{r})^* B(\underline{r}) \mathrm{d}\underline{r} \tag{107}$$

may be treated as a shape-based docking score, or pseudo interaction energy (the asterisk denotes complex conjugation of $A(\underline{r})$). While the functions $A(\underline{r})$ and $B(\underline{r})$ are initially entirely real, adopting the convention of conjugating one of these functions in the above overlap expression ensures that the docking score (taken as the real part of S) remains meaningful with complex functions. Indeed, by treating $A(\underline{r})$ and $B(\underline{r})$ as complex quantities, it is possible to accelerate the search over multiple candidate docking orientations using FFT techniques [215, 216]. In the subsequent analysis, we will use only the symbols $A(\underline{r})$ and $B(\underline{r})$ instead of the actual linear combinations for the sake of clarity.

In the rigid-body docking problem where the relative orientations of A and B are unknown, we adopt the convention that the centres of mass of proteins A and B are initially located at the origin, and we let the expression

$$A(\underline{r}) \longleftrightarrow \hat{T}(x, y, z)\hat{R}(\alpha, \beta, \gamma)B(\underline{r}) \tag{108}$$

represent a general interaction between protein A and a rotated and translated version of protein B. Consequently, the aim is to find the six parameters $(x, y, z, \alpha, \beta, \gamma)$ that give the most favourable interaction. The pair-wise docking score, S, that corresponds to the above interaction would be calculated as a 3D overlap integral of the form

$$S = \int A(\underline{r})^* \left[\hat{T}(x, y, z)\hat{R}(\alpha, \beta, \gamma)B(\underline{r})\right] \mathrm{d}\underline{r}. \tag{109}$$

Here, we represent protein shapes as SPF expansions of complex spherical harmonic, $Y_{lm}(\theta, \phi)$, and Gauss-Laguerre, $R_{nl}(r)$, radial basis functions

$$A(\underline{r}) = \sum_{nlm} A_{nlm} R_{nl}(r) Y_{lm}(\theta, \phi), \tag{110}$$

where the A_{nlm} are complex expansion coefficients (see [217]). Nonetheless, when working in the SPF domain, it is often more efficient to calculate one side of a given "docking equation" than the other. Thus, it is important to consider the most efficient order of operators for a given symmetry type.

3.2.14 *Docking Cyclic C_n Complexes*

With SPF basis functions, rotations and translations of SPF representations are most easily implemented with respect to the z axis. Hence, it is convenient to associate the z axis with the main (1D FFT) rotational and translational degrees of freedom (DOFs) and to associate the y axis with the principal rotational symmetry axis.

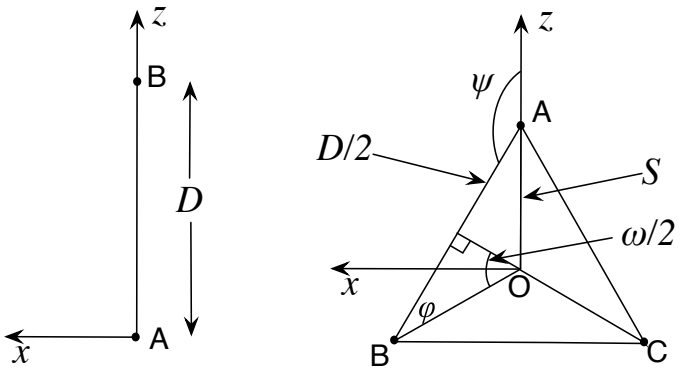


Figure 5: The coordinate systems used for pair-wise docking in C_n. The figure on the left shows the computational coordinate frame for a pair of monomers, A and B, with A at the origin in the xz plane and B at a distance D along the positive z axis. The figure on the right shows the symmetry frame of a C_3 trimer with the monomers arranged about the y axis (which points out of the plane towards the viewer). Here, $\omega = 2\pi/n$ is the C_n symmetry angle. From basic geometry, $S = D/(2\cos\phi) = D/(2\sin(\omega/2))$ is the distance from the principal symmetry axis to the centre of each monomer. We also have $\psi = \pi - \phi = (\pi/2 + \omega/2)$, which defines the rotation that relates the two coordinate systems.

Because an individual protein monomer is asymmetric, we normally have to assume that it can take any orientation in space relative to a set of fixed coordinate axes. Thus, describing a particular orientation of a given monomer, A, with respect to a random starting orientation will absorb three rotational DOFs. Let us suppose that the operator associated with that description is $\hat{R}(\alpha, \beta, \gamma)$. If we then copy the rotated A into an equally rotated monomer B, we can describe the docking interaction between of a pair of C_n symmetry mates by applying the following transformations:

$$\hat{R}_y(\omega_{j+1})\hat{T}_z(D)\hat{R}(\alpha,\beta,\gamma)B(\underline{r}) \longleftrightarrow \hat{R}_y(\omega_j)\hat{T}_z(D)\hat{R}(\alpha,\beta,\gamma)A(\underline{r}), \tag{111}$$

where the angles $\omega_j = 2\pi j/n$ are rotations around the principal symmetry axis. This equation highlights the fact that there exist only four degrees of freedom $(D, \alpha, \beta, \gamma)$ between the monomers in a complex with C_n symmetry. It is easy to demonstrate that the range of the α rotation angle must be restricted to $0 \leq \alpha < \pi$.

For a symmetric dimer or trimer, the above pair-wise A$\longleftrightarrow$B interaction is the only interaction that needs to be calculated. For $C_{n>3}$, there may also exist additional higher-order (i.e. 1$\longleftrightarrow$3, ... 1$\longleftrightarrow$(n/2+1)) interactions which should in principle be taken into account. However, these are likely to be small or negligible in most cases, and are ignored in the current work.

Nonetheless, a weakness of the above approach is that when n becomes large, it becomes necessary to translate each monomer far from the origin in order to achieve the desired separation between consecutive pairs of monomers. Such large translations can seriously reduce the resolution of the shape-density representations due to the exponential fall-off in the SPF radial basis functions. Therefore, in order to have expressions which involve only small translations, it is desirable to perform the SPF docking search near the origin, and to transform only the top solutions back to the symmetry frame. Figure 5 describes the problem graphically.

Thus, with the aid of Figure 5, it is preferable to begin instead with

$$\hat{T}_z(S)\hat{R}(\alpha,\beta,\gamma)A(\underline{r}) \longleftrightarrow \hat{R}_y(\omega)\hat{T}_z(S)\hat{R}(\alpha,\beta,\gamma)B(\underline{r}), \tag{112}$$

where $\omega = 2\pi/n$ and $S = D/(2\sin(\omega/2))$. To calculate this equation with A at the origin, we apply $\hat{T}_z(S)^{-1}$ to each side to give

$$\hat{R}(\alpha,\beta,\gamma)A(\underline{r}) \longleftrightarrow \hat{T}_z(S)^{-1}\hat{R}_y(\omega)\hat{T}_z(S)\hat{R}(\alpha,\beta,\gamma)B(\underline{r}). \tag{113}$$

Then, to locate B on the positive the z axis, we apply $R_y(-\psi)$ to each side, where $\psi = \pi/2 + \omega/2$ (see Figure 5), to obtain

$$\begin{aligned}&\hat{R}_y(-\psi)\hat{R}(\alpha,\beta,\gamma)A(\underline{r}) \longleftrightarrow \\ &\hat{R}_y(-\psi)\hat{T}_z(S)^{-1}\hat{R}_y(\omega)\hat{T}_z(S)\hat{R}(\alpha,\beta,\gamma)B(\underline{r}).\end{aligned} \tag{114}$$

It can then be shown that

$$\hat{R}_y(-\psi)\hat{T}_z(S)^{-1}\hat{R}_y(\omega)\hat{T}_z(S) = \hat{T}_z(D)\hat{R}_y(\omega)\hat{R}_y(-\psi), \tag{115}$$

where D is the distance between the two monomers. Furthermore, if we assume that we are starting from a random monomer orientation, we can "bury" the y-rotation by putting

$$\hat{R}_y(-\psi)\hat{R}(\alpha,\beta,\gamma) = \hat{R}(\alpha',\beta',\gamma') \tag{116}$$

to give

$$\hat{R}(\alpha',\beta',\gamma')A(\underline{r}) \longleftrightarrow \hat{T}_z(D)\hat{R}_y(\omega)\hat{R}(\alpha',\beta',\gamma')B(\underline{r}). \tag{117}$$

As shown below, we can use a 1D FFT search near the origin to determine the parameters $(D,\alpha',\beta',\gamma')$. We can then transform the solution back to the original coordinate frame by applying the operator $\hat{T}_z(S)\hat{R}_y(\psi)$ to each side. In other words, if the FFT search finds solutions $(D,\alpha',\beta',\gamma')_k$, the transformation matrix, $\underline{M}_k^A$, that should be applied to locate the A monomer on the positive z axis for the k^{th} docking solution is given by

$$\underline{M}_k^A = \underline{T}_z(S_k)\underline{R}_y(\pi/2+\omega/2)\underline{R}(\alpha'_k,\beta'_k,\gamma'_k). \tag{118}$$

Similarly, the docked B monomer may be located by applying the matrix

$$\underline{M}_k^B = \underline{T}_z(S_k)\underline{R}_y(\pi/2+\omega/2)\underline{T}_z(D_k)\underline{R}_y(\omega)\underline{R}(\alpha'_k,\beta'_k,\gamma'_k). \tag{119}$$

Because it can be seen that $\underline{M}_k^B = \underline{R}_y(\omega)\underline{M}_k^A$, it follows that all remaining symmetry mates may be generated from the coordinates of the A monomer.

Regarding the actual FFT calculation, by putting

$$A(\underline{r})' = B(\underline{r})' = \hat{R}(0,\beta',\gamma')A(\underline{r}), \tag{120}$$

and by exploiting the fact that $\hat{R}_z(\alpha')$ and $\hat{T}_z(D)$ commute, the docking equation in the computational frame becomes

$$\hat{T}_z(D)^{-1}A(\underline{r})' \longleftrightarrow \hat{R}_z(\alpha')^{-1}\hat{R}_y(\omega)\hat{R}_z(\alpha')B(\underline{r})' \tag{121}$$

or more simply

$$A(\underline{r})'' \longleftrightarrow \hat{R}_z(\alpha')^{-1}\hat{R}_y(\omega)\hat{R}_z(\alpha')B(\underline{r})'. \tag{122}$$

The Fourier series representation of the A monomer may be rotated and translated using

$$A'_{nlm} = \sum_{m'} D^{(l)}_{mm'}(0,\beta',\gamma')A_{nlm'} \tag{123}$$

and

$$A''_{nlm} = \sum_{kj} T^{|m|}_{nl,kj}(-D)A'_{kjm}, \tag{124}$$

where each $D^{(l)}_{mm'}(\alpha,\beta,\gamma)$ are matrix elements of the Wigner rotation matrices for the spherical harmonics [27] and each $T^{|m|}_{nl,kj}(D)$ is a translation matrix element for the SPF basis functions [214]. Then, writing the rotations for monomer B in terms of the Wigner rotation matrix elements (see eq. 18) gives

$$\begin{aligned}\hat{R}_z(\alpha')^{-1}\hat{R}_y(\omega)\hat{R}_z(\alpha')B(\underline{r})' = \sum_{nlm}\sum_{rpq} D^{(l)}_{mr}(-\alpha',0,0)\times \\ D^{(l)}_{rp}(0,\omega,0)D^{(l)}_{pq}(\alpha',0,0)B'_{nlq}R_{nl}(r)Y_{lm}(\theta,\phi),\end{aligned} \tag{125}$$

and hence

$$\begin{aligned}\hat{R}_z(\alpha')^{-1}\hat{R}_y(\omega)\hat{R}_z(\alpha')B(\underline{r})' = \sum_{nlmp} e^{-i(p-m)\alpha'}\times \\ d^{(l)}_{mp}(\omega)B'_{nlp}R_{nl}(r)Y_{lm}(\theta,\phi).\end{aligned} \tag{126}$$

Taking the complex conjugate of $A(\underline{r})''$ and integrating over the product with B then gives a $O(N^4)$ complexity docking score

$$S(\alpha';\omega,D,\beta',\gamma') = \sum_{nlmp} e^{-i(p-m)\alpha'} d^{(l)}_{mp}(\omega)B'_{nlp}A''^{*}_{nlm}. \tag{127}$$

Summing over n and l using

$$C_{mp} = \sum_{nl} d^{(l)}_{mp}(\omega)B'_{nlp}A''^{*}_{nlm}. \tag{128}$$

reduces this to

$$S(\alpha';\omega,D,\beta',\gamma') = \sum_{mp} C_{mp}e^{-i(p-m)\alpha'}. \tag{129}$$

The α' rotation (which here is restricted by symmetry to the range $0 \leq \alpha' < \pi$) may be scaled back onto the natural range of the FFT (see [216, 217]) by putting $\alpha'' = 2\alpha'$ and writing

$$e^{-is\alpha'} = \sum_{t} \lambda^{(\pi)}_{st} e^{-it\alpha''} \tag{130}$$

to obtain

$$S(\alpha';\omega,D,\beta',\gamma') = \sum_{mpt} C_{mp}\lambda^{(\pi)}_{p-m,t}e^{-it\alpha''}. \tag{131}$$

Finally, summing over m and p as

$$Q_t = \sum_{mp} C_{mp}\lambda^{(\pi)}_{p-m,t}. \tag{132}$$

gives a 1D Fourier series in α'':

$$S(\alpha';\omega,D,\beta',\gamma') = \sum_{t} Q_t e^{-it\alpha''}. \tag{133}$$

Because we now have a simple complex exponential on the right-hand side, this expression shows that for a given translation D and rotation (β',γ') the pair-wise docking score in an arbitrary C_n system may be calculated over a range of samples in α'' by using a 1D FFT.

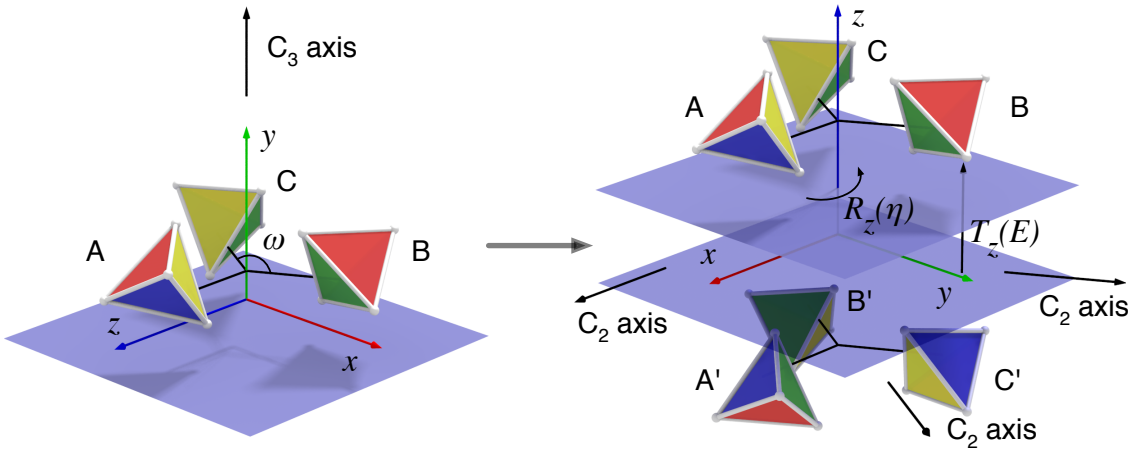


Figure 6: **Left**: Illustration of the C_3 point group symmetry with the y axis as the principal rotational symmetry axis and $\omega = 2\pi/n$. Each asymmetric protein monomer is represented by a tetrahedron having four differently coloured faces (red, green, blue, and yellow). **Right**: A D_3 system may be generated from two planar C_3 solutions but note the change of axes here with respect to the C_3 system on the left). When starting from a C_n solution, the D_n assembly problem has one translational and one rotational DOF, here denoted as $T_z(E)$ and $R_z(\eta)$, respectively. From symmetry, the rotational search range in $R_z(\eta)$ may be restricted to $0 \leq \eta < 2\pi/n$.

3.3 RESULTS AND DISCUSSION

3.3.1 *AnAnaS Computational Details*

We implemented the method using the C++ programming language. The method is called AnAnaS, which stands for Analytical Analysis of Symmetries. It is available as a standalone executable and also as a module with graphical user interface for the SAMSON software platform. We can also provide the source code upon request.

We have exhaustively assessed our program with all the structures labeled as symmetric in the PDB. This demonstrates the reliability and robustness of our method overall, and its heuristic for the discrete optimization steps in particular. Running the tests on all of these structures took us about 10 hours on a Windows laptop equipped with an Intel Core i7 @ 3.1 GHz CPU. For all the examples we tested, the running time was largely dominated by the multiple sequence alignment, which is required to compare the relevant alpha carbons in different subunits. Only in one case (2qzv) with a D_{48} symmetry, the computational bottleneck turned out to be the graph matching Indeed, the perception of dihedral and cubic groups is based on a robust determination of permutations between the assembly subunits corresponding to each rotation operator within the symmetry group.

This sequence alignment can be seen as a potential weakness in the procedure as it is not analytical. However, for homomeric assemblies, which are the most common ones, the alignment is trivial since all the chains have the same sequence. The alignment also prevents from comparing unrelated parts of different chains. Finally, it significantly reduces the number of possible matches between atoms in different chains and makes the method robust against inconsistencies in the input data.

Then, the formulation of the optimization problem takes time linear with the number of matched atoms, typically a few milliseconds. Finally, solution of the constrained quadratic optimization problems 81 takes only constant time and the solver of the trust-region subproblem converges to machine precision in 3-10 iterations, which takes a few microseconds.

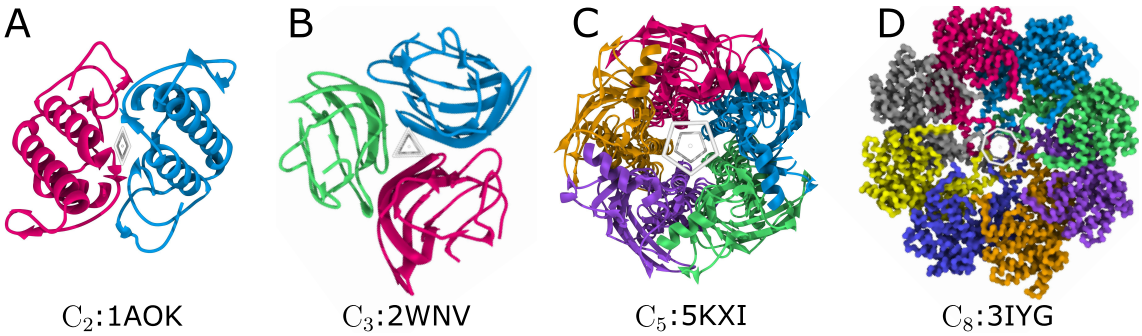


Figure 7: Example of symmetry detection of four pseudo-symmetrical assemblies with C_2, C_3, C_5, and C_8 symmetries. The determined symmetry axes are orthogonal to the screen. The order n of each axis is represented with a regular n-gone, except of order 2 shown with a rhombus. The corresponding RMSD symmetry measures are 1.406 Å, 2.226 Å, 1.613 Å, and 2.736 Å, respectively. This illustration and some of the illustrations below were produced in SAMSON (www.samson-connect.net).

We should also say that if no symmetry group is specified by a user, then the program exhaustively tests all the symmetry groups that are consistent with the number of chains in the input assembly. Also, we label an assembly as symmetric only if the corresponding RMSD measure is smaller than 7 Å and smaller than half of its radius of gyration. The second condition is added to filter out very small asymmetric assemblies.

3.3.2 *Pseudo-Symmetrical C_n detection*

We will first demonstrate our method on complete pseudo-symmetrical assemblies, for which we will determine the axis of symmetry and the RMSD measure. Pseudo-symmetrical assemblies are complexes that look symmetrical, however their sequences in different subunits are not the same. For the following example we have picked one pseudo-symmetrical assembly from each of C_2, C_3, C_5, and C_8 cyclic groups that are available in PDB. The PDB codes of these assemblies are 1AOK, 2WNV, 5KXI, and 3IYG, correspondingly. Figure 7 shows the output of our method. The RMSD symmetry measures for these assemblies are 1.406 Å, 2.226 Å, 1.613 Å, and 2.736 Å, correspondingly. The determined symmetry axes are shown with polygons.

3.3.3 *Reconstruction of assemblies with missing subunits*

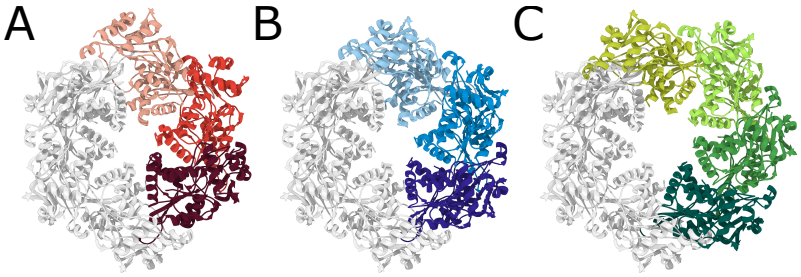


Figure 8: Cyclic reconstructions of the PDB structure 2GZA. The grey color corresponds to the asymmetric unit, which consists of three chains. **A**: In red we show the reconstruction of the assembly based on the crystallographic information. The corresponding RMSD measure is 4.85 Å. **B**: In blue we show the reconstruction made with the optimal C_6 axis. The corresponding RMSD measure is 2.74 Å. **C**: In green we show the reconstruction made with the optimal C_7 axis. The corresponding RMSD measure is 4.24 Å.

In the following example we will illustrate the possibility of finding the axis of symmetry of a partial assembly that does not pass through its COM. For this purpose we

Order	RMSD (Å)	Axis
C_4	12.39	(0.986, 0.161, -0.050)
C_5	5.61	(0.991, 0.129, -0.036)
C_6	2.34	(0.994, 0.110, -0.030)
C_7	3.93	(0.995, 0.097, -0.027)
C_8	6.20	(0.996, 0.089, -0.025)

Table 1: RMSD symmetry measures and the symmetry axes computed for several symmetry orders of the 2GZA structure.

will consider the PDB structure 2GZA. The asymmetric subunit of this structure contains three chains with identical sequence and crystallographic information explains that this subunit should be replicated two times around the x-axis to obtain the biological assembly.

From the three chains in the PDB file, we computed the RMSD for cyclic symmetries of different order. Table 1 lists the obtained results. We can see that the asymmetric unit present in the PDB file is consistent with a C_6 symmetry (RMSD of 2.34 Å), but a C_7 symmetry (RMSD of 3.93 Å) could also be possible. We should also mention that the found axes of symmetry are rather different from the x-axis provided by the crystallographic information. For example, for the C_6 case, the two axes have about 6 degrees of difference.

Using the computed axes, we can also reconstruct the C_6 and C_7 assemblies by a replication of the asymmetric unit for the C_6 case, and a replication of the asymmetric unit plus one more chain for the C_7 case. Figures 8B-C show the obtained assemblies. If we compute RMSDs for the reconstructed assemblies, we obtain the values of 2.74 Å for the C_6 reconstruction (Fig. 8B), 4.24 Å for the C_7 reconstruction (Fig. 8C), and 4.85 Å for the reconstruction from crystallographic information (Fig. 8A). The big difference between the symmetry measures obtained by reconstruction with and without the crystallographic information, and the fact that in a crystal this assembly is less symmetric than the C_7 reconstructed version, may suggest that this protein forms a C_7 assembly in solution and is forced to be in a C_6 conformation in a crystal.

3.3.4 *Completion by symmetry of AlphaFold2 predictions*

The AlphaFold2 (AF2) method has recently become the baseline not only for single-domain but also for multimeric predictions [35, 64, 66, 76, 108, 166, 276]. However, the complexity and the memory consumption of the AlphaFold2 algorithm scale quadratically with the size of the assembly. Therefore, it is often preferable to reconstruct only pair-wise or partial interactions within the assembly. As a result, a robust method is needed to reconstruct the full assembly from partial predictions. A few stochastic sampling approaches have just been proposed [35, 64]. However, we can exploit the known symmetry of the final assembly and reconstruct it from partial predictions using AnAnaS. We have applied this idea to a small benchmark composed of incomplete assemblies extracted from dihedral PDB structures of 1fo6, 1lk5, 1p8c, 1tqj. Then, we also tested it on several blind AF2 predictions that demonstrated multiple binding interfaces. Figure 9 shows one of the D_3 completions of AF2 partial predictions.

3.3.5 *Generation of perfectly symmetrical assemblies*

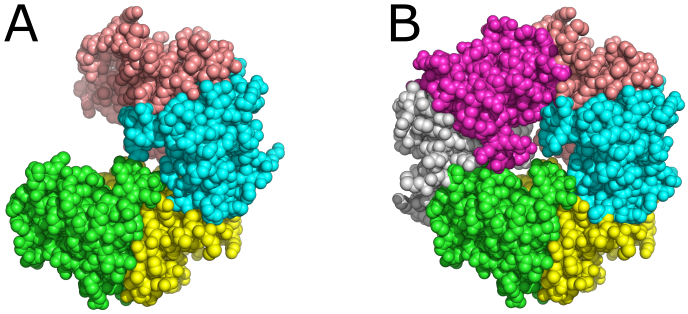


Figure 9: **A**: A partial prediction of a D3 assembly by AF2. **B**: Symmetric completion by AnAnaS.

A particularly interesting task in molecular modeling and crystallographic applications is to use an approximately symmetrical assembly as a starting model and generate a perfectly symmetrical structure from it. As a starting structure one can use an assembly from molecular dynamics simulations, a pseudo-symmetrical assembly, or the one with non-crystallographic symmetry, for example. Then, we proceed by computing the best C_n axis from the initial model. After, we choose one of the subunits as a 'master' subunit and replicate it around this axis to obtain the perfectly symmetrical assembly. Figure 10 illustrates this approach when using a pseudo-symmetrical C_3 assembly (PDB code 2IX2) as an input structure. This structure is composed of three chains with two different sequences. The RMSD measure of this structure is 6.20 Å. The symmetrized assembly is perfectly symmetrical and obviously has the RMSD measure of 0 Å.

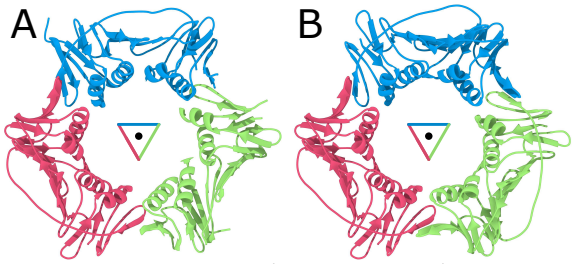


Figure 10: **A**: A pseudo-symmetrical C_3 assembly (PDB code 2IX2) with the axis of symmetry shown with the triangle. Its three chains are shown with three different colors and are slightly different from each other. **B**: The symmetrized version of this assembly. Here, we arbitrarily chose the red chain from the complex in A and replicated it to obtain the perfectly symmetrical assembly.

3.3.6 *Comparison of AnAnaS with other methods*

In order to demonstrate the efficiency of our approach, we compared it with two other published techniques. The first one was developed by Dryzun, Zait, and Avnir [61], and we will refer to it as to CSM (Continuous Symmetry Measure). It considers all the atoms in the input assembly and finds the symmetry axis by alternatively refining the axis of rotation and the permutation between the atoms. Table 3 lists all the cyclic examples found in the CSM article [61]. We should note that Dryzun, Zait, and Avnir [61] report either the symmetry measure or the computational time. The CSM symmetry measure can easily be converted to the RMSD symmetry measure by the following equation,

$$\mathrm{RMSD}^2 = \frac{\mathrm{CSM} \times R_g^2}{50}, \tag{134}$$

where R_g is the radius of gyration of the assembly. The second method is from Levy et al. [142], and will be called Levy. It exhaustively scans a finite set of axes of symmetry and chooses the best one. Unlike the previous technique, it has to be fed with lists of atoms organized in subunits. Therefore, to prepare the input, we used the same alignment procedure as we implemented in our method, and we used parameters suggested by the author.

Table 3 lists the execution time and the symmetry measure (RMSD value) for the three tested methods. It shows that our method scales with the size of the input assembly

PDB Code	Group	RMSD(AnAnaS)	RMSD(CSM)	RMSD(Levy)	AnAnaS Time[a]	CSM Time[b]	Levy Time[a]
1HPV	C_2	0.23 Å	-	0.23 Å	0.02 s	1.9 s	0.11 s
1LGN	C_5	0.20 Å	-	0.36 Å	0.15 s	34 s	1.02 s
1NN2[c]	C_4	0.00 Å	-	0.00 Å	0.19 s	77 s	0.77 s
2FKW	C_9	0.28 Å	-	0.81 Å	0.15 s	1175 s	3.9 s
2XE2	C_3	0.12 Å	0.23 Å	0.12 Å	0.11 s	-	0.42 s
3FV9	C_8	27.7 Å	19.8 Å	>7 Å	0.73 s	-	7.32 s
3FV9	C_4	0.48 Å	7.6 Å	0.60 Å	0.73 s	-	1.36 s
3KML	C_{17}	0.36 Å	0.45 Å	0.67 Å	1.7 s	-	74 s

[a] AnAnaS and Levy times were measured on a Windows laptop equipped with an Intel i7 @ 3.1 GHz.
[b] CSM times were taken from [61] with a different, a 7 year older, CPU. However, we believe that the order of magnitude of these timings is still correct.
[c] For this structure, the biological assembly was used.

Table 2: Comparative results between AnAnaS, CSM and Levy methods tested on cyclic examples collected from the CSM paper [61].

much better than the two other methods. Indeed, its runtime typically stays below one second, even for large assemblies.

On all the tested examples, our method is significantly faster than the one from Levy and it also produces a lower RMSD measure. In practice, we obtain the same RMSD when the actual symmetry axis is among the ones sampled by Levy's method. Comparison to CSM is a bit more difficult because this method considers more atoms (reference points) than we do, and also because we do not have the computed axes for the analysis. These additional atoms can explain small differences in the computed RMSD values. We should note that more freedom in choosing the correspondence between the atoms can significantly lower RMSD in poorly symmetrical assemblies. These two effects explain the small differences in the 2XE2 and 3KML examples, and also the difference in the 3FV9 example when measuring the C_8 symmetry. However, we believe that the iterative process of CSM was stuck in a local minimum when measuring the C_4 symmetry. Indeed, visual inspection reveals that the 3FV9 assembly has a D_4 symmetry that seems of a very high quality, thus it is not possible that the average deviation between the different dimers is more than 7Å, as reported by CSM. In this example, the dihedral symmetry makes the 4-fold axis much more difficult to detect by CSM, because several 2-fold axes are also present.

3.3.7 *High-order Symmetry Examples*

Figure 11 presents an example of symmetry axes detection for each of the cubic groups, i.e. tetrahedral, octahedral and icosahedral, and for a dihedral group of order 6. These assemblies do not possess any particular computational difficulty.

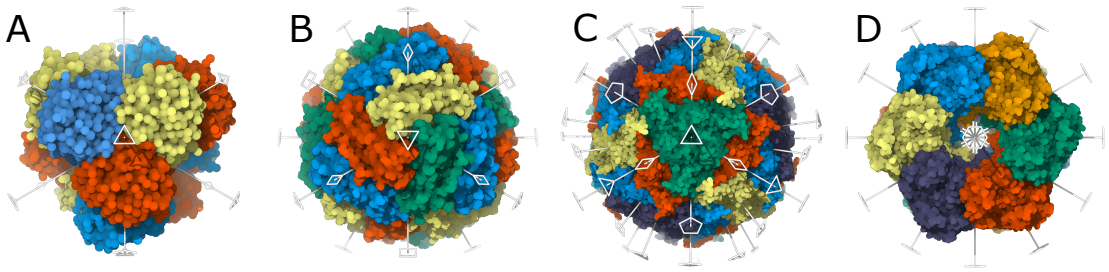


Figure 11: Four examples of symmetric assemblies with their axes. All of these are seen from a 3-fold axis except for the last one, seen from a 6 fold axis. The order n of each axis is represented with a regular n-gone, except of order 2 represented with a rhombus. **A**: A tetrahedral assembly (1doi) with the RMSD loss of 0.36 Å. **B**: An octahedral assembly (1bfr) with the RMSD loss of 0.22 Å. **C**: A perfect icosahedral assembly (1stm) with the RMSD loss of 0.0 Å. **D**: A dihedral D_6 assembly (1f52) with the RMSD loss of 0.20 Å.

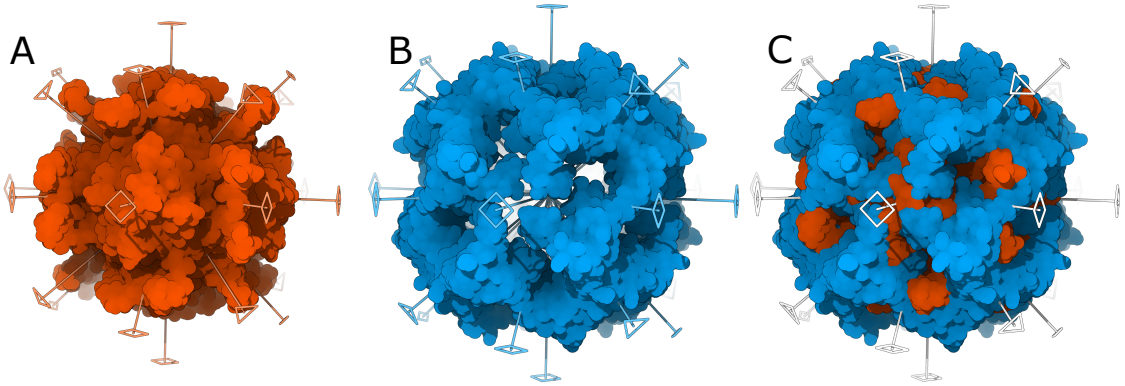


Figure 12: The 5tov octahedral assembly. The homologous chains are colored with the same color. **A**: The chains of the first type form an octahedral assembly with the RMSD loss of 2.94 Å. **B**: The chains of the second type also form an octahedral assembly with the RMSD loss of 2.67 Å. The axes are slightly different from the first assembly, with about 1° of difference. **C**: The axes are computed for the full assembly, with the RMSD loss of 2.83 Å.

Some assemblies contain more chains than the number of asymmetric subunits expected from their point group symmetry. Each subunit thus must be composed of several chains. For example, Figure 12 shows the 5tov structure, which is an octahedral assembly with 48 chains and a stoichiometry of A24B24. This example demonstrates that our method determines symmetry axes in assemblies where the asymmetric subunits are composed of multiple chains. We should also note that in this case it is important to rigorously take into account all the chains, since the angular difference in the axis determination can be as large as 1° if only chains A or B are considered.

While scanning the PDB, we found several assemblies that are classified with a low-order symmetry group, but can alternatively possess a higher symmetry group. For example, Figure 13 shows the 1ocw structure, which is a perfect C_4 assembly with a stoichiometry of A4B4 and the RMSD loss of 0 Å. Our algorithm also detects a D_4 pseudo-symmetry with the RMSD loss of 2.68 Å, which is rather low. The visual inspection of this protein confirms this possibility (see Fig. 13). Similarly, we also discovered some assemblies with cubic symmetries that were labelled as cyclic in the PDB database. Figure 14 shows two of such examples. One is the 4itv protein labelled as C_2 (RMSD loss of 4.44 Å), but also possessing a tetrahedral symmetry with the RMSD loss of 10.94 Å. The other is the 5hpn protein labelled as C_5 (RMSD loss of 0.68 Å), but also possessing an icosahedral symmetry with the RMSD loss of 0.56 Å.

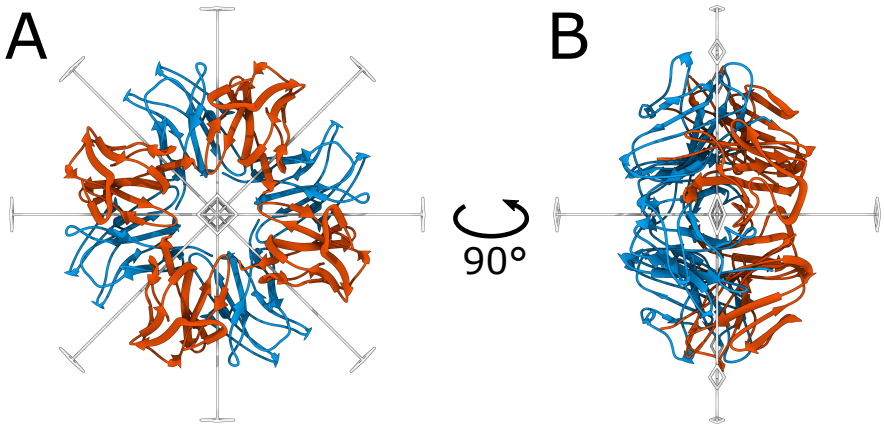


Figure 13: The 1ocw protein colored in blue for the A chains and red for the B chains. **A**: as seen from the 4-fold axis. **B**: as seen from a 2-fold axis computed with our method.

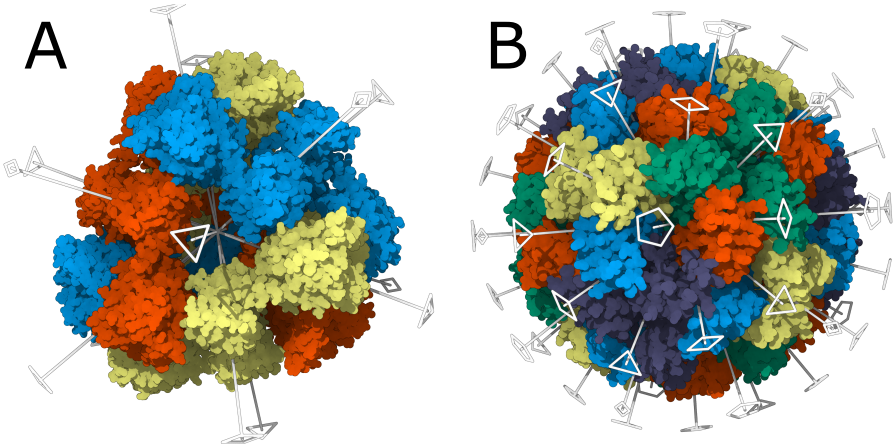


Figure 14: **A**: The 4itv protein classified in PDB as C_2 (RMSD loss of 4.44 Å), also has a tetrahedral symmetry with the RMSD loss = 10.94 Å. **B**: The 5hpn protein classified in PDB as C_5 (RMSD loss of 0.68Å), also has an icosahedral symmetry with the RMSD loss = 0.56 Å.

3.3.8 *Comparison with other methods*

We compared our approach with two other published methods following the comparison strategy from our previous work on symmetry detection in cyclic protein assemblies [186]. More precisely, we compared it to the results published by David Avnir and colleagues [61, 198]. We will refer to it as to CSM (Continuous Symmetry Measure). We also compared our method to the the one from Emmanuel Levy [142], and will refer to it as to Levy. Please refer to the first part of our paper [186] for more details.

For the comparison, we have selected all dihedral assemblies presented in the original CSM publications [61, 198]. These are listed in Table 3. We have also complemented these assemblies with three examples of cubic groups, 5x47 with tetrahedral symmetry, 4p18 with octahedral symmetry, and 4zor with icosahedral symmetry.

Table 3 lists the execution time and the symmetry measure (RMSD value) for the three tested methods. As in the cyclic case [186], it clearly shows that our method scales with the size of the input assembly much better than the two other methods. This is especially notecable for large assemblies. Regarding the accuracy of the obtained results, it is typically much better than in the Levy method for high-order symmetries. As we have mentioned in the first part of this work, comparison to CSM is trickier because this method considers more atoms than we do. Therefore, the additional atoms add more freedom to the CSM method when it chooses the correspondences between these, which can explain small differences in the computed RMSD values. For example, in the 1f52 case CSM reports a smaller RMSD measure than we do (0.15 Å vs. 0.19 Å).

3.3.9 *Exhaustive analysis of symmetric structures in the PDB*

To demonstrate the efficiency of our approach, we exhaustively analyzed all the structures labelled as symmetric in the PDB. To do so, we downloaded their biological assemblies (about 40,800 cyclic, 9,800 dihedral and 1,300 cubic examples as for January 2018) and assessed the symmetry for each of these. Figure 15 plots the distribution of the RMSD symmetry measures for assemblies with different types of symmetry. We should note that there

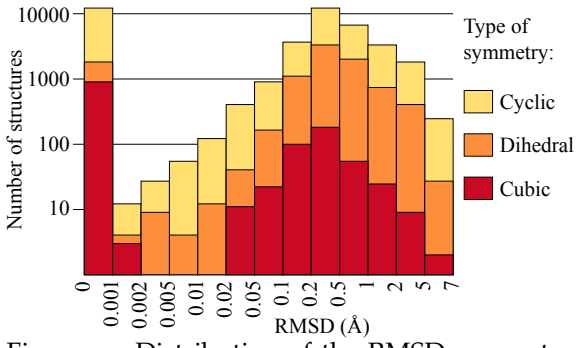


Figure 15: Distribution of the RMSD symmetry measure for different types of symmetry shown in a log-log scale.

PDB Code	Group	RMSD(AnAnaS)	RMSD(CSM)	RMSD(Levy)	AnAnaS Time[a]	CSM Time[b]	Levy Time[a]
1mso[c]	D_3	1.36Å	-	1.39Å	0.13s	3.7s	0.49s
2hhb	D_2	1.64Å	2.43Å	1.64Å	0.05s	12.2s	0.28s
2nwc	D_7	0.81Å	-	0.89Å	0.63s	3950s	2.3s
2rgw	D_3	0.34Å	0.39Å	0.47Å	0.23s	-	1.8s
1odi	D_3	0.35Å	0.50Å	0.47Å	0.14s	-	1.5s
1f52	D_6	0.19Å	0.15Å	0.54Å	1.21s	-	16.6s
5x47	T	0.85 Å	-	1.02 Å	0.32 s	-	5.62 s
4p18	O	0.19 Å	-	2.13 Å	3.1 s	-	131 s
4zor[c]	I	1.05 Å	-	2.38 Å	18.8 s	-	1118 s

[a] AnAnaS and Levy times were measured on a Windows laptop equipped with an Intel i7 @ 3.1 GHz.
[b] CSM times and CSM symmetry measures were taken from [61] and [198] with a different, 7 year older, CPU. However, we believe that the order of magnitude of these timings is still correct.
[c] For these structures, the biological assembly was used.

Table 3: Comparative results between AnAnaS, CSM and Levy methods for dihedral and cubic molecular assemblies.

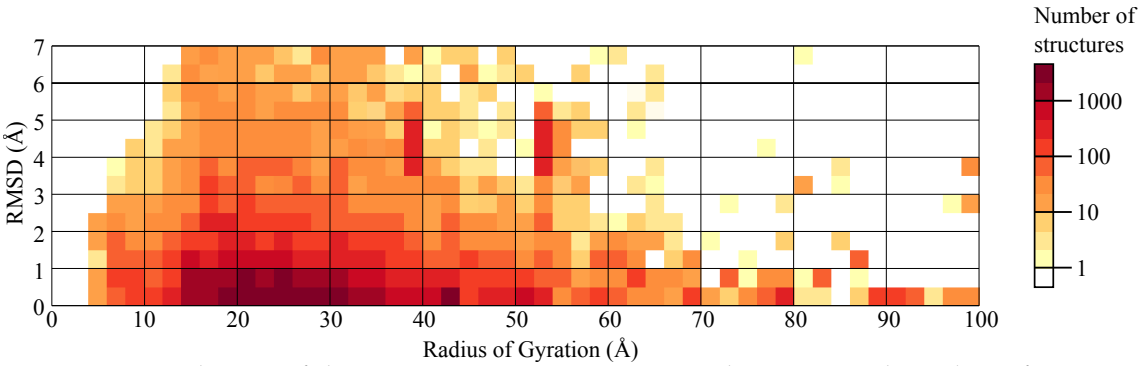


Figure 16: Distribution of the RMSD symmetry measure with respect to the radius of gyration for non-perfectly symmetric assemblies from PDB.

are many structures with a very low RMSD value (< 0.001Å), which is the precision of the pdb format. These are typically obtained by replicating subunits with crystallographic symmetry or BIOMT transforms, so they have a perfect symmetry. Regarding all other structures, we can see that all the three distributions of cyclic, dihedral, and cubic groups follow the same law in the log-log scale. The maxima of the distributions belong to the range of 0.2-0.5 Å, and there are no noticeable differences between the shapes of all of these.

Another interesting question we are able to answer using our tool is whether the degree of asymmetry is related to the size of the assembly under consideration. In other words, we can study if the RMSD symmetry measure is related with the radius of gyration of the symmetric assemblies. A geometrical intuition would suggest that as the angular uncertainty should stay constant with the size of the assembly, and of protein assembly grows larger, the imperfections of its symmetry become more pronounceable. Visually, we would expect a linear correlation between the RMSD symmetry measure and the radius of gyration of the assemblies. However, it is not the case in reality. Indeed, Figure 16 does not demonstrate any clear relation between the size and the imperfection of the PDB assemblies, and the correlation between these two variables is only about 0.1. Interestingly enough, large assemblies are very well organized with sufficiently small values of the RMSD measure. This is one of the reasons behind our choice of RMSD as the symmetry measure instead of its normalization by the size of the structure (as it is often done in other methods [61, 198]). We should specifically add that in the case of very small assemblies, we consider them symmetric only if the corresponding RMSD measure is smaller than half of the radius of gyration of the assembly.

3.3.10 *How good are symmetry annotations in the PDB?*

Our tool also allows to assess the overall quality of annotations of symmetric assemblies in the PDB. More precisely, we compared the highest symmetry group suggested by our method with the group provided in the PDB. If these two groups are different, there are two types of possible errors. First, one of the two groups can be a subgroup of the other one (e.g. C_4 is a subgroup of D_4). This type of errors simply results from a difference of sensibility between the annotation methods. We call the groups *compatible*. Second, the two groups may also be *incompatible* (e.g. C_4 and D_5). This case means that one of the two results is wrong and a careful visual inspection is generally required.

Table 4 lists the results for 51,358 PDB structures. In 50,378 cases (98.1% of all the cases), the symmetry group annotated by the PDB is the one found by our method. These cases are located at the green diagonal of the table. Red cells show the incompatible groups, while white cells show the compatible groups. Our method is generally more sensitive compared to the PDB annotation. Indeed, there are 845 structures (1.6%) for which it finds a higher order compatible group, while only in 125 cases (0.2%) the PDB annotated compatible group has a higher order. Finally, there are only 13 cases (0.03%) that present incompatible groups. We have visually inspected all of these structures. The two of these annotated as T and detected as C_5 are 4aod and 4aoe, for which the biological assemblies are indeed C_5. The 11 other cases have uncertainties between C_2 and C_3 annotation. In all of these cases, both symmetries are detected by our method, and the difference of RMSD between the two symmetries is smaller than 1 Å. Moreover, some of these examples have less than 5 amino acids in each chain, and are at the limit of the usability of the annotation techniques. We can also mention two particular cases. One is 3alz, for which both perfect C_3 and C_2 axes are detected, and is actually a part of a D_3 assembly. The other is 3aqq, which is annotated as C_2 in the PDB, but looks much more like a partial C_3 assembly.

The first column of Table 4 lists 75 structures for which AnAnaS was not able to detect symmetry. There are 4 reasons that explain this:

- For the 6 icosahedral structures, we ran out of memory at the discrete optimization step. Thus, no results were outputted and we considered these cases as assymetric.
- Some structures have missing or additional chains that are not supported by our program. For example, 2zl2 has a D_7 symmetry but contains 24 chains, 10 of them being very small peptides. AnAnaS expects a multiple of 14 chains as input to test a D_7 symmetry and, therefore, does not test it. However, if we remove these small peptides, we detect a D_7 symmetry with an RMSD of 0.35 Å.
- Some structures are at the edge of the threshold that we set up for the assemblies to be symmetric. More precisely, as we explain it below, RMSD must be smaller than 7 Å and also smaller than half of the radius of gyration.
- Finally, some structures do not possess the symmetry annotated in the PDB. For example, 2ol9 is the structure of two identical peptides translated with respect to each other, and these are annotated as C_2, while a C_2 symmetry would necessarily require a rotation between the two peptides.

3.3.11 *SAM Results*

To test our symmetry assembly approach, we selected a representative example structure of each complex symmetry type for which 3D structures exist in the 3D-Complex database. These examples are listed in Table 5. For each complex, we manually extracted

PDB \ AnAnaS	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	D_2	D_3	D_4	D_5	D_6	D_7	D_8	T	O	I	Total
C_2	54	33091	8	23		6			470	15	7	1	1	205	2				33883
C_3	2	3	4188			16				60									4269
C_4	1	2		1046				7			4								1060
C_5	6				561							1							568
C_6		2	2			411							1						416
C_7							104							6					110
C_8								34							3				37
D_2	3	26							6571		6		1			2			6609
D_3		8	5							1939									1952
D_4	1	1									654				5				661
D_5		1										236							237
D_6													106						106
D_7	1						1							99					101
D_8															34				34
T					2											359	3		364
O																	329		329
I	6															2		617	625

Table 4: Summary of the symmetry groups annotated in the PDB (rows) against the ones discovered by AnAnaS (columns). Red cells mark incompatible groups, while white cells mark compatible groups and green cells mark identical groups. For example, first cell shows that there are 54 structures annotated as C_2 in the PDB for which we found a C_1 symmetry (i.e. no symmetry).

the first monomer from the PDB file to serve as the A monomer, and we applied the SPF assembly algorithm for the given symmetry type using SPF expansions to polynomial order N=30.

More specifically, for the C_n correlation search (and for the initial trimeric search in the higher symmetry types), the (β, γ) angular samples were generated from an icosahedral tessellation of the sphere with 812 sample vertices with an angular separation between the vertices of approximately $7.5°$. The FFT search in α was calculated using 64 steps of approximately $2.8°$ in the first hemisphere, and up to 64 translational steps of 0.8 Å were applied starting from an initial inter-monomer distance estimated from the monomer radius. Thus, a total of approximately 6×10^6 trial $A_1 \longleftrightarrow B_1$ orientations were generated and scored in the FFT search. The B_1 monomers of the generated solutions were then clustered using a greedy clustering algorithm with a 3 Å RMSD cluster threshold in order to remove near-duplicate solutions, and the top-scoring member of each of the first 100 clusters were retained as distinct solutions. For the C_n complexes, any remaining monomer coordinates were generated by symmetry, and the top 100 solutions were saved as PDB files. When calculating the FFT correlations in parallel using these parameters, it takes approximately 30 seconds to generate 100 C_n complexes on a dual processor workstation with two 2.3 GHz E4510 Intel Xeon processors (8 cores in total).

For the D_n, T, O, and I complexes, similar angular and translational search parameters were then used again in the subsequent trimeric assembly search using the top 100 trimeric solutions. For these complexes, the calculation time is governed by the cost of constructing the trimeric pseudo-molecules and the cost of performing the subsequent correlation search explicitly, without the benefit of a FFT. Typical execution times are between 60 and 90 seconds per complex.

To assess the quality of the generated complexes, the coordinates of the crystallographically determined complex structure were used as a reference structure with which to calculate root-mean-squared deviations (RMSDs) between the calculated and reference monomer coordinates. For all of the examples in Table 5, the "Rank-C_n" and "RMSD-

			M-Zdock	M-Zdock	M-Zdock	SymmDock	SymmDock	SymmDock	SAM	SAM	SAM	SAM	SAM	SAM
PDB	#Res	Sym	Rank-C_n	RMSD-B_1	Time	Rank-C_n	RMSD-B_1	Time	Rank-C_n	RMSD-B_1	RMSD-B_2	Rank	RMSD	Time
1M4G	182	C_2	N/F	N/F	5963	26	21.47	6	1	1.82	–	1	1.82	45
1F7O	117	C_3	1	2.33	4641	1	2.32	14	1	2.82	–	1	2.82	48
1F8C	389	C_4	1	2.00	11171	1	2.37	62	1	2.04	–	1	2.04	40
1G8Z	104	C_5	1	1.87	3187	1	2.02	15	1	1.62	–	1	1.62	43
1GL7	412	C_6	1	1.41	14228	1	1.41	40	1	0.68	–	1	0.68	50
1I81	75	C_7	1	1.95	2571	1	4.02	7	1	1.17	–	1	1.17	43
1V5W	240	C_8	1	2.49	7354	1	2.93	14	1	2.51	–	1	2.51	44
1QAW	68	C_{11}	1	2.61	2196	1	1.75	5	1	1.09	–	1	1.09	43
1XIB	389	D_2	–	–	–	–	–	–	1	1.01	0.68	1	0.86	319
1GUN	68	D_3	–	–	–	–	–	–	2	1.35	0.99	1	1.19	308
1B9L	120	D_4	–	–	–	–	–	–	1	1.34	1.57	1	1.46	393
1L6W	221	D_5	–	–	–	–	–	–	1	1.26	3.61	5	2.70	479
1ZNN	246	D_6	–	–	–	–	–	–	1	1.34	1.92	1	1.66	439
1YG6	194	D_7	–	–	–	–	–	–	1	1.94	3.30	1	2.70	381
1Q3R	519	D_8	–	–	–	–	–	–	2	3.65	10.83	25	7.98	397
2CC9	65	T	–	–	–	–	–	–	1	1.97	2.63	1	2.32	199
1IES	175	O	–	–	–	–	–	–	1	1.24	0.94	1	1.10	201
1HQK	155	I	–	–	–	–	–	–	1	1.45	1.88	1	1.68	200

Table 5: Example symmetrical complexes assembled from a single monomer by the SAM algorithm with N=30. Here, #Res denotes the number of residues in one monomer of each structure, B_1 denotes the B monomer of the first C_n system, and B_2 denotes a B monomer of the second ring system in D_n complexes or of an adjoining C_3 trimer for T, O, and I complexes. All RMSD values are in Å units and all times are elapsed seconds for a Linux workstation with dual 6-core (2.67 GHz) Intel X5650 processors. "N/F" denotes not found. A hyphen denotes not applicable.

B_1" columns show the rank and RMSD for the first B monomer of the C_n complexes (or the trimeric component in the higher symmetry cases) found within 10 Å of the crystal structure. This column shows that in all but one case (1GUN), our 1D FFT search is correctly identifying a near-native interface between the A and B monomers. Given that this calculation is rigidly assembling monomers which should fit perfectly, these very good results are not especially surprising. Nonetheless, these figures confirm that our FFT correlation expressions are implemented correctly. Figure 17 shows cartoon representations of the first near-native solution found for each complex.

In order to compare the performance of SAM with some examples of existing symmetry docking algorithms, we selected M-Zdock [193] as a good example of a FFT-based algorithm and SymmDock because it is based on a geometric hashing technique [223]. Table 5 shows that these algorithms can also successfully find rank-1 solutions with low RMSDs for all of our C_n examples (both M-Zdock and SymmDock were designed only for C_n complexes) except for the first C_2 structure (1M4G) for which M-Zdock does not find a solution in its top 10 predictions and for which SymmDock finds a very poor solution only at rank 26. However, if we consider the 7 examples (C_3 to C_{11}) for which all three algorithms produce rank-1 solutions, Table 5 shows that SymmDock is approximately twice as fast as SAM, while SAM is approximately 130 times faster than M-Zdock, with average execution times of 23s for SymmDock, 44s for SAM, and 5,734s for M-Zdock. Furthermore, the RMSD-B_1 columns of this table show that SAM often gives considerably better quality solutions, with average RMSD values of 1.70 for SAM, 2.09 for M-Zdock, and 2.40 for SymmDock. These results show that SAM performs quite favourably when compared to these previous approaches.

In order to assess the trimeric pseudo-molecule assembly step for the D_n, T, O, and I complexes, the "RMSD-B_2" column of Table 5 reports the best RMSD found by SAM for the calculated coordinates of the B_2 monomer. This column shows that our strategy of scoring the interactions between trimeric pseudo-molecules works very well for all of the examples except for the D_8 complex (PDB code 1Q3R). Finally, the "Rank" and "RMSD" columns give the rank and overall RMSD of the first B_1 and B_2 solutions found within 10 Å of the crystal structure. These columns show that in 16 out of the 18 examples, the first solution calculated by SAM corresponds very closely to the crystal structure.

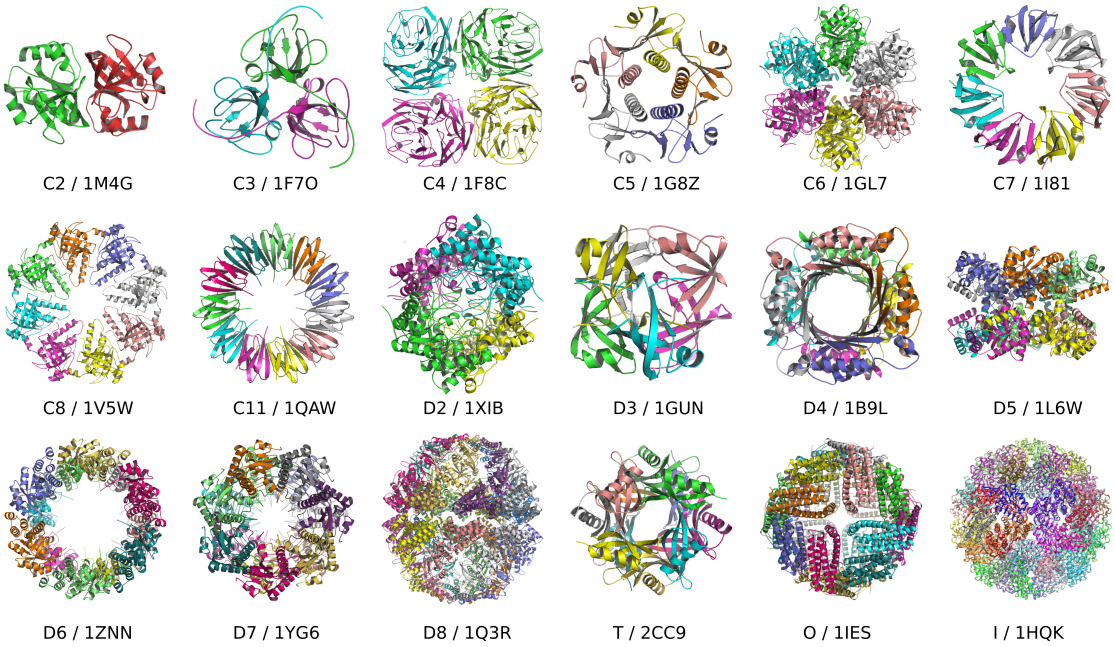


Figure 17: The example symmetrical complexes assembled by SAM, starting from a single monomer from the crystal structure. Computational details are provided in Table 5.

For the D_5 example (PDB code 1L6W), the first near-native structure is found at rank 5. Although a good trimer is found at rank 2 for the D_8 example (PDB code 1Q3R), the subsequent trimer assembly step finds a rather poor near-native orientation only at rank 25.

A limitation of the SPF approach is that most of the zeros in the basis functions appear within about 50Å from the origin. This means that very large protein domains, typically greater than about 500 residues, cannot be represented accurately by a single SPF expansion. We believe that this explains the poor performance of the 1Q3R example (519 residues per monomer). Taking into account the possibility that one monomer might consist of several chains, we have calculated that 87% (9,024/10,176) of C_n complexes, 91% (2,704/2,965) of D_n complexes, and 43% (60/139) of the T (43/86), O (12/47), and (non-viral) I (5/6) complexes in the 3D-Complex database have less than 500 residues per monomer. In other words, we estimate that SAM could be usefully be applied in approximately 89% of protein docking problems that involve point group symmetry. One way to circumvent the monomer size limitation would be to use a coarse-grained force-field model to perform the trimeric assembly step, for example. Indeed, since FFT correlation function used here is based on a simple surface skin density model of protein shape [215], it would be advisable to refine and re-score the SAM models using a conventional molecular mechanics force field if clash-free atomic models are required.

While this approach focuses on complexes having point group symmetry, we expect it would be relatively straight-forward to extend the SAM algorithm to deal with complexes having translational symmetry such as cylindrical and helical structures. Cylindrical structures could be made in the same way that we make a D_n complex from two C_n systems, but without applying a flip ($\hat{R}_y(\tau_1)$). Helical structures could be made by introducing an additional translational DOF in our C_n assembly algorithm. This would correspond to replacing $\hat{R}_y(\omega)$ with $\hat{T}_y(\eta)\hat{R}_y(\omega)$ throughout Section 3.2.14, where $\hat{T}_y(\eta)$ represents a translation along the major helical axis.

The SAM program may be downloaded for academic use at *http://sam.loria.fr/*.

3.4 CONCLUSION

This Chapter presents an efficient computational approach to assess the quality of point-group symmetries in macromolecular assemblies called AnAnaS. We express the quality through the symmetry measure using a Euclidian 3D distance. We showed that the problem of finding the best symmetry axis can be formulated as a constrained quadratic optimization problem and provided an efficient solution to it. More precisely, using the quaternion arithmetic, we expressed the rotation operators through quadratic forms with constraints. This allowed us to find the unique solution using efficient methods developed for the trust-region sub-problem. We have demonstrated the efficiency of the method on several examples including partial assemblies and pseudo symmetries. We have also compared the presented method with two other published techniques and showed that out method is significantly faster and generally much more robust and efficient on all the tested examples.

We have demonstrated the efficiency of our method on all the structures marked as symmetric in the PDB, including those with multiple chains per asymmetric subunit or with pseudo-symmetry. It allowed us to verify symmetry annotations in the PDB and detect several inconsistencies in the annotations. For example, in 1.6 % of the cases, we detected a higher symmetry group compared to those provided in the PDB. We have also compared structural organization of protein assemblies with different point group symmetries and concluded that these follow the same distribution laws. Finally, we have detected that the angular impurity in symmetry does not scale with the size of the assemblies. More precisely, very often these are the largest and high-order symmetry systems that are organized the most regularly. The method is available at https://team.inria.fr/nano-d/software/ananas/.

This Chapter also presents a novel FFT-based approach called SAM for building models of protein complexes with arbitrary point group symmetry. The basic approach relies on a novel and very fast 1D symmetry-constrained spherical polar FFT search to assemble cyclic C_n systems from a given protein monomer. Structures with higher order (D_n, T, O, and I) symmetries may be built by performing a subsequent symmetry-constrained Fourier domain search to assemble trimeric pseudo-molecules. Overall, our results demonstrate that the SAM algorithm can correctly and rapidly assemble protein complexes with arbitrary point group symmetry from a given monomer structure in 17 out of 18 test complexes. The main limitation of our approach is that the resolution of the SPF representation begins to degrade with monomers having more than about 500 residues, and this therefore sets a limit on the size of symmetrical complexes that can be modelled. We propose that one way to address this limitation would be to use a residue-based coarse-grained force field representation in place of the Fourier domain pseudo-molecules during the final trimeric assembly stage.

4

MODELING MOLECULAR MOTIONS

Macromolecular flexibility links protein structures with their function. I have been examining multiple ways how it can be efficiently described and predicted. For its description, we have proposed a scheme for the nonlinear Cartesian normal mode analysis of large macromolecules, such as proteins and their complexes [94]. It allows rapid computation and a very compact representation of complex molecular motions. The method is very CPU and memory efficient, it is typically two orders of magnitude faster compared to the state of the art. The tool is getting popular in the community and we have multiple ongoing methodological collaborations (Elodie Laine at University Sorbonne; the team of Randy Reed at MRC Cambridge; Nathalie Reuter at the University of Bergen; Pablo Chacon at IQFR-CSIC Madrid). The main idea of this method is to compute collective motions in a reduced rigid-body space using diagonalization of the reduced Hessian matrix $P^T HP$, where P is a rigid-body projector, and H is the all-atom Hessian. Then, we demonstrated that the obtained rigid-body linear motions $(\vec{\omega}, \vec{v})$ can be nonlinearly extrapolated to large amplitudes t and all-atom representation $\vec{A}'(t)$ using the following equation,

$$\vec{A}'(t) = R(\vec{\omega}t)(\vec{A} - \vec{r}_0) + \vec{r}_0 + \vec{v}_{||}t, \tag{135}$$

where $R(\vec{\omega}t)$ is the rotation matrix describing a rigid block's rotation about axis $\vec{\omega}$ by an angle ωt, $\vec{r}_0$ is the center of rotation of a rigid block, determined using values of $(\vec{\omega}, \vec{v})$, and $\vec{v}_{||}$ is a component of v collinear to $\vec{\omega}$.

I have applied this technique to some practical biological examples with my experimental collaborators [89, 107]. We then also combined this methodology with FFT-based shape matching [95], and applied it to flexible docking [177] and flexible fitting of templates into small-angle scattering profiles [83]. We used this technique to predict protein transition between multiple states [84, 130], and also as a component of protein structure prediction pipeline in data-assisted challenges in CASP13 [67, 101].

4.1 INTRODUCTION

Large macromolecules, including proteins and their complexes, are intrinsically flexible, and this flexibility is often linked with their function. A molecule in solution can be viewed as a structurally heterogeneous ensemble, where a finite number of conformational states (*e.g.* active-inactive, bound-unbound) may become stable under certain conditions to perform specific tasks. Identifying the molecular states relevant to protein functioning is necessary for our understanding of biological processes. Moreover, targeting protein functional motions bears a great potential to control and modulate proteins' activities and interactions in physio-pathological contexts.

Structural heterogeneity can be probed by various experimental techniques. These include X-ray crystallography, cryo-electron microscopy (cryo-EM), nuclear magnetic resonance (NMR), small-angle scattering and many others [23]. The two first methods allow obtaining large macromolecular structures at high resolution. While X-ray crystallography captures single stable states, cryo-EM allows observing conformational ensembles in solution. The resolution attained by cryo-EM is very often lower than that of X-ray structures, mainly due to the *structural heterogeneity* of the measured samples. However, the ongoing revolution in cryo-EM instrumentation [36] has supplied an exponentially growing body of near-atomic resolution structures. These techniques provide valuable

insights on proteins' functioning and interactions with their environment. Nevertheless, experimental protein structure determination remains a time consuming and costly process. The *systematic* description of the variety of shapes a protein adopts under particular environmental conditions, upon post-translational modifications and/or partner binding still remains out of reach. Hence, there is a need for computational tools able to efficiently and accurately predict functionally relevant protein conformations and macromolecular motions in general.

Several decades ago, Hayward and Go [92] observed that large-scale protein dynamics can be described with a set of just a few *collective coordinates*, accessible through the normal mode analysis (NMA). Thus, the latter provides an efficient way for reducing the dimensionality of the initial system and allows to study conformational transitions in proteins and their complexes. This has motivated the development of NMA-based tools for multiple biological applications, including flexible fitting of atomistic structures into cryo-EM maps [150, 227, 228, 242, 246, 247, 250, 279] or one-dimensional scattering profiles [79], prediction of crystallographic temperature factors [126, 163, 283], generation of structural ensembles for cross-docking [38, 172], prediction of protein hinge regions [65, 225], flexible docking [70, 161, 170, 177], refinement of crystallographic structures [52, 146] and docking solutions [145, 160, 264], and many others. The suitability of the NMA to model conformational dynamics varies widely depending on the system studied and on the type of motions involved [155]. The NMA was shown to better describe highly collective motions, compared to localized deformations [245].

Overall, the NMA is an old and well established technique [272] that has recently found many new applications in the field of structural biology and structural bioinformatics [18]. The internal motions of a protein have been a topic of great interest for a long time. One reason for this interest is the fact that some of these motions are known to play an important role in protein functions [17, 18, 118, 162, 271]. While molecular dynamics can nowadays accurately predict these motions, it is typically very computationally expensive, whereas NMA is relatively cheap and easily allows us to either extract the so-called *essential dynamics* of the protein from the MD trajectories [8], or to compute some low-frequency *collective motions* for a single structure [15, 33, 118, 139]. These low-frequency motions are particularly interesting to the structural biology community because they are commonly assumed to give more insight into protein function and dynamics [18, 93].

Atomistic molecular dynamics (MD) simulations represent an alternative to the NMA. They provide a practical tool to describe the structural heterogeneity around an equilibrium state and the flexibility exhibited by solvent-exposed small regions, such as loops. For instance, MD-based sampling has been applied to model the conformational diversity embedded in localized regions of cryo-EM maps [30]. In addition, the concept of collective coordinates has been extended to MD [7, 237, 238], which, as a result, have been applied to the study of free energy changes between different conformational states, and rare-event dynamics [69]. Nevertheless, MD simulations are much more costly than the NMA and the characterization of conformational transitions on a large scale with the former still remains computationally prohibitive.

This Chapter presents an efficient real-time method to compute *nonlinear* normal modes (the nonlinear rigid block, NOLB, NMA method [94]) and to predict biomolecular transitions involving a wide range of motions, from local deformations, *e.g.* of a small loop, to highly collective domain motions. It follows several of our publications [84, 94, 130, 177] and also presents some unpublished work. NOLB extends the classical NMA to describe nonlinear motions. Specifically, it extrapolates motions computed from instantaneous linear and angular velocities to large amplitudes. The resulting molecular motion is represented as a series of rigid block twists. We apply this nonlinear extrapolation to a combination of a few low-frequency normal modes to approximate conformational

transitions. Importantly, our approach is conceptually simple and explores the conformational space in the Cartesian coordinate system. The nonlinearity of the computed motions allows a better approximation of experimentally observed transitions.

So far, the computation of nonlinear transitions using the NMA formalism has only been possible by cutting them in small steps and recomputing the normal modes at each step, and/or by performing the NMA in the internal coordinate system [75, 150, 152, 163]. On average, the internal-coordinate NMA (iNMA) requires a smaller number of modes than the classical Cartesian-coordinate NMA to describe large structural transitions [163], and better predicts transitions upon protein docking [75]. Working with internal coordinates also allows for large dimensionality reduction through variable selection and model simplification [115, 140, 151, 153, 163, 179]. Despite these advantages, iNMA implies solving the generalized eigenvalue problem and dealing with necessarily dense interaction matrices. This makes it computationally costly and prevents its application on a large scale. Moreover, small changes in the internal coordinates may result in very large overall structural changes, which makes the approach less amenable to conformational space exploration, as it generates instability in the solution.

To demonstrate the advantages of the method reported here, we assess structural transitions computed with the classical linear normal modes, the Cartesian nonlinear normal modes, and an iterative scheme where the nonlinear modes are updated while progressing to the target state. For this purpose, we composed three test benchmarks of proteins exhibiting various types of structural transitions. The first test case presents examples of large domain motions, where 'open' and 'closed' conformations can be clearly identified [150]. The second one is comprised of proteins changing their conformation upon binding to other proteins [265]. The third one contains test cases from the Cryo-EM 2015/2016 Model Challenge, where the transition takes place between a crystal form and a conformation in solution [135]. We find that the classical linear NMA behaves well on the first set, where the motions are mostly collective, but is not suited to describe the more localized deformations and very small transitions exhibited by the two other sets. We show that our Cartesian nonlinear approach systematically obtains better transitions compared to the linear one. Indeed, the final predicted structures are closer to the experimentally known targets and display less distortions. The improvement is particularly significant on changes associated to partner binding. Also, the transitions are stereochemically correct, as highlighted by high Procheck [133] G-factors along the transitions. Moreover, structures along the transitions approach several experimentally validated intermediate states. We further demonstrate the usefulness of nonlinearity and mode updating to extend the applicability of the NMA to localized and disruptive motions. We also show that if the target structure is unknown and the amplitudes of the deformations along each mode are sampled randomly, there is still a sufficiently high success rate to predict the transition. Last, but not least, our approach is very computationally and memory efficient. It is implemented as a fully automated tool available at: https://team.inria.fr/nano-d/software/nolb-normal-modes/.

Our results allow revisiting the NMA-based description of biomolecular transitions. They pave the way to the systematic targeting and modulation of protein-protein interactions.

4.2 DATASETS

To assess the NOLB method, we have selected three types of tests. Fist, we chose three molecular systems for the visual inspection of the motions. These systems are the T7 large terminase (pdb code 4bij), the TAL effector PthXo1 bound to its DNA target (pdb code 3ugm), and the cytoplasmic domain of a bacterial chemoreceptor from thermotoga maritima (pdb code 2ch7). Our second test is the energy comparison between the linear

and nonlinear deformations along some low-frequency modes at different deformation amplitudes. For this test we have selected four structures of molecular systems from those provided in the 2015/2016 Cryo-EM Model Challenge [134]. These are the structure of the T7 large terminase described above, the structure of the human γ-secretase (pdb code 5a63), the structure of the capsaicin receptor TRPV1 (pdb code 3j9j), and the structure of the TRPV1 ion channel (pdb code 3j5p). Finally, in the third test we measured the memory and CPU consumption of our method with five molecular structures of increasing size ranging from 4,630 of atoms to 284,479 of atoms. These are the structure of the cytoplasmic domain of a bacterial chemoreceptor from thermotoga maritima (pdb code 2ch7 with 4,630 of atoms excluding hetero atoms), the structure of the human γ-secretase (pdb code 5a63 with 9,646 of atoms excluding hetero atoms), the structure of the T7 large terminase (pdb code 4bij with 18,855 of atoms excluding hetero atoms), the structure of the photosystem II complex (pdb code 5b5e, 40,908 of atoms excluding hetero atoms), and the structure of the E. coli 70S ribosome (pdb code 5j8a, 284,479 of atoms excluding hetero atoms). We should mention that the last structure is one of the largest that the protein data bank [23] currently contains.

The first test set for the assessment of structural transitions is comprised of structures from the iMod benchmark [151] prepared by Chacón and colleagues. It was recently used to assess three coarse-grained elastic network model-based flexible fitting methods [252]. It contains 23 proteins, each given in "open" and "closed" conformations, and represents a wide variety of macromolecular motions, mostly hinge motions, but also shear and other complex motions. The structures were extracted from the molecular motions database MolMovDB [63]. All of them have less than 3% Ramachandran outliers (as computed by the MolProbity program [44]), do not have any broken chain or missing atom. The average root mean square deviation (RMSD) for this set is 5.1 ± 3.0 Å.

For the second test set we have chosen some examples from the Protein-Protein Docking Benchmark v5 (PPDBv5) [265]. This benchmark contains 230 protein complexes with at least one of the partners solved in both bound (complexed) and unbound (free) states. All structures have a resolution better than 3.25 Å, and some of them contain more than one chain. We extracted 95 proteins with C_α RMSD between the two states above 2 Å. This test set is well suited for assessing the range of applicability of flexible docking methods [57]. We should also mention that some of the structure pairs can be classified as open-closed pairs. The average displacement for this test set is 4.0 ± 3.9 Å.

For the third test set we have selected seven cases from the Cryo-EM 2015/2016 Model Challenge [135]. The initial set was comprised of eight cases, but we decided not to consider one of them, namely the 70S ribosome. Each one of them comprises one or several starting structures solved by X-ray crystallography and one or several target structures corresponding to a Model Challenge map. In one case (γ-secretase) we did not find homologous X-ray structures for the starting state and used several cryo-EM structures instead. The map resolutions range from 2.2 to 4.3 Å. The average C_α RMSD displacement between the two states is 2.6 ± 3.2 Å.

4.3 MATERIALS AND METHODS

4.3.1 *Outline of the method*

Protein shapes and motions are governed by a multitude of interatomic forces, resulting from intra- and inter-molecular interactions. Despite this high complexity, many functional motions can be approximated by a few *low-frequency modes* characteristic of the protein's geometrical shape [127, 245, 256]. To compute these modes, we represent the protein as an elastic network (Fig. 18, top panel on the left), where each node stands for an atom and two nodes i and j are connected by a spring whenever the distance

d_{ij} between the corresponding atoms is smaller than a cutoff value, typically 5 Å. The normal modes are obtained by diagonalizing the mass-weighted Hessian matrix of the potential energy of this network. To reduce the dimensionality of this diagonalization problem, we consider each protein residue as a rigid block, according to the *rotation translation blocks* (RTB) approach [62, 248] (Fig. 18, middle panel on the left). With this coarse-grained representation, the computed normal modes are composed of *instantaneous linear velocities* $\vec{v}$ and *instantaneous angular velocities* $\vec{\omega}$, defining translations and rotations for each block/residue.

A straightforward way to compute normal-mode guided structural transitions is to calculate instantaneous displacements of each atom in a residue and then linearly extrapolate these up to a given amplitude a. However, at large amplitudes, this will distort interatomic distances and produce unrealistic molecular conformations. To circumvent this problem, we apply a nonlinear extrapolation (Fig. 18, bottom panel on the left), where each residue undergoes a *screw* (or a *twist*) motion. Specifically, the linear velocity $\vec{v}$ is decomposed in two terms, namely $\vec{v}_{||}$, which is collinear to $\vec{\omega}$, and $\vec{v}_{\perp}$, which is orthogonal to $\vec{\omega}$. We further represent the pair of $\vec{\omega}$ and $\vec{v}_{\perp}$ as a pure rotation around a new center $\vec{r}_0$. Hence, instead of rotating about the axis defined by $\vec{\omega}$ passing through its center of mass, each residue is rotated about the new axis defined by $\vec{\omega}$ passing through $\vec{r}_0$ and translated only in the direction of $\vec{v}_{||}$. This nonlinear extrapolation guarantees preservation of the topology or the protein structure subject to the motion.

Our method computes normal mode-guided nonlinear conformational transitions, starting from an experimentally determined structure or a high-quality 3D model. Specifically, normal modes are computed from the starting structure, which is then deformed along a selection of these modes up to a given amount of conformational deviation (Fig. 18, right panel). The simulated conformational change can be potentially very large (several tens of Å). The algorithm may be run in an iterative mode, where the normal modes are re-computed on intermediate conformations. This allows modifying the topology of the network representing the structure and going further away from the starting structure (Fig. 18, right panel, compare orange and red conformations). The method guarantees producing *plausible* physics-based motions and conformations.

4.3.2 *Equilibrium dynamics*

Let us consider a molecular system with N atoms near an equilibrium position. Let $V(x)$ be a potential energy function of our system evaluated at a position $x \in \mathbb{R}^{3N}$. The near-equilibrium motion of our system can be described with Newton's equation of motion in the harmonic approximation as

$$M\ddot{x} + \nabla V(x) \approx M\ddot{x} + Hx = 0, \tag{136}$$

where M is a $3N \times 3N$ diagonal mass matrix, and H is a $3N \times 3N$ *Hessian matrix* of the potential energy V evaluated at the equilibrium position. It is very convenient to work with mass-weighted Cartesian coordinates x^w [272] defined as

$$x^w = M^{\frac{1}{2}}x. \tag{137}$$

In these coordinates, the motion equation is simplified to

$$\ddot{x}^w + H^w x^w = 0, \tag{138}$$

where H^w is a *mass-weighted* Hessian matrix $H^w = M^{-\frac{1}{2}}HM^{-\frac{1}{2}}$. The solution of this equation is obtained by the diagonalization of matrix H^w as

$$H^w = L^w \Lambda L^{wT}, \tag{139}$$

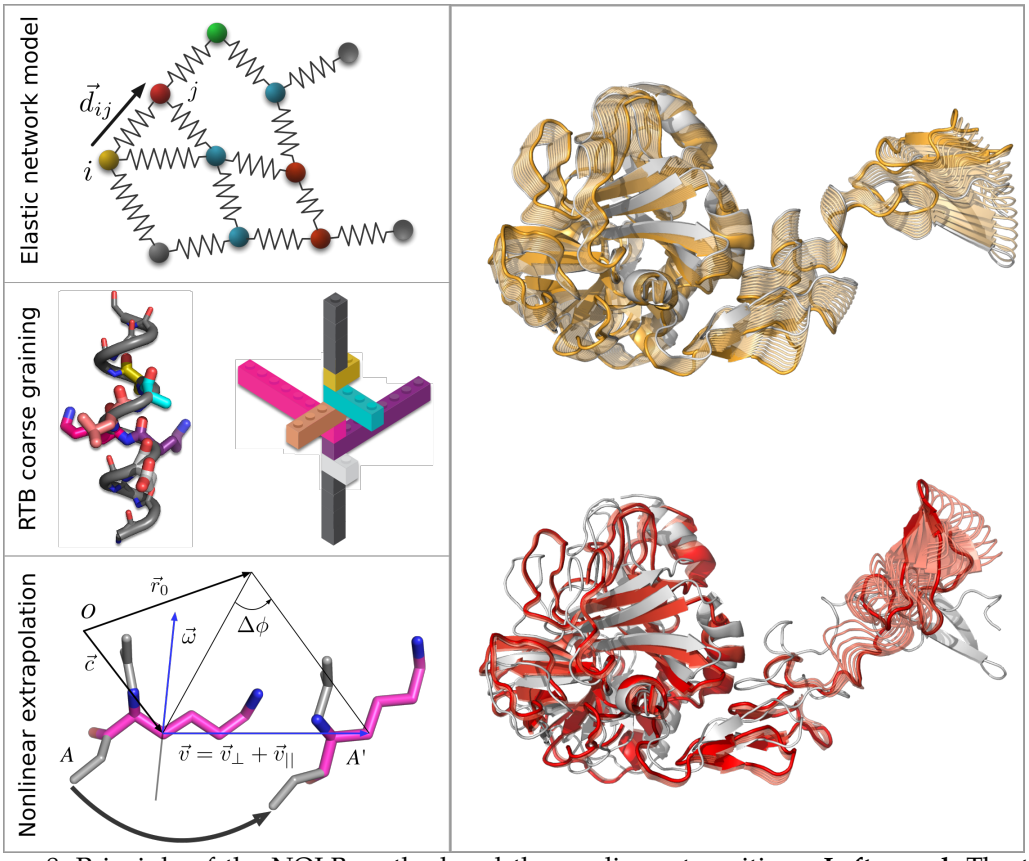


Figure 18: Principle of the NOLB method and the nonlinear transitions. **Left panel**: The three main ingredients of the method are depicted: the elastic network model, the rotation translation blocks (RTB) projection and the nonlinear extrapolation of motions. The protein is represented as an elastic network (on top), where all the atom pairs within a certain cutoff distance are connected with harmonic springs. Coarse-graining is achieved by replacing each protein residue by a rigid block (in the middle). The color code indicates the one-to-one correspondence between residues (on the left) and blocks (on the right). Each block has six degrees of freedom, three for rotation and three for translation. At each step of the transition, each residue/block undergoes a screw (twist) motion (at the bottom) defined from the instantaneous linear and angular velocities $\vec{v}$ and $\vec{\omega}$ obtained by the NMA. The initial and final atomic positions are denoted as A and A', respectively. O is the origin of the coordinate system and $\vec{c}$ is the residue's center of mass. **Right panel**: Examples of nonlinear transitions computed for coagulation factor VIIa upon binding to tissue factor. The intermediate structures in orange were determined from the normal modes of the known unbound structure (1qfk:HL, in grey). Those in red were further obtained by updating the normal modes three times. The final predicted structure (in opaque) is 1.3 Å from the known bound structure (1fak:HL).

where L^w are the mass-weighted eigenvectors of H^w, also referred to as to *mass-weighted normal modes*, and Λ is a diagonal matrix of associated eigenvalues. Assuming that in the equilibrium positions $x = 0$, the harmonic Cartesian motions as a function of time along a i-th normal mode L_i of frequency w_i will be given as

$$x(t) = M^{-\frac{1}{2}} L_i^w \frac{\sqrt{2k_BT}}{w_i} \sin w_i t, \tag{140}$$

where k_BT is the temperature factor. Here we have also used the equipartition theorem to calculate the amplitude of the motion, and assumed the energy of each mode to be $k_BT/2$. Accordingly, the instantaneous atom velocities in the equilibrium position at time $t = 0$ corresponding to this mode will be

$$\dot{x}|_{t=0} = M^{-\frac{1}{2}} L_i^w \sqrt{2k_BT}. \tag{141}$$

This equation connects the *2-norm* of mass-weighted normal mode vectors L_i^w with the *kinetic energy* of this mode K at a temperature T. Indeed,

$$L_i^{wT} L_i^w = \frac{K}{k_B T}, \tag{142}$$

where the kinetic energy is defined as $K = \frac{1}{2}\dot{x}^T M \dot{x}$. We will refer to the columns of the $M^{-1/2}L$ matrix as to *Cartesian linear normal modes*. We should specifically mention that these normal modes are not generally orthogonal, unless all the masses in M are equal to each other.

4.3.3 *Motions of rigid bodies and the RTB projection method*

Many methods have been proposed to reduce the dimensionality of the NMA diagonalization problem. For example, Noguti and Gõ [179] and Levitt et al. [140], and later Ma et al. [153], Mendez and Bastolla [163], and Chacón et al. [151] explored the NMA approach in internal coordinates. However, an orthogonal idea of reducing the dimensionality of the original system by coarse-graining its representation has gained much more popularity. One of the first coarse-graining methods was the *rotation translation blocks* (RTB) approach introduced by Durand et al. [62] and further developed by Tama et al. [248] and Li and Cui [144]. In this method, individual or several consecutive amino residues are considered as rigid blocks that can only exhibit rotational and translational motions [62, 248].

Similarly to the above case, Newton's equation of motion can also be written for a system composed of *rigid bodies*, or a composition of rigid bodies and individual atoms. Each rigid body is parametrized with a 3-vector of its centre of mass $\vec{c}$, a 3-vector of its orientation $\vec{\theta}$, its mass m, and its *inertia tensor* I, computed relatively to the center of mass. As before, it is very convenient to introduce mass-weighted rigid-body coordinates,

$$\vec{c^w} = \sqrt{m}\vec{c} \tag{143}$$

$$\vec{\theta^w} = I^{\frac{1}{2}}\vec{\theta}. \tag{144}$$

The kinetic energy K_{RB} of a rigid body will then be given as

$$K_{RB} = \frac{1}{2}(\dot{\vec{c^w}})^2 + \frac{1}{2}(\dot{\vec{\theta^w}})^2. \tag{145}$$

It is also useful to mention momentum conservation laws in these coordinates, assuming that a rigid body is composed of individual atoms at positions $\vec{x}_k$ with masses m_k,

$$\sqrt{m}\dot{\vec{c^w}} = \sum_k \sqrt{m_k}\dot{\vec{x}}_k^w \tag{146}$$

$$I^{\frac{1}{2}}\dot{\vec{\theta^w}} = \sum_k (\vec{x}_k^w - \sqrt{\frac{m_k}{m}}\vec{c^w}) \times \dot{\vec{x}}_k^w. \tag{147}$$

The momentum conservation laws provide a linear relationship between *instantaneous* motions in the Cartesian and the rigid-body spaces. It can be compactly represented with a $6B \times 3N$ *projector matrix* P, which translates Cartesian instantaneous motions of N atoms into the rigid-body space composed of B rigid blocks. [62, 248] Projector matrix P is composed of B positional P^c and orientational P^θ matrices of size $3 \times 3N_b$ each,

where N_b is the number of atoms in the b-th rigid body. Each of them is composed of N_b 3×3 square matrices,

$$P_k^c = \sqrt{\frac{m_k}{m}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad P_k^\theta = \sqrt{m_k} I^{-\frac{1}{2}} [\vec{x}_k - \vec{c}]_\times \tag{148}$$

where k is one of N_b atom indices.

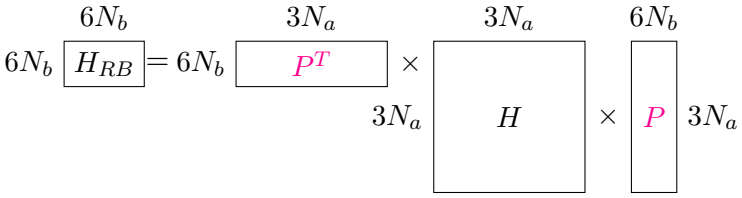
Now, one can project instantaneous motions into the rigid-body (RB) space,

$$L_{RB}^w = PL^w. \tag{149}$$

Alternatively, we can apply projectors P directly to the Hessian matrix,

$$H_{RB}^w \equiv PH^w P^T = PL^w \Lambda L^{wT} P^T = L_{RB}^w \Lambda L_{RB}^{wT}, \tag{150}$$

obtaining an RB-projected mass-weighted Hessian matrix H_{RB}^w with the corresponding eigenvectors L_{RB}^w. Schematically, this projection can be illustrated as



We should note that generally, the 2-norm of L_{RB}^w is smaller or equal to the 2-norm of L^w. Indeed, from the momentum conservation laws, for each mode i we again obtain a relation equivalent to eq. 151,

$$L_{RB_i}^{wT} L_{RB_i}^w = \frac{K_{RB}}{k_B T}, \tag{151}$$

where the kinetic energy K_{RB} for a set of rigid bodies is defined in this case by eq. 145. The kinetic energy before and after application of the rigid constraints is not preserved, as some of the energy is stored inside the constraints. Therefore, strictly speaking, vectors L_{RB}^w need to be normalized to become eigenvectors of H_{RB}^w.

We should explain the physical meaning and the properties of the projection matrix. As it was stated above, the projection matrix P projects the instantaneous motions of the initial system into the RB subspace. Its transpose P^T maps rigid-body motions to the original atomic representation *linearizing* them. The two satisfy the following identity by construction,

$$PP^T = I_{6B}, \tag{152}$$

where I_{6B} is a $6B \times 6B$ identity matrix. We should specifically note that $P^T P$ is $3N \times 3N$ matrix, which is not identity. Matrix PP^T can be seen as a *rigid-body liniarization operator*, which transforms initial all-atom motions to the motions compatible with the separation of the system into a set of rigid bodies. For example, we can obtain approximate all-atom mass-weighted normal modes L^w by applying this linearization to mass-weighted RB normal modes L_{RB}^w,

$$L^w \approx P^T L_{RB}^w. \tag{153}$$

4.3.4 *The NOLB method*

Molecular vibrations in a multi-dimensional harmonic oscillator are all uncoupled and can be found by solving eq. 138. Diagonalization of the RTB-projected mass-weighted Hessian gives a set of eigenvectors that are composed of *instantaneous linear velocities* $\vec{v}_w$ and *instantaneous angular velocities* $\vec{\omega}_w$ of individual rigid blocks. For a rigid block with mass M_b and inertia tensor I, we first compute these in non-mass weighted coordinates as follows,

$$\begin{aligned} \vec{v} &= M_b^{-1/2}\vec{v}_w \\ \vec{\omega} &= I^{-1/2}\vec{\omega}_w. \end{aligned} \tag{154}$$

Then, given a deformation amplitude a, the translational increment in the rigid block's position $\Delta\vec{x}$ and the angular increment in its orientation $\Delta\phi$ can be computed as

$$\begin{aligned} \Delta\vec{x} &= a\vec{v} \\ \vec{n} &= \vec{\omega}/||\vec{\omega}||_2 \\ \Delta\phi &= a||\vec{\omega}||_2, \end{aligned} \tag{155}$$

where the rigid block's rotation is described with a unit axis $\vec{n}$ passing though its center of mass (COM) $\vec{c}$, and an angle ϕ. Finally, we rewrite the increment in the rigid block's position $\Delta\vec{x}$ as a sum of two orthogonal vectors,

$$\Delta\vec{x} = \Delta\vec{x}_\perp + \Delta\vec{x}_{||}, \tag{156}$$

where $\Delta\vec{x}_\perp$ is orthogonal to $\vec{n}$, and $\Delta\vec{x}_{||}$ is collinear to $\vec{n}$. We then represent the $\Delta\vec{x}_\perp$-related motion as a pure rotation about a new center $\vec{r}_0$ given as

$$\vec{r}_0 = \vec{c} + (\vec{n} \times \vec{v}_\perp)/||\vec{\omega}||_2, \tag{157}$$

such that the final rigid block's positions $\vec{A}'$ is expressed through the initial positions $\vec{A}$ as

$$\vec{A}' = R(\Delta\phi, \vec{n})(\vec{A} - \vec{r}_0) + \vec{r}_0 + \Delta\vec{x}_{||}, \tag{158}$$

where $R(\Delta\phi, \vec{n})$ is the rotation matrix describing rigid block's rotation about an axis $\vec{n}$ by an angle $\Delta\phi$. More details can be found in the original NOLB publication [94]. It is easy to demonstrate that this is the only type of rigid-body motion that conserves the original kinetic energy. Indeed, using the parallel axis theorem it is readily seen that the initial energy contribution of linear velocity $v_{w\perp}^2/2$ is transformed into equivalent contribution from the angular velocity.

4.3.5 *Linear structural transitions*

Let us assume we know two conformations of the same molecular system and the correspondence between their atoms. The latter can be robustly deduced from sequence alignment if the two systems are composed of not fully identical proteins. Let us also assume we are given the displacement vector $\Delta\vec{r}$ between the two conformations after their optimal rigid superposition. It is easy to demonstrate that in this case, the COMs of the two conformations match. We can now find the minimum RMSD between the two conformations, if one of them is allowed to deform along its M lowest normal modes $L \in \mathbb{R}^{3N\times M}$, which are not necessarily orthonormal, as

$$\text{RMSD}^2 = \frac{1}{N}\left(\Delta r - La\right)^2 = \frac{1}{N}\Delta r^T\left[I - L(L^TL)^{-1}L^T\right]\Delta r, \tag{159}$$

where N is the number of atoms in the system, I is the identity matrix, and a are the optimal amplitudes of linear deformations given as

$$a = (L^T L)^{-1} L^T \Delta r. \tag{160}$$

If the normal modes L are orthonormal (which may happen if the mass matrix in eq. 137 is identity), the above equation simplifies to

$$\text{RMSD}^2 = \frac{1}{N} \Delta r^T \left[I - LL^T \right] \Delta r. \tag{161}$$

It can be readily seen that if all the $3N$ modes are activated, the matrix L becomes square, LL^T turns into an identity, and the RMSD reduces to zero.

4.3.6 *Nonlinear structural transitions*

The NOLB method produces nonlinear deformations. Therefore, eq. 159 would not be exact in this case. However, given the displacement vector $\Delta\vec{r}$ between the two conformations as in the previous case, we can still construct a deterministic deformation trajectory and compute the corresponding RMSD. We should specifically mention that rotation operators do not commute, and thus the result of applying two rotations will generally depend on the order of these operators. Therefore, to make the method deterministic, when combining several modes, we always order them from the lowest to the highest frequencies. This choice is dictated by the fact that slower modes result in larger amplitudes of thermal fluctuations.

To produce a nonlinear deformation towards the target structure, we use an iterative procedure (see Algorithm 1 in Supplementary Material). At each step of the iteration we approximate the amplitudes of the nonlinear deformation by the analytically computed linear amplitudes using eq. 160. This approximation will not be valid at large deformation amplitudes a. Therefore, if the RMSD computed for the linear approximation (eq. 159) is larger than a certain threshold (we have chosen 0.1 Å), we split the deformation into smaller pieces. Each piece is computed based on the values of the linear amplitudes scaled in such a way that the total linear RMSD of the deformation equals to the threshold value of 0.1 Å. We terminate the algorithm when the maximum number of iterations is exceeded (100 by default), or if the relative deformation becomes smaller than a tolerance of $1e-6$. This algorithm can be iterated multiple times, the elastic network model being updated and the normal modes recomputed at each iteration (see Algorithm 2 in Supplementary Material). On-the-fly normal mode re-computation has been previously proposed in the context of cryo-EM fitting and morphing applications [150–152].

Our nonlinear model and the way we assess the predicted transitions naturally overcome the limitations of classical NMA schemes highlighted in Jernigan *et al.* [234, 274] when the transition involves a substantial protein domain rotation.

4.3.7 *Nonlinear random sampling*

If one of the two conformations is not known, which is the case in many practical applications, the NOLB method samples the conformational space around the known structure up to a given RMSD. In this case, the amplitudes of the selected modes are chosen randomly [177]. To test whether such random exploration could be useful to recapitulate functional states, we implemented a simulation protocol producing 10 000 conformations (see Algorithm 3 in Supplementary Material). In this protocol, the starting structure is first deformed along its 3 slowest modes, then the modes are recomputed and the new starting structures are deformed along their 10 slowest modes. An intermediate

step with 5 modes is added in case of large deformations (>4.5 Å). The biggest part of the displacement is accomplished in the first step. Keeping the number of modes very small (3) at this step allows limiting the combinatorics of the conformational search.

4.3.8 *Potential function*

Classical NMA methods can use any potential function, provided that it corresponds to the equilibrium position of the molecular system. Some recent developments can also assume non-equilibrium state of the initial system [279]. In our method we use an all-atom anisotropic network model (ANM) [15, 58], where the initial structure is always at equilibrium. The all-atom ANM has the following potential function,

$$V(q) = \sum_{d_{ij}^0 < R_c} \frac{\gamma}{2}(d_{ij} - d_{ij}^0)^2, \tag{162}$$

where d_{ij} is the distance between the i^{th} and the j^{th} atoms, d_{ij}^0 is the reference distance between these atoms, as found in the original structure, γ is the spring constant, and R_c is a cutoff distance, typically between 3.5 Å and 15 Å. By default we let this value to 5 Å. However, if there are loosely connected structural fragments in the system, it makes sense to increase this value to 10 Å or even more. The Hessian matrix corresponding to this potential function is composed of the following blocks [15, 18, 58],

$$\begin{aligned} H_{ij} &\equiv \frac{\partial^2 U}{\partial \vec{x}_i \partial \vec{x}_j^T}|_0 = -\frac{\gamma}{(d_{ij}^0)^2}\vec{x}_{ij}\vec{x}_{ij}^T \quad i \neq j \\ H_{ii} &\equiv \frac{\partial^2 U}{\partial \vec{x}_i \partial \vec{x}_i^T}|_0 = \sum_{j \neq i} \frac{\gamma}{(d_{ij}^0)^2}\vec{x}_{ij}\vec{x}_{ij}^T \quad i = j \end{aligned}, \tag{163}$$

where $\vec{x}_{ij} = \vec{x}_i - \vec{x}_j$. To rapidly compute this matrix, we use an efficient neighbor search algorithm [13].

4.3.9 *Extension for symmetric systems*

Let us assume that a replica of the original molecular system is rotated with a matrix R and then translated by a vector $\vec{T}$. Then, the interaction energy between the system and its replica will be written as

$$V(q) = \sum_{ij} \frac{\gamma}{2}(|R\vec{x}_i + \vec{T} - \vec{x}_j| - d_{ij}^0)^2, \tag{164}$$

where d_{ij} is the distance between the i^{th} atom in the replica and the j^{th} atom in the original system, d_{ij}^0 is the reference distance between these atoms, and γ is the stiffness constant. The gradient elements will be

$$\begin{aligned} \frac{\partial U}{\partial \vec{x}_i} &= \sum_{ij} \frac{\gamma}{d_{ij}}(d_{ij} - d_{ij}^0)R^T(R\vec{x}_i + \vec{T} - \vec{x}_j) \quad i \neq j \\ \frac{\partial U}{\partial \vec{x}_i} &= \frac{\gamma}{d_{ij}}(d_{ij} - d_{ij}^0)(R - I)^T(R\vec{x}_i + \vec{T} - \vec{x}_i) \quad i = j \end{aligned}. \tag{165}$$

And the Hessian elements will be

$$\begin{aligned} &\frac{\partial^2 U}{\partial \vec{x}_i \partial \vec{x}_j^T}|_0 = -\frac{\gamma}{(d_{ij}^0)^2}R^T\vec{x}_{ij}\vec{x}_{ij}^T \quad i \neq j \\ &\frac{\partial^2 U}{\partial \vec{x}_j \partial \vec{x}_i^T}|_0 = -\frac{\gamma}{(d_{ij}^0)^2}\vec{x}_{ij}\vec{x}_{ij}^T R \quad i \neq j \\ &\frac{\partial^2 U}{\partial \vec{x}_i \partial \vec{x}_i^T}|_0 = \frac{\gamma}{(d_{ij}^0)^2}(R - I)^T\vec{x}_{ij}\vec{x}_{ij}^T(R - I) \end{aligned}, \tag{166}$$

where $\vec{x}_{ij} = R\vec{x}_i + \vec{T} - \vec{x}_j$. We should mention that for the inverse transform $\{R_{\text{inv}}, \vec{T}_{\text{inv}}\} \equiv \{R^T, -R^T\vec{T}\}$, we get the following Hessian elements,

$$\frac{\partial^2 U}{\partial \vec{x}_i \partial \vec{x}_j^T}|_0 = -\frac{\gamma}{(d_{ij}^0)^2}\vec{x}_{ji}\vec{x}_{ji}^T R \quad i \neq j, \tag{167}$$

such that the final Hessian matrix is symmetric.

4.3.10 *Assessment of the transitions*

Transition coverage

To assess the ability of NOLB to reach the target structure by deforming the starting structure along its lowest normal modes, we compute the transition coverage, expressed as

$$\text{Coverage} = \frac{\text{RMSD}_i - \text{RMSD}_f}{\text{RMSD}_i}, \tag{168}$$

where RMSD_i is the initial root mean square deviation between the starting and target structures, and RMSD_f is the deviation between the final structure obtained from the computed transition and the target structure. The coverage varies between 0 (null prediction) and 1 (perfect prediction).

To assess the ability of NOLB to recapitulate known intermediate structures, we computed the improvement score described in [270] and expressed as

$$\text{Improvement} = \frac{\min(\text{RMSD}_{SI}, \text{RMSD}_{TI}) - \min_j(\text{RMSD}_{P_jI})}{\min(\text{RMSD}_{SI}, \text{RMSD}_{TI})}, \tag{169}$$

where S, I and T are the starting, intermediate and target structures, respectively, and P_j is the jth conformation predicted by NOLB. In the best-case scenario, one of the conformations predicted by NOLB is identical to the known intermediate structure, leading to an improvement of 100%. In the worst-case scenario, all conformations predicted by NOLB are further away from the intermediate than the starting and target structures, leading to a negative value.

Collectivity

Collective motions can be characterised by their *collectivity* κ, which is proportional to the exponential of the information entropy [34]. The collectivity of a transition between two structures of a molecule with N atoms can be computed [245] as

$$\kappa = \frac{1}{N}\exp\left(-\sum_{i=1}^{N} q_i^2 \log q_i^2\right), \tag{170}$$

where q_i are scaled Cartesian displacements of individual atoms, $q_i = \alpha\Delta r_i^2$, with the normalization factor α taken such that $\sum_{i=1}^{N} q_i^2 = 1$. $N\kappa$ gives an effective number of nonzero displacements q_i^2. Thus, κ is confined to the interval $\{1/N;1\}$. If $\kappa = 1$, then the corresponding transition is maximally collective and has all the displacements q_i^2 identical, which happens for rigid-body motions, for example. In the limit of an extremely localized motion, where only one single atom is affected, κ is minimal and equals to $1/N$. In a similar way, one can estimate the degree of collectivity of a normal mode. For example, collectivity of the jth mode is given by the same equation above provided that q_i are now the scaled normal mode's displacements,

$$q_i^2 = \alpha\frac{(M^{-1/2}L)_{j,3i}^2 + (M^{-1/2}L)_{j,3i+1}^2 + (M^{-1/2}L)_{j,3i+2}^2}{m_i}. \tag{171}$$

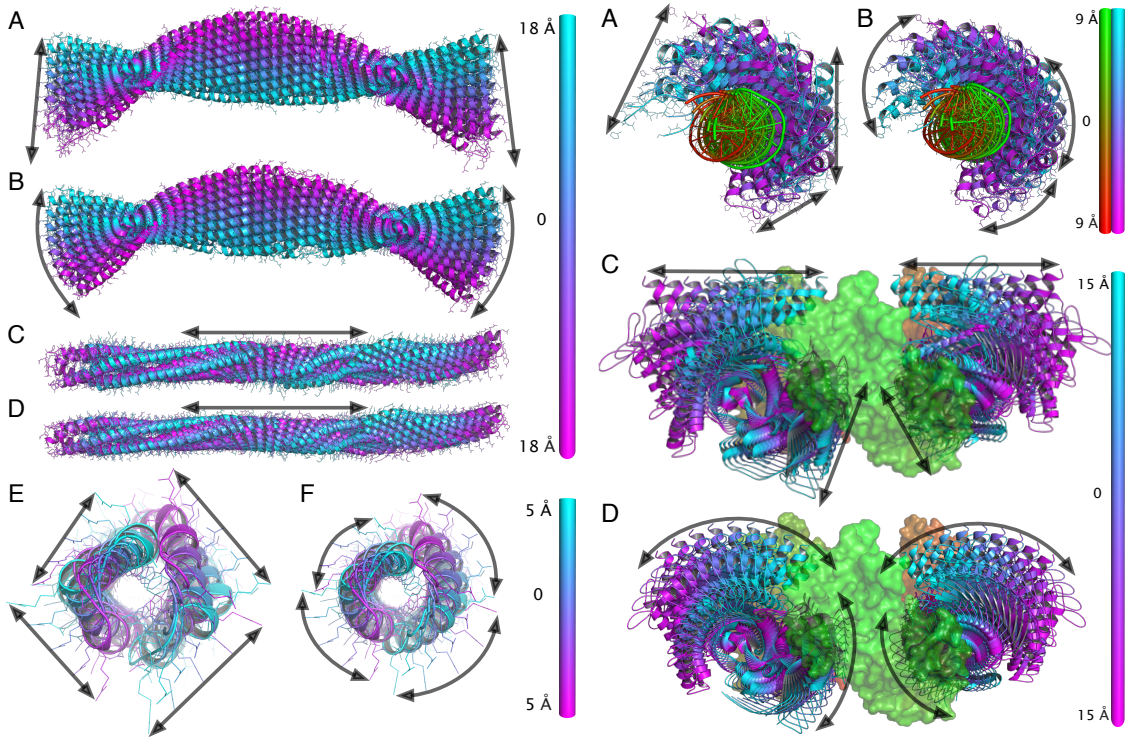


Figure 19: **Left**: Comparison of linear (A, C, E) and nonlinear (B, D, F) motion extrapolations of a coiled coil protein (pdb code 2ch7). Three types of motions are shown, bending (A, B), stretching (C, D), and twisting (E, F). Several snapshots at different deformation amplitudes are superposed to each other. These are colored according to the values of the overall deformation, as measured by the RMSD. The colorbars show the RMSD with respect to the initial position. The arrows follow the trajectories of individual atoms. **Right**: Comparison of linear (A, C) and nonlinear (B, D) motions computed for two molecular systems. Several snapshots at different deformation amplitudes are superposed to each other. These are colored according to the values of the overall deformation, as measured by the RMSD. The colorbars show the overall RMSD with respect to the initial positions. The arrows follow the trajectories of individual atoms. (A, B). Sliding of a DNA-binding protein (colored from cyan to purple) in the groove of the surface of the DNA (pdb code 3ugm). (C, D). Motion of two subunits of a terminase pentamer protein (pdb code 4bij). Three other subunits are shown in the surface representation. The 5-fold symmetry axis points towards the top of the figure.

4.4 RESULTS AND DISCUSSION

4.4.1 *Visual inspection of the nonlinear motions*

For the first test we have computed some lowest-frequency normal modes for several molecular systems and present the difference between the linear and the nonlinear extrapolation approaches, as it is described below. The first molecular system demonstrates three basic types of internal motions (see Figure 19 Left) and the other two systems illustrate some biologically relevant motions (see Figure 19 Right). Overall, Figure 19 clearly demonstrates that the nonlinear extrapolation produces visually better and physically more realistic motions than the standard approach. We should mention that in this test we used a single residue as a rigid block. We have additionally performed experiments with a larger number of residues per block, up to 10, and the results are very similar with the same conclusions as stated below.

There are, generally, three basic types of internal motions that a molecular system may exhibit. These are bending, stretching and twisting. All of these motions can be clearly seen with symmetric elongated rod-like objects. Therefore, for the first illustration we

have chosen a coiled-coil water-soluble protein from the cytoplasmic domain of a bacterial chemoreceptor (pdb code 2ch7). For this protein, we have computed its ten lowest normal modes and specifically selected those that correspond to the described basic types of motions. Then, we have computed the linear and nonlinear motion extrapolations at different amplitudes. These are presented in Figure 19 Left. The difference between the two types of extrapolations is especially apparent for motions with a large portion of involved rotation. For example, Figures 19 Left A-B show a bending type of motion and Figures 19 Left E-F show a twisting motion. For these two types of motions the difference between the two extrapolation approaches is visually clear. This is because for these types of motions the translational component is typically negligible with respect to the rotational component, which is given as a pure rotation of rigid blocks about a certain center. Thus, the nonlinear extrapolation produces a very different result at large deformation amplitudes. However, for the stretching motion, which is shown in Figures 19 Left C-D, there is no noticeable visual difference between the two types of motion extrapolation. This is because in this case the motion is mostly represented by its translation component and there is almost no difference between the two extrapolation approaches.

Another interesting type of motion where the nonlinear extrapolation produces a noticeable different result is the spiral sliding of a transcription activator-like effector (TALE) protein in a surface groove of its DNA target. This motion, both using linear and nonlinear extrapolations at large amplitudes, is shown in Figures 19 Right A-B. Here, we can see very similar motions of the DNA molecule (colored from green to red), while the extrapolated motions of the TALE protein (colored from cyan to purple) look more physically realistic in the nonlinear case. We should note that the maximum overall RMSD, as measured for the linear extrapolation, is about 9 Å. At such large deformation amplitudes, the linear extrapolation significantly perturbs the structure, as can be illustrated by broken covalent bonds. We should also emphasize that this sliding motion, as computed by the NOLB analysis around the system's equilibrium position, is biologically relevant, as has been recently demonstrated by the direct observation of TALE protein dynamics [50]. More precisely, the TALE proteins are capable of rapid diffusion along DNA using a combination of sliding and hopping.

Finally, as the last example, we have chosen a pentameric assembly of terminase proteins with the C_5 cyclic symmetry. The terminase is a powerful motor that converts ATP hydrolysis into mechanical movement of the DNA [51]. Similar to the previous examples, we have computed the lowest normal modes for the whole assembly and chosen the one that is responsible for the opening and closing of the channel in the middle of the assembly. More precisely, here each of the five subunits rotates symmetrically such that the channel in the middle changes its shape. Figures 19 Right C-D show the difference between both the linear and the nonlinear extrapolations of this motion. In order to make the figure more comprehensible, we show the motion of only two out of five subunits, colored from cyan to purple according to the amplitude of the deformation. The three remaining subunits (shown in surface representation) are static. Again, we can see that at large amplitudes the nonlinear extrapolation looks more physically realistic than the linear one. Similar to the previous example, this motion composed of symmetric rotations of each of the five subunits, as computed by the NOLB analysis, is biologically relevant and has been noticed during the cryo-electron microscopy reconstruction of the T7 large terminase [51]. More precisely, the five terminase subunits rotate to adapt the channel in such a way that it can accommodate the guest DNA.

4.4.2 *Comparison of local deformations*

As we have discussed above, the nonlinear normal modes approach demonstrated *visually* better results on all the tested examples. However, both linear and nonlinear extrapolation methods result in physically unrealistic local geometries at large deformation amplitudes. Thus, an additional energy minimization is typically required to relax the locally disturbed molecular geometries. Therefore, in this test we estimate the computational difficulty of such a minimization, which should be proportional to the deformation energy of the final structure. More precisely, we assume that the covalent bonds in the initial molecular structure are represented by harmonic springs with a force constant of 500 kcal/(mol Å^2), which is a typical value in classical force fields [156, 267], and we also assume that the total potential energy in the system is given by the sum of the bond contributions.

For this test, we measured the potential energy of the molecular structures generated by both linear and nonlinear extrapolations at various deformation amplitudes. Figures 20(a,d,g,j) show potential energy for several molecular structures averaged over ten lowest normal modes as a function of the overall RMSD of the final structure with respect to the initial one. We can see that for all the systems the nonlinear normal modes approach produces geometries with a lower bond energy than the standard linear NMA method, at least for deformations that do not exceed 25 Å in RMSD. This means that, in principle, it will be computationally more efficient to optimize the structures produced by the NOLB approach compared to the standard one.

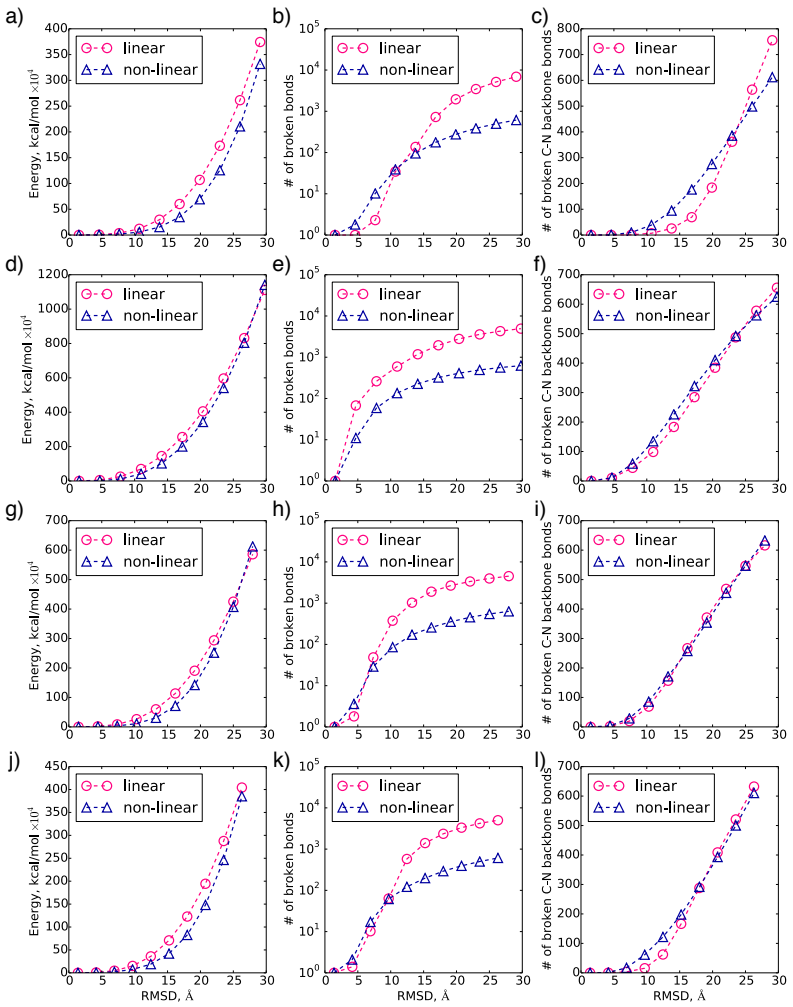


Figure 20: Comparison of linear and nonlinear deformations averaged over the 10 lowest normal modes computed for the following systems, (a-c) 4bij, (d-f) 5a63, (g-i) 3j9j, and (j-l) 3j5p. In (a,d,g,j) the bond harmonic energy as a function of the deformation amplitude is shown. In (b,e,h,k) the total number of broken bonds as a function of the deformation amplitude is shown (in a log scale). In (c,f,i,l) the number of broken bonds between individual amino acids is shown as a function of the deformation amplitude. See the main text for details.

To extend the analysis of the produced molecular topologies, we compared the number of broken covalent bonds in the final molecular structures. We define a covalent bond between two atoms as broken if its length exceeds the sum of the corresponding van der Waals radii multiplied by a factor of 0.6. Figures 20 (b,e,h,k) show the total number of broken covalent bonds for the two approaches and

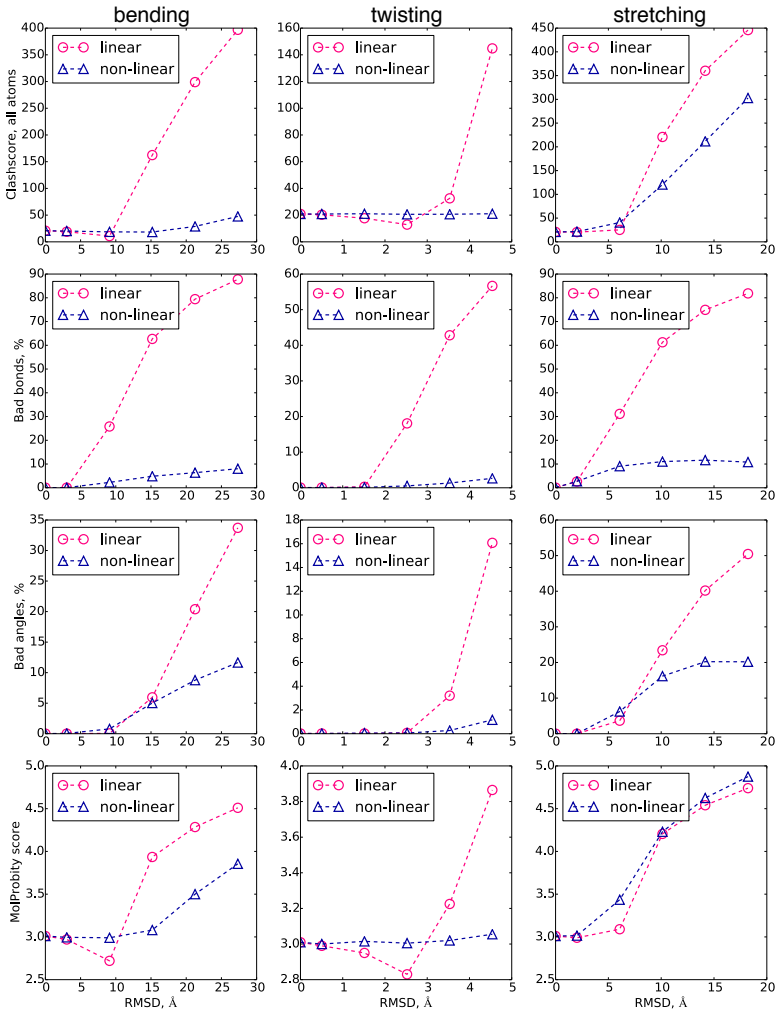


Figure 21: Comparison of linear and nonlinear deformations of the coiled-coil cytoplasmic domain of a bacterial chemoreceptor (pdb code 2ch7) assessed by the MolProbity server [44]. Results for the bending motion are shown in the left column, for the twisting motion are shown in the middle column, and for the stretching motion are shown in the right column. Multiple MolProbity statistics are plotted as a function of the deformation amplitude. The 'clashscore' is the number of serious clashes (atomic overlap ≥ 0.4 Å) per 1,000 atoms. Bad bonds and angles are those that are further away than four standard deviations from the expected values. The MolProbity score is a log-weighted combination of the clash-score, the percentage of not favored Ramachandran angles, and the percentage of bad side-chain rotamers, giving one number that reflects the crystallographic resolution at which those values should be expected.

clearly demonstrate that the linear extrapolation perturbs local molecular geometries much more compared to the NOLB method. Indeed, we can see that the gap between the two curves increases when the deformation amplitudes become larger. However, since the NOLB NMA approach relies on the rigid body dynamics and all the individual amino acids are treated as rigid blocks, we additionally compared the number of broken covalent bonds between individual amino acids for the two extrapolation approaches. Figures 20 (c,f,i,l) show these comparisons. For this case we can see that at small deformation amplitudes, the NOLB method breaks more covalent bonds, which should be expected. At large deformation amplitudes, however, the NOLB method performs better than the standard approach. Nonetheless, we should only consider the total number of broken bonds, or the total deformation energy of the system. In all the cases, as Figure 20 demonstrates, the NOLB NMA approach produces much better results compared to the standard method. In Supporting Information we also provides individual tables that list the data for each of the normal modes individually for all the described molecular structures.

To complete the analysis, we have also evaluated the quality of several selected structures using a popular MolProbity server [44]. For this evaluation we chose three types of deformations of a coiled coil cytoplasmic domain of a bacterial chemoreceptor presented in Fig. 19 LEft, namely, bending, twisting, and stretching. MolProbity is a structure validation web service widely used to evaluate the quality of X-ray or NMR structures. For the analysis it uses a variety of physics- and knowledge-based algorithms. Figure 21 presents the computed MolProbity statistics. More precisely, it shows the amount of serious clashes (with atomic overlap ≥ 0.4 Å), the percentage of statistically abnormal bonds and angles, and finally, the cumulative 'MolProbity score', which reflects the crystallographic resolution at which these structures should be expected. As before, we can see that at large deformation amplitudes the NOLB method produces consistently better structures than the standard linear approach. This conclusion is true for all studied types of motions. At small deformation amplitudes, the linear NMA approach performs slightly better if we consider the total number of serious clashes in the structures. Interestingly enough, this number can even decrease compared to the crystallographic structure, presumably because of its moderate resolution.

The presented examples demonstrate that the NOLB approach is able to generate structures with a fewer number of geometric distortions compared to the linear NMA method. However, after a certain amplitude of deformation, our method will also produce topological artefacts. This amplitude will generally depend on the type of motion, or, more technically, on the amount of the involved rotation compared to the translation (see eq. 158). For a pure rotational motion, for example, trajectories of all the rigid blocks will be located on certain circles and thus the maximum geometrical distortion of the structure will be always bounded by the circles radii regardless the deformation amplitude. Figure 19 Left F gives a fare approximation of such a motion. For the other extreme case of a pure translational motion, there will be no difference between the two approaches and the distortions produced by the NOLB method will be the same as in the standard approach, as it is shown in Fig. 19 Left D.

We would like to conclude this section mentioning that structural distortions presented above are not a serious obstacle for the applicability of the Cartesian NMA approaches. Indeed, the produced molecular structures can be straightforwardly optimized using standard techniques, for example, gradient-based minimizers and classical force-fields. However, as we hinted above, it will be computationally more efficient to optimize a structure produced by the NOLB approach compared to the linear one due to a typically lower energy of the NOLB structure. Also, at large NMA deformation amplitudes, the result of such an optimization for the linear technique will be generally different from the one of the nonlinear technique. Thus, the presented NOLB approach is a computationally cheap alternative to the other NMA methods when large deformation amplitudes are required.

4.4.3 *Memory and CPU consumption*

Here, we demonstrate the scalability of our method on five molecular structures of various sizes and geometries, as we have described in more detail above. We should specifically mention that these results only demonstrate the performance of our RTB NMA implementation. The subsequent nonlinear analysis of the motions takes only a marginal piece of the total time, which can be ignored. More technically, our method uses sparse data representation and the Lanczos scheme to find a subset of eigenvectors of the Hessian matrix. As a reference, we also provide results of other state-of-the-art NMA methods. These are the RTB module of the ProDy package [19] and the iMod method that performs NMA in internal coordinates [151]. Both of these methods operate with dense matrices and use LAPACK routines for the partial diagonalization. ProDy

Table 6: Memory consumption of the NOLB NMA method on the tested molecular structures. All the computations were performed using the double precision variables. We set the interatomic interaction cutoff to 10 Å. The number of atoms is listed without the heteroatoms. The size of the matrices is given as the number of rows (or columns) they contain.

Name	PDB code	Number of atoms	All-atom Hessian size	RTB Hessian size	Memory required
Chemoreceptor	2ch7	4,630	13,890	3,702	123 Mb
Human γ-secretase	5a63	9,646	28,938	7,338	310 Mb
Terminase	4bij	18,855	56,565	14,220	570 Mb
Photosystem II	5b5e	40,908	122,724	31,494	1,3 Gb
70S ribosome	5j8a	284,479	853,437	123,804	9,3 Gb

computes a subset of eigenvectors of a real symmetric matrix, whereas iMod seeks for a subset of eigenvectors of the generalized symmetric definite eigenvalue problem. We should mention that we also tested the original RTB NMA implementation of Yves-Henri Sanejouand and colleagues [62, 248], but it turned out to be much slower than the other tested methods because of the full Hessian diagonalization. Also, the CHARMM program used to have a generalized RTB method called block-normal-modes (BNM) [144], but it disappeared from the recent CHARMM releases and we could not assess its performance. We present the numerical results measured on a MacBook Pro Mid 2015 laptop with a 2.8 GHz Intel Core i7 processor and 16 GB 1600 MHz DDR3 RAM. The same interaction cutoff value of 10 Å was used in all the tested methods. The rigid blocks in both ProDy and NOLB were constructed on a single residue basis. For the iMod method, we chose all the dihedral angles as degrees of freedom. Table 6 lists the memory consumption of the NOLB method on the tested structures. We can see that even the structure of the E. coli 70S ribosome with ~300,000 of atoms, which is one of the largest in the protein data bank, can be computed with our method on all the modern computers. Figure 22 shows the total execution time of the NOLB, ProDy and iMod methods to compute the first 10, 100, and 1,000 normal modes for five systems of increasing size. We should mention that the NOLB method spends almost all of its time in the diagonalization of the Hessian matrix, thus its total time can be generally attributed solely to the diagonalization procedure. Also, in these tests, we have disabled the output of the computed normal modes, as this might take a significant portion of time. Overall, the timing for our method scales linearly with the size of the molecular structure and nonlinearly with the number of the computed normal modes. Regarding the other two methods, we can draw several observations. First of all, in terms of speed ProDy and iMod are very similar to each other despite the fact that one uses the RTB model in the Cartesian space, while the other uses model representation in the internal coordinates. Second, the performance of these two methods is almost indepen-

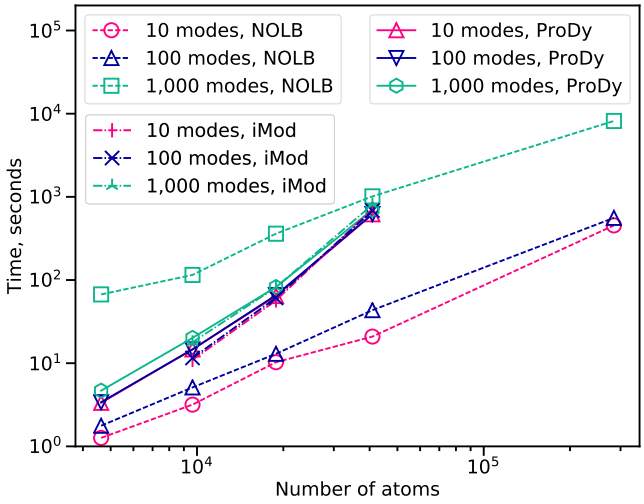


Figure 22: Total time taken by the NOLB, ProDy and iMod methods to compute first 10, 100, and 1,000 normal modes for five molecular structures as a function of their size in a log-log scale. Several data points are missing because ProDy failed on the largest system and iMod failed on the smallest and the largest systems. See the main text for details.

dent of the number of requested modes. We should mention that iMod failed on the smallest chemoreceptor system outputting zero eigenvectors, so we removed these data from the plot. Finally, both methods failed with the segmentation fault on the largest system during the computation of the Hessian matrix. Therefore, we repeated the test removing all the RNA chains from the ribosome molecule, such that the final structure contained only 90,587 atoms, but the two methods failed again. To conclude, if only a few normal modes are required (up to 100), then the sparse iterative scheme based on the Lanczos diagonalization algorithm seems to be advantageous over the other strategies. The difference becomes very significant for mid- to large-size systems starting at about 20,000 of atoms. On the other hand, if all the modes are required, then the dense diagonalization methods are much more effective. Finally, for molecular systems of a very large size starting from about 100,000 of atoms, only the sparse method implemented in NOLB completed the job. We should mention here that, of course, more aggressive coarse-graining schemes can be used for large systems such that dense diagonalization methods will be very efficient as well. Also, our test case is far from being exhaustive and more rigorous comparisons of different diagonalization techniques can be found elsewhere, for example in a recent study from the authors of iMod [149], where they drew the same conclusions regarding the advantage of the iterative Krylov subspace techniques. Overall, this test demonstrated that modern NMA algorithms compute the slowest normal modes for mid-size molecular systems in a very reasonable time, typically in less than a minute, and in many cases these are computed in several seconds almost at the interactive rates.

4.4.4 *NOLB nonlinear transitions better predict a wide range of functional motions*

We assessed the nonlinear transitions computed by NOLB against 132 pairs of experimentally determined structures displaying a wide range of biologically relevant conformational changes. The root mean square deviation (RMSD) between the two structures ranges from 0.5 Å to 33 Å and the motions involve up to 80% of the protein atoms. For each pair, we defined a starting structure and a target structure. For a subset of 23 pairs (open-closed set, see below), each structure alternatively played the role of the starting structure and the target structure, resulting in a total of 155 predicted transitions. The transitions were computed by deforming the starting structure along its lowest-frequency modes, with the mode amplitudes being inferred from the displacement between the starting and target structures. This allows obtaining the optimal (or close-to-optimal) transitions within our framework. Nevertheless, we should stress that the knowledge of the target structure is only used to determine the sense and extent of the deformation along each mode, not the modes themselves. Hence, our approach is markedly different from linear interpolation or other morphing approaches implemented in popular tools [73, 125, 152, 168, 178, 180]. We set the number of selected modes to 10, as it was shown to be sufficient to describe 90% of open-to-closed conformational transitions [151]. Moreover, this allows performing real-time calculations. To compute all transitions reported here, it took us less than 5 minutes with one iteration, and about 15 minutes with five iterations, on a single CPU.

The quality of the conformations produced by NOLB was assessed by computing Procheck [133] G-factor (Fig. 23A). A model resembling experimental structures deposited in the PDB should have a G-factor greater than -0.5 (red dotted line) and the higher the better. The vast majority of NOLB conformations are as good as an experimental structure. By comparison, the quality of the conformations produced by the classical linear extrapolation is much more variable, with a significant proportion displaying very low G-factors. Moreover, about three quarters of the predicted transitions are exclusively comprised of high-quality conformations when we use NOLB. This pro-

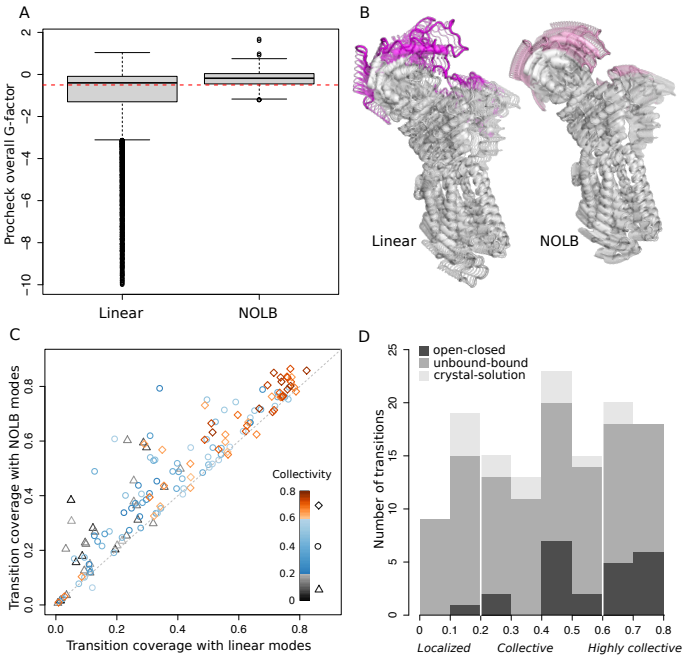


Figure 23: Transition quality, coverage and collectivity. **A.** Comparison of the overall G-factors computed by Procheck [133] on the conformations produced with the classical NMA (linear extrapolation) and with the NOLB method (nonlinear extrapolation). The distributions contain 7 676 and 8 769 conformations, respectively, corresponding to 155 transitions between 132 structure pairs (see text for details). **B.** Closing of the calcium ATPase pump (1su4–1t5s). Conformations predicted by the classical NMA and by NOLB are shown on the left and the right, respectively. The residues undergoing the highest displacements are highlighted in color. **C.** Comparison of the coverage achieved by the NOLB nonlinear modes with 5 iterations (y-axis) versus the classical linear modes (x-axis) for the 155 transitions. The colors indicate the degrees of collectivity of the experimental transitions. **D.** Histogram of the collectivity degrees for all structure pairs from the three test sets. The transitions are labelled as localized (below 0.2), collective (between 0.2 and 0.6) and highly collective (above 0.6).

portion drops to 21% when we use the classical linear extrapolation. Let us stress that 7 (out of 155) transitions start from an experimental structure of poor quality (G-factor below -0.5). The better quality of the NOLB conformations can also be appreciated by directly looking at them, and is particularly visible when dealing with large displacements. For instance, the calcium ATPase pump (1su4-1t5s) undergoes a large domain motion of 13.5 Å, taking place during active transport. While the nonlinear transition computed by NOLB very well preserves the structure of the protein (Fig. 23B, on the right, the linear transition visibly distorts the cytoplasmic headpiece, where the closing motion takes place (Fig. 23B, on the left).

We also evaluated how close the final conformations produced by NOLB were to the target structures. For this, we computed the *transition coverage*, *i.e.* the relative RMSD between the initial and target structures explained by the predicted transition. To give an example, if the initial RMSD is of 5 Å, a prediction achieving a coverage of 70% will produce a final conformation 1.5 Å away from the target structure. On average, the NOLB predictions, computed with five iterations, covered 48% of the transitions. For comparison, the average coverage obtained with the classical linear modes was 40%. Moreover, the nonlinear predictions better approximated the transitions in 92% of the cases (Fig. 23C). The superiority of the NOLB predictions was also found significant without any update of the modes along the transition. Hence, beyond producing conformations with better stereo-chemical properties, the NOLB method also better exploits the information contained in the starting structure's geometry to get closer to the target structure. The anticoagulation factor VIIa (Fig. 18, right panel) gives an illustrative example of

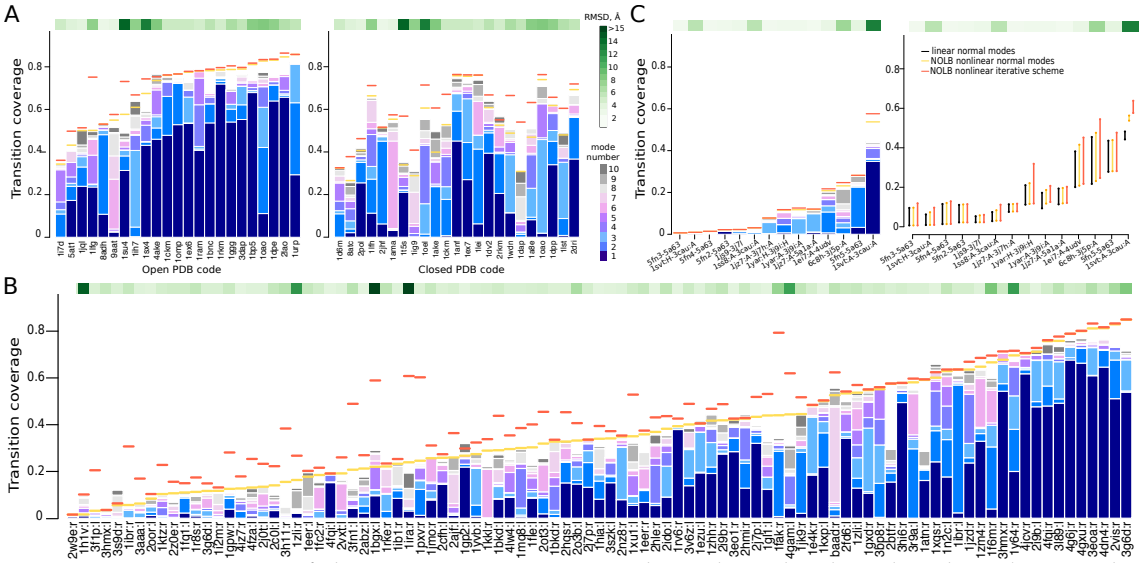


Figure 24: Comparison of the transition coverages achieved on the three benchmark sets. The strips on top show C_α RMSD between the two known structures. The y-axes show the transition coverage achieved by the 10 lowest-frequency linear normal modes (bars in blue tones), the NOLB nonlinear normal modes (in orange) and the NOLB nonlinear iterative scheme (in red). The x-axes list the PDB codes, ordered according the NOLB normal mode predictions' quality **A.** Open-to-closed (on the left) and closed-to-open (on the right) transitions. **B.** Unbound-to-bound transitions. **C.** Crystal-to-solution transitions. The plot on the right shows the improvement of the predictions when increasing the number of active normal modes from 10 to 40.

a partner-binding associated large but localized transition (6.2 Å) that is clearly better described by NOLB. The transition involves a complex motion of an "arm" comprising about 20% of the protein. The classical linear modes covered one third of the transition, producing a conformation 4.1 Å away from the target. The nonlinear NOLB normal modes achieved 44% coverage (Fig. 18, conformations in orange) and 79% after updating the modes 3 times (conformations in red). The final conformation is only 1.3 Å away from the target.

Noticeably, 8 transitions are poorly predicted by classical NMA and by NOLB iterative scheme (coverage below 3.5%, see the points on the diagonal on Fig. 23C). The majority of these cases (5 out of 8) correspond to very small transitions (see also below). Visual inspection revealed that 2 of the remaining cases may be explained by ambiguities or errors in the experimental data (transitions involving the C-α only 4.2 Å resolution cryo-EM structure 3cau) and 1 case displays drastic rearrangements that linear and nonlinear normal modes fail to describe correctly. Let us stress that we did not observed any significant correlation between the transition coverage and the resolution of the starting and/or final structure(s).

4.4.5 *NOLB extends the applicability of the normal mode analysis to localized motions*

We collected the pairs of experimental structures from three benchmark sets designed for different practical applications, namely NMA, docking and cryo-EM fitting. The first set comprises 23 proteins undergoing opening/closing motions. The vast majority of these transitions involve more than 40% of the protein atoms (Fig. 23D, dark grey bars). They can be explained by a few low-frequency normal modes (typically 1-3) computed from the open form (Fig. 24A, see bars in blue tones on the left). The second set contains 95 structural transitions associated to the binding of a protein partner. Such transitions are particularly challenging for protein docking applications [57, 70, 158, 170, 176, 177]. Indeed, they are often induced by the spatial proximity of the partner (induced-fit mechanism), which makes them very difficult to estimate starting only from the knowledge

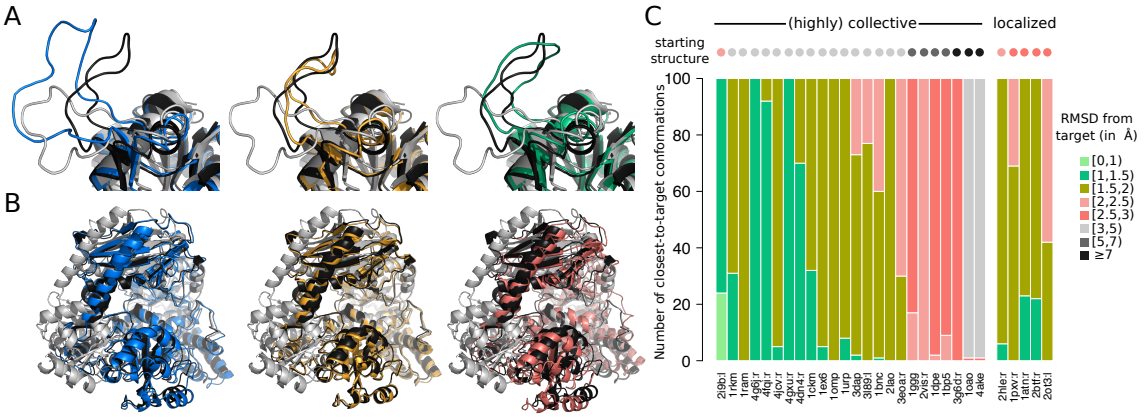


Figure 25: Examples of predictions and random sampling assessment. **A-B.** Superimposition of the predicted conformations (in color) onto the known starting (in grey) and target (in black) structures. The conformations in blue were produced by classical NMA while those in other colors were produced by NOLB. The amplitude of the modes were either determined using the knowledge of the target state (on the left and in the middle) or randomly sampled (on the right, with colors matching panel C). **A.** Transition of actin upon binding to DNase I (1atn:r). The initial RMSD is 2.7 Å and the motion involves 10% of the protein atoms. **B.** Closing of the diaminopimelate dehydrogenase (3dap–1dap). The initial RMSD is 4.2 Å and the motion involves about half of the protein atoms. **C.** RMSDs computed between the 100 closest-to-target randomly sampled conformations and the target structure. In total, 10 000 conformations were generated for each protein. The proteins are split into two groups, depending on the type of transition. Within each group, they are ordered according to the RMSD between the starting and target structures, indicated by the colored dots. The x-axis lists the PDB codes.

of the unbound state. This set includes a great variety of motions, from highly localized to highly collective ones (Fig. 23D, medium grey bars). The transition coverage achieved by the classical linear normal modes is rather low (below 40%) for the majority of transitions (Fig. 24B, see colored bars). The few transitions explained by the first three modes (see right part of the plot) involve more than 70% of the protein atoms and are all antibodies. The third set comprises 14 transitions between either a crystal structure and a solution structure solved by cryo-EM or between two cryo-EM solution structures. Contrary to the other two sets, it is dominated by very small transitions, even below 1 Å. The explanative power of the 10 first modes is very poor on this set (Fig. 24C, left panel). Using 40 active modes significantly improves the coverage (Fig. 24C, right panel). Nonetheless, the small transitions remain difficult to predict.

Overall, the ability of the classical NMA to predict transitions is largely determined by the transitions' collectivity degrees (Fig. 23C, see the color gradient along the x-axis). Highly collective motions tend to be very well predicted while localized motions tend to be poorly predicted, in agreement with previous works [163, 245, 274]. We found that our nonlinear scheme permits to *go beyond this observation and extends the applicability of the NMA*. Indeed, the highest improvement of NOLB predictions over the classical NMA is observed for localized transitions, involving less than 20% of the protein atoms (Fig. 23C, grey dots). The transition coverage is more than twice as big, on average, reaching a maximum value of 60% (versus 40% for the linear normal modes). As illustrative examples, let us mention Ephrin B4 receptor (2hle:r), Cystein protease (1pxv:r), actin (1atn:r and 2btf:r) and Rabex-5 VPS9 domain (2ot3:l), which undergo localized motions upon binding to their partners (Fig. 24B, see the location of the orange and red segments). While the linear modes predict between 23 and 36% of their transitions, our nonlinear scheme predicts between 43 and 60% of them. The linear and nonlinear transitions predicted for actin are illustrated in Fig. 25A (in blue and orange).

4.4.6 *Updating of the modes allows relaxing the elastic network's constraints*

The transitions predicted by the classical NMA strongly depend on the geometrical shape of the starting structure. This is particularly visible on the first test set, where the closed-to-open transitions are significantly worse than the open-to-closed ones (compare the two plots in Fig. 24A). Moreover, the number of transitions explained (at more than 40%) by the first three modes reduces from 18 to 8 upon starting from the closed structure. This effect was observed previously [245] and has a clear physical explanation connected to the limitations of the elastic network model. Indeed, the low-frequency modes are a consequence of the shape of the protein, and the shape of an open structure provides more information about its dynamical potential.

By re-computing the modes along the transition, our iterative scheme permits to overcome this limitation. Namely, it increases the coverage in the closed-to-open direction from 53% to 61%, on average (Fig. 24A, see the location of the red segments on the right). This result can be explained by the fact that, at each iteration, some elastic links are removed, alleviating some of the constraints that exert on the closed structure. As a consequence, the discrepancy between open-to-closed and closed-to-open predictions is largely reduced (compare the left and right plots). In four cases, namely the aspartate amino transferase (9aat-1ama), the maltodextrin binding protein (1omp–1anf), the alcohol dehydrogenase (8adh–2jhf) and the guanylate kinase (1ex6–1ex7), the coverage achieved in the two directions even becomes equivalent. The highest increase in coverage is obtained for the diaminopimelate dehydrogenase (1dap–3dap), from 31% without any update to 54% after one update.

4.4.7 *NOLB recapitulates known intermediates*

Beyond stereochemical realism, we investigated whether the transitions predicted by NOLB could recapitulate known intermediate states. We selected four proteins undergoing large conformational transitions (>6 Å) for which at least one intermediate structure is known (Fig. 26). Three of these proteins were previously studied in similar contexts [180, 270]. We recorded the RMSD from the experimental structures along the predicted transitions (Fig. 26A-D) and quantified the extent to which these transitions spontaneously approached the known intermediate states by computing the *improvement* measure proposed in [270]. It reflects the relative improvement of the predicted transition in recapitulating a given intermediate structure, compared to the starting and target structures.

For the ribose binding protein, the NOLB transition allowed reaching the two intermediate states and the target state with a deviation smaller than 1 Å (Fig. 26A). The improvements are of 56% and 69% for the first and second intermediates, respectively. For the 5'-Nucleotidase, NOLB produced conformations less than 2 Å away from the intermediate structures and covered 75% of the complete 9 Å transition (Fig. 26B). The two intermediates are very similar and the improvement is in the 54-57% range. For the calcium ATPase pump, we focused on the transition from the open E1-2Ca^{2+} state, with a splayed-headpiece, to the closed-headpiece ATP-bound E1-2Ca^{2+} state (Fig. 23B). NOLB covered 76% of the transition, with a final RMSD to the target structure of 3.3 Å, and produced conformations about 4 Å away from sarcolipin-bound E1-Mg^{2+} structures (Fig. 26C). The latter might represent distorted intermediates due to the presence of sarcolipin, known to interfere with the transition by stabilizing the E1-Mg^{2+} state [258]. This may explain the fact that the predicted conformations remain relatively distant from them, with improvement values of 19 and 26%.

Our last case study is that of the chaperone HSP90, which undergoes dramatic conformational changes upon binding to nucleotides. Many states have been characterized but

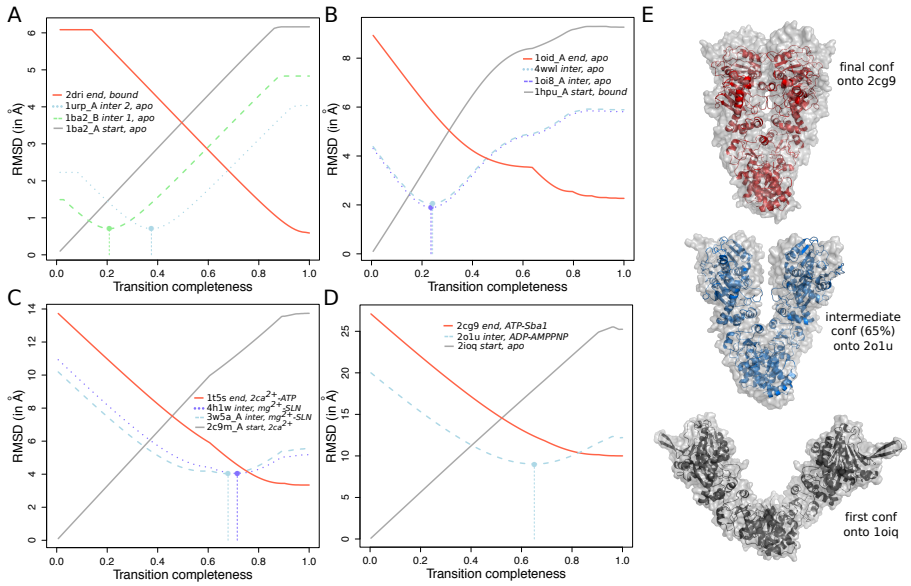


Figure 26: Prediction of transitions with known intermediates. **A.** Bacterial ribose binding protein. **B.** Bacterial 5'-Nucleotidase. **C**. Mammalian calcium ATPase pump. **D-E.** HSP90 homologs from bacteria (HTPG, 2ioq), yeast (HSP82, 2cg9) and mammals (GRP94, 2o1u). The transitions were predicted with 5 iterations of NOLB. **A-D.** RMSD computed along the predicted transition with respect to some experimental structures. The gray and red curves correspond to the starting and target structures respectively, while the curves in blue tones correspond to intermediate structures. The dots indicate the predicted conformations being the closest to the intermediate states. **E.** Superimposition of predicted conformations (in colored cartoons) onto experimental structures (in transparent grey surface) for HSP90.

for some of them only SAXS-based or EM-based low-resolution models are available. Here, we considered three crystallographic structures of HSP90 homologs, suggested to correspond to different steps in the conformational cycle of the chaperone [128]. NOLB covers 63% of the transition from the starting open apo structure to the target closed ATP-bound structure and approaches a semi-open ADP-bound structure along the way (Fig. 26D). The latter was suggested to represent an intermediate between the two others, or a non-catalytic conformation [128]. Although NOLB conformations remain relatively far from this structure (about 12 Å away), it is significantly closer than the two extreme structures, with an improvement of 29%. We should stress that this case is particularly challenging as the transition is of several tens of Å and we are dealing with proteins coming from different organisms and sharing about 40% sequence identity. Sequence divergence may be accompanied by local conformational rearrangements increasing the RMSD. Visual inspection of the conformations shows a good match with the experimental structures (Fig. 26E).

4.4.8 *NOLB produces near-target conformations by random sampling*

In the general case, the target is not known and one has to sample the amplitudes of the modes. In most of the practical applications, however, the sampling is guided by additional information, which can be docking scores, small-angle scattering profiles, Cryo-EM envelopes, cross-linking constraints, etc. To test whether this case could be dealt with in practice, we devised a conformational sampling strategy and applied it to a subset of 29 proteins from our datasets (Fig. 25). These proteins either display collective or highly collective transitions that are very well predicted (>70% coverage with NOLB), or localized motions poorly described by the linear modes (<30%) but well described by our iterative scheme (>40%). For each protein, we generated 10 000 conformations by progressively deforming the starting structure along its slowest normal modes using

NOLB. The relative amplitudes of the modes were randomly sampled and fitted to a given RMSD. The simulation was decomposed into two or three steps, depending on the expected extent of the deformation. The first step performs most of the expected displacement using only the three slowest modes. For large transitions (>4.5Å), an additional second step is performed using 5 modes and a smaller displacement. The final step consists in exploring the space around the previously generated conformations within 1 Å and exploiting all 10 slowest modes.

Depending on the protein, the simulation was able to produce conformations as close as 0.8-3 Å to the target structure (Fig. 25C). Moreover, for a large majority of proteins (22 out of 29), the simulation produced several conformations with deviations smaller than 2 Å. The localized motions are more difficult to recapitulate than the collective ones (Fig. 25C, compare left and right subpanels) but some generated conformations are still as close as 1.3-1.8 Å to the target. For instance, the closest-to-target conformation generated for actin (1atn:r) deviates by 1.3 Å from the target (Fig. 25A, in green). This is better than the conformation produced by target-informed classical NMA (1.9 Å, in blue) and only slightly worse than that produced by target-informed NOLB (1.1 Å, in orange). For the collective transitions, there is a clear tendency for further away targets to be more difficult to reach (Fig. 25C, see the correlation between the bars and dots color gradients). For example, in the case of carbon monoxide dehydrogenase (1oao), the best conformation is found 2.9 Å away from the target state (Fig. 25B, in pink). This is about 1.5 Å more than the conformations predicted using the knowledge of the target state (in blue and orange). Nevertheless, we can see that the randomly sampled conformation superimposes well onto the target structure and recapitulates most of the transition. We performed four additional simulations replicates, for each protein, using different random seeds for sampling the modes' amplitudes, and they produced similar results.

4.5 CONCLUSION

This Chapter revisits the formalism of normal modes and demonstrates its applicability to the previously inaccessible cases of localized motions. Firstly, we present a conceptually simple and computationally efficient method for the *nonlinear normal mode analysis*. It relies on the rotation-translation of rigid blocks theoretical basis developed by Y.-H. Sanejouand and colleagues [62, 248]. Secondly, this Chapter critically assesses the relevance of the normal mode analysis to the computation of various structural transitions in biological macromolecules.

Our results challenge the long-standing belief that the lowest-frequency modes can only describe collective transitions. Indeed, we show that nonlinear normal modes can also approximate local deformations such as loop motions. Moreover, iterative recomputation of the normal modes relaxes constraints imposed by the geometry of the protein and allows pushing the transitions even further. Another important advantage of our method is that the predicted conformations have a much better local geometry than those resulting from linear NMA perturbations. We demonstrated it by computing the quality of the transition intermediate states using the MolProbity and Procheck scores. We also showed that the predicted transitions recapitulate the known intermediate states solved experimentally, and that we can predict the transitions by randomly sampling the amplitudes of the lowest normal modes.

Small structural changes, for example those present in the Cryo-EM 2015/2016 Model Challenge benchmark, despite our recent efforts [130], still remain very difficult to predict with the NMA formalism. Indeed, in this case adding nonlinearity and iterative computations did not improve the results significantly. Activating a much larger number of modes can help approximating the transitions, but at the expense of a signifi-

cant computational cost. Indeed, the full diagonalization of the Hessian matrix scales as $O(N^3)$ with the number of degrees of freedom N. Therefore, it becomes preferable to use MD-based or other stochastic optimization techniques, *i.e.* simulated annealing, with the full range of degrees of freedom.

Our method is very CPU and memory efficient – it took us about 9 minutes to compute the nonlinear structural transitions for all proteins from the PPDBv5 (460 in total) set on a desktop computer. This implies that the method can be applied on a very large scale. For instance, it can be used to model flexibility in docking calculations or to generate putative conformations that can be targeted by small molecules.

5 SCATTERING

My interest in *Spherical Harmonics* and the *plane-wave expansion* has led me to the development of a near-linear-scaling method for the *calculation of small-angle X-ray and neutron scattering profiles* [83]. This is the fastest method so far (Pepsi-SAXS and Pepsi-SANS), and it has started being used by the community. Together with my experimental colleagues Anne Martel and Sylvain Prevost from ILL Grenoble, we have integrated this tool into a modeling platform at https://pepsi.app.ill.fr. I have applied this method to practical questions within CASP12/13 protein structure modeling exercises [101] and also to the molecular systems of my experimental collaborators [71]. In particular, I contributed to the development of a SAXS- and SANS-based ensemble refinement method using a Bayesian/Maximum Entropy approach [132].

5.1 INTRODUCTION

Small-angle scattering is one of the fundamental techniques for structural studies of biological systems. Small-angle X-ray scattering (SAXS) is a type of small-angle scattering where X-rays scatter elastically from the sample and are then collected at very small angles. Compared to other structure determination methods, SAXS experiments are very simple conceptually and thanks to advances in instrumentation [236], the SAXS technique, particularly, solution-state SAXS, is becoming very popular in the recent years as a complement to other methods in structural biology [80, 211]. SAXS also allows to overcome some restrictions of other experimental techniques, for example, it is applicable to all system's sizes, it allows to study particles in solution, it is relatively fast and destroy the sample only marginally. On the downside, SAXS can only determine the electron density's distance distribution function at a supra-nm resolution, however, it can distinguish conformations of a protein at a sub-nm resolution [280].

Over the years, a number of computational tools have been developed for the analysis of the solution-state SAXS curves, calculation of theoretical profiles and low-resolution reconstruction of model shapes. The most prominent of them is the ATSAS package developed at EMBL Hamburg [192]. To test a structural hypothesis or to construct a model system based on a SAXS experiment, an accurate and rapid calculation of a model SAXS profile is required. The running time of a method depends, among others, on the number of atoms in a model N, and the number of points in the scattering curve M. Tools that directly use the Debye equation have the cost of $O(N^2)$, whereas methods that use a linear approximation to the scattering equation have the cost of $O(N)$. Generally speaking, the same type of calculations should be repeated for each point in the scattering curve, which determines the worst-case performance of $O(N^2M)$. Keeping in mind the typical values of M and N to be of several thousands, this running time is usually prohibitive to do any kind of multiple model assessment. Thus, many efforts have been put in recent years to reduce the running time of SAXS computational tools without degrading the quality of their approximations. Below we give a brief overview of the most notable computational methods for the calculation of theoretical SAXS profiles given an atomic model as input. A deeper discussion of different computational techniques can be found elsewhere [212].

The most popular method is the CRYSOL program developed by Svergun and colleagues [243]. The method uses the theory of multipole expansions of scattering intensity initially developed by Stuhrmann [240]. The running time of the initial implementation

of the method had linear dependence on both the number of atoms in a molecule N, and the number of points in the scattering curve M as $O(NM)$. A more recent version of the program, however, maps experimental scattering intensities and associated errors onto a sparser grid [192], thus, reducing the computational cost to $O(N + M)$. The method is, generally, very fast, but has the major disadvantage (as of CRYSOL2) of a simplistic representation of the sample's hydration shell using a two-dimensional angular function [241]. CRYSOL3 introduces a better approximation of the hydration shall, being, however, significantly more computationally expensive, and also having one additional adjustable parameter [72]. The SASSIM method is very similar to CRYSOL, but the hydration shell is defined in terms of spherical harmonics and is calculated using a Lebedev grid [164].

Another popular FoXS program uses a linear approximation to the Debye scattering equation, which decouples the dependency of the running time on the number of atoms N in a model and the number of points in the scattering curve M as $O(N^2 + MN)$ [224, 226]. When created, the program was notably faster compared to the initial implementation of CRYSOL if tested on experimental curves with several thousands of points. However, the later development of CRYSOL, as we demonstrate below, outperforms FoXS for nearly all test cases.

A logical extension of the multipole expansion method is the computational scheme that uses three-dimensional Zernike polynomials for the representation of the electron density [148]. Here, the angular dependence of scattering amplitudes on the scattering vector is described, similarly to CRYSOL, using the spherical harmonics, but the radial dependence is expanded using a set of orthogonal functions. The computational complexity of this method is $O(N + M)$, however, the hidden time-limiting step is the computation of the three-dimensional Zernike moments. In order to calculate them, atomic models are mapped onto a three-dimensional grid, whose size can be adjusted according to the resolution of data.

Recently, some other linear-scaling schemes have been proposed. The golden-ratio scheme by Watson and Curtis [268] uses the Euler's formula to compute the rotationally averaged scattering intensity $I(q)$ by evaluating $I(\underline{q})$ in several scattering directions using the exact expression for $I(\underline{q})$ at a given wave-vector $\underline{q}$. The orientations of the $\underline{q}$ vectors are taken from a quasi-uniform spherical grid generated by the golden ratio. The hierarchical algorithm for fast summation of the Debye equation by Gumerov et al. [88] is similar to the fast multipole method (FMM) and is based on a hierarchical spatial decomposition of electron density using local harmonic expansions and translation operators for these expansions. Its computational cost is $O(N \log N)$.

Some efforts have been spent to a more precise description of solvation. The AXES method uses explicit water molecules equilibrated in a water box using molecular dynamics (MD) simulations to accurately model the scattering amplitudes of the surface and displaced solvent [82]. Another method calculated hydration shell intensities from MD trajectories of water molecules around a fixed protein [187]. The AquaSAXS method models non-uniform hydration shell of a protein by taking advantage of recently developed methods that compute the solvent-distribution around a given solute on a 3D grid such as the Poisson–Boltzmann–Langevin formalism or the three-dimensional reference interaction site model [201].

Finally, to increase the speed of calculations, several coarse-grained schemes have been proposed. For example, one recent method is based on the Debye formula and a set of scattering form factors for dummy atom representations of amino acids [239]. The FastSAXS-pro [275] algorithm uses the Debye-based approach and coarse-grained residue- and nucleotide- level structure factors. The method explicitly takes into account the non-homogeneous distribution within the hydration layer by assigning a different scaling factor for dummy water molecules according to their proximity to protein and DNA/RNA.

Finally, the method by Zheng and Tekpinar [280] uses a one-bead-per-residue coarse-grained protein representation coupled with the elastic network model. The hydration shell is modelled implicitly by combining each residue and its nearby implicit water molecules into a composite representation.

This Chapter present Pepsi-SAXS (Pepsi stands for Polynomial Expansions of Protein Structures and Interactions) [83], a new implementation of the multipole-based scheme proposed by Stuhrmann [240]. Overall, our method is significantly faster compared to CRYSOL, FoXS, and the 3D-Zernike implementation from the SAStbx package [147], as we demonstrate below using an excessive number of test cases. We use a very fast model for the hydration shell computation based on a uniform grid of points. We also use the adaptive order of the multipole expansion. More precisely, according to the Nyquist–Shannon–Kotelnikov sampling theorem [157], we determine the required expansion order using the *radius-of-gyration of the model's hydration shell* and the value of the maximum scattering vector q_{max}. Then, we represent the scattering intensity curve using a cubic spline interpolation, which allows us to significantly speed up the running time of our method. Finally, we introduce partial scattering intensities to rapidly fit the theoretical curve to the experimental one using exhaustive search in two adjustable parameters. We should also mention that we paid particular attention when deriving parameters for the form factors, especially those for charged and resonance groups.

5.2 MATERIALS AND METHODS

5.2.1 *The Multipole Expansion Theory*

The scattering theory presented here closely follows the works of H. B. Stuhrmann [240] and D. Svergun [243, 244], as it has been presented in our original Pepsi-SAXS method [83]. The spherically averaged scattering intensity $I(q)$ from a single molecule immersed in a solvent with bulk scattering density ρ can be written as

$$I(q) = \left\langle |A_a(\underline{q}) - \rho A_c(\underline{q}) + \delta\rho A_b(\underline{q})|^2 \right\rangle_\Omega , \quad (172)$$

where $A_a(\underline{q})$ is the scattering amplitude from the molecule in vacuum, $A_c(\underline{q})$ is the scattering amplitude from the excluded volume, and $A_b(\underline{q})$ is that from the hydration shell, which is assumed to have the scattering density different from the bulk value by $\delta\rho$ [243]. The scattering vector q is defined as $q = 4\pi \sin\theta/\lambda$, where 2θ is the scattering angle and λ is the wavelength of the incident X-ray beam. Due to the spherical averaging of the intensity, it is very convenient to introduce the multipole expansion of the scattering intensities and amplitudes in the spherical coordinates system [240]. Using this expansion up to the maximum expansion order L, we can re-write the intensity as

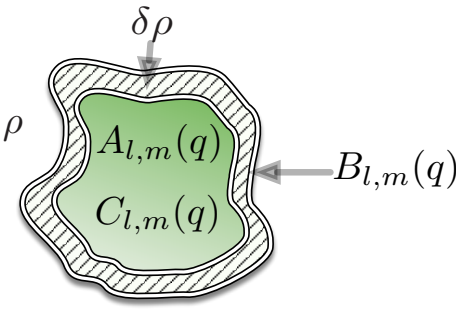


Figure 27: Schematic representation of a molecular geometry and the three scattering contributions $A_{lm}(q)$, $B_{lm}(q)$, and $C_{lm}(q)$. The bulk density is ρ and the difference near the molecular surface compared to the bulk is $\delta\rho$.

$$I(q) \approx \frac{1}{4\pi} \sum_{l=0}^{L} \sum_{m=-l}^{l} |A_{lm}(q) - \rho C_{lm}(q) + \delta\rho B_{lm}(q)|^2, \quad (173)$$

where $A_{lm}(q)$, $B_{lm}(q)$, and $C_{lm}(q)$ are the expansion coefficients of the amplitudes $A_a(\underline{q})$, $A_b(\underline{q})$, and $A_c(\underline{q})$, respectively [243]. Given atomic coordinates of a molecule consisting of N atoms expressed in the spherical coordinate system $\underline{r_i} \equiv (r_i, \omega_i)$, and the corre-

sponding form factors $f_i(q)$, we can write the vacuum scattering amplitude expansion coefficients as

$$A_{lm}(q) = 4\pi i^l \sum_{i=1}^{N} f_i(q) j_l(qr_i) Y_{lm}^*(\omega_i), \tag{174}$$

where $j_l(qr_i)$ are the spherical Bessel functions and $Y_{lm}^*(\omega_i)$ are the complex conjugated spherical harmonics. Similarly, given coordinates of the hydration shell of the molecule sampled at N_{hs} points, its expansion coefficients can be written as

$$B_{lm}(q) = 4\pi i^l h(q) \sum_{i=1}^{N_{hq}} j_l(qr_i) Y_{lm}^*(\omega_i), \tag{175}$$

where $h(q)$ is the form factor of a water molecule scaled with the ratio of the bulk water density to the density of the sampling points in the hydration shell. Finally, the excluded volume contribution can be written as

$$C_{lm}(q) = 4\pi i^l \sum_{i=1}^{N} g_i(q) j_l(qr_i) Y_{lm}^*(\omega_i), \tag{176}$$

where $g_i(q)$ are the form factors of the dummy atoms centered at the positions of molecular atoms $\underline{r_i}$. Figure 27 schematically illustrates the contributions $A_{lm}(q)$, $B_{lm}(q)$, and $C_{lm}(q)$ to the total scattering intensity of a molecule.

5.2.2 *Scattering from multiple particles*

Let $A_t(\underline{q})$ be the the *total* scattering amplitude of a single particle,

$$A_t(\underline{q}) = A_a(\underline{q}) - \rho A_c(\underline{q}) + \delta\rho A_b(\underline{q}), \tag{177}$$

and $O_{lm}(q)$ be the multipole expansion coefficients of the total amplitude,

$$O_{lm}(q) = A_{lm}(q) - \rho C_{lm}(q) + \delta\rho B_{lm}(q). \tag{178}$$

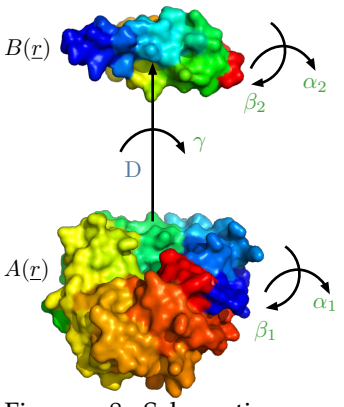


Figure 28: Schematic representation of a molecular geometry between 2 rigid particles.

Let now move this particle to a new position in space by applying a composition of spatial operators, first, a rotation about the origin $\hat{R}$ by a set of angles Λ_1, then, a translation along axis z, $\hat{T}_z$, by amount Δ, and then another rotation about the origin $\hat{R}$ by a set of angles Λ_2. Figure 28 schematically shows the geometry of the scattering particles. The expansion coefficients will transform according to following equations,

$$(\hat{R}(\Lambda_2)\hat{T}_z(\Delta)\hat{R}(\Lambda_1)O)_{lm}(q) = \sum_{m_1=-l}^{l} D_{mm_1}^{l}(\Lambda_2) \sum_{l_1=|m_1|}^{L} T_{l,l_1}^{m_1}(q\Delta) \sum_{m_2=-l_1}^{l_1} D_{m_1m_2}^{l_1}(\Lambda_1) O_{l_1m_2}(q), \tag{179}$$

where $T_{l,l'}^{m}$ are translation matrix elements given by

$$T_{l,l'}^{m}(\rho\Delta) = \sum_{p} i^p j_p(\rho\Delta) 4\pi \sqrt{\frac{2p+1}{4\pi}} c^p(l', m, l, m), \tag{180}$$

and $c^{l_2}(l,m,l_1,m_1)$ are Slater coefficients defined through the triple spherical harmonics integrals [16, 278],

$$c^{l_2}(l,m,l_1,m_1) = \int_{4\pi} Y_l^{m*}(\Omega) Y_{l_1}^{m_1}(\Omega) Y_{l_2}^{m-m_1}(\Omega) d\Omega. \tag{181}$$

These equations substitute the basis for the rapid rigid-body modeling of proteins and their complexes.

5.2.3 *Analytical modeling of particle aggregation effects*

Now, let us express the total scattering from multiple identical particles that have a certain radial distribution from the central one, and that also have a uniform angular distribution with respect to three Euler angles Λ. The total scattering intensity will be

$$\begin{aligned}
&I(q) \approx \frac{1}{4\pi}\sum_{l=0}^{L}\sum_{m=-l}^{l} O_{lm}(q)O_{lm}^*(q) + \\
&\frac{1}{4\pi}\sum_{l=0}^{L}\sum_{m=-l}^{l}\left\langle \left[\sum_{m_1=-l}^{l}\sum_{l_1=|m_1|}^{L}\sum_{m_2=-l_1}^{l_1} D_{mm_1}^{l}(\Lambda_2) T_{l,l_1}^{m_1}(q\Delta) D_{m_1m_2}^{l_1}(\Lambda_1) O_{l_1m_2}(q)\right]\right. \\
&\left.\left[\sum_{m_1'=-l}^{l}\sum_{l_1'=|m_1'|}^{L}\sum_{m_2'=-l_1'}^{l_1'} D_{mm_1'}^{l*}(\Lambda_2) T_{l,l_1'}^{m_1'*}(q\Delta) D_{m_1'm_2'}^{l_1'*}(\Lambda_1) O_{l_1'm_2'}^*(q)\right]\right\rangle_{\Lambda_1\Lambda_2\Delta} + \\
&2\frac{1}{4\pi}\sum_{l=0}^{L}\sum_{m=-l}^{l}\left\langle \left[\sum_{m_1=-l}^{l}\sum_{l_1=|m_1|}^{L}\sum_{m_2=-l_1}^{l_1} D_{mm_1}^{l}(\Lambda_2) T_{l,l_1}^{m_1}(q\Delta) D_{m_1m_2}^{l_1}(\Lambda_1) O_{l_1m_2}(q)\right] O_{lm}^*(q)\right\rangle_{\Lambda_1\Lambda_2\Delta} \\
&= \frac{1}{4\pi}\sum_{l=0}^{L}\sum_{m=-l}^{l}|O_{lm}(q)|^2 + \frac{1}{4\pi}\sum_{l=0}^{L}\sum_{m=-l}^{l}\frac{1}{1+2l} \\
&\left\langle \sum_{m_1=-l}^{l}\sum_{l_1=|m_1|}^{L}\sum_{m_2=-l_1}^{l_1}\frac{1}{1+2l_1}|T_{l,l_1}^{m_1}(q\Delta)|^2|O_{l_1m_2}(q)|^2\right\rangle_{\Delta} + 2\frac{1}{4\pi}|O_{00}(q)|^2\langle T_{0,0}^0(q\Delta)\rangle_{\Delta}.
\end{aligned} \tag{182}$$

This relation follows from the orthogonality of Wigner matrices, please see Section 2.6 for more detail, and the fact that all cross-terms $\langle D_{mm_1}^{l}(\Lambda_2) T_{l,l_1}^{m_1}(q\Delta) D_{m_1m_2}^{l_1}(\Lambda_1) O_{l_1m_2}(q) O_{lm}^*(q)\rangle_{\Lambda_1\Lambda_2\Delta}$

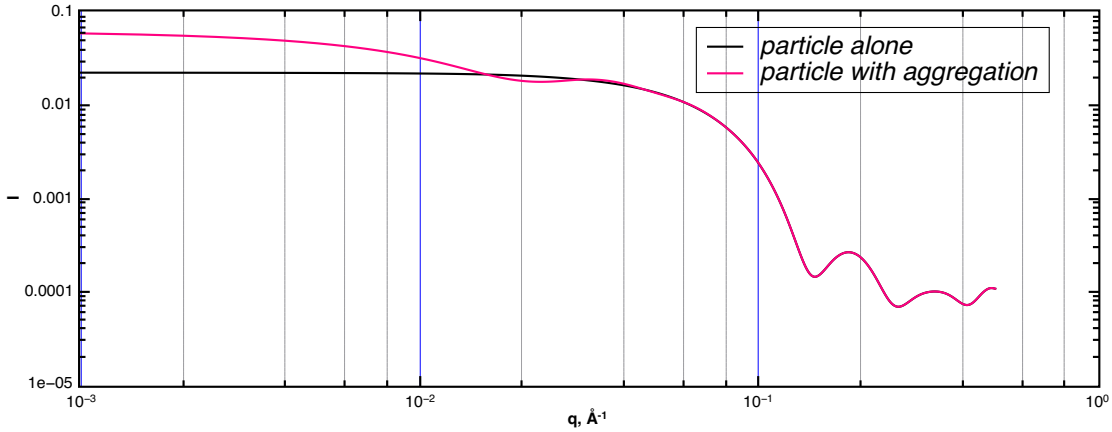


Figure 29: Comparison of a scattering profile of single particle and the same particle with the aggregation effect.

vanish except for $l = m = 0$, please see Section 2.8 . Let us look closer to the space-average term, regrouping the terms we obtain

$$\begin{aligned}
&\sum_{l_1=0}^{L}\left[\sum_{m_2=-l_1}^{l_1}\frac{|O_{l_1m_2}(q)|^2}{1+2l_1}\right]\sum_{l=0}^{L}\sum_{m_1=-\min(l,l_1)}^{+\min(l,l_1)}\left\langle |T^{m_1}_{l,l_1}(q\Delta)|^2\right\rangle_\Delta \\
&=\sum_{l_1=0}^{L}\left[\sum_{m_2=-l_1}^{l_1}\frac{|O_{l_1m_2}(q)|^2}{1+2l_1}\right]\sum_{l=0}^{L}\sum_{m_1=-\min(l,l_1)}^{+\min(l,l_1)} \\
&\sum_{p,q}\left\langle j_p(\rho\Delta)j_q(\rho\Delta)\right\rangle_\Delta 4\pi\sqrt{(2p+1)(2q+1)}c^p(l,m_1,l_1,m_1)c^q(l,m_1,l_1,m_1) \\
&=\sum_{l_1=0}^{L}\left[\sum_{m_2=-l_1}^{l_1}\frac{|O_{l_1m_2}(q)|^2}{1+2l_1}\right]\sum_{l=0}^{L}\sum_{p}\left\langle j_p^2(\rho\Delta)\right\rangle_\Delta 4\pi c^p(l,0,l_1,0)\sqrt{(2l+1)(2l_1+1)} \\
&=4\pi\sum_{l_1=0}^{L}\left[\sum_{m_2=-l_1}^{l_1}\frac{|O_{l_1m_2}(q)|^2}{\sqrt{2l_1+1}}\right]\sum_{p}\left\langle j_p^2(\rho\Delta)\right\rangle_\Delta\sum_{l=0}^{L}\sqrt{2l+1}c^p(l,0,l_1,0) \\
&=4\pi\sum_{l_1=0}^{L}\left[\sum_{m_2=-l_1}^{l_1}\frac{|O_{l_1m_2}(q)|^2}{\sqrt{2l_1+1}}\right]\sum_{l=0}^{L}\sqrt{2l+1}\sum_{p=|l_1-l|}^{l_1+l}\left\langle j_p^2(\rho\Delta)\right\rangle_\Delta c^p(l,0,l_1,0).
\end{aligned} \tag{183}$$

We should note that $c^p(l,0,l_1,0) = c^p(l_1,0,l,0)$. To proceed further, we need to compute radial averages for a given *fractal dimension* d of the form

$$\left\langle j_p^2(\rho\Delta)\right\rangle_{\Delta,d} = \frac{\int_{r_{\min}}^{r_{\max}} j_p^2(\rho r)r^{d-1}dr}{\int_{r_{\min}}^{r_{\max}} r^{d-1}dr}. \tag{184}$$

Next sections provide the computation of the radial averages under different approximations of the particle distributions with respect to the *fractal dimension* d.

5.2.4 *Fractal dimension 3*

Let us first consider the case of fractal dimension $d = 3$. In other words, this is the case when scattering particles are evenly distributed within 3D clusters. The result can be expressed through the closed-form integral $H_l^2(x)$ (see eq. 48 from [28]),

$$H_l^2(x) \equiv \int x^2 j_l(x)^2 dx = \frac{x^3}{2}\left(j_l(x)^2 - j_{l-1}(x)j_{l+1}(x)\right). \tag{185}$$

Then,

$$\left\langle j_p^2(\rho\Delta)\right\rangle_{\Delta,d=3} = \frac{3}{4\pi(r_{\max}^3 - r_{\min}^3)}\int_{r_{\min}}^{r_{\max}} j_p^2(\rho r)r^2 dr = \frac{3}{4\pi(r_{\max}^3 - r_{\min}^3)}$$
$$\left[r_{\max}^3\left(j_p^2(\rho r_{\max}) - j_{p-1}(\rho r_{\max})j_{p+1}(\rho r_{\max})\right) - r_{\min}^3\left(j_p^2(\rho r_{\min}) - j_{p-1}(\rho r_{\min})j_{p+1}(\rho r_{\min})\right)\right]. \tag{186}$$

For the special case of the 0th order we have (also see eq. 41 from [28]),

$$\left\langle j_0^2(\rho\Delta)\right\rangle_{\Delta,d=3} = \frac{3}{4\pi\rho^3(r_{\max}^3 - r_{\min}^3)}\left[0.5\rho r_{\max} - 0.25\sin(2\rho r_{\max}) - 0.5\rho r_{\min} + 0.25\sin(2\rho r_{\min})\right]. \tag{187}$$

Figure 29 shows a difference between profiles with and without aggregation.

5.2.5 *Fractal dimension 2*

For the fractal dimension $d = 2$ we will obtain a recurrent relation for the integral $H_l^1(x)$ using eq. 46 from [28] and a closed-form expression 47 for $H_l^{-1}(x)$,

$$H_l^1(x) = H_{l-1}^1(x) - \frac{1}{2}(2l-2)H_{l-1}^{-1}(x) + \frac{1}{2}j_{l-1}(x)^2 - xj_{l-1}(x)j_l(x) \tag{188}$$

5.2.6 *Fractal dimension 1*

Unfortunately, the above recurrent relation can not be applied to the evaluation of $H_l^0(x)$. Instead, we will combine equations 9b, 61 and 72 from [28] to obtain

$$H_{l+1}^0(x) = \frac{2l+1}{2l+3}H_l^0(x) + \frac{H_{l+2}^2(x) - H_l^2(x)}{(2l+3)^2} - \frac{1}{2l+3}xj_l(x)^2, \tag{189}$$

with

$$H_0^0(x) = \mathrm{Si}(2x) - xj_0(x)^2, \tag{190}$$

and

$$H_0^2(x) = \frac{1}{2}x - \frac{1}{4}\sin 2x. \tag{191}$$

5.2.7 *Cylindrical averaging of the scattering intensity*

Let us now introduce an extension of the previous method to cylindrically-averaged scattering profiles. In the case where all the particles are oriented in the same direction (thats to some external electric field or water flow, for example), the scattering intensity in eq. 172 is not spherically averaged anymore . Instead, it is only cylindrically averaged over the φ angle in the reciprocal space. Then, this equation becomes

$$I(q,\theta) = \left\langle |A_t(\underline{q})A_t^*(\underline{q})\right\rangle_\varphi \sin(\theta), \tag{192}$$

where we again introduced the total scattering amplitude $A_t(\underline{q}) = A_a(\underline{q}) - \rho A_c(\underline{q}) + \delta\rho A_b(\underline{q})$ to simplify the notations. Then, it is again useful to introduce its expansion coefficients $O_{lm}(q)$ from eq. 178. After substituting the expansions 174-176 to the previous equation, we obtain

$$I(q,\theta) = \int_{\varphi=0}^{2\pi}\sum_{l=0}^{L}\sum_{m=-l}^{l} O_{lm}(q)Y_{lm}(\theta,\varphi)\sum_{l'=0}^{L}\sum_{m'=-l'}^{l'} O_{l'm'}^*(q)Y_{l'm'}^*(\theta,\varphi)\sin\theta d\varphi. \tag{193}$$

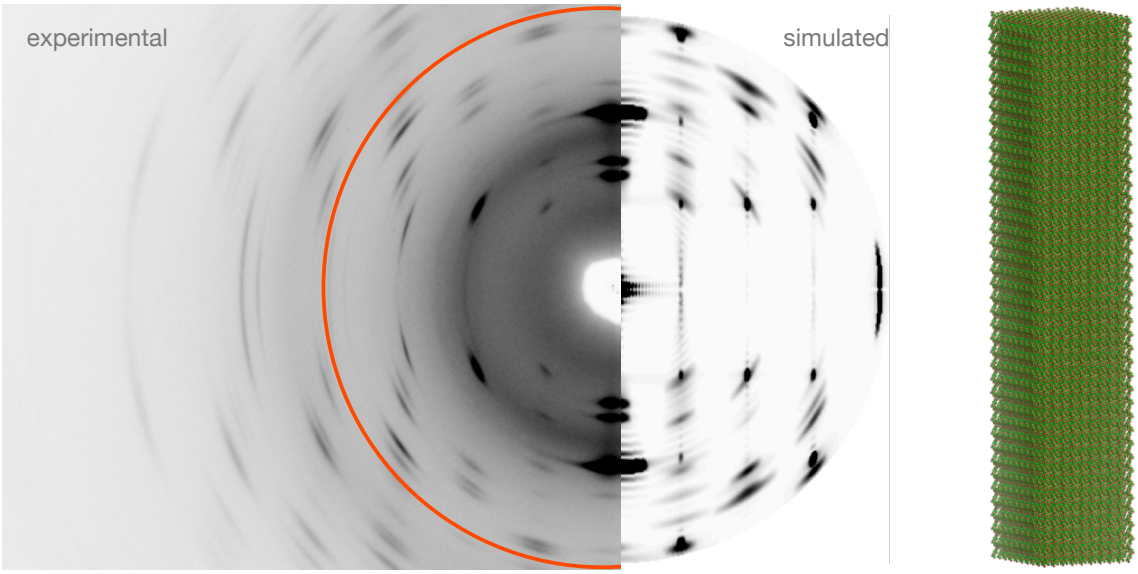


Figure 30: Experimental versus calculated scattering profiles for a cylindrically-averaged intensities of a fibril cristal.

Thanks to the orthogonality of spherical harmonics, this integral simplifies to

$$I(q,\theta) = 2\pi \sin\theta \sum_{m=-L}^{L} \sum_{l=|m|}^{L} \sum_{l'=|m|}^{L} O_{lm}(q) K_{lm} P_{lm}(\cos\theta) O^*_{l'm}(q) K_{l'm} P_{l'm}(\cos\theta), \quad (194)$$

with constants

$$K_{lm} = \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-m)!}{(l+m)!}}. \quad (195)$$

5.2.8 *Angular distribution along the orientation axis*

For the cylindrical averaging case, we have assumed that the particle's orientation is fixed along a single axis. In reality, however, there is a certain statistical distribution of the orientations around this axis. Here we aim to model this effect in a computationally efficient way. Let angle β represent the deviation from the cylindrical axis, and let us also assume β distributed following a Gaussian distribution with the mean $\mu = 0$ and the standard deviation σ.

Let us first introduce the angular dependence to the expansion coefficients $O_{lm}(q)$ using eq. (18,

$$O_{lm}(q,\alpha,\beta,\gamma) = \sum_{k=-l}^{l} O_{lk}(q) e^{-ik\alpha} d^l_{k,m}(\beta) e^{-im\gamma}. \quad (196)$$

Now, we can compute a spherical average of the intensity $I(q,\theta,\alpha,\beta,\gamma)$ weighted with the Gaussian distribution over β. This requires the evaluation of the following integrals,

$$\int_\beta O_{lm}(q,\beta) O^*_{l'm}(q,\beta) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\beta^2}{2\sigma^2}} \sin\beta d\beta = \sum_{k=0}^{L} O_{lk}(q) O_{l'k}(q) \int_\beta d^l_{m,k}(\beta) d^{l'}_{m,k}(\beta) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\beta^2}{2\sigma^2}} \sin\beta d\beta. \quad (197)$$

This integral is not analytical and we will use a Gaussian quadrature scheme to approximate it numerically. Figure 30 shows a fibril cristal and an overlap of the experimental and simulated scattering profiles.

5.2.9 *Form factors and unified atomic groups*

Computation of the expansion coefficients $A_{lm}(q)$, $B_{lm}(q)$, and $C_{lm}(q)$ requires knowledge of form factors $f_i(q)$, $g_i(q)$, and $h(q)$. For the calculation of form factors for the individual atoms, we use the five-gaussian approximation with coefficients taken from [266],

$$f(q) = c + \sum_{i=1}^{5} a_i e^{-b_i q^2}. \tag{198}$$

However, structural databases such as the Protein Data Bank (PDB) [25] typically provide coordinates of only non-hydrogen atoms. Therefore, it is useful to introduce *unified atomic groups* with the positions located at the centers of the heavy atom's nuclei and the corresponding scattering parameters computed for the heavy atoms with the covalently bonded hydrogen atoms. For example, the form factor for such a group f_{CH_n} with n H-atoms attached to the C heavy atom can be computed using the Debye equation as follows,

$$f_{CH_n}(q)^2 = f_C(q)^2 + n^2 f_H(q)^2 + 2n f_H(q) \frac{\sin(qr_H)}{qr_H}, \tag{199}$$

where f_C and f_H are the atomic form factors for C and H atoms given by the five-gaussian approximation (eq. 198), and r_H is the distance between C and H atoms. Distances r_H between the heavy atom and hydrogens in various atomic groups typical for biological molecules are taken from [6].We should note that a simpler approximation holds for practical values of scattering vector q [91],

$$f_{CH_n}(q) = f_C(q) + n f_H(q) \frac{\sin(qr_H)}{qr_H}, \tag{200}$$

which can also be derived from the spherical averaging of the scattering amplitudes instead of the scattering intensities.

We explicitly introduced individual form factors for charged groups of carboxylate, phosphate, guanidine, and ammonium. Form factors for NH^+, NH_2^+, NH_3^+ from guanidine and ammonium groups were approximated according to the model of the electron distribution in the ammonium ion [21]. More specifically, we modelled the central spherical charge cloud with six electrons around the N nucleus with unperturbed hydrogen electron distributions centered not on the protons but inwards along the N-H bonds at 0.76 of the N-H separation distance. We also paid a particular attention to the resonance forms of charged groups of carboxylate, phosphate, and guanidine. More precisely, we modelled form factors of the resonance groups as a linear combination of the non-resonance form factors. Given the analytic form of the atomic form factors for unified atomic groups (eq. 200), we computed their five-gaussian approximation, which were tabulated for a later use.

5.2.10 *Form factors for dummy atoms*

Following Fraser et al. [74], we express the form factors of the dummy atoms through the observed displaced solvent volumes V_i as

$$g_i(q) = V_i \exp\left(-\pi q^2 V_i^{2/3}\right). \tag{201}$$

We should specifically add that this is a very crude approximation with some parameters generally valid only for globular proteins (see, e.g., the discussion in Chatzimagas and

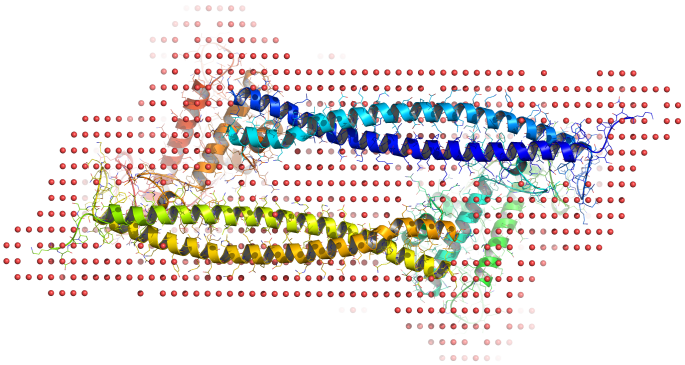


Figure 31: A schematic grid representation of the hydration shell with the resolution of 4 Å for the SASDAW3 model from the SASBDB database. Red dots represent the positions of the sampled points in the hydration shell. The effective width of the shell in this case is 5 Å.

Hub [41]). Following Svergun [243], we introduce the effective atomic radius r_0, an adjustable parameter that scales the observed displaced solvent volumes according to

$$V_i(r_0) = \frac{4}{3}\pi r_i^3 \frac{r_0^3}{r_m^3}, \tag{202}$$

where r_i are the tabulated actual values of atomic group radii, and r_m are the actual average radii of atomic groups. Changing the adjustable parameter to $\delta r \equiv r_0 - r_m$, we can expand the previous expression to the first order in δr using the Maclaurin series as

$$g_i(q, \delta r) = V_i \exp\left(-\pi q^2 V_i^{2/3}\right)\left[1 + \frac{\delta r}{r_m}\left(3 - 2\pi q^2 V_i^{2/3}\right)\right] + O\left(\delta r^2\right). \tag{203}$$

This equation can be further simplified to the form of expressions (12-13) from [243] as

$$g_i(q, \delta r) = V_i \exp\left(-\pi q^2 V_i^{2/3}\right)\left[1 + \frac{\delta r}{r_m}\left(3 - (4\pi/3)^{2/3} 2\pi q^2 r_m^2\right)\right] + O\left(\delta r^2\right), \tag{204}$$

with the term independent of δr being the *reference* dummy atoms form factor $g_i(q)$, and the term in the square brackets being the adjustable overall expansion factor $G(q, \delta r)$. We can even simplify the second term further dropping the q dependency. Using the last expression, the excluded volume amplitudes $C_{lm}(q, \delta r)$ can be adjusted through the reference values $C_{lm}(q)$ as

$$C_{lm}(q, \delta r) = C_{lm}(q) G(q, \delta r), \tag{205}$$

where the reference amplitudes $C_{lm}(q)$ are computed only once using the reference dummy atoms form factors $g_i(q)$. To compute excluded volumes and radii of the unified atomic groups, we used parameters provided in Svergun et al. [243].

5.2.11 *Hydration shell*

To compute the scattering contribution of the molecule's hydration shell (eq. 175), we first constructed its grid approximation using the linked-cell approach [14]. Figure 31 shows an example of our hydration shell model. More precisely, we constructed a grid with the cell size of 3 to 4 Å padded by at least 12 Å in each direction and associated each atom of the molecule with a cell in the grid. Then, we removed those grid cells, whose centers are closer to any atom within the corresponding cell and its 26 direct neighbours than 3 Å or further to all of these atoms than 3 Å plus the width of the shell. Finally, we used the centers of the remaining cells as the grid approximation of the

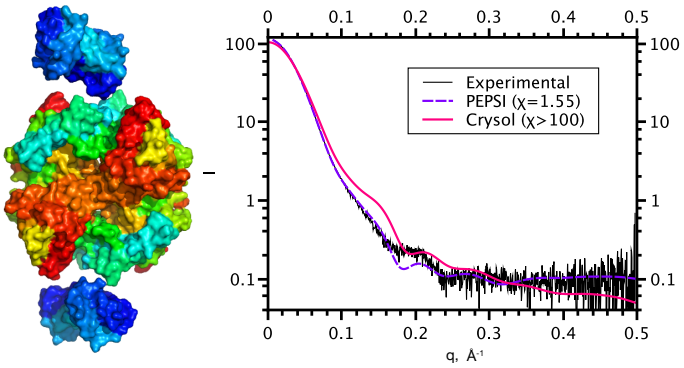


Figure 32: A comparison of scattering profiles between Pepsi-SAXS and CRYSOL-2 for a molecule with a very complex shape.

hydration shell. For the width of the shell we adopted the value of 3 Å for molecules with the radius of gyration smaller than 15 Å, the value of 5 Å for molecules with the radius of gyration larger than 20 Å, and used a linear interpolation in between. The width value of 5 Å is somewhat larger compared to what is usually assumed to be the width of the hydration shell. However, our numerical experiments demonstrated the least overfitting of the experimental data with this value. We should mention that the actual effect of the hydration shell depends not only on its width, but also on its contrast. Thus, the critical parameter that defines the potential overfitting is the product of the width of the hydration shell with its maximum contrast. In our case, this parameter equals to $0.167\ e/\text{Å}^2$, which is smaller than, for example, $0.180\ e/\text{Å}^2$ used in CRYSOL. We have also experimented with a lower resolution of the grid representation by decreasing the linear density of grid points by a factor of two. This did not demonstrate any significant change in the quality of fitting of the modelled profiles into experimental data, however, the execution time improved on average by about 10%, thus we optionally provide this possibility for the user with the '-fast' flag. Figure 32 shows a comparison in scattering profiles for a multi-domain protein with a very complex shapes computed with Pepsi-SAXS and CRYSOL-2, which uses a simplistic manifold representation of the solvation surface. We can see a drastic difference in the obtained results.

5.2.12 *Adaptivity*

We adapt the maximum expansion order L of the multipole expansion according to the *radius-of-gyration of the hydration shell* R_g and the maximum scattering vector of the experimental curve q_{max}. More precisely, we can estimate the value of L from the Nyquist–Shannon–Kotelnikov sampling theorem [157], which defines the angular resolution of encoding with complex spherical harmonics of order L to be $2\pi/L$. On the other hand, the spatial resolution of experimental data is $R = 2\pi/q_{max}$, thus we can relate the two resolutions using the radius-of-gyration R_g as

$$L = 2\pi\frac{R_g}{R} = R_g q_{max}. \tag{206}$$

This expression provides the default value of the maximum expansion order for our method. We use the same idea to approximate the radial functions in eqs. 174-176. There, the radial basis set is given by the spherical Bessel functions of maximum order L. Therefore, we sample expansion coefficients $A_{lm}(q)$, $B_{lm}(q)$, and $C_{lm}(q)$ in $2L$ equidistant points and after use the cubic spline interpolation [208] to reconstruct the values of the expansion coefficients at any point q.

5.2.13 *Fitting*

If the experimental curve $I_{exp}(q)$ is provided, we adjust two parameters δr and $\delta\rho$ such that the goodness of fit χ^2 is minimized,

$$\chi^2 = \frac{1}{N-1}\sum_j \left(\frac{I_{exp}(q_j) - cI_{theor}(q_j)}{\sigma(q_j)}\right)^2, \tag{207}$$

where N is the number of points in the experimental curve, $\sigma(q)$ are the experimental errors, $I_{theor}(q)$ is the theoretical intensity calculated according to eq. 173, and c is the scaling factor given as [243]

$$c = \left(\sum_j \frac{I_{exp}(q_j)I_{theor}(q_j)}{\sigma(q_j)^2}\right) / \left(\sum_j \frac{I_{theor}(q_j)^2}{\sigma(q_j)^2}\right). \tag{208}$$

If the errors are not provided, we model them as $\sigma(q) = 0.01 \times I_{exp}(q)$. To speed-up the calculations of the theoretical scattering intensity curve at different values of δr and $\delta\rho$, we re-write it as a sum of partial intensities, as it shown in Supporting Information. This allows us to reduce the computational cost of the theoretical scattering intensity curve by a factor of $O(L^2)$. We assume the bulk scattering density ρ to be constant and equal to 334 e/nm^3. We then exhaustively search for the optimal values of δr and $\delta\rho$ parameters on a grid of size 100×100. The values of δr are searched in the range of $-0.05 \leq \delta r/r_m \leq 0.05$. This effectively means $0.95r_m \leq r_0 \leq 1.05r_m$, with the mean r_m value over our dataset of 1.64 Å. The range of values of $\delta\rho$ is $0\ e/nm^3 \leq \delta\rho \leq 33.4\ e/nm^3$. We should note that upon request from the user we allow the contrast of the hydration shell $\delta\rho$ to be slightly negative up to $-15\ e/nm^3$. Indeed, as it has been demonstrated by X-ray diffraction, neutron and more recently X-ray reflectivity studies of water-hydrophobic interfaces, there is an unambiguous and distinguishable density-depleted interfacial region near hydrophobic interfaces [40, 106, 165, 259]. At these interfaces, water density drops below the bulk values. There is, however, a certain controversy about the width and the density of this depletion region [259]. We should admit that protein surfaces are never fully hydrophobic. Nonetheless, we allow negative $\delta\rho$ values upon request from the user. Below, we report the results for the two cases. We should also note that some experimental measurements have a systematic error in the determination of the intensity values. To account for this error, we can optionally introduce the offset constant κ and re-write the goodness of fit as it is shown below.

5.2.14 *Fitting with a constant*

Some experimental measurements have a systematic error in the determination of the intensity values. To account for this error, we introduce the offset constant κ and re-write the goodness of fit χ^2 as

$$\chi^2 = \frac{1}{N-1}\sum_j \left(\frac{I_{exp}(q_j) + \kappa - c(\kappa)I_{theor}(q_j)}{\sigma(q_j)}\right)^2, \tag{209}$$

where the scaling factor $c(\kappa)$ is

$$c(\kappa) = c + \kappa b, \tag{210}$$

with the constant b given as

$$b = \frac{\sum_j I_{theor}(q_j)/\sigma(q_j)^2}{\sum_j I_{theor}(q_j)^2/\sigma(q_j)^2}. \tag{211}$$

Now we can compute the optimal offset constant κ for each sampled value of δr and $\delta\rho$ of the theoretical intensity curve $I_{theor}(q)$ by analytically minimizing the least-square discrepancy χ^2 as follows,

$$\kappa = -\frac{\sum_j \left(I_{exp}(q_j) - cI_{theor}(q_j)\right)\left(1 - bI_{theor}(q_j)\right)/\sigma(q_j)^2}{\sum_j (1 - bI_{theor}(q_j))^2/\sigma(q_j)^2} = -\frac{\sum_j \left(I_{exp}(q_j) - cI_{theor}(q_j)\right)/\sigma(q_j)^2}{\sum_j (1 - bI_{theor}(q_j))^2/\sigma(q_j)^2}. \tag{212}$$

5.2.15 *Flexible fitting to experimental profiles*

We have extended the Pepsi-SAXS method for the flexible optimization of the initial molecular shapes along the lowest nonlinear normal modes, as described in the previous Chapter [84, 94, 130]. By default, we precompute the ten lowest modes, sample each of them sequentially in 10 points, such that the maximum deformation is about 5Å, and choose the conformation with the minimum value of χ^2. We also apply a simple optimization of the local geometry, such that bond lengths and angle values stay around the initial positions. The optimization is rather rapid, the whole pipeline takes a few minutes for a single-chain protein with multiple domains. The method has been successfully tested in CASP12 and CASP13 blind challenges and our team produced top-ranked models in the data-assisted sub-challenges [101, 249].

5.2.16 *Benchmarks*

We tested our methods using two benchmarks constructed from the structural models with the corresponding experimental SAXS profiles. We collected the experimental data from two large databases dedicated to the study of biological molecules by SAXS experiments. The first database is BioIsis, which was designed by Dr. Robert P. Rambo at the Lawrence Berkeley National Lab [100]. It contained 99 SAXS scattering profiles of biological molecules and their complexes with both known and unknown structure. The first entry in BioIsis is dated by 2009. The second database is the Small Angle Scattering Biological Data Bank (SASBDB), powered by the Biological Small Angle Scattering Group at European Molecular Biology Laboratory, Hamburg Outstation [261]. This database contained 125 scattering profiles. The first data for SASBDB were collected in 1998. For our tests we collected all those experimental scattering profiles from the two databases that had the corresponding atomic models. Overall, we use 28 entries from BioIsis and 23 entries from SASBDB. Models from BioIsis range from 424 to 23,149 atoms, having on average 6,676 atoms. Models from SASBDB range from 602 to 25,761 atoms, having on average 6,443 atoms.

5.2.17 *Implementation Details*

The presented method is implemented using the C++ programming language and compiled with the gcc-4.8 compiler on Linux, the clang compiler on Mac OS, and the MSVC compiler on Windows systems. To speed up computations of the expansion coefficients in eqs. 174-176, we use single-instruction-multiple-data (SIMD) instructions when possible. We also use multi-threaded computations for the evaluation of the expansion coefficients, as well as for the fitting procedure, if multiple CPU cores are available.

The test benchmarks were run on a MacBook Pro Mid 2015 laptop with a 2.8 GHz Intel Core i7 processor and 16 GB 1600 MHz DDR3 RAM. Pepsi-SAXS can optionally provide the output formatted using JSON, and change the initially guessed angular units of the experimental profile. On demand from the user, we allow for the negative contrast of

```
=======================Reading PDB file=======================
Read PDB file..................................................... : I52-33.pdb
Number of chains read............................................. : 2
Number of atoms read.............................................. : 162120
====================Analysing input PDB data====================
Radius of gyration................................................ : 106.919 A
Maximum extention from the center................................. : 143.247 A
Number of residues................................................ : 346
Molecular weight.................................................. : 38.2875 kDa
==================Reading experimental curve====================
Experimental curve read from...................................... : I52_33.dat
Data smearing..................................................... : off
Number of experimental points..................................... : 380
Minimum angle in the data file.................................... : 0.007855
Maximum angle in the data file.................................... : 0.199909
Estimated resolution.............................................. : 31.4302 A
==================Adaptive memory allocation====================
Estimated minimal expansion order of spherical harmonics......... : 23
================Fitting to the experimental curve================
Best d_rho found.................................................. : 0.006012 e/
A^3
Best r0 found..................................................... : 1.66752 e/A^3
Chi............................................................... : 11.6785
Chi^2............................................................. : 136.387
Displaced Volume.................................................. : 3.14874e+06
A^3
========================== Timing : ==========================
Reading PDB file.................................................. : 0.131131 s
Computing the hydration layer..................................... : 0.120579 s
Computing amplitudes for the hydration shell...................... : 0.379462 s
Computing atomic and excluded volume amplitudes................... : 2.82286 s
Fitting to the experimental curve................................. : 0.010125 s
.....................................................................
Total time : ..................................................... : 3.47689 s
=============================================================
```

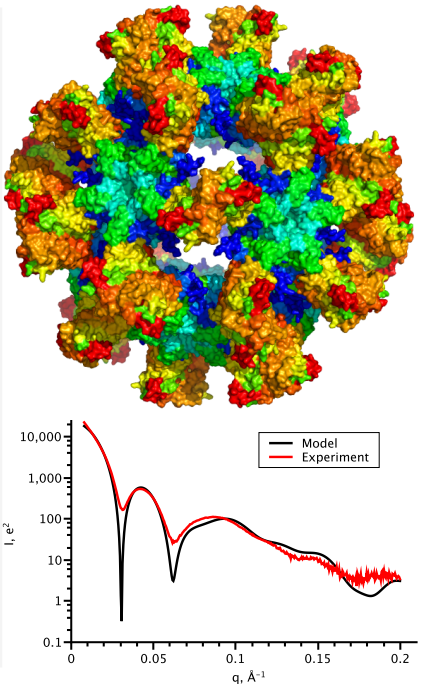


Figure 33: Pepsi-SAXS output from a nanocage particle. **Left panel**: Some of the standard output information. **Right panel**: An illustration of the nanocage PDB file and the corresponding scattering profiles.

the hydration shell using the '-neg' flag. We also provide a coarser representation of the hydration shell with the '-fast' flag, which also improves the execution time by about 10%. By default, the maximum scattering angle is set to 0.5 $Å^{-1}$. The user can change it using the '-ms' flag. Finally, the user can optionally require fitting experimental profile with a constant background noise using the '-cst' flag. Figure 33 shows a typical output and a scattering curve of the method applied to a nanocage composed of 162,120 heavy atoms [20].

5.3 RESULTS AND DISCUSSION

To demonstrate speed and accuracy of the present method, we conducted seven numerical experiments using excessive experimental data. In the experiments, we compared the performance of Pepsi-SAXS with three widely used methods, CRYSOL version 2.8.2 [192, 243], FoXS [224] and SAStbx [147]. We should note that SAStbx provides implementations of three different methods, but we have specifically chosen the novel 3D-Zernike technique, with the 'data_reduct' and 'solvent_scale' options set to 'true'. We did not use more computational methods for the comparison because a recent study of the FoXS method [226] demonstrated an advantage in speed and accuracy of FoXS and CRYSOL over other tested programs.

5.3.1 *BioIsis database*

In the first series of tests, we aimed to compare the four methods on the data from the BioIsis database. More precisely, we measured the goodness of fit for the modelled intensities to the experimental SAXS profiles (eq. 207) and the corresponding timings. Table 7 lists the results of the tests. Pepsi-SAXS outperforms the other methods in running time for all the test profiles. On average, Pepsi-SAXS is about 7 times faster compared to CRYSOL, and 29 and 36 times faster compared to FoXS and SAStbx, correspondingly. As can be expected, for small molecules the difference in running time between Pepsi-

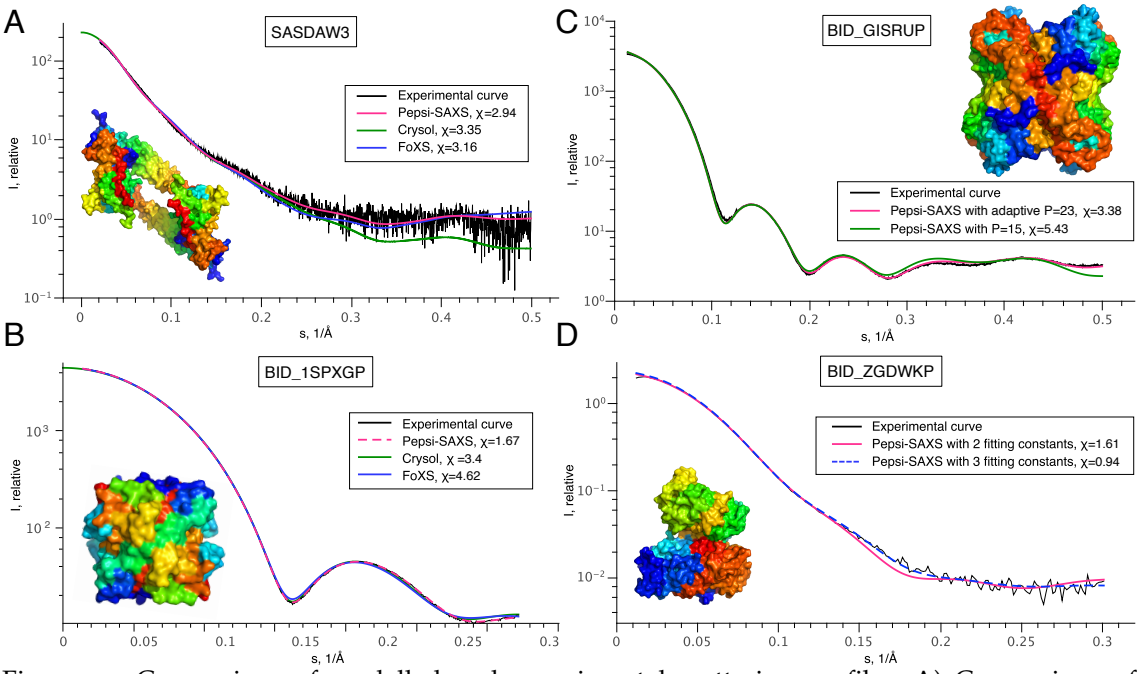


Figure 34: Comparison of modelled and experimental scattering profiles. **A**) Comparison of Pepsi-SAXS, CRYSOL, and FoXS on the SASDAW3 profile from the SASBDB database. **B**) Comparison of Pepsi-SAXS, CRYSOL, and FoXS on the BID_1SPXGP profile from the BioIsis database. **C**) Effect of the adaptive expansion order on the model quality of Pepsi-SAXS applied to the BID_GISRUP profile from the BioIsis database. **D**) Comparison of Pepsi-SAXS modelled scattering profiles without the subtraction (two fitting constants) and with the subtraction (three fitting constants) of the constant systematic error from the experimental data calculated for the BID_ZGDWKP model from the BioIsis database.

SAXS and CRYSOL becomes larger, and the difference between Pepsi-SAXS and FoXS becomes smaller. For large molecules, however, FoXS is significantly slower compared to CRYSOL and Pepsi-SAXS. For example, Pepsi-SAXS computes the scattering profile for model BID_MnmGEP about 62 times faster compared to FoXS. We should mention that the reported speed-up critically depends on the number of available CPU cores. Thus, we did an additional artificial test and executed Pepsi-SAXS on a single CPU core. The average running time over the BioIsis dataset in this case was 0.36 seconds, still several times smaller compared to CRYSOL and other methods.

Regarding the accuracy of the modelled profiles, on average Pepsi-SAXS produces scattering curves very similar to the ones computed by CRYSOL and FoXS with approximately the same χ values, if these are computed for the same range of the scattering angles. We should specifically mention that in all the tests we have restricted the maximum scattering angles of Pepsi-SAXS to the default value of 0.5 $Å^{-1}$. This was done for the rigorous comparison of the χ values with the results of CRYSOL and FoXS. We should also mention that in some cases of noisy experimental data FoXS restricts the maximum scattering angles even further, thus producing on average higher χ values. This, however, does not necessarily mean that the quality of the computed profiles is worse compared to the results of Pepsi-SAXS and CRYSOL. SAStbx generally provides a significantly worse quality of fit and is thus excluded from the detailed comparison. Figures 34B-D show three examples of modelled scattering profiles from the BioIsis database. Generally, we can conclude that Pepsi-SAXS computes scattering profiles that are comparable to the other two methods. Below, we will also study the effect of the adaptive resolution in comparison with CRYSOL in more detail.

We should also mention that a smaller χ value achieved with a certain method for a scattering profile does not necessarily mean a better quality of the computed profile. Generally, one should be concerned about possible flexibility and conformational hetero-

geneity of the modelled proteins. Also, some of the models from the two benchmarks are not crystallographic structures but were produced with molecular dynamics simulations or MODELLER [269], for example. Therefore, small values of χ for some of the models would rather indicate a potential overfitting of experimental profiles than be a demonstration of the superiority of the fitting method. Finally, we should add that different methods use different ranges of fitting parameters, and also different models for the hydration shell, which, consequently, contribute differently to a potential overfitting of experimental data.

Ideally, a reference dataset of native structures supplemented with experimental SAXS profiles along with non-native decoys should be established for the evaluation of SAXS algorithms. Then, different methods can be tested on this dataset by scoring the non-native decoys. The absence of overfitting in a SAXS method can be confirmed, for example, if the native structures will have the least χ values among all the scored decoys. To support our method, we should say that we use a small range of adjustable parameters as compared to other methods such as CRYSOL and FoXS. Thus, we believe that Pepsi-SAXS does not have any significant overfitting of experimental data.

			CRYSOL		FoXS		SAStbx		Pepsi-SAXS	
	# structures	N. of atoms	χ	Time, s	χ	Time, s	χ	Time, s	χ	Time, s
Average	28	6678	2.79	0.81	2.73	3.51	4.92	4.38	2.55	0.12
Average with constant background	28	6678	2.55	0.86	2.98	3.5	–	–	2.34	0.12

Table 7: Comparison of four methods, CRYSOL, FoXS, SAStbx (using the 3D-Zernike technique and data reduction option), and Pepsi-SAXS, when fitting modelled intensity profiles to experimental data collected from the BioIsis database. For each method, we provide the value of χ and the running time measured in seconds for each of the scattering profiles. We also list the number of atoms in the models along with the average values of χ and running time.

5.3.2 *BioIsis database with a systematic error*

In the second series of tests, we compared the three methods, excluding SAStbx, on the same data from BioIsis, but this time measuring the goodness of fit for the modelled intensities to the experimental SAXS profiles and the corresponding timings for data with a constant systematic error (see eq. 209). Table 7 lists the detailed results of the tests. For all the three methods, the running time becomes larger only marginally. Regarding the accuracy of the models, both Pepsi-SAXS and CRYSOL on average improve the value of χ by about 9%, and FoXS unexpectedly worsens the averaged value of χ. Figure 34D shows an example of fitting for two profiles calculated by Pepsi-SAXS with and without the constant systematic error into the experimental curve. We can see a drastic improvement of the model when subtracting the constant noise from experimental data.

5.3.3 *SASBDB database*

In the third series of tests, we compared the four methods on the data from SASBDB. Here, we again first measured the goodness of fit for the modelled intensities to the experimental SAXS profiles (eq. 207) and the corresponding timings. Table 8 lists the detailed results of the tests. Similarly to the previous tests, Pepsi-SAXS significantly outperforms the other methods in running time. Here, on average, Pepsi-SAXS is about 5 times faster compared to CRYSOL, and 21 and 25 times faster compared to FoXS and SAStbx, correspondingly. The speed-up in the running time of Pepsi-SAXS compared to

the other methods is somewhat smaller compared to the previous tests due to on average higher expansion orders used here. More precisely, for SASBDB, Pepsi-SAXS uses the average expansion order of 19 and for BioIsis it uses the order of 14. Regarding the accuracy of the modelled profiles, on average Pepsi-SAXS, CRYSOL and FoXS achieve the same values of χ if these are computed using the same range of scattering angles. SAStbx was not able to process a half of the scattering profiles. Anyway, it was again the slowest method among the four. Figure 34A shows an example of modelled scattering profiles from this database. The model (SASDAW3) has a complex shape, thus, we expected the quality of the CRYSOL's modelled profile to be lower compared to profiles built with FoXS and Pepsi-SAXS.

			CRYSOL		FoXS		SAStbx		Pepsi-SAXS	
	# of structures	N. of atoms	χ	Time, s	χ	Time, s	χ	Time, s	χ	Time, s
Average	23	6443	2.32	0.81	2.71	3.56	2.26	4.17	2.33	0.17
Average with constant background	23	6443	2.19	0.89	2.93	3.55	–	–	2.16	0.19

Table 8: Comparison of four methods, CRYSOL, FoXS, SAStbx (using the 3D-Zernike technique and data reduction option), and Pepsi-SAXS, when fitting modelled intensity profiles to experimental data collected from the SASBDB database. For each method, we provide the value of χ and the running time measured in seconds for each of the scattering profiles. We also list the number of atoms in the models along with the average values of χ and running time. SAStbx failed for some of the profiles, the corresponding values of χ and time are marked with a dash.

5.3.4 *SASBDB database with a systematic error*

In the forth series of tests, we again compared the three methods, excluding SAStbx, on the data from SASBDB and measured the goodness of fit with a constant systematic error (see eq. eq:chi2k) and the corresponding timings. Table 8 lists the detailed results of the tests. As before, the running time becomes larger only marginally. Regarding the accuracy of the models, Pepsi-SAXS on average improves the value of χ by 8%, CRYSOL improves it by 6%, and, FoXS again shows no improvement of fit.

5.3.5 *Running times*

For the fifth test we decided to compare the running time of the four methods if a user computes a scattering profile without fitting it into the experimental data. Here, we considered two scenarios, a profile with 51 points, as it used to be the default option in CRYSOL, and a profile with 512 points, which better corresponds to the modern experimental measurements. Table 9 lists the timings for all the four methods run on atomic models from the BioIsis and SASBDB databases. For the 51 points-profiles, Pepsi-SAXS is on average about 3 times faster compared to CRYSOL, and 19 and 27 times faster compared to FoXS and SAStbx, correspondingly. With the 512 points-profile, Pepsi-SAXS, FoXs and SAStbx increase the timings only marginally. However, the running time of CRYSOL depends linearly on the number of points in the scattering profile. Therefore, its timing increases by about ten times.

			Time for 512 points, s / Time for 51 points, s			
	# of structures	N. of atoms	CRYSOL	FoXS	SAXStbx	Pepsi-SAXS

Average	51	6572.0	3.59 / 0.45	3.51 / 3.25	4.63 / 4.6	0.18 / 0.17

Table 9: Comparison of four methods, CRYSOL, FoXS, SAStbx (using the 3D-Zernike technique), and Pepsi-SAXS, when calculating intensity profiles for models collected from the BioIsis and SASBDB databases. No fitting to experimental data is performed. For each method, we provide two running times measured in seconds when calculating the intensity profile with 512 points and with 51 points, correspondingly. We also list the number of atoms in the models along with the average values of running times. SAStbx failed for some of the profiles, the corresponding values of timmings are marked with dashes.

5.3.6 *Adaptive choice of the multipole expansion order*

In the sixth series of tests, we compared the effect of the adaptive choice of the multipole expansion order using data from the BioIsis and SASBDB databases. To do so, we first fixed the expansion order to the value of 15, which is used by default in CRYSOL, and ran Pepsi-SAXS in comparison with CRYSOL. Then, we chose the value of the expansion order adaptively according to eq. 206 and ran the two programs again. Table 10 lists the details of the comparisons. As we can see from this table, using the default expansion order of 15, Pepsi-SAXS demonstrates very similar quality of models compared to CRYSOL, with a slightly smaller value of χ. Adaptive resolution lowers the value of χ for the two methods, by about 1% for CRYSOL and about 2% for Pepsi-SAXS. We attribute the more pronounced effect of the adaptive resolution in Pepsi-SAXS to the different model of the hydration shell in our method. Figure 34C shows an example of scattering profiles plotted at a different expansion order in comparison with the experimental curve for a large molecule (BID_GISRUP). We can see a pronounced difference between the curves at large values of q, which corresponds to a fine resolution in the real space that is not well encoded using low multipole expansion orders.

Finally, in the sevens series of tests, we compared the values of two adjustable parameters r_0 and $\delta\rho$ for the three methods, excluding SAStbx, on data from the BioIsis and SASBDB databases. In case of FoXS, we computed the values of r_0 and $\delta\rho$ by rescaling its internal fitting parameters c_1 and c_2 as suggested by the authors [224]. Table 11 lists the adjustable parameters along with the mean values and the standard deviations for the three methods. We can see that all the methods agree on the average value of the effective atomic radius r_0 of 1.64 Å. However, the standard deviation of this parameter in FoXS and Pepsi-SAXS is only 0.05 Å, which constitutes 3% of the average value and is several times smaller compared to the standard deviation of 0.18 Å in CRYSOL. We should note that if we double the width of the search window for the r_0 parameter to make it more comparable with the CRYSOL settings, the quality of fit to experimental data improves only marginally.

Regarding the second adjustable parameter, the contrast of the hydration shell $\delta\rho$, all the methods provide different mean values. More precisely, CRYSOL allows variation of $\delta\rho$ between 0 and 60 e/nm^3, with the average of 22.4 ± 21.7 e/nm^3. FoXS allows negative values of $\delta\rho$ in the range of -27 $e/nm^3 \leq \delta\rho \leq 54$ e/nm^3. Thus, its average $\delta\rho$ is lower compared to the one computed by CRYSOL and equals to 16.6 ± 22.2 e/nm^3. In our model, by default, we allow only positive values of $\delta\rho$ up to one tenth of the bulk density value of 33.4 e/nm^3. As a results, our mean value of the contrast of the hydration shell $\delta\rho$ lies in between those computed by CRYSOL and FoXS, however, with a significantly lower standard deviation, $\delta\rho = 18.4 \pm 11.2$ e/nm^3. Allowing for a negative contrast of the hydration shell or for a larger width of the search window in the $\delta\rho$ parameter provides slightly better fits with the experimental profiles. However, this choice of adjustable parameters might overfit the actual experimental data.

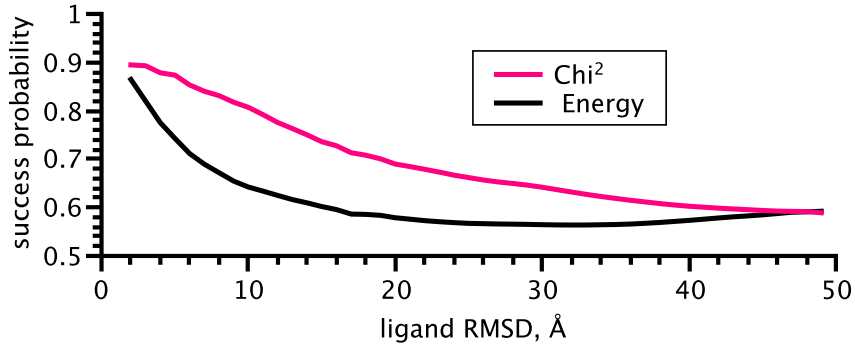


Figure 35: Probability for a random docking decoy from the Protein-Protein Benchmark v4 with RMSD $<$ cutoff to have a better score than a random decoy with RMSD $>$ cutoff.

	# of structures	Number of atoms	Expansion order=15				Adaptive expansion order				
			CRYSOL		Pepsi-SAXS			CRYSOL		Pepsi-SAXS	
			χ	Time, s	χ	Time, s	Order	χ	Time, s	χ	Time, s
Average	51	6572.0	2.58	0.81	2.5	0.11	16.18	2.55	0.94	2.45	0.14

Table 10: Comparison of CRYSOL with Pepsi-SAXS when using adaptive multipole expansion orders. Experimental data is collected from the BioIsis and SASBDB databases. For each method, we provide the value of χ and the running time measured in seconds when using the default expansion order of 15 and the adaptive expansion order. We also list the number of atoms in the models, the order of the adaptive multipole expansion, along with the average values of χ and running time.

	# of structures	Fitting Parameters								
		CRYSOL			FoXS			Pepsi-SAXS		
		χ	r_0, Å	$\delta\rho$, e/nm	χ	r_0, Å	$\delta\rho$, e/nm	χ	r_0, Å	$\delta\rho$, e/nm
Average	51	2.58	1.64 $\pm$0.18	22.4 $\pm$21.7	2.72	1.64 $\pm$0.05	16.6 $\pm$22.2	2.45	1.64 $\pm$0.05	18.4 $\pm$11.2

Table 11: Comparison of three methods, CRYSOL, FoXS, and Pepsi-SAXS, when fitting modelled intensity profiles to experimental data for models collected from the BioIsis and SASBDB databases. For each method, we provide the value of χ, the fitted value of r_0 parameter, and the fitted value of the contrast of the hydration shell parameter $\delta\rho$. We also list the average values of χ along with the average values and the standard deviations of the fitting parameters.

5.3.7 *Applications to the rigid-body docking*

To test the rigid-body formalism, we used the Protein-Protein Benchmark v4 with 176 protein-protein complexes in both bound and unbound states [103]. We simulated scattering profiles using the bound states of the complexes, and used rigid-body docking predictions from the unbound states. The docking predictions were generated with the Zdock software. We used precomputed docking decoys with 6 degree sampling, corresponding to 54,000 conformations per protein complex. We then compared scoring the docking poses with the Zdock score and χ^2. The whole experiment took about 2 hours on a MacBook Pro Mid 2015 laptop with a 2.8 GHz Intel Core i7 processor and 16 GB 1600 MHz DDR3 RAM. Figure 35 shows the average results of the experiments. This is the probability for a random docking decoy with RMSD $<$ cutoff having a better score than a random decoy with RMSD $>$ cutoff. One can see that scoring by χ^2 significantly increases the probability to pick a near-native decoy structure.

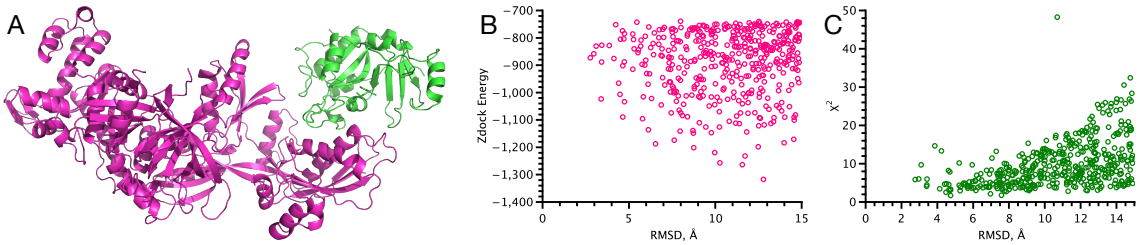


Figure 36: **A**: Bound conformation of the eEF2-ETA-bTAD complex, PDB code 1ZM4. **B**: Zdock scores of the unbound docking decoys for this complex. **C**: Goodness of fit of the scattering profiles of the docking decoys compared with the bound-state scattering profile.

Scoring by the protein profiles appears to be especially effective for the complexes with non-trivial shapes. Figure 36A shows one of such examples with the bound state of the eEF2-ETA-bTAD complex, (PDB code 1ZM4). Zdock scores of the docking decoys for this examples have no correlation with the ligand-RMSD to the bound state (see Fig. 36B). However, χ^2 goodness of fit to the bound-state scattering profile has a much better correlation with the ligand-RMSD and can serve as a proxy for the selection of near-native docking decoys (see Fig. 36C).

5.3.8 *Scattering profiles of MD trajectories*

We have extended the Pepsi-SAXS method for the computation of scattering profiles on molecular trajectories. We aimed to achieve two goals. Firstly, we have extended the method to be used in fitting the MD simulations trajectories and the calculation of trajectories profiles. Secondly, we explored the possibility of explaining experimental data with multiple molecular conformations. In our approach, we compute the solvation shell and the maximum expansion order individually for each of the trajectory snapshots. Standard MD trajectory formats are also supported. An example of such a calculation can be found in multiple studies from my collaborators [1–3, 31, 119, 132, 159, 189, 190, 218, 253–255].

5.4 CONCLUSION

We developed a new method called Pepsi-SAXS that calculates small angle X-ray scattering profiles from atomistic models. Our method is based on the multipole expansion scheme and has quite a number of distinct features. Firstly, we use a very fast model for the scattering contribution from the hydration shell based on a uniform grid of points. Secondly, we use the adaptive resolution of the multipole expansion estimated according to the Nyquist–Shannon–Kotelnikov sampling theorem. Then, we introduce partial scattering intensities to rapidly fit the modelled profiles to the experimental data using exhaustive search in two adjustable parameters. Finally, we introduce individual form factors for charges and resonance groups, which increase the quality of the modelled scattering profiles.

Overall, the Pepsi-SAXS method is significantly faster compared to CRYSOL, FoXS and SAStbx (with the 3D-Zernike option) methods with on average the same quality of scattering profiles. Thanks to its speed and modular architecture, it has been already well adapted in the bioinformatics and biophysics community.

Pepsi-SAXS is freely available for the academic community. We have also developed a web-interface at https://pepsi.app.ill.fr, which also provides access to Pepsi-SANS, a method for the computation of small-angle scattering neutron profiles. Finally, we have been working on several novel methodological developments of the method, e.g., the computation of scattering profiles from flexible particles.

6

MACHINE LEARNING

For quite a long time, I have been interested in how classical *convex optimization* and more recent *machine-learning* techniques based on *neural networks* can help solve very challenging sampling problems in statistical physics. This led us to designing several knowledge-based potentials for 3D structures of biological molecules and their complexes and also allowed us to prototype convex relaxations for combinatorial sampling problems [200, 213]. We developed descriptor-based methods [86, 110–114, 117, 176, 205], and also methods based on features learned with deep convolutional neural networks [53, 104, 105, 182, 184, 278]. For example, thanks to the ideas of my student Georgy Derevyanko, we proposed to learn the functional form of the interaction potentials $f^{kl}(r)$ by expanding them along with the geometrical features $n^{kl}(r)$ into orthogonal polynomial bases $\psi_q(r)$, which leads to an easy expression for the interaction energy E,

$$x_q^{kl} = \int_0^{r_{max}} n_{kl}(r)\psi_q(r)\,dr, \quad w_q^{kl} = \int_0^{r_{max}} f_{kl}(r)\psi_q(r)\,dr, \quad E = \sum_{kl}\int_0^{r_{max}} n^{kl}(r)f^{kl}(r)\,dr = \sum_{klq} x_q^{kl} w_q^{kl} \tag{213}$$

We then pushed forward these ideas to other applications with Petr Popov, Georgy Cheremovskiy, Emilie Neveu, and Maria Kadukova. I should also add that thanks to my students Georgy Derevyanko, Guillaume Pagès, Benoit Charmettant, Dmitrii Zhemchuzhnikov, Ilia Igashov, and Nikita Pavlichenko, we were one of the first teams worldwide who started designing *deep-learning* applications for molecular data in 3D.

6.1 CONVEX OPTIMISATION AND POLYNOMIAL EXPANSIONS FOR KNOWLEDGE-BASED POTENTIALS

6.1.1 *Problem Formulation*

Let us consider N native 3D structures of proteins or protein complexes, $\mathbb{P}_i^{nat}$, $i = 1...N$. Let us also assume that for each protein or protein complex number i, we can generate its D non-native structures (decoys), $\mathbb{P}_{ij}^{nonnat}$, $j = 1...D$, where the first index runs over different proteins / complexes and the second index runs over the decoys. Our goal is to find a *scoring functional* F, defined for all possible protein or protein-protein complex structures $\mathbb{P}$, such that for each native structure i and its nonnative decoy j, the following inequality holds:

$$F(\mathbb{P}_i^{nat}) < F(\mathbb{P}_{ij}^{nonnat}) \tag{214}$$

Without loss of generality, from now on we will only consider the problem os scoring protein-protein complexes. This is a very difficult problem in such a general formulation. In order to simplify it, we can assume the following. First, the functional F depends only on the interface between the proteins in a complex. We define the interface as a set of all atom pairs at a distance smaller than a certain cutoff distance r_{max}, such that the first atom in each pair belongs to the first protein and the second atom in each pair belongs to the second protein. Second, the protein is represented as a set of discrete interaction sites located at the centers of the atomic nuclei. All interaction sites are split into M

interaction types according to the properties of the corresponding atomic nuclei. Here we choose $M = 20$. More generally, we may say that all the atoms in all amino acids have different properties, which leads to $M = 167$. Third, the functional F depends only on the distribution of the distances between the interaction sites $F(\mathbb{P}) \equiv F(n(r))$, where $n^{kl}(r)$ is the *number density of site-site pairs* separated by a distance r, with site k located on the first protein, and site l located on the second protein. Finally, we assume that F is a linear functional, $F(\alpha n_1(r) + \beta n_2(r)) = \alpha F(n_1(r)) + \beta F(n_2(r))$. This is a very strong assumption, but we demand it for an efficient optimization scheme. One of the simplest functionals $F(n(r))$ fulfilling these assumptions can be written as:

$$F(n(r)) = \sum_{k=1}^{M} \sum_{l=k}^{M} \int_0^{r_{max}} n^{kl}(r) U^{kl}(r)\, dr. \tag{215}$$

It contains unknown functions $U^{kl}(r)$ that can be determined from the training set of protein complexes. From now on, we will call these functions *scoring potentials*. [1] Once the scoring potentials are known, to compute the value of F, we only need to define site-site number densities $n^{kl}(r)$. In practice, we compute them as a sum of all kl-distances in a given protein complex using the following equation,

$$n^{kl}(r) = \sum_{ij} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r-r_{ij})^2}{2\sigma^2}}, \tag{216}$$

where each distance distribution is represented as a Gaussian centered at r_{ij} with the standard deviation of σ. The sum is taken over all kl-site pairs i and j separated by a distance r_{ij} smaller than r_{max}, with site k located on the first protein of the complex, and site l located on the second protein. In the limiting case of σ tending to zero, eq. 216 turns into a sum over Dirac delta functions. In the present study we assume the value of σ to be fixed for all site-site distributions. However, if one has additional information about individual distance distributions, e.g., Debye-Waller factors, molecular dynamics trajectories, etc., it can be used for more precise parametrization of the standard deviation or even instead of the Gaussian approximation in eq. 216. Finally, we compute the score of each conformation as a sum of pair-wise contributions $\Upsilon^{kl}(r_{ij})$ taken over all pairs of atoms i and j separated by the distance r_{ij} smaller than r_{max},

$$F = \sum_{ij} \Upsilon^{kl}(r_{ij}) \equiv \sum_{ij} \frac{1}{\sqrt{2\pi\sigma^2}} \int_0^{r_{max}} e^{-\frac{(r-r_{ij})^2}{2\sigma^2}} U^{kl}(r)\, dr, \tag{217}$$

with atom i of type k located on the first protein of the complex, and atom j of type l located on the second protein. We will refer to functions $\Upsilon^{kl}(r)$ as to the *scoring functions*.

6.1.2 *Expansion of $U(r)$ and $n(r)$ in an orthogonal basis*

Given a set of functions $\phi_p(r)$ orthogonal on the interval $[r_1; r_2]$ with a nonnegative weight function $\Omega(r)$ such that

$$\int_{r_1}^{r_2} \phi_{p_1}(r) \phi_{p_2}(r) \Omega(r)\, dr = \delta_{p_1 p_2}, \tag{218}$$

1 Though the scoring function 215 is similar by the structure to e.g. the excess internal energy [90], our scoring potentials $U^{kl}(r)$ are not equal to the potential energy functions between sites k and l.

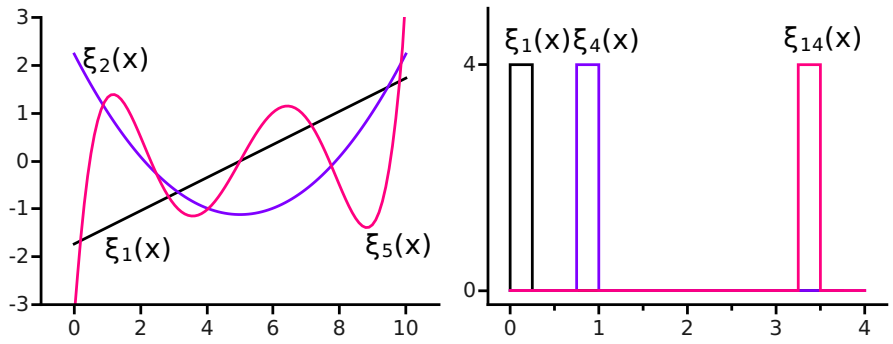


Figure 37: Two types of orthogonal functions. Left: shifted Legendre polynomials orthogonal on the interval $[0; 10]$. Right: shifted rectangular functions.

where $\delta_{p_1 p_2}$ is the Kronecker delta function, scoring potentials $U_{kl}(r)$ and number densities $n_{kl}(r)$ can be expanded on $[r_1; r_2]$ as

$$U_{kl}(r) = \sum_p w_p^{kl} \phi_p(r) \sqrt{\Omega(r)}, \quad r \in [r_1; r_2] \tag{219}$$

$$n_{kl}(r) = \sum_p x_p^{kl} \phi_p(r) \sqrt{\Omega(r)}, \quad r \in [r_1; r_2]. \tag{220}$$

Expansion coefficients w_p^{kl} and x_p^{kl} can be determined from the orthogonality condition 218 as

$$w_p^{kl} = \int_{r_1}^{r_2} U_{kl}(r) \phi_p(r) \sqrt{\Omega(r)}\, dr \tag{221}$$

$$x_p^{kl} = \int_{r_1}^{r_2} n_{kl}(r) \phi_p(r) \sqrt{\Omega(r)}\, dr. \tag{222}$$

Here we use two types of functions $\phi_p(r)$ orthogonal on the interval $[0; 10]$ with a unit weight, (i) shifted Legendre polynomials and (ii) traditionally used shifted rectangular functions. These two types of functions are plotted in Figure 37. Other types of orthogonal functions can also be used. If the functions $\phi_p(r)$ are chosen to be negligibly small outside the interval $[0; r_{max}]$ or if their interval of orthogonality $[r_1; r_2]$ coincides with the interval $[0; r_{max}]$, as is the case for two sets of our functions, then using eqs. 219-220, the scoring functional $F(n(r))$ can be expanded up to the expansion order P as

$$F(n(r)) \approx \sum_{k=1}^{M} \sum_{l=k}^{M} \sum_{p=1}^{P} w_p^{kl} x_p^{kl} = (\mathbf{w} \cdot \mathbf{x}), \quad \mathbf{w}, \mathbf{x} \in \mathbb{R}^{\frac{P \times M \times (M+1)}{2}}. \tag{223}$$

We will refer to the vector $\mathbf{w}$ as to the *scoring vector* and to the vector $\mathbf{x}$ as to the *structure vector*. Formulas 216 and 222 provide the projection from a protein complex structure into the *scoring space* $\mathbb{R}^{P \times M \times (M+1)/2}$. Using these formulas, we can project structural information of each protein complex into a certain structure vector $\mathbf{x}$ on $\mathbb{R}^{P \times M \times (M+1)/2}$.

6.1.3 *Connection to convex optimization*

Now we can reformulate the scoring problem 214 as follows – given N native structure vectors $\mathbf{x}_i^{nat}$ and $N \times D$ nonnative structure vectors $\mathbf{x}_{ij}^{nonnat}$, find a scoring vector $\mathbf{w} \in \mathbb{R}^{P \times M \times (M+1)/2}$ such that:

$$\forall i = 1...N, \forall j = 1...D \quad (\mathbf{x}_i^{nat} \cdot \mathbf{w}) < (\mathbf{x}_{ij}^{nonnat} \cdot \mathbf{w}), \tag{224}$$

or, equivalently,

$$\forall i = 1...N, \forall j = 1...D \quad ([\mathbf{x}_{ij}^{nonnat} - \mathbf{x}_i^{nat}] \cdot \mathbf{w}) > 0), \tag{225}$$

which defines $N \times D$ *half-spaces* in $\mathbb{R}^{P \times M \times (M+1)/2}$ with a common normal $\mathbf{w}$. Thus, *finding the scoring vector is equivalent to finding the common normal* $\mathbf{w}$ to the planes in eq. 225. Geometrical representation of three groups of structure vectors separated by three parallel hyperplanes with the common normal $\mathbf{w}$ is given in Fig. 38.

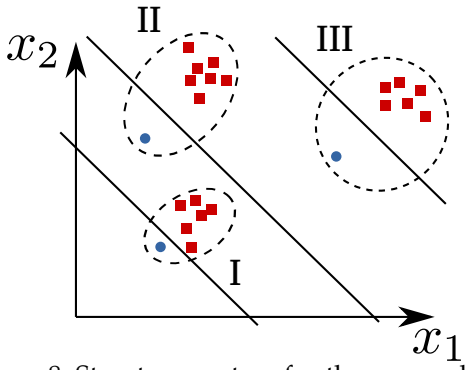


Figure 38: Structure vectors for three complexes are shown. Native structure vectors are plotted as blue circles. Nonnative structure vectors are plotted as red squares. Native structure vectors in each complex are separated from nonnative ones by three hyperplanes with a common normal. This normal is the scoring vector $\mathbf{w}$ we are aiming to find.

In the training set, some decoy structures can be very close to the native structures. In practice, we define the native structure as a structure with ligand root-mean-square deviation (lRMSD) smaller than 2 Å. Therefore, for each complex we may have several native structure vectors along with several nonnative structure vectors. Fig. 39A presents an example of a single complex when infinitely many hyperplanes can separate the two classes of structure vectors. Fig. 39B presents an example when no hyperplane can separate the two classes of structure vectors for a single complex. Similar examples can be constructed for the case with multiple complexes. Given two classes of vectors, Vapnik proposed to use the *optimal separating hyperplane* [263], which is unique and maximizes the distance to the closest point from either class. For the non-separable case, Cortes and Vapnik proposed to relax the condition for the optimal separating hyperplane [49], including an additional term. This term minimizes the sum of penalties for misclassified vectors. Following these ideas, we introduce for each decoy set *slack variables* ξ_{ij}, which are positive for misclassified structure vectors and zero otherwise. A non-zero value of ξ_{ij} allows the structure vector x_{ij} to overcome inequality conditions 225 at a cost proportional to the value of ξ_{ij} (see Fig. 39B). The resulting quadratic optimization problem reads:

$$\begin{aligned} &\text{Minimize } (\text{ in } \mathbf{w}, b_j, \xi_{ij}) : && \tfrac{1}{2}\mathbf{w} \cdot \mathbf{w} + \textstyle\sum_{ij} C_{ij}\xi_{ij} \\ &\text{Subject to: } && y_{ij}\left[\mathbf{w} \cdot \mathbf{x}_{ij} - b_j\right] - 1 + \xi_{ij} \geq 0, \; i = 1...N, \; j = 1...D \\ & && \xi_{ij} \geq 0 \end{aligned} \quad . \tag{226}$$

The solution of this problem provides a trade-off between how large will be the separation between the two classes of structure vectors for each complex and how many misclassified vectors will be in the solution. Parameters C_{ij} can be regarded as *regularization parameters*. Small values of C_{ij} maximize the structure vector separation whereas large values of C_{ij} minimize the number of misclassified structure vectors. We choose parameters C_{ij} to be different for native and nonnative structure vectors of each complex because fewer native structure vectors should have the larger weight (see e.g. [5]). The following observation provides the foundation for the numerical scheme used in this work:

Observation 1. *The optimal scoring vector is unique and given by the solution of problem 226.*

Remark. *Here, the scoring vector is optimal in the sense that it maximizes the separation between native and nonnative structure vectors and minimizes the number of misclassified vectors. Regularization parameters C_{ij} in 226 tune the importance of either factors.*

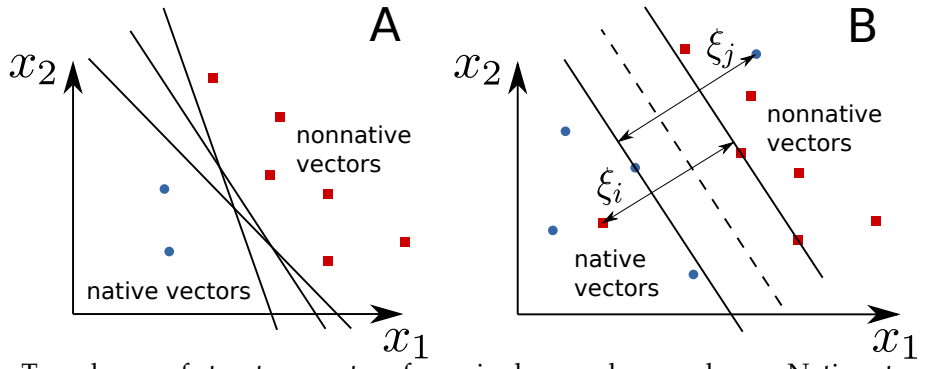


Figure 39: Two classes of structure vectors for a single complex are shown. Native structure vectors are plotted as blue circles. Nonnative structure vectors are plotted as red squares. A) The case when infinitely many hyperplanes can separate the two classes. B) The case when no optimal separating hyperplane exists. Slack variables ξ_i and ξ_j for misclassified structure vectors are added, which are the distances to the corresponding margin hyperplanes. The optimal hyperplane, which maximizes the separation between the two classes, is plotted as a dashed line. Two margin hyperplanes are plotted as solid lines.

6.1.4 *The dual form*

Optimization problem 226 can be solved by the classical method of *Lagrange multipliers* [32, 49]. If we introduce $N \times D$ nonnegative Lagrange multipliers λ_{ij} associated with the first set of inequality constraints from 226 and $N \times D$ nonnegative Lagrange multipliers ν_{ij} associated with the second set of inequality constraints from 226, the solution of problem 226 is equivalent to determining the *saddle point* of the following *Lagrangian* function:

$$\begin{aligned}\mathcal{L} =& \frac{\mathbf{w}\cdot\mathbf{w}}{2} + \sum_{ij} C_{ij}\xi_{ij} - \sum_{ij}\lambda_{ij}\left(y_{ij}\left[\mathbf{w}\cdot\mathbf{x}_{ij} - b_j\right] - 1 + \xi_{ij}\right) \\ & - \sum_{ij}\nu_{ij}\xi_{ij},\end{aligned} \tag{227}$$

with $\mathcal{L} = \mathcal{L}(\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\nu})$, where $\mathbf{b} = (b_1, b_2, \ldots, b_D)$, $\boldsymbol{\xi} = (\xi_{11}, \xi_{12}, \ldots, \xi_{ND})$, $\boldsymbol{\lambda} = (\lambda_{11}, \lambda_{12}, \ldots, \lambda_{ND})$, and $\boldsymbol{\nu} = (\nu_{11}, \nu_{12}, \ldots, \nu_{ND})$. At the saddle point, $\mathcal{L}$ has a minimum with respect to $\mathbf{w}$, $\mathbf{b}$ and $\boldsymbol{\xi}$ and a maximum with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$. According to the classical *Karush-Kuhn-Tucker* (KKT) conditions [32, 129], which is a generalization of the method of Lagrange multipliers to inequality constraints, the saddle point of the Lagrangian function 227 satisfies the four following conditions:

1. Stationarity conditions:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{ij} y_{ij}\lambda_{ij}\mathbf{x}_{ij} = 0 \tag{228}$$

$$\frac{\partial L}{\partial b_j} = \sum_{i} y_{ij}\lambda_{ij} = 0 \tag{229}$$

$$\frac{\partial L}{\partial \xi_{ij}} = C_{ij} - \lambda_{ij} - \nu_{ij} = 0 \tag{230}$$

2. Complementary slackness conditions:

$$\lambda_{ij}\left(y_{ij}\left[\mathbf{w}\cdot\mathbf{x}_{ij} - b_j\right] - 1 + \xi_{ij}\right) = 0 \tag{231}$$

$$\nu_{ij}\xi_{ij} = 0 \tag{232}$$

3. Primal feasibility conditions:

$$y_{ij}\left[\mathbf{w}\cdot\mathbf{x}_{ij} - b_j\right] - 1 + \xi_{ij} \geq 0 \tag{233}$$

$$\xi_{ij} \geq 0 \tag{234}$$

4. Dual feasibility conditions:

$$\lambda_{ij} \geq 0 \tag{235}$$
$$\nu_{ij} \geq 0 \tag{236}$$

Using equation 227 along with the aforementioned KKT conditions 228-236, we can rewrite the original optimization problem 226 as:

$$\begin{aligned} &\text{Maximize } \mathcal{L}(\lambda_{ij}) = \textstyle\sum_{ij} \lambda_{ij} - \frac{1}{2} \sum_{ij} \sum_{kl} y_{ij} y_{kl} \lambda_{ij} \lambda_{kl} \mathbf{x}_{ij} \cdot \mathbf{x}_{kl} \\ &\text{Subject to:} \qquad\qquad 0 \leq \lambda_{ij} \leq C_{ij} \\ &\qquad\qquad\qquad\quad \textstyle\sum_i y_{ij} \lambda_{ij} = 0 \end{aligned} \quad . \tag{237}$$

6.1.5 *The optimization algorithm*

Properties and solutions of quadratic optimization problems similar to the one stated above 226 have been extensively studied in the theory of convex optimization [32, 263]. For instance, using the *Lagrangian formalism*, the optimization problem 226 can be converted into its dual form (see subsection above), and the resulting dual optimization problem is *convex*:

$$\begin{aligned} &\text{Maximize } \mathcal{L}(\lambda_{ij}) = \textstyle\sum_{ij} \lambda_{ij} - \frac{1}{2} \sum_{ij} \sum_{kl} y_{ij} y_{kl} \lambda_{ij} \lambda_{kl} \mathbf{x}_{ij} \cdot \mathbf{x}_{kl} \\ &\text{Subject to:} \qquad\qquad 0 \leq \lambda_{ij} \leq C_{ij}, \quad \forall i, j \\ &\qquad\qquad\qquad\quad \textstyle\sum_i y_{ij} \lambda_{ij} = 0, \quad \forall j \end{aligned} \quad , \tag{238}$$

where the maximization is performed with respect to the *Lagrange multipliers* λ_{ij}. This dual problem is similar to the the soft-margin SVM optimization problem [49]. The difference lies in the constraints. For the soft margin SVM, conditions on the parameters written in the same two-indexed form as in eq. 238, are $\sum_{ij} y_{ij} \lambda_{ij} = 0$. Vectors $\mathbf{x}_{ij}$ for which $\lambda_{ij} > 0$ are called *support vectors*. Once the dual problem 238 is solved and the Lagrange multipliers λ_{ij} are found, we can express the solution of the original primal problem 226 as a linear combination of the support vectors:

$$\mathbf{w} = \sum_{\text{support vectors}} y_{ij} \lambda_{ij} \mathbf{x}_{ij}. \tag{239}$$

The dual representation 238 of the original primal problem 226 allows us to break the original large problem into a series of smaller sub-problems. Due to its enormous size, the problem 238 can not be easily solved by standard techniques. The quadratic form in 238 involves a matrix with number of elements proportional to the squared number of the training structure vectors. This matrix often exceeds the size of available RAM, for instance, explicit storage of the matrix used in the present study requires about 20GB of memory. Nonetheless, algorithms that deal with large datasets are widely used in machine learning. More precisely, various decomposition techniques have been developed to reduce the requirements of solvers to the size of available RAM [136, 181, 199, 262]. Here, we employ the *block-decomposition technique* and propose the *block sequential minimal optimization* (BSMO) algorithm, which is described below.

6.1.6 *The BSMO algorithm*

Here we explain the *block sequential minimal optimization* (BSMO) algorithm. Briefly, we partition the training set into N *blocks*, each comprising $D + 1$ structure vectors, both *native* and *nonnative* ones. Then, for each block i, we iteratively optimize each pair of

Lagrange multipliers (λ_1, λ_2), preserving the equality constraint $y_1\lambda_1 + y_2\lambda_2 = const$. To do this, we write the Lagrangian 238-237 as a function of λ_1 and λ_2:

$$\begin{aligned}\mathcal{L}(\lambda_1, \lambda_2) = &\frac{1}{2}\eta\lambda_2^2 - \eta\lambda_2\lambda_2^{old} + \lambda_2 y_2(y_2 - y_1) \\ &+ \lambda_2 y_2(\mathbf{x}_{i1} - \mathbf{x}_{i2}) \cdot \mathbf{w}^{old} + \text{Const.},\end{aligned} \tag{240}$$

with

$$\eta = 2\mathbf{x}_{i1} \cdot \mathbf{x}_{i2} - \mathbf{x}_{i1} \cdot \mathbf{x}_{i1} - \mathbf{x}_{i2} \cdot \mathbf{x}_{i2}. \tag{241}$$

Then, we analytically maximize this Lagrangian with respect to λ_1 and λ_2 according to the *sequential minimal optimization* (SMO) algorithm [199]. We provide details about the SMO algorithm in the next subsection. After the minimization, we obtain new values of λ_1 and λ_2. After each iteration, we recompute the current scoring vector $\mathbf{w}^{new}$ (see equation 239) according to:

$$\mathbf{w}^{new} = \mathbf{w}^{old} + \Delta\lambda_1 y_1 \mathbf{x}_{i1} + \Delta\lambda_2 y_2 \mathbf{x}_{i2}. \tag{242}$$

For each block i, we continue the iterative optimization of the Lagrangian 240 until the relative change in its value between two successive inner cycles of iterations is less than the desired tolerance. Each inner cycle consists in the optimization of all pairs of Lagrange multipliers for a given block i. Globally, we terminate the optimization when the relative change in the value of the Lagrangian 238,237 between two successive outer cycles is less than the desired tolerance. Each outer cycle consists in the optimization of all the blocks of the training set.

As it is seen from equation 241, our BSMO algorithm requires only scalar products of the structure vectors within the *same* block. Therefore, it is sufficient to load each block into RAM sequentially, which results in memory efficiency of our method. Precisely, RAM required for our implementation of the block-decomposition solver is N^2 times less compared to the standard quadratic programming solvers.

6.1.7 *The SMO algorithm*

Here we describe how the SMO algorithm [199] solves the problem 238,237 for two Lagrange multipliers λ_1 and λ_2. All quantities that refer to the first multiplier have a subscript 1 and all quantities that refer to the second multiplier have a subscript 2. SMO first computes the constraints on these multipliers and then solves the problem 238,237 for the constrained maximum. The inequality constraints in 238,237 force the two multipliers to lie within a box $[0,\ C_1] \times [0,\ C_2]$, while the equality constraints in 238,237 force the two multipliers to lie on a diagonal line segment:

$$y_1\lambda_1 + y_2\lambda_2 = \gamma. \tag{243}$$

This equation explains why one needs to optimize the two Lagrange multipliers simultaneously. Precisely, it is not possible to optimize a single multiplier without breaking the equality constraints in 237-238, and, subsequently, breaking the constraints 243.

Without loss of generality, SMO first computes the second Lagrange multiplier λ_2 and then expresses the ends of the diagonal line segment in terms of λ_2. The following lower and upper bounds, L_2 and H_2, apply to λ_2:

1. if $y_1 = y_2$:

$$\begin{aligned} L_2 &= \max(0,\ \gamma y_2 - C_1) \\ H_2 &= \min(C_2,\ \gamma y_2) \end{aligned}$$

2. if $y_1 \neq y_2$:

$$\begin{aligned} L_2 &= \max(0,\ \gamma y_2) \\ H_2 &= \min(C_2,\ \gamma y_2 + C_1) \end{aligned}$$

On the next step, SMO computes the location of the unconstrained maximum of the Lagrangian with respect to λ_2:

$$\frac{\partial \mathcal{L}(\lambda_1, \lambda_2)}{\partial \lambda_2} = 0. \tag{244}$$

The corresponding unconstrained λ_2 will be:

$$\lambda_2^{\text{new}} = \lambda_2^{\text{old}} + y_2 \frac{(\mathbf{x}_2 - \mathbf{x}_1) \cdot \mathbf{w}^{\text{old}} + y_1 - y_2}{\nu}. \tag{245}$$

Next, SMO computes the constrained maximum by clipping the unconstrained maximum to the ends of the line segment:

$$\lambda_2^{\text{new,clipped}} = \begin{cases} L, & \text{if } \lambda_2^{\text{new}} \leq L \\ \lambda_2^{\text{new}}, & \text{if } L < \lambda_2^{\text{new}} < H \\ H, & \text{if } \lambda_2^{\text{new}} \geq H \end{cases}. \tag{246}$$

Finally, SMO determines the value of λ_1 from the new clipped value of λ_2:

$$\lambda_1^{\text{new}} = \lambda_1^{\text{old}} - y_1 y_2 (\lambda_2^{\text{new,clipped}} - \lambda_2^{\text{old}}). \tag{247}$$

6.1.8 *Training database*

Here, we used the training database of 851 non-redundant protein-protein complex structures prepared by Huang and Zou [99]. This database contains protein-protein complexes extracted from the PDB [23] and includes 655 homodimers and 196 heterodimers. We updated three PDB structures from the original training database: 2Q33 supersedes 1N98, 2ZOY supersedes 1V7B, and 3KKJ supersedes 1YVV. The training database contains only crystal dimeric structures determined by X-ray crystallography at resolution better than 2.5 Å. Each chain of the dimeric structure has at least 10 amino acids, and the number of interacting residue pairs (defined as having at least 1 heavy atom within 4.5 Å) is at least 30. Each protein-protein interface consists only of 20 standard amino acids. No homologous complexes with the sequence identity $> 70\%$ were included in the training database.

Our algorithm requires as input native and nonnative structure vectors (see, e.g., eq. 225). Native structure vectors can be computed from the native protein-protein contacts in the training database using eq. 222. However, for the computation of the nonnative structure vectors for each protein-protein complex from the training database, we need to generate decoys. Since our optimization algorithm is very general and has no special requirements for nonnative protein-protein contacts, we generated them by "rolling" a smaller protein (ligand) over the surface of a bigger protein (receptor) using the Hex protein docking software [215]. To do so, we initialized Hex exhaustive search algorithm with the radial search step of 1.5 Å and expansion order of the shape function equal to 31. We used only the shape complementarity energy function from Hex (i.e., electrostatic contribution was omitted). Afterwards, we clustered Hex docking results with a root mean square (RMS) threshold of 8 Å. The top 200 clusters, ranked by Hex surface complementarity function, plus the native protein-protein complex conformation

(giving in total 201 structures) were then used to compute the distance distribution functions 216. Then, we computed the structure vectors using eq. 222 and labeled them as "native" if the RMSD of the corresponding ligand was < 2 Å from its native position. Otherwise, the structure vector was labeled as "nonnative". On average, we obtained about 2.5 native structure vectors per protein-protein complex. To each structure vector $\mathbf{x}_{ij}$, we assigned a regularization parameter C_{ij} according to

$$\begin{aligned} C_{ij}^{native} &= CD_j^{nonnative}/D_j \\ C_{ij}^{nonnative} &= CD_j^{native}/D_j \end{aligned}. \tag{248}$$

We repeated the same procedure for each protein-protein complex from the training database. We used $M = 20$ atom-centered interaction sites based on the atom types definitions provided by Huang and Zou [99], resulted in total of 210 pair potentials.

6.1.9 *Results : Overfitting and Convergence*

Various methods of derivation of the knowledge-based potentials usually produce results biased towards the training data set. Typically, such algorithms maximize the predictive accuracy of the corresponding potential on a set of training data, which does not imply that the same potential will perform equally well on a new set of data. Indeed, fitting the potential to the training data set also fits the noise in the data. Thus, very often a knowledge-based potential memorizes noisy features of the training data instead of deducing general predictive concepts from it. This phenomenon is usually referred to as *overfitting* [56]. A clear indication of an "overfitted" potential can be, for example, the need for post-smoothing techniques applied to the initial knowledge-based potential. Overfitting is clearly not desirable. In order to avoid it, many *regularization* techniques have been successfully proposed to penalize the initial objective function with various additional terms [12, 120]. These terms serve to achieve a better predictive accuracy on the off-training data owing to the predictions on the training data.

To avoid overfitting, we introduced two regularization parameters, σ for the width of the Gaussian distribution of distances in eq. 216, and C for the amplitude of the hinge loss function in eq. 226. To find the best values of these parameters we used the following *cross-validation* procedure. Firstly, we divided the training set into two parts, consisting of 200 complexes (temporary training set) and 651 complexes (temporary test set). Then, for each value of σ and C, we obtained the scoring potentials using the temporary training set and verified it on the temporary test set. Finally, we chose those values of σ and C that correspond to the maximum number of guessed structures in the temporary test set. We define the structure as guessed if its native complex has the score better than all of its decoys. Figure 40 shows the predictive performance of the scoring potential on the two sets as a function of σ and C. Obviously, the maximum predictive performance on the training set is achieved at the highest values of C (Figure 40Left). However, the validation on the test set highlights the best choice of these values to be $C = 10^6 \ldots 10^7$ and $\sigma = 0.4$ Å (Figure 40Center).

Figure 40Right shows the convergence of the success rate on the training set with the number of iterations of the BSMO algorithm. The success rate was measured as the number of guessed structures divided by the total number of protein-protein complexes. We can see a fast convergence of the method. In principle, a hundred optimization steps is sufficient to obtain the final result. We have also observed that increasing the regularization parameter C leads to a slower convergence and vice versa. We should note that thanks to the convexity of our optimization problem, its solution is unique and does not depend on the starting point and the optimization method used.

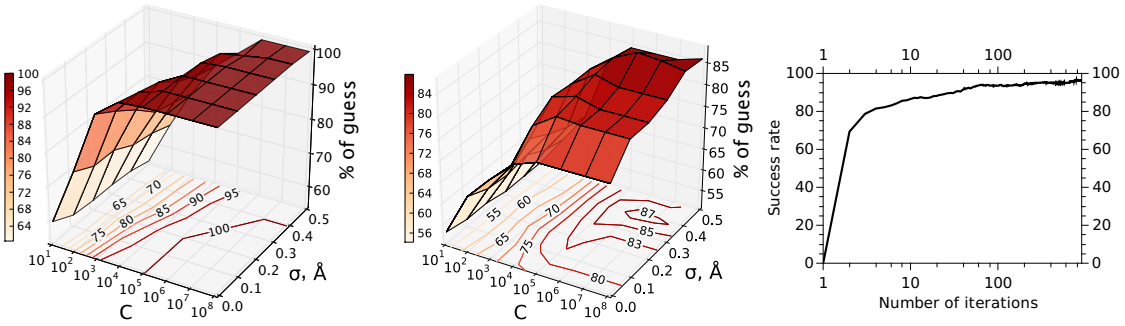


Figure 40: Left and Center : Predictive performance of the Convex-PP scoring potential as a function of the smoothing parameter σ and the regularization parameter C. Left: Performance obtained if the scoring functions are trained on the whole training set and verified on the same set. Center: Performance obtained if the scoring functions are trained on 200 protein complexes and verified on the other 651 complexes from the training set. Here the best performance is obtained with $\sigma = 0.4$ Å and $C = 10^6 \dots 10^7$. Right : Success rate of the scoring potentials on the training set versus the number of iterations of the BSMO algorithm. The scoring potentials were obtained using the Legendre basis and the whole training set, without excluding any homologous proteins. Parameters σ and C were set to the optimal values of $\sigma = 0.4$ Å and $C = 10^5$.

6.1.10 *Results : Extracted Potentials*

Our method can in principle use any type of orthogonal polynomials to decompose the structural statistics and reconstruct the potentials. However, since rectangular functions are the simplest and the most widely used ones, we employed them as a reference. Additionally, we did computational experiments using the Legendre basis orthogonal on the interval [0;10]. We chose this basis because of its simplicity, in particular because its weight function is distance-independent.

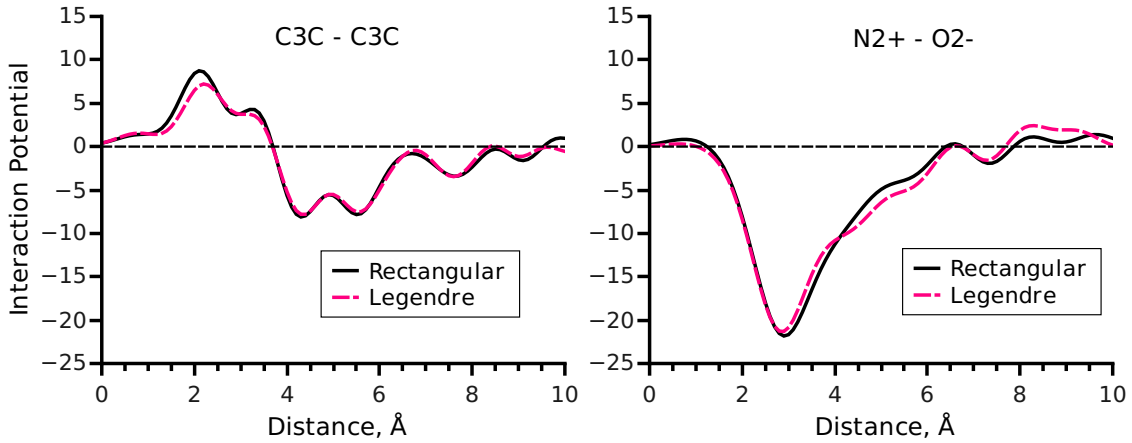


Figure 41: Scoring functions trained in two different polynomial bases. Solid lines correspond to the potentials obtained using the rectangular basis functions. Dashed lines correspond to the potentials obtained using the Legendre basis functions. Left: Potential between aliphatic carbons bonded to carbons or hydrogens only. Right: Potential between a guanidine nitrogen with two hydrogens and an oxygen in carboxyl groups.

From now on, we call the obtained scoring potentials the Convex Protein Protein (Convex-PP) potentials. Figure 41 shows typical scoring potentials derived using the two different orthogonal bases. Obtained potentials are smooth by construction, thanks to the Gaussian kernel in eq. 216. We can see that the shape of the potentials does not depend on the basis set that was used to derive it. This is the consequence of the global convergence of the optimization problem (see Observation 1). We can also see that the obtained potentials tend to zero as the interaction distance increases. On the other hand, all the potentials approach zero at short distances. The latter is the consequence of the

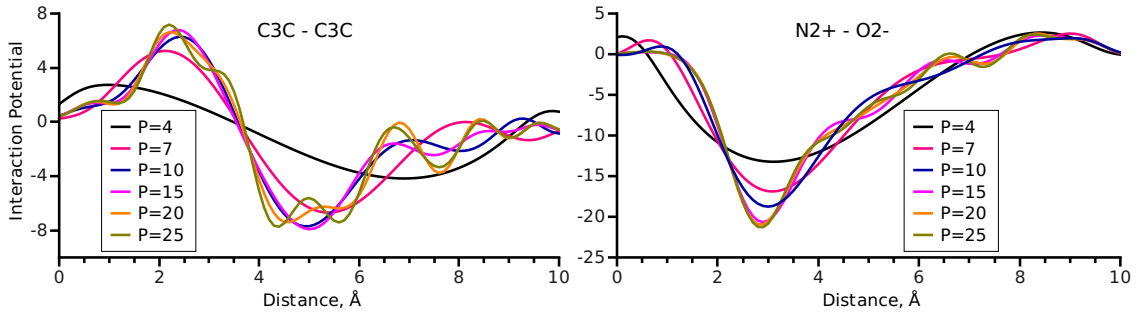


Figure 42: Dependence of the extracted scoring functions on the decomposition order P in the Legendre basis. After order $P = 25$, the potentials are indistinguishable from each other and thus not shown for clarity. Left: Potential between aliphatic carbons bonded to carbons or hydrogens only. Right: Potential between a guanidine nitrogen with two hydrogens and an oxygen in carboxyl groups.

absence of statistics for the native structures at short separation distances and the result of the $\mathbf{w} \cdot \mathbf{w}$ regularisation term in optimization problem 226. We discuss this behaviour in more detail below.

Due to the Gaussian smoothing of statistics, it is sufficient to use the maximum expansion order of $P_{max} = r_{max}/\sigma$. For $\sigma = 0.4$ Å and $r_{max} = 10$ Å, the estimate on the number of basis functions is $P_{max} = 25$. However, due to the adjustment of σ with the cross-validation procedure, in our experiments we used a larger expansion order, $P = 40$. Figure 42 demonstrates how the resulting potentials depend on the expansion order. We should note that decompositions of orders above 25 are almost indistinguishable and thus are not shown.

6.1.11 *Results : Protein-Protein docking benchmarks*

First we tested the Convex-PP scoring function on the protein-protein docking benchmark version 3.0. It consists of 124 crystallographic structures of protein-protein complexes extracted from the PDB database [102]. These are divided into three groups: rigid, medium and difficult cases. The decoys for the scoring were generated using ZDOCK 3.0 [194] with the sampling step equal to 6 degrees (we call this set of docking position *ZDOCK benchmark* below). We also compared our scoring function with the well established ZRANK reranking protocol [195]. Figure 43A shows ROC curves (success rate vs the number of top predictions considered). We see that Convex-PP scoring functions outperform ZRANK and ZDOCK if the number of considered predictions is more than eight.

We also assesed our scoring function using the Rosetta benchmark. Baker, Gray *et al* generated the Rosetta benchmark from 54 complexes of the protein-protein docking benchmark version 0.0 [43] using a flexible docking protocol, which is a part of the RosettaDock suite [81]. Figure 43B compares the results of RosettaDock[81], ITScore-PP[99] and our Convex-PP scoring functions. It shows that our potentials significantly improve prediction rate over the ITScore-PP and RosettaDock scoring functions while also outperforming them according to the other criteria. Unlike the results on the ZDock benchmark, the results on the Rosetta unbound benchmark slightly decrease when we remove homologous complexes from the training set.

6.1.12 *Discussion : Short Distances.*

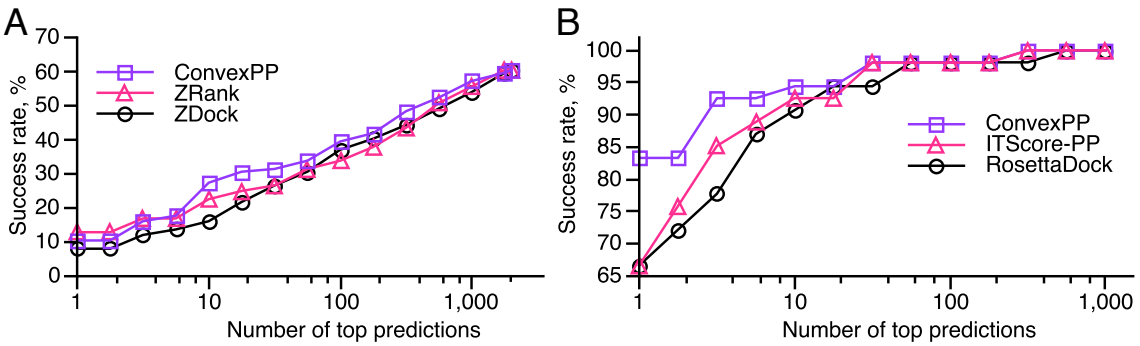


Figure 43: A) Dependence of the success rate on ZDock benchmark on the number of top predictions in consideration for the three methods. B) The same for the Rosetta protein docking benchmark. The data for ITScore-PP and RosettaDock were taken from the original publications [81, 99].

The key property of a scoring function is the existence of the correlation between the score of a structure and its similarity to the corresponding native structure. Conventionally, the ligand-RMSD is taken as the measure of similarity of the decoys to the native structure. Ligand-RMSD is the ligand (the smaller protein in a complex) root mean square deviation of C_α atoms of a decoy relative to the native complex structure when receptors (the larger proteins in the complexes) are superposed. To verify that our potentials indeed correlate with the similarity to the native structures, we plotted the Convex-PP score of each decoy versus the ligand-RMSD for all decoys from the ZDOCK and Rosetta benchmarks. Figure 44 shows some typical plots for the complexes from the training set and the two benchmarks. Typically, in the training set we see a wide separation between native and non-native structures. This happens because decoys in the training set have only few *near-native* structures with ligand-RMSD<10Å. On the contrary, about 28% of the Rosetta decoys are the near-native structures. The ZDOCK benchmark has few near-native decoys compared to Rosetta, only 1.5% of the decoys have the near-native conformations.

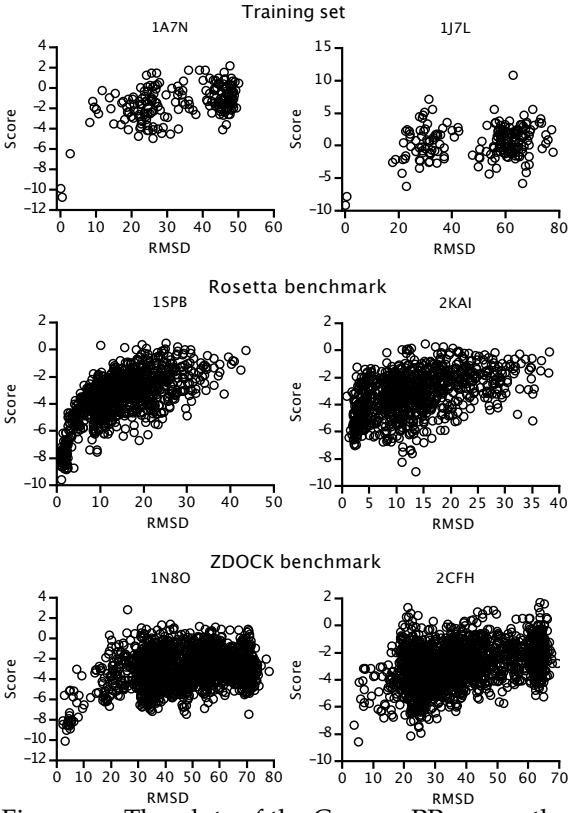


Figure 44: The plots of the Convex-PP versus the ligand-RMSD for the decoy structures from the training set (1A7N, 1J7L), Rosetta benchmark (1SPB, 2KAI) and ZDOCK benchmark(1N8O, 2CHF). On the left we show the plots that exhibit funnel-like behaviour near the frame origin. On the right side the plots without obvious funnels are shown.

Figure 45 plots normalized atom-pairs distance distributions for 1CGI and 1PPE complexes. From this figure we see that the distance distributions for the Rosetta benchmark are much closer to the native distributions compared to the ZDOCK and training set distributions. We can also see that the Hex docking program [215], which we used for the generation of the training set, produces fewer short-distance atom contacts compared to ZDOCK. Since Rosetta decoys were additionally minimized using the Rosetta scoring function, they do not have short-distance atom contacts and generally their distance distributions resemble the native statistics. Native structures neither have statistics

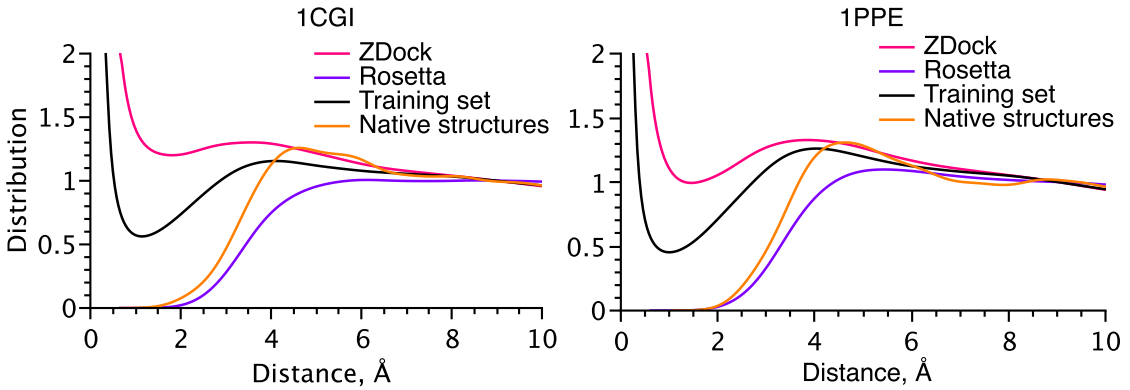


Figure 45: The normalized atom-pairs distance distributions for two complexes 1CGI and 1PPE. For each complex, four plots are shown: average ZDock distribution, average Rosetta distribution, average training set distribution and the native distribution. The average is taken over all decoys from the two benchmarks and the training set.

at short distances. Therefore, reconstructed potentials in the vicinity of zero are not reliable and can not provide fair scores for e.g. decoys generated with ZDOCK, since these decoys have many short-distance contacts. Ideally, one needs to additionally penalize short-distance contacts using, e.g., empirical scores that cannot be obtained with statistics from the native structures. This was one of our motivations to develop scoring potentials further, see, e.g., the KSENIA potential below.

6.1.13 *Discussion : Filtering*

Some knowledge-based potentials are smoothed with a smoothing filter *a posteriori*. For example, Mitchel et al. [167] and Huang et al. [99] used a "1:2:4:2:1" filter, DOPE potential is smoothed using cubic polynomials [231], etc. On the contrary, our method introduces an assumption about interaction pair distance uncertainty *a priory*. More specifically, we collect statistics using gaussian events 216 with the standard deviation of σ. We determine the value of σ from the afore-mentioned cross-validation procedure. Then, according to eq. 217, Convex-PP scoring function is smooth by construction. In other words, we do not need to apply a smoothing filter to the obtained potentials, since we introduce the uncertainty when we collect statistics. Another parameter that indirectly influences the smoothness of the resulting potential is the regularization parameter C (se eq. 248). According to eq. 239, the scoring vector $\mathbf{w}$, from which the scoring potentials are derived, is a weighted sum of the support structure vectors $\mathbf{x}_{ij}$. The more support structure vectors are in the sum, the more regular the scoring vector $\mathbf{w}$ will be. On the other hand, this number equals to the number of non-zero Lagrange multipliers λ_{ij} 239, which is uniquely defined by the value of the regularization parameter C [202]. Decreasing C results in the increase of the number of non-zero λ_{ij} therefore resulting in smoother scoring potentials. We also determine the value of this parameter by the cross-validation procedure. The consistent determination of the two parameters σ and C allows us to obtain smooth potentials according to eq. 217 directly as the solution of the optimization problem 226.

6.1.14 *Discussion : Uniqueness of the solution and the reference state*

The concept of the statistical knowledge-based potentials is based on the definition of two states: the observed state and the reference state [169, 233, 251]. The observed state is usually the state when a single protein or a complex has the native conformation. It can be derived from the crystal structures. Reference state was introduced as an atom

pair distance distribution when the interactions between the atom pairs are absent. The knowledge-based potential is then expressed in terms of these two states as:

$$u_{ij}(r) = -RT \ln \left(\frac{N_{ij}^{obs}(r)/N_{ij}^{obs}}{N_{ij}^{ref}(r)/N_{ij}^{ref}} \right), \tag{249}$$

where $N_{ij}^{ref}(r)$ and $N_{ij}^{obs}(r)$ are the numbers of atomic pairs i, j at a distance r in the reference and observed states, correspondingly, and numbers N_{ij}^{ref} and N_{ij}^{obs} are the total numbers of pairs i, j in these states. Some widely used approaches to derive the reference state for protein folding are the ideal-gas approximation [282], the shuffling of atoms [222], a random-walk chain [277], etc. For protein docking Chuang *et al.* used decoys as the reference state [47], Bernard and Samudrala took the average over the atomic pairs and a cumulative distribution function for all pairs as two reference states [26], etc. The very wide variety of approaches to derive the reference state has its roots in the loose definition and the complexity of the problem.

Recently, the new algorithms that avoid the reference state calculation appeared. We should mention the iterative scheme used by Huang et al. [99] and the neural network classifier by Chae et al. [39]. These algorithms indeed avoid the definition of the reference state. However, they do not guarantee the uniqueness of their solution. On the contrary, we showed that our algorithm converges to the global minimum of the function 226. Thus, we avoid dependence on the initial guess of the interaction potential.

6.2 IDENTIFICATION OF WATER MOLECULES AROUND A PROTEIN

We used the methodology described above to discover water molecules around X-ray protein structures. Figure 46 shows an example of our potential trained and assessed for a blind predictions of water positions at protein-protein interface, performed as part of the critical assessment of predicted interactions (CAPRI) community-wide experiment for the CAPRI Target 47. Our method was the only one that predicted near zero false positive water positions [137].

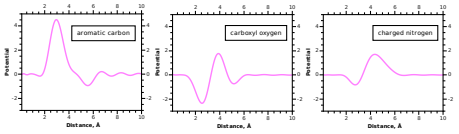


Figure 46: Knowledge-based solvation scoring functions between the oxygen of a water molecule and protein atoms as a function of the separation distance. Left: water – aromatic carbon. Middle: water – carboxyl oxygen (like in aspartic and glutamic acids). Right: water – charged nitrogen (like in lysins). Was developed for [137].

6.3 THE KSENIA DOCKING POTENTIAL

We later fixed the shortcoming of the Convex-PP scoring function that did not penalize interactions at very short interaction distances not seen during the training phase. To do so, we extrapolated the scoring potentials $U_{kl}(r)$ from eq. 220 using a cubic spline interpolation with a number of reference points. Figure 47 shows some of the obtained potentials. Another novelty of the scoring functions was the fact that we only used information about the native protein-protein interfaces during training. To solve the binary classification problem, for the second class labels, we generated random near-native conformations with coordinate deformations along

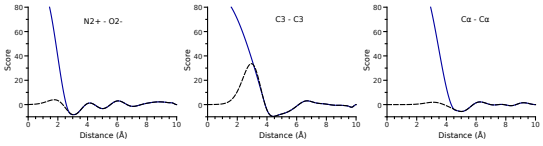


Figure 47: Examples of the KSENIA distance-dependent scoring functions between atoms of types N2+ – O2- , C3 – C3 and $C_\alpha - C_\alpha$, respectively. Here, N2+ are guanidine nitrogens with two hydrogens, O2- are oxygens in carboxyl groups, C3 are aliphatic carbons bonded to carbons or hydrogens only and C_α are the backbone C_α atoms. Black, dashed: initially derived scoring functions without taking into account the absence of statistics at short distances. Blue, solid: redefined scoring functions that take into account the absence of statistics at short distances.

the low-frequency normal modes. During the scoring phase we also introduced a rigid-body minimazion algorithm [207].

Then, I integrated the obtained potentials and optimization algorithms into an *interactive* molecular docking environment. Figure 48 shows an example of this environment during CAPRI session predictions.

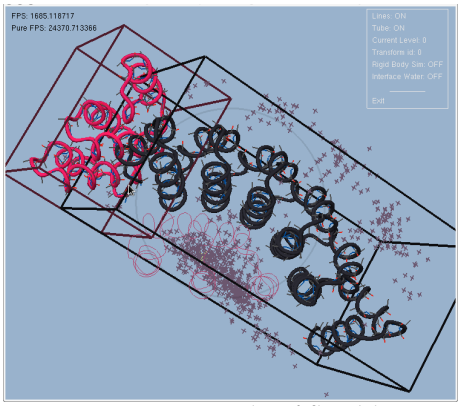


Figure 48: An example of flexible interactive docking for CAPRI Target 53.

6.4 PEPSI-DOCK ML-BASED SYSTEMATIC DOCKING

Finally, we demonstrated that an arbitrary-shaped ML-based potential can be integrated into an FFT-accelerated systematic docking engine. More concretely, we developed Pepsi-Dock [176], a protein docking algorithm that uses a very precise energy function [205] to explore the 6D search space. It combines a very fast FFT-accelerated exhaustive search with a detailed data-driven model of the binding free energy. This was the first demonstration how computation of a distance-dependent knowledge-based pairwise energy function can be accelerated by FFT. The method is available at team.inria.fr/nano-d/software/PEPSI-Dock/.

The main methodological idea of the method is the fact that a potential field in 3D acting on an object, if the field is created by a set of points with spherically-symmetric potentials, which are additive, can be also represented as a sum of 1D integrals. Let us assume a receptor molecule composed of R_i atoms of type i exerting a potential field $f_{ij}(\mathbf{x})$ that acts on a ligand molecule composed of L_j atoms of type j at positions $g(\mathbf{x})$:

$$\begin{aligned} E &= \sum_{ij} \iiint_V f_{ij} \sum_{R_i} \sum_{L_j} (\mathbf{x} - \mathbf{x}_{R_i})\, g(\mathbf{x} - \mathbf{x}_{L_j})\, dV \quad \text{used in systematic FFT-based search} \\ &= \sum_{ij} \sum_{R_i} \sum_{L_j} \int_r f_{ij}(r) g_{|1D}(r - \mathbf{x}_{L_j}) dr \qquad\qquad \text{used in the ML optimization} \end{aligned} \tag{250}$$

Then, the total interaction energy, given us a sum $\sum_{ij}$ of 3D volumetric contributions over all atom type combinations, $\iiint_V \sum_{R_i} \sum_{L_j} f_{ij}(\mathbf{x} - \mathbf{x}_{R_i})\, g(\mathbf{x} - \mathbf{x}_{L_j})\, dV$, can be also seen as a sum of 1D radial-dependent integrals, where symmetrical angular-dependent degrees of freedom have been integrated out, $\int_r f_{ij}(r) g_{|1D}(r - \mathbf{x}_{L_j}) dr$. It is convenient to use the latter expression in the ML task, as it coincides with the optimization problem 226. And the former expression is identical to those used in systematic 3D FFT-based docking engines. Figure 49 shows a schematic pipeline of the method.

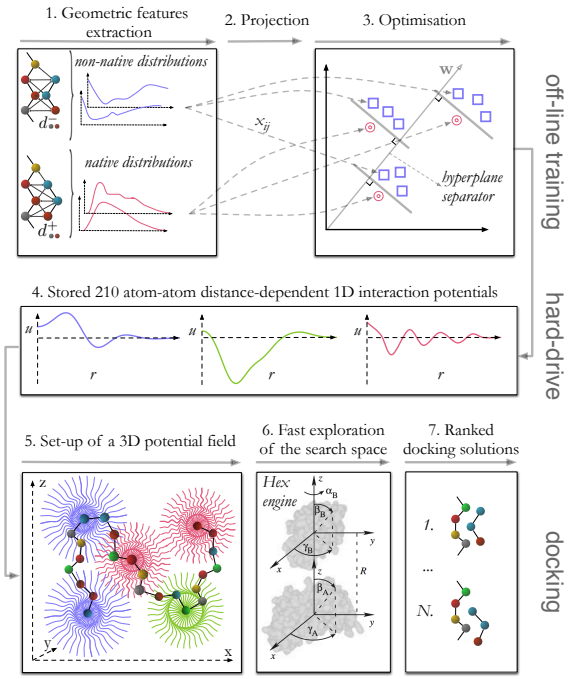


Figure 49: Schematic representation of the Pepsi-Dock algorithm. Firstly, we learn the interaction potentials. Then, we use them in a systematic FFT-accelerated docking.

6.5 DOCKING OF SMALL MOLECULES

In molecular biology and pharmacology, a small molecule is a low molecular weight organic compound that may help regulate a biological process, with a size on the order of 1 nm. Most drugs are small

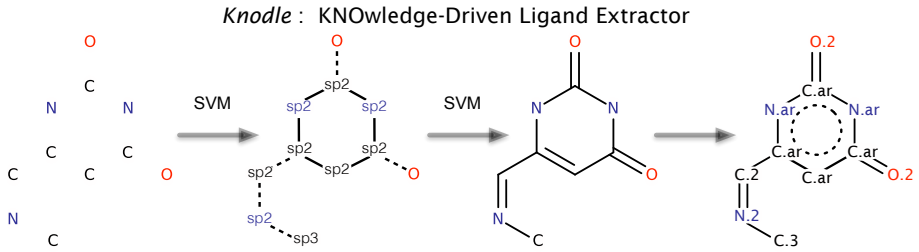


Figure 50: Schematic workflow of the Knodle method.

molecules. Current computational challenges in this field include prediction of protein-small molecule docking poses, virtual screening of drug-like compounds, and rational drug design.

Our initial goal was to extend ML-based potentials developed for protein-protein interactions to protein-ligand tasks. However, we did not aim using classical parametrization of small drug-like molecules. Instead, we wanted to learn everything from available data. This lead us to the problem of the assignment of atom types and bond orders in low molecular weight compounds, for which we developed a prediction model based on nonlinear Support Vector Machines (SVM). Figure 50 shows a schematic illustration of the Knodle pipeline. We implemented the developed methods in a KNOwledge-Driven Ligand Extractor called Knodle, a software library for the recognition of atomic types, hybridization states and bond orders in the structures of small molecules [111]. We trained the model using an excessive amount of structural data collected from the PDBbindCN database. Accuracy of the results and the running time of our method is comparable with other popular methods, such as NAOMI, fconf, and I-interpret. Overall, our study demonstrated the efficiency of nonlinear SVM in structure perception tasks. Knodle is available at https://team.inria.fr/nano-d/software/Knodle.

6.6 THE CONVEX-PL AND CONVEX-PL^R POTENTIALS

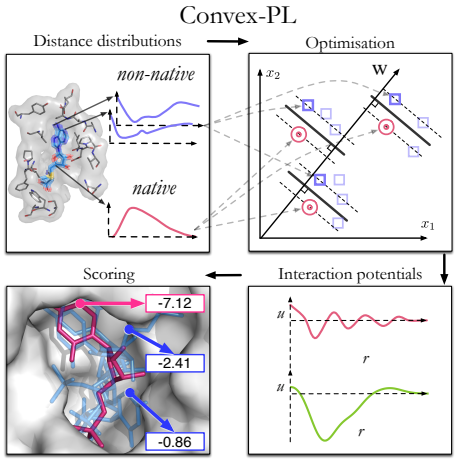


Figure 51: Schematic workflow of the ConvexPL derivation.

Grounded on the Knodle typization of small molecules, we then derived a novel protein-ligand interaction potential called Convex-PL [86, 87, 112]. We did not impose any functional form of the scoring function. Instead, we decomposed it into a polynomial basis and deduced the expansion coefficients from the structural knowledge base using a convex formulation of the optimization problem 226. Here, our optimization problem had the dimensionality of $\sim$ 50,000 with about 150,000 of linear constraints. Also, for the training set we did not generate false poses with molecular docking packages, but use constant RMSD rigid-body deformations of the ligands inside the binding pockets. This allowed the obtained scoring function to be generally applicable to scoring structural ensembles generated with different docking methods. The method was also patented [46]. We assessed the Convex-PL scoring function using data from D3R Grand Challenge 2 submissions and the docking test of the CASF 2013 study. We demonstrated that our results outperformed the other 20 methods previously assessed in CASF 2013, as it is shown in Fig. 52. The method is available at http://team.inria.fr/nano-d/software/Convex-PL/.

Later, we analyzed scoring functions' performance in the CASF benchmarks and discovered that the vast majority of them have a strong bias towards predicting larger binding interfaces. This motivated us to extend our protein-ligand interaction potential

and develop a physical model with additional entropic terms with the aim of penalizing such a preference. We parameterized the new model using affinity and structural data, solving a classification problem followed by regression. The new model, called Convex-PL^R, demonstrated high-quality results on multiple tests and a substantial improvement over its predecessor Convex-PL. Convex-PL^R can be used for molecular docking together with VinaCPL, our version of AutoDock Vina, with Convex-PL integrated as a scoring function. Convex-PL^R, Convex-PL, and VinaCPL are available at https://team.inria.fr/nano-d/convex-pl/.

6.7 PROTEIN-LIGAND DOCKING METHODS

We then integrated the Knodle parametrization and the Convex-PL and Convex-PL^R potential into a docking engine that predicts binding poses of small molecules with respect to their protein receptors [110]. Molecular interactions are precomputed on a rigid grid, small molecules are treated in the dihedral angle subspace with all the rotatable bonds active, and the search is made with Monte-Carlo and rapidly exploring random trees (RRT) techniques. This method has been used in a number of blind assessment exercises and is available for download on our website at https://team.inria.fr/nano-d/convex-pl/.

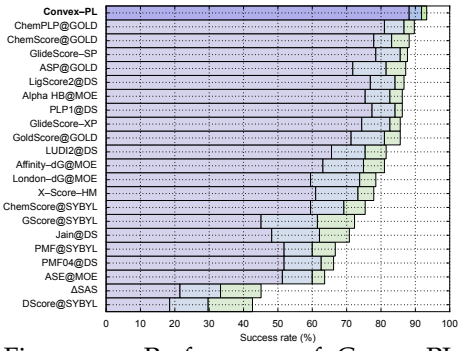


Figure 52: Perfomance of ConvexPL potential on the docking test of the CASF 2013 benchmark.

6.8 KORP-PL POTENTIAL

Despite the progress made in studying protein–ligand interactions and the widespread application of docking and affinity prediction tools, improving their precision and efficiency still remains a challenge. Computational approaches based on the scoring of docking conformations with statistical potentials constitute a popular alternative to more accurate but costly physics-based thermodynamic sampling methods. In this context, a minimalist and fast sidechain-free knowledge-based potential with a high docking and screening power is extremely useful when screening a big number of putative docking conformations. This observation motivated us to explore the idea of implicit (coarse-grained) representations of protein molecules in protein-ligand interactions. As a result, we developed KORP-PL [114], a novel coarse-grained potential defined by a 3D joint probability distribution function that only depends on the pairwise orientation and position between protein backbone and ligand atoms . Figure 53 explains the geometrical description used in KORP-PL. Despite its extreme simplicity, our approach yields very competitive results with the state-of-the-art scoring functions, especially in docking and screening tasks. For example, we observed a twofold improvement in the median 5% enrichment factor on the DUD-E benchmark compared to the state-of-the-art Autodock Vina results. Moreover, our results prove

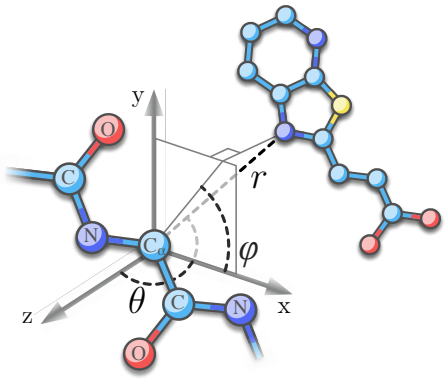


Figure 53: KORP-PL coarse-graind potential. Schematic view defining the relative orientation and position of a ligand relative to a protein. The relative orientation of a ligand atom is described by two spherical angles, i) θ, the angle between the r and z vectors, and ii) φ, the angle between x and the projection of r into the xy plane.

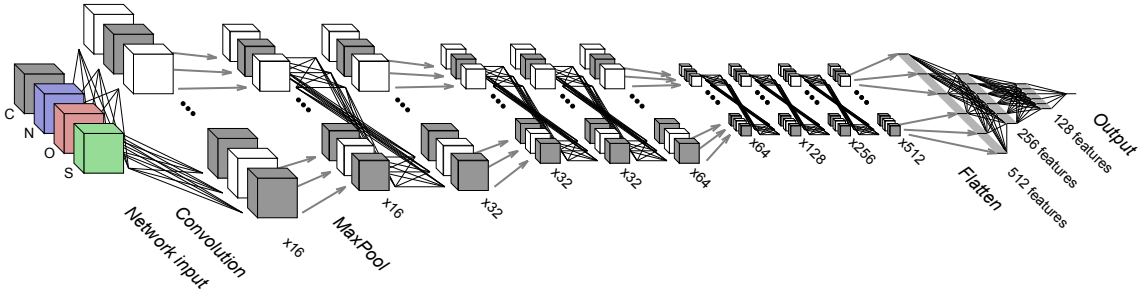


Figure 55: A schematic representation of the convolutional neural network architecture from [53]. Unless otherwise specified, line connections across boxes denote the consecutive application of a 3D convolutional layer ('Convolution'), a batch normalization layer ('BatchNorm') and a ReLU layer. Grey arrows between boxes denote maximum pooling layers ('MaxPooling'). Labels 'M' denote the number of 3D grids and the number of filters used in the corresponding convolutional layer. The grey stripes denote one-dimensional vectors and crossed lines between them stand for fully-connected layers with ReLU nonlinearities.

that a coarse sidechain-free potential is sufficient for a very successful docking pose prediction. We also used the developed method in two blind challenges (GPCR Dock 2021 and CASP15). This work was carried out with Pablo Chacon from IQFR-CSIC Madrid, Spain with the support from FlexMol Inria associate team (2019-2022). The standalone version of KORP-PL with the corresponding tests and benchmarks are available at https://team.inria.fr/nano-d/korp-pl/ and https://chaconlab.org/modeling/korp-pl.

6.9 SBROD PROTEIN SINGLE-MODEL QUALITY ASSESSMENT METHOD

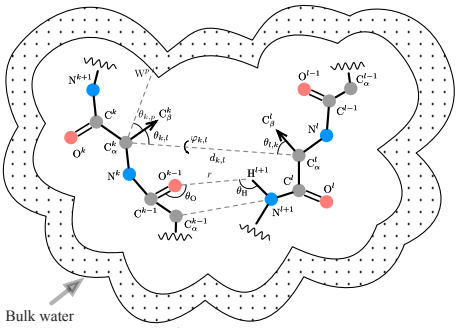


Figure 54: Schematic representation of physical and geometrical features used in the SBROD potential.

Our work on ML applied to protein interactions motivated us to extend the application domain of our methods and develop a technique specifically for recognition of protein folds. Thus we created a novel protein single-model quality assessment method called SBROD [117]. The SBROD (Smooth Backbone-Reliant Orientation-Dependent) method uses only the conformation of the protein backbone, and hence it can be applied to scoring the coarse-grained protein models. The proposed method deduces the scoring function from a training set of protein 3D models. It is smooth with respect to atomic coordinates, and is composed of four terms related to different structural features, residue-residue orientations, contacts between the backbone atoms, hydrogen bonding, and solvent-solvate interactions. The method is available at https://team.inria.fr/nano-d/software/SBROD.

6.10 DEEP LEARNING

6.10.1 *3D CNNs*

Our deep-learning campaign started with the 3D convolutional neural network (3D CNN) developed by my student Georgy Derevyanko [53]. We were generally motivated by the computational prediction of a protein structure from its sequence and were looking into multiple methods to assess the quality of protein models. Early-stage ML-based methods, like SBROD developed in our team, even being very competitive at the CASP

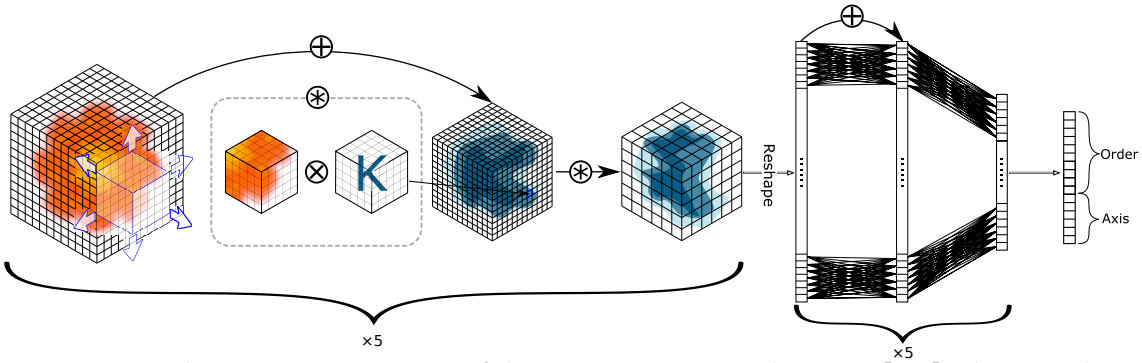


Figure 57: A schematic representation of the DeepSymmetry architecture [184]. The input layer containing a 3D density map is followed by five residual convolutional layers. The output is then reshaped into a linear array and five residual layers of a fully connected network are added. The output vector contains information about the order and the axis of the putative symmetry.

blind assessments, still relied on engineered structural features, defined as some functions of the atomic coordinates [117]. Very few methods had attempted to learn these features directly from the data. As a result, we demonstrated for the first time that deep CNNs can be used to predict the ranking of protein model structures solely on the basis of their raw three-dimensional atomic densities, without any feature tuning. Figure 55 schematically shows our architecture. We trained the network on decoy protein models from the the CASP7 to CASP10 datasets, tuned it parameters using the CASP11 dataset, and validated the performance on CASP12, CAMEO, 3DRobot, and blind testing in CASP13 datasets, where it performed on par with the state-of-the-art algorithms.

ORNATE: The previous 3D CNN architecture, despite being very innovative, had a number of fundamental flaws. Most importantly, it was not invariant to the orientation of the initial 3D model and had to be trained on multiple orientations of the same data. Also, it operated on volumetric grids of a predefined size and used an external parametrization (channels) of protein atoms. These challenges motivated us to develop Ornate (Oriented Routed Neural network with Automatic Typing) – a novel method for single-model protein quality assessment (QA) [182]. Ornate is a residue-wise scoring function that takes as input 3D density maps. It predicts the local (residue-wise) and the global model quality through a deep 3D CNN. Specifically, Ornate aligns the input density map, corresponding to each residue and its neighborhood, with the backbone topology of this residue. This circumvents the problem of ambiguous orientations of the initial models. Also, Ornate includes automatic identification of atom types and dynamic routing (gating) of the data in the network. Established benchmarks (CASP 11 and CASP 12) demonstrated the state-of-the-art performance of our approach among single-model QA methods. It was also a very competitive architecture in the subsequent blind CASP challenges CASP13 and CASP14. The method is available at https://team.inria.fr/nano-d/software/Ornate/.

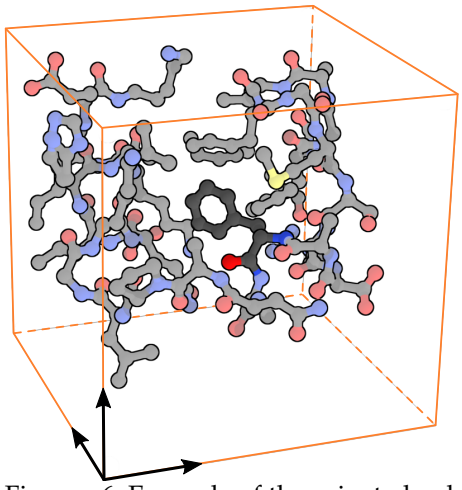


Figure 56: Example of the oriented volumetric input corresponding to one protein residue, as it it implemented in the Ornate architecture [182]. The orientation is fixed by the topology of the central residue's backbone. The atoms of the considered residue are shown in dark colors and the atoms of its neighborhood are shown in light colors. The orange box shows the boundaries of the residue's neighborhood. Only the atoms within this neighborhood are shown and considered by the architecture.

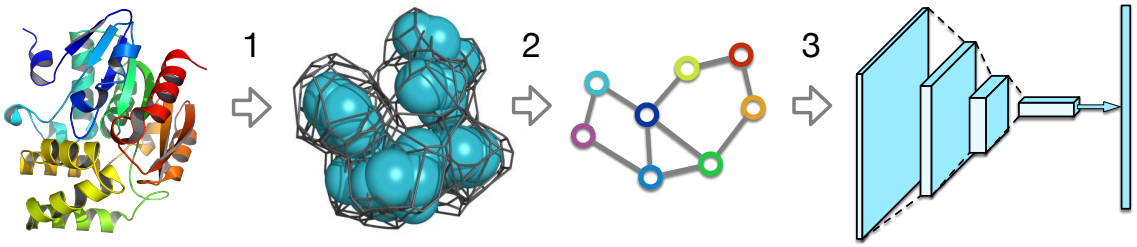


Figure 58: Schematic representation of the VoroCNN Voronoi-based geometric learning algorithm. Firstly, a Voronoi tessellation of a 3D-model is computed with the Voronota library. Then, based on Voronoi 3D-tessellation, a graph is built. Finally, a graph neural network predicts local CAD-scores of all residues in the initial model.

DEEPSYMMETRY: Motivated by the initial successes of 3D convolutional neural networks in multiple tasks, we applied the same ideas for the detection of structural repetitions in proteins and their density maps and created a deep neural architecture called DeepSymmetry [184]. We designed our method to identify tandem repeat proteins, proteins with internal symmetries, symmetries in the raw density maps, their symmetry order, and also the corresponding symmetry axes. Detection of symmetry axes is based on learning six-dimensional *Veronese mappings* of 3D vectors, and the median angular error of axis determination is less than one degree. Figure 57 shows a schematic workflow of our architecture. We demonstrated the capabilities of our method on benchmarks with tandem repeated proteins and also with symmetrical assemblies. For example, we have discovered about 7,800 putative tandem repeat proteins in the PDB. According to our tests, the method is able to detect the order of a cyclic symmetry with a $> 90\%$ accuracy, and guesses the direction of the axis of symmetry with an average error of $< 1°$. The method is available at https://team.inria.fr/nano-d/software/deepsymmetry/.

6.10.2 *Voronoi tessellations and geometric learning*

Learning molecular representations in three dimensions (3D) poses numerous algorithmic challenges. These include rotational invariance of the representation, rotational dependence of the geometric features, learning chemical-geometrical features, and many more. One of the ideas to crack this problem was to descibe a molecular shape using irregular 3D tessellations, such as Voronoi diagrams or molecular graphs, and then to apply geometric deep learning to them. This motivated us to create the first geometric learning methods operating on Voronoi tessellations of molecular shapes – VoroCNN and its extension to angular filters S-GCN [104]. It turned out that geometric learning is very efficient, as two of these models, VoroCNN and S-GCN, were ranked among the top-3 methods in the recent assessment of protein model quality prediction tasks in CASP 14 (December 2020). Figure 58 and Figure 59 schematically show our ideas.

VOROCNN: VoroCNN (Fig. 58) was the first deep convolutional neural network (CNN) constructed on a Voronoi tessellation of 3D molecular structures [105]. Despite the irregular data domain, our data representation allowed to efficiently introduce both convolution and pooling operations of the network. We trained our model to predict local qualities of 3D protein folds. The prediction results were competitive to the state of the art and superior to the previous 3D CNN architectures built for the same task (the model was ranked in top-3, among over 70 methods, for the blind CASP14 challenge). In the manuscript, we also discussed practical applications of VoroCNN, for example, in the recognition of protein binding interfaces. The method is available at https://team.inria.fr/nano-d/software/vorocnn/ and in gitlab repository. This project was conducted in a close collaboration with the team of f Ceslovas Venclovas from Vilnius Uni-

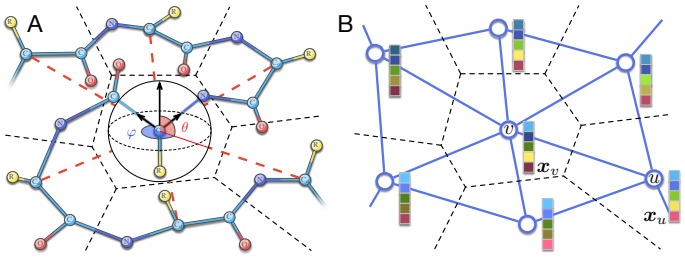


Figure 59: Illustration of a molecular graph representation used by the S-CGN scheme. (A) 3D protein structure is partitioned into Voronoi cells, shown with the dashed lines. The central amino acid has the associated coordinate system, which is built according to the topology of its backbone (atoms C, C_α, N) with the center at the position of the C_α atom. R symbols denote amino acid residues. The spherical angles φ and θ of the neighboring residues are computed with respect to the local coordinate system of the central residue. (B) Graph corresponding to the Voronoi tessellation, v is the central node, u is its neighbor, $\mathbf{x}_v$ and $\mathbf{x}_u$ are the corresponding feature vectors, which are also shown with colored boxes. A graph-learning network is then constructed on this graph, such that graph convolutional filters have angular dependence on spherical angles φ and θ.

versity, Lithuania, supported by the PHC Gilibert 2019-2020 grant. VoroCNN was also presented as a highlight talk at the ICML 2021 Workshop on Computational Biology.

S-CGN: Then, we extended VoroCNN with Spherical Graph Convolutional filters [104]. In a protein molecule, individual amino acids have common topological elements. This allowed us to unambiguously associate each amino acid with a local coordinate system and construct rotation-equivariant spherical filters that operate on angular information between graph nodes (Fig. 59). More technically, our main idea was to approximate spherical convolutional filters (matrix functions) $\mathbf{F} : S_1 \to \mathbb{R}^{d_1 \times d_2}$ trough a finite expansion series in spherical harmonics $Y_l^m(\theta, \varphi)$,

$$\mathbf{F}(\theta, \varphi) \approx \hat{\mathbf{F}}(\theta, \varphi) = \sum_{l=0}^{L} \sum_{m=-l}^{l} \mathbf{W}_l^m Y_l^m(\theta, \varphi), \tag{251}$$

where matrices $\mathbf{W}_l^m$ denote expansion coefficients of the function $\mathbf{F}$ in the Y_l^m basis. This allowed us to introduce the spherical convolution operation for the vertex v in a graph in the following way,

$$\mathbf{F} \circ v = \sum_{u \in \mathcal{N}(v)} \hat{\mathbf{F}}(\theta_v^u, \varphi_v^u) \mathbf{x}_v. \tag{252}$$

Considering matrices $\mathbf{W}_l^m$ to be optimized parameters, we thus learn a spherical filter. We should specifically emphasise that matrices $\mathbf{W}_l^m$ are rotation-equivariant by construction.

Within the framework of the protein model quality assessment problem, we demonstrated that the proposed spherical convolution method significantly improves the quality of model assessment compared to the standard message-passing approach. It is also comparable to state-of-the-art methods, as we demonstrated on Critical Assessment of Structure Prediction (CASP) benchmarks and in the CASP14 blind challenge, where the model was ranked in top-3 among over 70 methods. The proposed technique operates only on geometric features of protein 3D models. This makes it universal and applicable to any other geometric-learning task where the graph structure allows constructing local coordinate systems. The method is available at https://team.inria.fr/nano-d/software/s-gcn/ and in gitlab repository. S-GCN was also presented as a highlight talk at the ICML 2021 Workshop on Computational Biology.

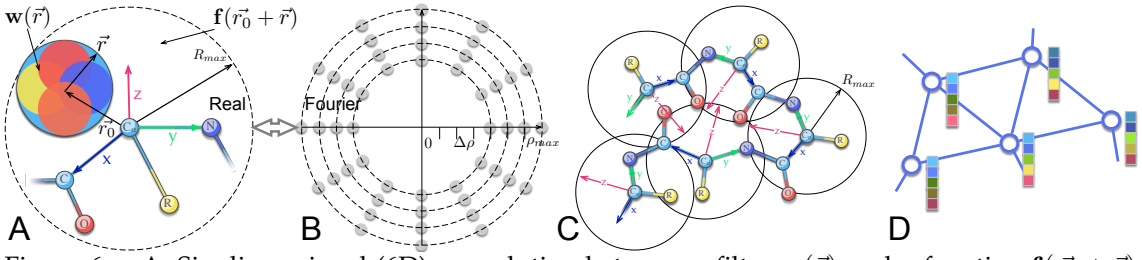


Figure 60: A. Six-dimensional (6D) convolution between a filter $\mathbf{w}(\vec{r})$ and a function $\mathbf{f}(\vec{r_0}+\vec{r})$. The function $\mathbf{f}(\vec{r_0}+\vec{r})$ describes the local environment of a protein residue and is defined within a certain radius R_{max} from the corresponding C_α atom. The local coordinate system xyz is built on the backbone atoms C_α, C, N of each protein residue. R denotes the location of a residue's side-chain. B. The spherical Fourier space with the reciprocal spacing $\Delta\rho$ and the maximum resolution of ρ_{max}. Grey dots schematically illustrate points where the Fourier image is stored. C. An illustration of a protein chain representation. Each protein residue has its own coordinate system xyz and the corresponding local volumetric description $\mathbf{F}_l^k(\rho)$ within a certain sphere of R_{max} radius. Spheres of different residues may overlap. Two residues are considered as neighbors in the graph representation if their C_α atoms are located within a certain threshold R_n. D. The graph representation of the protein structure. The node features are learned by the network and are represented with colored rectangles. The edge features are assigned based on the types of the corresponding residues and the topological distance of the protein graph.

6.10.3 *6DCNN, local equivariance, and physics-based neural layers*

3D molecular data turns out to be rather different from classical 2D images in that respect, that in big molecules we may have multiple identical 3D pattens with different orientations. The challenge in novel convolutional networks will be to find relations between these pattens by using operations preserving *local equivariance*. Classical CNN operators would require sampling 6 degrees of freedom for each new convolutional filter in 3D. This is certainly out of reach for today's CPU and RAM hardware, as the dimensionality of network's parameters would increase *exponentially* with the number of layers in the network.

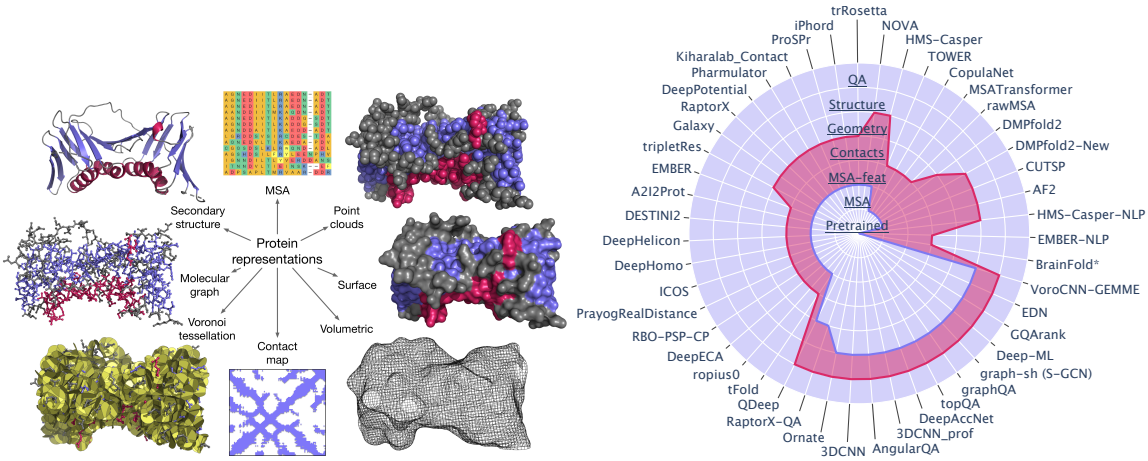
This motivated us to develop six-dimensional (6D) Convolutional Neural Network (6DCNN) designed to tackle the problem of detecting relative positions and orientations of local patterns when processing three-dimensional volumetric data [278]. Technically, the main idea was to extend the convolution operation with an integration of all possible filter rotations. Let $\mathbf{f}(\vec{r}) : \mathcal{R}^3 \to \mathcal{R}^{d_i}$ and $\mathbf{w}(\vec{r}) : \mathcal{R}^3 \to \mathcal{R}^{d_i} \times \mathcal{R}^{d_o}$ be the initial signal and a spatial filter, correspondingly. We proposed to extend the classical convolution as follows,

$$\int_{\vec{r}} d\vec{r}\, \mathbf{f}(\vec{r_0}+\vec{r})\mathbf{w}(\vec{r}) \to \int_{\Lambda} d\Lambda \int_{\vec{r}} d\vec{r}\, \mathbf{f}(\vec{r_0}+\Lambda^{-1}\vec{r})\mathbf{w}(\Lambda\vec{r}), \tag{253}$$

where $\Lambda \in \mathrm{SO}(3)$ is a 3D rotation (see Fig. 60A). 6DCNN also includes SE(3)-equivariant message-passing and nonlinear activation operations constructed in the Fourier space. Working in the Fourier space allows significantly reducing the computational complexity of our operations. Indeed, let the functions $\mathbf{f}(\vec{r})$ and $\mathbf{w}(\vec{r})$ be *finite-resolution* and have spherical Fourier expansion coefficients $F_l^k(\rho)$ and $W_l^k(\rho)$, correspondingly, which are nonzero for $l \leq$ than some maximum expansion coefficient L. Then, the result of the 6D convolution has the following Fourier coefficients,

$$[\mathbf{F}_{\text{out}}]_l^k(\rho) = \sum_{l_1=0}^{L} \sum_{k_1=-l_1}^{l_1} \frac{8\pi^2}{2l_1+1} \mathbf{W}_{l_1}^{-k_1}(\rho) \sum_{l_2=|l-l_1|}^{l+l_1} c^{l_2}(l,k,l_1,-k_1)\mathbf{F}_{l_2}^{k+k_1}(\rho), \tag{254}$$

where c^l are the products of three spherical harmonics. For a single reciprocal distance ρ, the complexity of this operation is $O(L^5)$, where L is the maximum order of the



(a) (b)

Figure 61: Left : Comparison between protein representations for human PCNA (PDB code:1AXC, chain A). Right : Schematic representation of the inputs and outputs of deep learning-based methods in CASP14, excluding pipelines compiling several methods coming from different sources, and methods lacking a clear description. The blue and red lines indicate the input and output levels, respectively. Pretrained: sequence embeddings determined from NLP models pre-trained on huge amounts of sequence data. MSA: raw multiple sequence alignement. MSA-feat: MSA features (such as PSSMs, covariance and precision matrices). Contacts: contact or distance matrix. Geometry: geometrical features, typically including contacts/distances and torsion angles. Structure: 3D coordinates. QA: model quality. In case of several inputs and/or outputs, we report those closest to the "end".

spherical harmonics expansion. We demonstrated the properties of the 6D convolution and its efficiency in the recognition of spatial patterns. We also assessed the 6DCNN model on several datasets from the recent CASP protein structure prediction challenges. There, 6DCNN improved over the baseline architecture and also outperformed the state-of-the-art.

6.10.4 *Review for the CASP14 special issue on the progress of deep learning*

After the end of the CASP14 protein structure prediction blind challenge in December 2020, the CASP organizers invited me to contribute with an overview paper on the DL-related methodological advances in the field for the CASP14 special issue. I invited 3 of my colleagues and we compiled our opinion of the novel deep-learning approaches developed between 2018-2020 and widely used in CASP14 . We specifically reviewed novel representations of protein structures, such as tessellations, surfaces, molecular graphs, and point clouds (see Fig. 61Left), carefully listed and explained all recent DL architectures (see Fig. 61Right), and also provided our outlook on the current impact of DL on structural biology and the future in the field [131].

6.11 CONCLUSION

The potential of deep learning has been recognized in the structural bioinformatics community for already some time and became indisputable after CASP13 in 2018. In CASP14 (2020) and CASP 15 (2022) blind experiments, deep learning has boosted the field to unanticipated levels reaching near-experimental accuracy of single-domain predictions (CASP14) and protein assemblies (CASP15). This success comes from ideas and advances transferred from other machine learning areas and methods specifically designed to deal with protein sequences and structures and their abstractions. The future of structural bioinformatics seems to orient toward exploiting vast collections of data

in a mostly unsupervised fashion. Current technological advances, both in experimental and computational sciences, bring us to a new level of understanding of how cellular machinery works and what will be the algorithmic needs in the nearest future. Open challenges include understanding the principles of functioning very flexible or disordered macromolecules, such as proteins and RNAs, and also various aspects of molecular interactions beyond stable protein-protein assemblies. I would argue that physics-based and geometrical priors will be very useful in future method developments. Thus, I am confident that physics-based and engineering approaches will not disappear from structural bioinformatics.

7

OUTLOOK

Following the enormous progress in sequencing techniques and instrumentation, we have just witnessed the revolution in Cryo-EM, sub-Ångström protein crystallography and microscopy, and finally, massive protein structure prediction on the genomic scale. Similar technological advances take place in other disciplines – I can only mention the unprecedented quality of language translation and text generation, generative models in image and video processing and automatic speech recognition. How do all these breakthroughs impact the future of structural biology and bioinformatics? At first glance, it may seem we will only need big data from now on to train deep-learning models, and then these models will answer all types of biological questions. However, the reality may not be so bright for the data science. Big data would not be available for many questions in hand, its interpretation would not be straightforward, and the biological community will continue assessing the quality of machine learning models with new experimental measurements.

What are the possible solutions when collecting large corpus of data is out of the question in the near future? My guess is – we will still use classical physics-based tools, or at least very strong physical and geometrical priors on statistical models. Somewhat ironically, the state-of-the-art crystallographic and cryo-EM data processing pipelines follow the opposite scenario. Indeed, they impose statistical priors and Bayesian inference to optimize the free parameters of the models built on experimental measurements.

Finally, as I have mentioned above, many classical questions in bioinformatics and biology have just been answered and we are ready to move toward new boundaries. For example, we seem to understand the structure and function of most globular single-domain proteins and even some of their stable assemblies. Now it is time to shift our main attention to multi-domain proteins, their complexes with other molecules, the role and function of weak transient interactions, and also highly flexible or even disordered macromolecules.

Protein flexibility and their observed structural heterogeneity is the question that has puzzled me for already some time. The problem is not easy – we do not have well-annotated data, it is very inhomogeneous, some observations are sparse, and classical models based on stochastic sampling or the theory of linear elasticity do not seem to fully explain experimental observations. In the future, I aim to combine my developments for physics-based predictions of protein motions with deep-learning architectures. The first goal would be fixing potential flaws in the parametrization of the physical models. Then I would like to go beyond simple physical descriptions, including linear elasticity, elastic network models, etc. I ultimately intend to learn novel mean-field motion laws that correspond to some complex physics that would not be practical to apply directly because of too many degrees of freedom in the system that need to be integrated over.

Thanks to the large collections of available protein models, we can now deepen our understanding of their organizational complexity and function. We can do it, e.g., by including in our models the effects of post-translational modifications (phosphorylation, acetylation, glycosylation, lipidation, etc.), non-covalent interactions with small molecules, lipids, RNAs, and DNAs, and also the effect of physiological environments, such as different levels of molecular concentration, pH, salt, and temperature. This shall ultimately allow us to go as far as whole-cell modeling and simulation at the atomic resolution. As whole-cell modeling still seems a rather far perspective, I nonetheless

intend to model proteins in crowded environments. Crowding is an essential physical phenomenon. It shall allow us to describe proteins' structure, dynamics, and function in physiological conditions. To study and parameterize the crowding effect, I will model a part of the Escherichia coli cytoplasm system, which has a rich experimental characterization. Then, I plan to model the crowded behavior of antibodies and some molecular motors, e.g., proteasome and dynein.

Another fascinating area of research is molecular (e.g., protein) design. We have seen steady progress in the experimental optimization of protein's affinity and thermostability, optimization of protein-protein interfaces, de-novo design of very rigid protein domains and then multi-component systems, and recent advances in the level of natural sequences' recovery. Currently, we can achieve levels of natural sequence recovery higher than 50%, which was not possible even several years ago when protein folds were optimized for their stability. The future challenges will be to transfer these successes to other types of molecules – RNAs, DNAs, peptides, and drug-like small molecules.

BIBLIOGRAPHY

[1] Mustapha Carab Ahmed, Ramon Crehuet, and Kresten Lindorff-Larsen. "Analyzing and comparing the radius of gyration and hydrodynamic radius in conformational ensembles of intrinsically disordered proteins." In: *bioRxiv* (2019), p. 679373.

[2] Mustapha Carab Ahmed, Ramon Crehuet, and Kresten Lindorff-Larsen. "Computing, analyzing, and comparing the radius of gyration and hydrodynamic radius in conformational ensembles of intrinsically disordered proteins." In: *Intrinsically Disordered Proteins*. Springer, 2020, pp. 429–445.

[3] Mustapha Carab Ahmed, Line K Skaanning, Alexander Jussupow, Estella A Newcombe, Birthe B Kragelund, Carlo Camilloni, Annette E Langkilde, and Kresten Lindorff-Larsen. "Refinement of α-synuclein ensembles against SAXS data: Comparison of force fields and methods." In: *Frontiers in molecular biosciences* 8 (2021), p. 654333.

[4] Sebastian E Ahnert, Joseph A Marsh, Helena Hernández, Carol V Robinson, and Sarah A Teichmann. "Principles of assembly reveal a periodic table of protein complexes." In: *Science* 350.6266 (2015), aaa2245.

[5] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. "Applying support vector machines to imbalanced datasets." In: *In Proceedings of the 15th European Conference on Machine Learning (ECML)*. 2004, pp. 39–50.

[6] FH Allen, DG Watson, L Brammer, AG Orpen, and R Taylor. "Typical interatomic distances: organic compounds." In: *International Tables for Crystallography Volume C: Mathematical, physical and chemical tables*. Springer, 2004, pp. 790–811.

[7] A. Amadei, A. B. Linssen, and H. J. Berendsen. "Essential dynamics of proteins." In: *Proteins: Struct., Funct., Genet.* 17.4 (1993), pp. 412–425.

[8] Andrea Amadei, Antonius B M Linssen, and Herman J C Berendsen. "Essential Dynamics of Proteins." In: *Proteins: Struct., Funct., Genet.* 17.4 (1993), pp. 412–425. DOI: 10.1002/prot.340170408. URL: https://doi.org/10.1002{\%}2Fprot.340170408.

[9] I. André, P. Bradley, C. Wang, and D. Baker. "Prediction of the structure of symmetrical protein assemblies." In: *PNAS* 104 (2007), pp. 17656–17661.

[10] Ingemar André, Charlie EM Strauss, David B Kaplan, Philip Bradley, and David Baker. "Emergence of symmetry in homooligomeric biological assemblies." In: *Proc Natl Acad Sci USA* 105.42 (2008), pp. 16148–16152.

[11] Christian B Anfinsen. "Principles that govern the folding of protein chains." In: *Science* 181.4096 (1973), pp. 223–230.

[12] Sylvain Arlot and Alain Celisse. "A survey of cross-validation procedures for model selection." In: *Statistics Surveys* 4 (2010), pp. 40–79.

[13] Svetlana Artemova, Sergei Grudinin, and Stephane Redon. "A Comparison of Neighbor Search Algorithms for Large Rigid Molecules." In: *J. Comput. Chem.* 32.13 (2011), pp. 2865–2877.

[14] Svetlana Artemova, Sergei Grudinin, and Stephane Redon. "A comparison of neighbor search algorithms for large rigid molecules." In: *Journal of Computational Chemistry* 32.13 (2011), pp. 2865–2877.

[15] A R Atilgan, S R Durell, R L Jernigan, M C Demirel, O Keskin, and I Bahar. "Anisotropy of Fluctuation Dynamics of Proteins with an Elastic Network Model." In: *Biophys. J.* 80.1 (2001), pp. 505–515. DOI: 10.1016/s0006-3495(01)76033-x. URL: https://doi.org/10.1016{\%}2Fs0006-3495{\%}2801{\%}2976033-x.

[16] Natalie Baddour. "Operational and convolution properties of three-dimensional Fourier transforms in spherical polar coordinates." In: *JOSA A* 27.10 (2010), pp. 2144–2155.

[17] Ivet Bahar, Chakra Chennubhotla, and Dror Tobi. "Intrinsic Dynamics of Enzymes in the Unbound State and Relation to Allosteric Regulation." In: *Curr. Opin. Struct. Biol.* 17.6 (2007), pp. 633–640. DOI: 10.1016/j.sbi.2007.09.011. URL: https://doi.org/10.1016{\%}2Fj.sbi.2007.09.011.

[18] Ivet Bahar, Timothy R Lezon, Ahmet Bakan, and Indira H Shrivastava. "Normal Mode Analysis of Biomolecular Structures: Functional Mechanisms of Membrane Proteins." In: *Chem. Rev.* 110.3 (2010), pp. 1463–1497. DOI: 10.1021/cr900095e. URL: https://doi.org/10.1021{\%}2Fcr900095e.

[19] Ahmet Bakan, Lidio M Meireles, and Ivet Bahar. "ProDy: Protein Dynamics Inferred from Theory and Experiments." In: *Bioinformatics* 27.11 (2011), pp. 1575–1577.

[20] Jacob B Bale, Shane Gonen, Yuxi Liu, William Sheffler, Daniel Ellis, Chantz Thomas, Duilio Cascio, Todd O Yeates, Tamir Gonen, Neil P King, et al. "Accurate design of megadalton-scale two-component icosahedral protein complexes." In: *Science* 353.6297 (2016), pp. 389–394.

[21] KE Banyard and NH March. "The electron distribution in the ammonium ion." In: *Acta crystallographica* 14.4 (1961), pp. 357–360.

[22] A. Berchanski and M. Eisenstein. "Construction of molecular assemblies via docking: modeling of tetramers with D_2 symmetry." In: *Proteins* 53 (2003), pp. 817–829.

[23] 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." In: *Nucleic Acids Res.* 28 (2000), pp. 235–242.

[24] Helen M Berman. "The protein data bank: a historical perspective." In: *Acta Crystallographica Section A* 64.1 (2008), pp. 88–95.

[25] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, TN Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. "The protein data bank." In: *Nucleic acids research* 28.1 (2000), pp. 235–242.

[26] Brady Bernard and Ram Samudrala. "A generalized knowledge-based discriminatory function for biomolecular interactions." In: *Proteins: Structure, Function, and Bioinformatics* 76.1 (2009), pp. 115–128.

[27] L. C. Biedenharn and J. C. Louck. *Angular Momentum in Quantum Physics*. Reading, MA: Addison-Wesley, 1981.

[28] Jolyon K Bloomfield, Stephen HP Face, and Zander Moss. "Indefinite integrals of spherical bessel functions." In: *arXiv preprint arXiv:1703.06428* (2017).

[29] Tom L Blundell and N Srinivasan. "Symmetry, stability, and dynamics of multidomain and multicomponent protein systems." In: *Proc Natl Acad Sci USA* 93.25 (1996), pp. 14243–14248.

[30] M. Bonomi, R. Pellarin, and M. Vendruscolo. "Simultaneous Determination of Protein Structure and Dynamics Using Cryo-Electron Microscopy." In: *Biophys. J.* 114.7 (Apr. 2018), pp. 1604–1613.

[31] Sandro Bottaro, Tone Bengtsen, and Kresten Lindorff-Larsen. "Integrating molecular simulation and experimental data: a Bayesian/maximum entropy reweighting approach." In: *Structural Bioinformatics*. Springer, 2020, pp. 219–240.

[32] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[33] Bernard Brooks and Martin Karplus. "Harmonic Dynamics of Proteins: Normal Modes and Fluctuations in Bovine Pancreatic Trypsin Inhibitor." In: *Proc. Natl. Acad. Sci. U.S.A.* 80.21 (1983), pp. 6571–6575.

[34] Rafael Brüschweiler. "Collective protein dynamics and nuclear spin relaxation." In: *J. Chem. Phys.* 102.8 (1995), pp. 3396–3403.

[35] Patrick Bryant, Gabriele Pozzati, Wensi Zhu, Aditi Shenoy, Petras Kundrotas, and Arne Elofsson. "Predicting the structure of large protein complexes using alphafold and sequential assembly." In: *bioRxiv* (2022).

[36] Ewen Callaway. "the Revolution Will Not Be Crystallized: A New Method Sweeps Through Structural Biology." In: *Nature* 525.7568 (2015), pp. 172–174. DOI: 10.1038/525172a. URL: http://dx.doi.org/10.1038/525172a.

[37] Ewen Callaway. "'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures." In: *Nature* 588 (2020), pp. 203–204. ISSN: 0028-0836.

[38] Claudio N. Cavasotto, Julio A. Kovacs, and Ruben A. Abagyan. "Representing Receptor Flexibility in Ligand Docking through Relevant Normal Modes." In: *J. Am. Chem. Soc.* 127.26 (2005), pp. 9632–9640.

[39] Myong-Ho Chae, Florian Krull, Stephan Lorenzen, and Ernst-Walter Knapp. "Predicting protein complex geometries with a neural network." In: *Proteins: Structure, Function, and Bioinformatics* 78.4 (2010), pp. 1026–1039.

[40] Sudeshna Chattopadhyay, Ahmet Uysal, Benjamin Stripe, Young-geun Ha, Tobin J Marks, Evguenia A Karapetrova, and Pulak Dutta. "How water meets a very hydrophobic surface." In: *Physical review letters* 105.3 (2010), p. 037803.

[41] Leonie Chatzimagas and Jochen S Hub. "Predicting solution scattering patterns with explicit-solvent molecular simulations." In: *arXiv preprint arXiv:2204.04961* (2022).

[42] Hanlin Chen, Yanzhao Huang, and Yi Xiao. "A simple method of identifying symmetric substructures of proteins." In: *Comput Biol Chem* 33.1 (2009), pp. 100–107.

[43] R. Chen and Z. Weng. "Docking unbound proteins using shape complementarity, desolvation, and electrostatics." In: *Proteins: Structure, Function, and Bioinformatics* 47.3 (2002), pp. 281–294.

[44] Vincent B Chen, W Bryan Arendall, Jeffrey J Headd, Daniel A Keedy, Robert M Immormino, Gary J Kapral, Laura W Murray, Jane S Richardson, and David C Richardson. "MolProbity: All-Atom Structure Validation for Macromolecular Crystallography." In: *Acta Crystallogr., Sect. D: Biol. Crystallogr.* 66.1 (2010), pp. 12–21.

[45] Yifan Cheng, Nikolaus Grigorieff, Pawel A Penczek, and Thomas Walz. "A primer to single-particle cryo-electron microscopy." In: *Cell* 161.3 (2015), pp. 438–449.

[46] Georgy Cheremovsky, Petr Popov, Deorgy Derevyanko, and Sergey Grudini. *Interaction parameters for the input set of molecular structures*. US Patent App. 15/529,774. 2017.

[47] Gwo-Yu Chuang, Dima Kozakov, Ryan Brenke, Stephen R Comeau, and Sandor Vajda. "DARS (Decoys As the Reference State) potentials for protein-protein docking." In: *Biophysical Journal* 95.9 (2008), pp. 4217–4227.

[48] S. R. Comeau and C. J. Camacho. "Predicting oligomeric assemblies: N-mers a primer." In: *Journal of Structural Biology* 150 (2004), pp. 233–244.

[49] C. Cortes and V. Vapnik. "Support-vector networks." In: *Machine Learning* 20.3 (1995), pp. 273–297.

[50] Luke Cuculis, Zhanar Abil, Huimin Zhao, and Charles M Schroeder. "Direct Observation of TALE Protein Dynamics Reveals a Two-State Search Mechanism." In: *Nat. Commun.* 6 (2015).

[51] M. I. Dauden, J. Martin-Benito, J. C. Sanchez-Ferrero, M. Pulido-Cid, J. M. Valpuesta, and J. L. Carrascosa. "Large Terminase Conformational Change Induced by Connector Binding in Bacteriophage T7." In: *J. Biol. Chem.* 288.23 (2013), pp. 16998–17007. DOI: 10.1074/jbc.m112.448951. URL: https://doi.org/10.1074%2Fjbc.m112.448951.

[52] Marc Delarue and Philippe Dumas. "On the use of low-frequency normal modes to enforce collective movements in refining macromolecular structural models." In: *Proc. Natl. Acad. Sci. U.S.A.* 101.18 (2004), pp. 6957–6962.

[53] G. Derevyanko, S. Grudinin, Y. Bengio, and G. Lamoureux. "Deep convolutional networks for quality assessment of protein folds." In: *Bioinformatics* 34.23 (2018), pp. 4046–4053. DOI: 10.1093/bioinformatics/bty494.

[54] Georgy Derevyanko and Sergei Grudinin. "HermiteFit : Fast-Fitting Atomic Structures into a Low-Resolution Density Map Using Three-Dimensional Orthogonal Hermite Functions." In: *Acta Crystallogr., Sect. D: Biol. Crystallogr.* 70.8 (2014), pp. 2069–2084. DOI: 10.1107/s1399004714011493. URL: http://dx.doi.org/10.1107/s1399004714011493.

[55] Robert Diamond. "A note on the rotational superposition problem." In: *Acta Crystallogr A* 44.2 (1988), pp. 211–216.

[56] T. Dietterich. "Overfitting and undercomputing in machine learning." In: *ACM Computing Surveys (CSUR)* 27.3 (1995), pp. 326–327.

[57] Sara E. Dobbins, Victor I. Lesk, and Michael J. E. Sternberg. "Insights into Protein Flexibility: the Relationship Between Normal Modes and Conformational Change upon Protein–Protein Docking." In: *Proc. Natl. Acad. Sci. USA* 105.30 (2008), pp. 10390–10395.

[58] Pemra Doruker, Ali Rana Atilgan, and Ivet Bahar. "Dynamics of Proteins Predicted by Molecular Dynamics Simulations and Analytical Approaches: Application to α-Amylase Inhibitor." In: *Proteins: Struct., Funct., Bioinf.* 40.3 (2000), pp. 512–524.

[59] Jan Drenth. *Principles of protein X-ray crystallography*. Springer Science & Business Media, 2007.

[60] Chaim Dryzun. "Continuous symmetry measures for complex symmetry group." In: *J Comput Chem* 35.9 (2014), pp. 748–755.

[61] Chaim Dryzun, Amir Zait, and David Avnir. "Quantitative symmetry and chirality—A fast computational algorithm for large structures: Proteins, macromolecules, nanotubes, and unit cells." In: *J Comput Chem* 32.12 (2011), pp. 2526–2538.

[62] Philippe Durand, Georges Trinquier, and Yves-Henri Sanejouand. "A New Approach for Determining Low-Frequency Normal Modes in Macromolecules." In: *Biopolymers* 34.6 (1994), pp. 759–771. DOI: 10.1002/bip.360340608. URL: http://dx.doi.org/10.1002/bip.360340608.

[63] Nathaniel Echols, Duncan Milburn, and Mark Gerstein. "MolMovDB: Analysis and Visualization of Conformational Change and Structural Flexibility." In: *Nucleic Acids Res.* 31.1 (2003), pp. 478–482. ISSN: 1362-4962.

[64] Arne Elofsson, Patrick Bryant, Gabriele Pozzati, Wensi Zhu, Aditi Shenoy, and Petras Kundrotas. "Predicting the structure of large protein complexes using AlphaFold and Monte Carlo tree search." In: (2022).

[65] U. Emekli, D. Schneidman-Duhovny, H. Wolfson, R. Nussinov, and T. Haliloglu. "HingeProt: Automated Prediction of Hinges in Protein Structures." In: *Proteins: Struct., Funct., Bioinf.* 70.4 (2008), pp. 1219–1227.

[66] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew W Senior, Timothy Green, Augustin Žídek, Russell Bates, Sam Blackwell, Jason Yim, et al. "Protein complex prediction with AlphaFold-Multimer." In: *BioRxiv* (2021).

[67] J. Eduardo Fajardo et al. "Assessment of chemical-crosslink-assisted protein structure modeling in CASP13." In: *Proteins: Structure, Function and Bioinformatics* 87.12 (2019), pp. 1283–1297. ISSN: 10970134. DOI: 10.1002/prot.25816.

[68] LA Feigin, Dimitrij I Svergun, and George W Taylor. *Structure analysis by small-angle X-ray and neutron scattering*. Springer, 1987.

[69] Giacomo Fiorin, Michael L Klein, and Jérôme Hénin. "Using collective variables to drive molecular dynamics simulations." In: *Mol. Phys.* 111.22-23 (2013), pp. 3345–3362.

[70] Sébastien Fiorucci and Martin Zacharias. "Binding site prediction and improved scoring during flexible protein–protein docking with ATTRACT." In: *Proteins: Struct., Funct., Bioinf.* 78.15 (2010), pp. 3131–3139.

[71] Giulia Fonti et al. "KAP1 is an antiparallel dimer with a functional asymmetry." In: *Life Science Alliance* 2.4 (2019). ISSN: 25751077. DOI: 10.26508/lsa.201900349.

[72] D Franke, MV Petoukhov, PV Konarev, A Panjkovich, A Tuukkanen, HDT Mertens, AG Kikhney, NR Hajizadeh, JM Franklin, CM Jeffries, et al. "ATSAS 2.8: a comprehensive data analysis suite for small-angle scattering from macromolecular solutions." In: *Journal of applied crystallography* 50.4 (2017), pp. 1212–1225.

[73] Joel Franklin, Patrice Koehl, Sebastian Doniach, and Marc Delarue. "MinActionPath: maximum likelihood trajectory for large-scale structural transitions in a coarse-grained locally harmonic energy landscape." In: *Nucleic Acids Res.* 35.suppl_2 (2007), W477–W482.

[74] RDB Fraser, TP MacRae, and E Suzuki. "An improved method for calculating the contribution of solvent to the X-ray diffraction pattern of biological molecules." In: *Journal of Applied Crystallography* 11.6 (1978), pp. 693–694.

[75] Elisa Frezza and Richard Lavery. "Internal normal mode analysis (iNMA) applied to protein conformational flexibility." In: *J. Chem. Theory Comput.* 11.11 (2015), pp. 5503–5512.

[76] Mu Gao, Davi Nakajima An, Jerry M Parks, and Jeffrey Skolnick. "AF2Complex predicts direct physical interactions in multimeric proteins with deep learning." In: *Nature communications* 13.1 (2022), pp. 1–13.

[77] David S Goodsell and Arthur J Olson. "Structural symmetry and protein function." In: *Annu Rev Bioph Biom* 29.1 (2000), pp. 105–153.

[78] *Google colab for accurate protein design by integrating structure prediction networks and diffusion generative models*. 2023. URL: https://colab.research.google.com/github/sokrypton/ColabDesign/blob/v1.1.1/rf/examples/diffusion_beta.ipynb.

[79] Christian Gorba, Osamu Miyashita, and Florence Tama. "Normal-mode flexible fitting of high-resolution structure of biological molecules toward one-dimensional low-resolution data." In: *Biophys. J.* 94.5 (2008), pp. 1589–1599.

[80] Melissa A Graewert and Dmitri I Svergun. "Impact and progress in small and wide angle X-ray scattering (SAXS and WAXS)." In: *Current opinion in structural biology* 23.5 (2013), pp. 748–754.

[81] J.J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C.A. Rohl, and D. Baker. "Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations." In: *Journal of Molecular Biology* 331.1 (2003), pp. 281–300.

[82] Alexander Grishaev, Liang Guo, Thomas Irving, and Ad Bax. "Improved fitting of solution X-ray scattering data to macromolecular structures and structural ensembles by explicit water modeling." In: *Journal of the American Chemical Society* 132.44 (2010), pp. 15484–15486.

[83] Sergei Grudinin, Maria Garkavenko, and Andrei Kazennov. "Pepsi-SAXS: an adaptive method for rapid and accurate computation of small-angle X-ray scattering profiles." In: *Acta Crystallogr., Sect. D: Biol. Crystallogr.* 73.5 (2017), pp. 449–464.

[84] Sergei Grudinin, Elodie Laine, and Alexandre Hoffmann. "Predicting protein functional motions: an old recipe with a new twist." In: *Biophys. J.* (2020).

[85] Sergei Grudinin and Stephane Redon. "Practical modeling of molecular systems with symmetries." In: *Journal of Computational Chemistry* 31.9 (2010), pp. 1799–1814. ISSN: 0192-8651.

[86] Sergei Grudinin, Maria Kadukova, Andreas Eisenbarth, Simon Marillet, and Frédéric Cazals. "Predicting Binding Poses and Affinities for Protein - Ligand Complexes in the 2015 D3R Grand Challenge Using a Physical Model with a Statistical Parameter Estimation." In: *J. Comput.-Aided Mol. Des.* 30.9 (2016), pp. 791–804. ISSN: 1573-4951. DOI: 10.1007/s10822-016-9976-2.

[87] Sergei Grudinin, Petr Popov, Emilie Neveu, and Georgy Cheremovskiy. "Predicting Binding Poses and Affinities in the CSAR 2013-2014 Docking Exercises Using the Knowledge-Based Convex-PL Potential." In: *J. Chem. Inf. Model.* 56.6 (2016), pp. 1053–1062. ISSN: 15205142. DOI: 10.1021/acs.jcim.5b00339.

[88] Nail A Gumerov, Konstantin Berlin, David Fushman, and Ramani Duraiswami. "A hierarchical algorithm for fast Debye summation with applications to small angle scattering." In: *Journal of computational chemistry* 33.25 (2012), pp. 1981–1996.

[89] Ivan Gushchin, Igor Melnikov, Vitaliy Polovinkin, and Others. "Mechanism of Transmembrane Signaling by Sensor Histidine Kinases." In: *Sience* ().

[90] J.P. Hansen and I.R. McDonald. *Theory of simple liquids*. Academic Press, 2006.

[91] David Harker. "The meaning of the average of— F— 2 for large values of the interplanar spacing." In: *Acta Crystallographica* 6.8-9 (1953), pp. 731–736.

[92] Steven Hayward and Nobuhiro Go. "Collective variable description of native protein dynamics." In: *Annu. Rev. Phys. Chem.* 46.1 (1995), pp. 223–250.

[93] Konrad Hinsen. "Analysis of Domain Motions by Approximate Normal Mode Calculations." In: *Proteins: Struct., Funct., Genet.* 33.3 (1998), pp. 417–429.

[94] Alexandre Hoffmann and Sergei Grudinin. "NOLB: Nonlinear Rigid Block Normal-Mode Analysis Method." In: *J. Chem. Theory Comput.* 13.5 (2017), pp. 2123–2134.

[95] Alexandre Hoffmann, Valérie Perrier, and Sergei Grudinin. "A novel fast Fourier transform accelerated off-grid exhaustive search method for cryo-electron microscopy fitting." In: *J. Appl. Crystallogr.* 50.4 (2017).

[96] Berthold KP Horn. "Closed-form solution of absolute orientation using unit quaternions." In: *J Opt Soc Am A* 4.4 (1987), pp. 629–642.

[97] Yang Hsia, Jacob B Bale, Shane Gonen, Dan Shi, William Sheffler, Kimberly K Fong, Una Nattermann, Chunfu Xu, Po-Ssu Huang, Rashmi Ravichandran, et al. "Design of a hyperstable 60-subunit protein icosahedron." In: *Nature* 535.7610 (2016), pp. 136–139.

[98] P.-S. Huang, J. J. Love, and S. L. Mayo. "Adaptation of a Fast Fourier Transform-Based Docking Algorithm for Protein Design." In: *Journal of Computational Chemistry* 26 (2005), pp. 1222–1232.

[99] S.Y. Huang and X. Zou. "An iterative knowledge-based scoring function for protein–protein recognition." In: *Proteins: Structure, Function, and Bioinformatics* 72.2 (2008), pp. 557–579.

[100] Greg L Hura, Angeli L Menon, Michal Hammel, Robert P Rambo, Farris L Poole Ii, Susan E Tsutakawa, Francis E Jenney Jr, Scott Classen, Kenneth A Frankel, Robert C Hopkins, et al. "Robust, high-throughput solution structural analyses by small angle X-ray scattering (SAXS)." In: *Nature methods* 6.8 (2009), pp. 606–612.

[101] Greg L Hura, Curtis D Hodge, Daniel Rosenberg, Dmytro Guzenko, Jose M Duarte, Bohdan Monastyrskyy, Sergei Grudinin, Andriy Kryshtafovych, John A Tainer, Krzysztof Fidelis, et al. "Small angle X-ray scattering-assisted protein structure prediction in CASP13 and emergence of solution structure differences." In: *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1298–1314.

[102] Howook Hwang, Brian Pierce, Julian Mintseris, Joël Janin, and Zhiping Weng. "Protein–protein docking benchmark version 3.0." In: *Proteins: Structure, Function, and Bioinformatics* 73.3 (2008), pp. 705–709.

[103] Howook Hwang, Thom Vreven, Joël Janin, and Zhiping Weng. "Protein–protein docking benchmark version 4.0." In: *Proteins: Structure, Function, and Bioinformatics* 78.15 (2010), pp. 3111–3114.

[104] Ilia Igashov, Nikita Pavlichenko, and Sergei Grudinin. "Spherical convolutions on molecular graphs for protein model quality assessment." In: *Machine Learning: Science and Technology* 2 (2021), p. 045005.

[105] Ilia Igashov, Kliment Olechnovič, Maria Kadukova, Česlovas Venclovas, and Sergei Grudinin. "VoroCNN: deep convolutional neural network built on 3D Voronoi tessellation of protein structures." In: *Bioinformatics* 37.16 (Feb. 2021), pp. 2332–2339. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab118. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/16/2332/39947203/btab118.pdf. URL: https://doi.org/10.1093/bioinformatics/btab118.

[106] T Iiyama, K Nishikawa, T Otowa, and K Kaneko. "An ordered water molecular assembly structure in a slit-shaped carbon nanospace." In: *The Journal of Physical Chemistry* 99.25 (1995), pp. 10075–10076.

[107] Andrii Ishchenko et al. "New Insights on Signal Propagation by Sensory Rhodopsin II/Transducer Complex." In: *Sci. Rep.* 7 (2017), p. 41811. DOI: 10.1038/srep41811. URL: https://hal.inria.fr/hal-01458744.

[108] Isak Johansson-Åkhe and Björn Wallner. "Benchmarking Peptide-Protein Docking and Interaction Prediction with AlphaFold-Multimer." In: *BioRxiv* (2021).

[109] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold." In: *Nature* 596 (2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2. URL: https://doi.org/10.1038%2Fs41586-021-03819-2.

[110] Maria Kadukova, Vladimir Chupin, and Sergei Grudinin. "Convex-PL R–Revisiting affinity predictions and virtual screening using physics-informed machine learning." In: *bioRxiv* (2021), pp. 2021–09.

[111] Maria Kadukova and Sergei Grudinin. "Knodle: A Support Vector Machines-Based Automatic Perception of Organic Molecules from 3D Coordinates." In: *J. Chem. Inf. Model.* 56.8 (2016), pp. 1410–1419. ISSN: 1549-960X. DOI: 10.1021/acs.jcim.5b00512.

[112] Maria Kadukova and Sergei Grudinin. "Convex-PL: a novel knowledge-based potential for protein-ligand interactions deduced from structural databases using convex optimization." In: *Journal of Computer-Aided Molecular Design* 31.10 (2017), pp. 943–958. ISSN: 15734951. DOI: 10.1007/s10822-017-0068-8.

[113] Maria Kadukova and Sergei Grudinin. "Docking of small molecules to farnesoid X receptors using AutoDock Vina with the Convex-PL potential: lessons learned from D3R Grand Challenge 2." In: *Journal of computer-aided molecular design* 32 (2018), pp. 151–162.

[114] Maria Kadukova, Karina dos Santos Machado, Pablo Chacón, and Sergei Grudinin. "KORP-PL: a coarse-grained knowledge-based scoring function for protein–ligand interactions." In: *Bioinformatics* 37.7 (2021), pp. 943–950.

[115] Kenshu Kamiya, Yoko Sugawara, and Hideaki Umeyama. "Algorithm for Normal Mode Analysis with General Internal Coordinates." In: *J. Comput. Chem.* 24.7 (2003), pp. 826–841. DOI: 10.1002/jcc.10247. URL: https://doi.org/10.1002%2Fjcc.10247.

[116] E. Karaca, A. S. J. Melquiond, S. J. de Vries, P. L. Kastritis, and A. M. J. J. Bonvin. "Building Macromolecular Assemblies by Information-driven Docking – Introducing the Haddock Multibody Docking Server." In: *Molecular & Cellular Proteomics* 9 (2010), pp. 1784–1794.

[117] Mikhail Karasikov, Guillaume Pagès, and Sergei Grudinin. "Smooth orientation-dependent scoring function for coarse-grained protein quality assessment." In: *Bioinformatics* 35.16 (Aug. 2019), pp. 2801–2808. DOI: 10.1093/bioinformatics/bty1037.

[118] M Karplus and J A McCammon. "Dynamics of Proteins: Elements and Function." In: *Annu. Rev. Biochem.* 52.1 (1983), pp. 263–300. DOI: 10.1146/annurev.bi.52.070183.001403. URL: https://doi.org/10.1146{\%}2Fannurev.bi.52.070183.001403.

[119] Noah Kassem, Raul Araya-Secchi, Katrine Bugge, Abigail Barclay, Helena Steinocher, Adree Khondker, Yong Wang, Aneta J Lenard, Jochen Bürck, Cagla Sahin, et al. "Order and disorder – An integrative structure of the full-length human growth hormone receptor." In: *Science Advances* 7.27 (2021), eabh3805.

[120] M. Kearns, Y. Mansour, A.Y. Ng, and D. Ron. "An experimental and theoretical comparison of model selection methods." In: *Machine Learning* 27.1 (1997), pp. 7–50.

[121] Simon K Kearsley. "On the orthogonal transformation used for structural comparisons." In: *Acta Crystallogr A* 45.2 (1989), pp. 208–210.

[122] Neil P King, William Sheffler, Michael R Sawaya, Breanna S Vollmar, John P Sumida, Ingemar André, Tamir Gonen, Todd O Yeates, and David Baker. "Computational design of self-assembling protein nanomaterials with atomic level accuracy." In: *Science* 336.6085 (2012), pp. 1171–1174.

[123] Neil P King, Jacob B Bale, William Sheffler, Dan E McNamara, Shane Gonen, Tamir Gonen, Todd O Yeates, and David Baker. "Accurate design of co-assembling multi-component protein nanomaterials." In: *Nature* 510.7503 (2014), pp. 103–108.

[124] Donald E. Knuth. "Computer Programming as an Art." In: *Communications of the ACM* 17.12 (1974), pp. 667–673.

[125] Patrice Koehl. "Minimum action transition paths connecting minima on an energy surface." In: *J. Chem. Phys.* 145.18 (2016), p. 184111.

[126] Dmitry A Kondrashov, Adam W Van Wynsberghe, Ryan M Bannen, Qiang Cui, and George N Phillips Jr. "Protein structural variation in computational models and crystallographic data." In: *Structure* 15.2 (2007), pp. 169–177.

[127] W. G. Krebs, V. Alexandrov, C. A. Wilson, N. Echols, H. Yu, and M. Gerstein. "Normal mode analysis of macromolecular motions in a database framework: developing mode concentration as a useful classifying statistic." In: *Proteins* 48.4 (2002), pp. 682–695.

[128] Kristin A Krukenberg, Timothy O Street, Laura A Lavery, and David A Agard. "Conformational dynamics of the molecular chaperone Hsp90." In: *Q. Rev. Biophys.* 44.2 (2011), pp. 229–255.

[129] H.W. Kuhn and A.W. Tucker. "Nonlinear programming." In: *Second Berkeley symposium on mathematical statistics and probability*. Vol. 1. 1951, pp. 481–492.

[130] Elodie Laine and Sergei Grudinin. "HOPMA: Boosting protein functional dynamics with colored contact maps." In: *The Journal of Physical Chemistry B* 125.10 (2021), pp. 2577–2588.

[131] Elodie Laine, Stephan Eismann, Arne Elofsson, and Sergei Grudinin. "Protein sequence-to-structure learning: Is this the end(-to-end revolution)?" In: *Proteins: Structure, Function, and Bioinformatics* 89.12 (2021), pp. 1770–1786. DOI: https://doi.org/10.1002/prot.26235. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26235. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.26235.

[132] Andreas Haahr Larsen, Yong Wang, Sandro Bottaro, Sergei Grudinin, Lise Arleth, and Kresten Lindorff-Larsen. "Combining molecular dynamics simulations with small-angle X-ray and neutron scattering data to study multi-domain proteins in solution." In: *PLoS computational biology* 16.4 (2020), e1007870.

[133] R. A. Laskowski, M. W. MacArthur, D. S. Moss, and J. M. Thornton. "PROCHECK: a program to check the stereochemical quality of protein structures." In: *J. Appl. Crystallogr.* 26.2 (Apr. 1993), pp. 283–291. ISSN: 1600-5767. DOI: 10.1107/s0021889892009944. URL: http://dx.doi.org/10.1107/s0021889892009944.

[134] Catherine L Lawson et al. "EMDataBank Unified Data Resource for 3DEM." In: *Nucleic Acids Res.* 44.D1 (2016), pp. D396–D403. ISSN: 1362-4962. DOI: 10.1093/nar/gkv1126.

[135] Catherine Lawson et al. "CryoEM Models and Associated Data Submitted to the 2015/2016 EMDataBank Model Challenge." In: *Zenodo* (Jan. 2018). https://doi.org/10.5281/zenodo.1165
DOI: 10.5281/zenodo.1165999. URL: https://doi.org/10.5281/zenodo.1165999.

[136] Y.J. Lee and O.L. Mangasarian. "RSVM: Reduced support vector machines." In: *Proceedings of the First SIAM International Conference on Data Mining*. 2001, pp. 00–07.

[137] M.F. Lensink et al. "Blind prediction of interfacial water positions in CAPRI." In: *Proteins: Structure, Function and Bioinformatics* 82.4 (2014). ISSN: 10970134. DOI: 10.1002/prot.24439.

[138] Cyrus Levinthal. "Mossbauer spectroscopy in biological systems." In: *Proceedings of a meeting held at Allerton House. P. Debrunner, JCM Tsibris, and E. Munck, editors. University of Illinois Press, Urbana, IL.* 1969.

[139] Michael Levitt, Christian Sander, and Peter S Stern. "the Normal Modes of a Protein: Native Bovine Pancreatic Trypsin Inhibitor." In: *Int. J. Quantum Chem.* 24.S10 (1983), pp. 181–199.

[140] Michael Levitt, Christian Sander, and Peter S Stern. "Protein Normal-Mode Dynamics: Trypsin Inhibitor, Crambin, Ribonuclease and Lysozyme." In: *J. Mol. Biol.* 181.3 (1985), pp. 423–447.

[141] Emmanuel D Levy and Sarah Teichmann. "Structural, evolutionary, and assembly principles of protein oligomerization." In: *Prog Mol Biol Transl* 117 (2013), pp. 25–51.

[142] Emmanuel D Levy, Jose B Pereira-Leal, Cyrus Chothia, and Sarah A Teichmann. "3D complex: a structural classification of protein complexes." In: *PLoS Comput Biol* 2.11 (2006), e155.

[143] Emmanuel D Levy, Elisabetta Boeri Erba, Carol V Robinson, and Sarah A Teichmann. "Assembly reflects evolution of protein complexes." In: *Nature* 453.7199 (2008), pp. 1262–1265.

[144] Guohui Li and Qiang Cui. "A Coarse-Grained Normal Mode Approach for Macromolecules: An Efficient Implementation and Application Ca^{2+}-ATPase." In: *Biophys. J.* 83.5 (2002), pp. 2457–2474. DOI: 10.1016/s0006-3495(02)75257-0. URL: http://dx.doi.org/10.1016/s0006-3495(02)75257-0.

[145] Erik Lindahl and Marc Delarue. "Refinement of Docked Protein–Ligand and Protein–DNA Structures Using Low Frequency Normal Mode Amplitude Optimization." In: *Nucleic Acids Res.* 33.14 (2005), pp. 4496–4506.

[146] Erik Lindahl, Cyril Azuara, Patrice Koehl, and Marc Delarue. "NOMAD-Ref: Visualization, Deformation and Refinement of Macromolecular Structures Based on All-Atom Normal Mode Analysis." In: *Nucleic Acids Res.* 34.2 (2006), W52–W56.

[147] Haiguang Liu, Alexander Hexemer, and Peter H Zwart. "The Small Angle Scattering ToolBox (SASTBX): an open-source software for biomolecular small-angle scattering." In: *Journal of Applied Crystallography* 45.3 (2012), pp. 587–593.

[148] Haiguang Liu, Billy K Poon, Augustus JEM Janssen, and Peter H Zwart. "Computation of fluctuation scattering profiles via three-dimensional Zernike polynomials." In: *Acta Crystallographica Section A: Foundations of Crystallography* 68.5 (2012), pp. 561–567.

[149] José R López-Blanco, Ruymán Reyes, José I Aliaga, Rosa M Badia, Pablo Chacón, and Enrique S Quintana-Ortí. "Exploring Large Macromolecular Functional Motions on Clusters of Multicore Processors." In: *J. Comput. Phys.* 246 (2013), pp. 275–288.

[150] José Ramón Lopéz-Blanco and Pablo Chacón. "iMODFIT: efficient and robust flexible fitting based on vibrational analysis in internal coordinates." In: *J. Struct. Biol.* 184.2 (2013), pp. 261–270.

[151] José Ramón Lopéz-Blanco, José Ignacio Garzón, and Pablo Chacón. "iMod: Multipurpose Normal Mode Analysis in Internal Coordinates." In: *Bioinformatics* 27.20 (2011), pp. 2843–2850.

[152] José Ramón López-Blanco, José I Aliaga, Enrique S Quintana-Ortí, and Pablo Chacón. "iMODS: internal coordinates normal mode analysis server." In: *Nucleic Acids Res.* 42.W1 (2014), W271–W276.

[153] Mingyang Lu, Billy Poon, and Jianpeng Ma. "A New Method for Coarse-Grained Elastic Normal-Mode Analysis." In: *J. Chem. Theory Comput.* 2.3 (2006), pp. 464–471. DOI: 10.1021/ct050307u. URL: https://doi.org/10.1021%2Fct050307u.

[154] DB Lukatsky, BE Shakhnovich, J Mintseris, and EI Shakhnovich. "Structural similarity enhances interaction propensity of proteins." In: *J Mol Biol* 365.5 (2007), pp. 1596–1606.

[155] J. Ma. "Usefulness and limitations of normal mode analysis in modeling dynamics of biomolecular complexes." In: *Structure* 13.3 (2005), pp. 373–380.

[156] Alexander D MacKerell, Nilesh Banavali, and Nicolas Foloppe. "Development and Current Status of the CHARMM Force Field for Nucleic Acids." In: *Biopolymers* 56.4 (2000), pp. 257–265.

[157] Robert J Marks II. *Handbook of Fourier analysis and its applications*. Oxford University Press, 2008.

[158] Joseph A Marsh, Sarah A Teichmann, and Julie D Forman-Kay. "Probing the diverse landscape of protein flexibility and binding." In: *Curr. Opin. Struct. Biol.* 22.5 (2012), pp. 643 –650. ISSN: 0959-440X. DOI: https://doi.org/10.1016/j.sbi.2012.08.008. URL: http://www.sciencedirect.com/science/article/pii/S0959440X1200142X.

[159] Erik W Martin, F Emil Thomasen, Nicole M Milkovic, Matthew J Cuneo, Christy R Grace, Amanda Nourse, Kresten Lindorff-Larsen, and Tanja Mittag. "Interplay of folded domains and the disordered low-complexity domain in mediating hnRNPA1 phase separation." In: *Nucleic acids research* 49.5 (2021), pp. 2931–2945.

[160] Efrat Maschiach, Ruth Nussinov, and Wolfson Haim. "FiberDock: Flexible Induced-Fit Backbone Refinement in Molecular Docking." In: *Proteins: Struct., Funct., Bioinf.* 78.6 (2010), pp. 1503–1519.

[161] Andreas May and Martin Zacharias. "Energy minimization in low-frequency normal modes to efficiently allow for global flexibility during systematic protein–protein docking." In: *Proteins: Struct., Funct., Bioinf.* 70.3 (2008), pp. 794–809.

[162] J Andrew McCammon and Martin Karplus. "the Dynamic Picture of Protein Structure." In: *Acc. Chem. Res.* 16.6 (1983), pp. 187–193. DOI: 10.1021/ar00090a001. URL: https://doi.org/10.1021{\%}2Far00090a001.

[163] R. Mendez and U. Bastolla. "Torsional network model: normal modes in torsion angle space better correlate with conformation changes in proteins." In: *Phys. Rev. Lett.* 104.22 (2010), p. 228103.

[164] Franci Merzel and Jeremy C Smith. "SASSIM: a method for calculating small-angle X-ray and neutron scattering and the associated molecular envelope from explicit-atom models of solvated proteins." In: *Acta Crystallographica Section D: Biological Crystallography* 58.2 (2002), pp. 242–249.

[165] Markus Mezger, Harald Reichert, Benjamin M Ocko, Jean Daillant, and Helmut Dosch. "Comment on "How Water Meets a Very Hydrophobic Surface"." In: *Physical review letters* 107.24 (2011), p. 249801.

[166] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. "ColabFold: making protein folding accessible to all." In: *Nature Methods* (2022), pp. 1–4.

[167] John BO Mitchell, Roman A Laskowski, Alexander Alex, and Janet M Thornton. "BLEEP—potential of mean force describing protein–ligand interactions: I. Generating potential." In: *Journal of computational chemistry* 20.11 (1999), pp. 1165–1176.

[168] Osamu Miyashita, José Nelson Onuchic, and Peter G Wolynes. "Nonlinear elasticity, proteinquakes, and the energy landscapes of functional transitions in proteins." In: *Proc. Natl. Acad. Sci. U.S.A.* 100.22 (2003), pp. 12570–12575.

[169] S. Miyazawa and R.L. Jernigan. "Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation." In: *Macromolecules* 18.3 (1985), pp. 534–552.

[170] Iain H. Moal and Paul A. Bates. "SwarmDock and the Use of Normal Modes in Protein-Protein Docking." In: *Int. J. Mol. Sci.* 11 (2010), pp. 3623–3648.

[171] Kevin B Murray, William R Taylor, and Janet M Thornton. "Toward the detection and validation of repeats in protein structure." In: *Proteins* 57.2 (2004), pp. 365–380.

[172] Diana Mustard and David W Ritchie. "Docking essential dynamics eigenstructures." In: *Proteins: Struct., Funct., Bioinf.* 60.2 (2005), pp. 269–274.

[173] Douglas Myers-Turnbull, Spencer E Bliven, Peter W Rose, Zaid K Aziz, Philippe Youkharibache, Philip E Bourne, and Andreas Prlić. "Systematic detection of internal symmetry in proteins using CE-Symm." In: *J Mol Biol* 426.11 (2014), pp. 2255–2268.

[174] J. Navaza. "Implementation of molecular replacement in *AMoRe*." In: *Acta Crystallographica* D57 (2001), pp. 1367–1372.

[175] Stefan Neumark. *Solution of cubic and quartic equations*. Elsevier, 2014.

[176] Emilie Neveu, David W. Ritchie, Petr Popov, and Sergei Grudinin. "PEPSI-Dock: A Detailed Data-Driven Protein-Protein Interaction Potential Accelerated by Polar Fourier Correlation." In: *Bioinformatics* 32.17 (2016), pp. i693–i701. ISSN: 14602059. DOI: 10.1093/bioinformatics/btw443.

[177] Emilie Neveu, Petr Popov, Alexandre Hoffmann, Angelo Migliosi, Xavier Besseron, Gregoire Danoy, Pascal Bouvry, and Sergei Grudinin. "RapidRMSD: Rapid determination of RMSDs corresponding to motions of flexible molecules." In: *Bioinformatics* 34.16 (Mar. 2018), pp. 2757–2765. DOI: 10.1093/bioinformatics/bty160. URL: https://hal.inria.fr/hal-01735214.

[178] Minh Khoa Nguyen, Léonard Jaillet, and Stéphane Redon. "Generating conformational transition paths with low potential-energy barriers for proteins." In: *J Comput Aid Mol Des* 32.8 (2018), pp. 853–867.

[179] Tosiyuki Noguti and Nobuhiro Gō. "Dynamics of Native Globular Proteins in Terms of Dihedral Angles." In: *J. Phys. Soc. Japan* 52.9 (1983), pp. 3283–3288.

[180] Laura Orellana, Ozge Yoluk, Oliver Carrillo, Modesto Orozco, and Erik Lindahl. "Prediction and validation of protein intermediate states from structurally rich ensembles and coarse-grained simulations." In: *Nat. Commun.* 7.1 (2016), pp. 1–14.

[181] E. Osuna, R. Freund, and F. Girosi. "An improved training algorithm for support vector machines." In: *Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop*. 1997, pp. 276–285.

[182] Guillaume Pagès, Benoit Charmettant, and Sergei Grudinin. "Protein model quality assessment using 3D oriented convolutional neural networks." In: *Bioinformatics* 35.18 (2019), pp. 3313–3319.

[183] Guillaume Pagès and Sergei Grudinin. "Analytical symmetry detection in protein assemblies. II. Dihedral and cubic symmetries." In: *Journal of Structural Biology* 203.3 (2018), pp. 185–194. ISSN: 1047-8477.

[184] Guillaume Pagès and Sergei Grudinin. "DeepSymmetry: using 3D convolutional networks for identification of tandem repeats and internal symmetries in protein structures." In: *Bioinformatics* 35.24 (2019), pp. 5113–5120. ISSN: 1367-4803.

[185] Guillaume Pagès and Sergei Grudinin. "AnAnaS: software for analytical analysis of symmetries in protein structures." In: *Protein Structure Prediction* (2020), pp. 245–257.

[186] Guillaume Pagès, Elvira Kinzina, and Sergei Grudinin. "Analytical symmetry detection in protein assemblies. I. Cyclic symmetries." In: *Journal of Structural Biology* 203.2 (2018), pp. 142–148. ISSN: 10958657. DOI: 10.1016/j.jsb.2018.04.004.

[187] Sanghyun Park, Jaydeep P Bardhan, Benoît Roux, and Lee Makowski. "Simulated x-ray scattering of protein solutions using explicit-solvent models." In: *The Journal of chemical physics* 130.13 (2009), p. 134114.

[188] W. Park, G. Leibon, D.N. Rockmore, and G.S. Chirikjian. "Accurate image rotation using Hermite expansions." In: *IEEE Transactions on Image Processing* 18.9 (2009), pp. 1988–2003.

[189] Francesco Pesce and Kresten Lindorff-Larsen. "Refining conformational ensembles of flexible proteins against small-angle x-ray scattering data." In: *Biophysical journal* 120.22 (2021), pp. 5124–5135.

[190] Francesco Pesce, Estella A Newcombe, Pernille Seiffert, Emil E Tranchant, Johan G Olsen, Birthe B Kragelund, and Kresten Lindorff-Larsen. "Assessment of models for calculating the hydrodynamic radius of intrinsically disordered proteins." In: *bioRxiv* (2022).

[191] Michel Petitjean. "On the root mean square quantitative chirality and quantitative symmetry measures." In: *J Math Phys* 40.9 (1999), pp. 4587–4595.

[192] Maxim V Petoukhov, Daniel Franke, Alexander V Shkumatov, Giancarlo Tria, Alexey G Kikhney, Michal Gajda, Christian Gorba, Haydyn DT Mertens, Petr V Konarev, and Dmitri I Svergun. "New developments in the ATSAS program package for small-angle scattering data analysis." In: *Journal of applied crystallography* 45.2 (2012), pp. 342–350.

[193] B. Pierce, W. Tong, and Z. Weng. "M-ZDOCK: a grid-based approach for C_n symmetric multimer docking." In: *Bioinformatics* 21 (2005), pp. 1472–1478.

[194] Brian G Pierce, Yuichiro Hourai, and Zhiping Weng. "Accelerating protein docking in ZDOCK using an advanced 3D convolution library." In: *PloS One* 6.9 (2011), e24657.

[195] Brian Pierce and Zhiping Weng. "ZRANK: reranking protein docking predictions with an optimized energy function." In: *Proteins: Structure, Function, and Bioinformatics* 67.4 (2007), pp. 1078–1086.

[196] Mark Pinsky, Chaim Dryzun, David Casanova, Pere Alemany, and David Avnir. "Analytical methods for calculating continuous symmetry measures and the chirality measure." In: *J Comput Chem* 29.16 (2008), pp. 2712–2721.

[197] Mark Pinsky, Amir Zait, Maayan Bonjack, and David Avnir. "Continuous symmetry analyses: Cnv and Dn measures of molecules, complexes, and proteins." In: *J Comput Chem* 34.1 (2013), pp. 2–9.

[198] Mark Pinsky, Amir Zait, Maayan Bonjack, and David Avnir. "Continuous symmetry analyses: Cnv and Dn measures of molecules, complexes, and proteins." In: *J Comput Chem* 34.1 (2013), pp. 2–9.

[199] John C Platt. "Fast training of support vector machines using sequential minimal optimization." In: *Advances in kernel methods*. MIT press. 1999, pp. 185–208.

[200] Roman Pogodin, Alexander Katrutsa, and Sergei Grudinin. "Quadratic Programming Approach to Fit Protein Complexes into Electron Density Maps." In: *Information Technology and Systems 2016*. Repino, St. Petersburg, Russia, 2016, pp. 576–582. URL: https://hal.inria.fr/hal-01419380.

[201] Frédéric Poitevin, Henri Orland, Sebastian Doniach, Patrice Koehl, and Marc Delarue. "AquaSAXS: a web server for computation and fitting of SAXS profiles with non-uniformally hydrated atomic models." In: *Nucleic acids research* 39.suppl 2 (2011), W184–W189.

[202] Massimiliano Pontil and Alessandro Verri. "Properties of support vector machines." In: *Neural Computation* 10.4 (1998), pp. 955–974.

[203] P. Popov and S. Grudinin. "Eurecon: Equidistant uniform rigid-body ensemble constructor." In: *Journal of Molecular Graphics and Modelling* 80 (2018), pp. 313–319. ISSN: 18734243. DOI: 10.1016/j.jmgm.2018.01.015.

[204] Petr Popov and Sergei Grudinin. "Rapid Determination of RMSDs Corresponding to Macromolecular Rigid Body Motions." In: *J. Comput. Chem.* 35.12 (2014), pp. 950–956. ISSN: 1096987X. DOI: 10.1002/jcc.23569.

[205] Petr Popov and Sergei Grudinin. "Knowledge of Native Protein-Protein Interfaces Is Sufficient to Construct Predictive Models for the Selection of Binding Candidates." In: *J. Chem. Inf. Model.* 55.10 (2015), pp. 2242–2255. ISSN: 15205142. DOI: 10.1021/acs.jcim.5b00372.

[206] Petr Popov, David W. Ritchie, and Sergei Grudinin. "DockTrina: Docking Triangular Protein Trimers." In: *Proteins: Struct., Funct., Bioinf.* 82.1 (2014), pp. 34–44. ISSN: 08873585. DOI: 10.1002/prot.24344.

[207] Petr Popov, Sergei Grudinin, Andrii Kurdiuk, Pavel Buslaev, and Stephane Redon. "Controlled-advancement rigid-body optimization of nanosystems." In: *Journal of Computational Chemistry* 40.27 (2019), pp. 2391–2399.

[208] William H Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes: The art of scientific computing. Third edition*. Cambridge university press, 2007.

[209] Andreas Prlić, Andrew Yates, Spencer E Bliven, Peter W Rose, Julius Jacobsen, Peter V Troshin, Mark Chapman, Jianjiong Gao, Chuan Hock Koh, Sylvain Foisy, et al. "BioJava: an open-source framework for bioinformatics in 2012." In: *Bioinformatics* 28.20 (2012), pp. 2693–2695.

[210] Ali Punjani and David J Fleet. "3D variability analysis: Resolving continuous flexibility and discrete heterogeneity from single particle cryo-EM." In: *Journal of structural biology* 213.2 (2021), p. 107702.

[211] Christopher D Putnam, Michal Hammel, Greg L Hura, and John A Tainer. "X-ray solution scattering (SAXS) combined with crystallography and computation: defining accurate macromolecular structures, conformations and assemblies in solution." In: *Q. Rev. Biophys.* 40.3 (2007), pp. 191–285.

[212] Robert P Rambo and John A Tainer. "Super-resolution in solution X-ray scattering and its applications to structural systems biology." In: *Annual review of biophysics* 42 (2013), pp. 415–441.

[213] Andrii Riazanov, Mikhail Karasikov, and Sergei Grudinin. "Inverse Protein Folding Problem Via Quadratic Programming." In: *Information Technology and Systems 2016*. Repino, St. Petersburg, Russia, 2016, pp. 561–568. URL: https://hal.inria.fr/hal-01419374.

[214] D. W. Ritchie. "High-order analytic translation matrix elements for real-space six-dimensional polar Fourier correlations." In: *J. Appl. Cryst.* 38 (2005), pp. 808–818.

[215] D. W. Ritchie and G. J. L. Kemp. "Protein Docking Using Spherical Polar Fourier Correlations." In: *Proteins* 39.2 (2000), pp. 178–194.

[216] D W Ritchie, D Kozakov, and S Vajda. "Accelerating and Focusing Protein-Protein Docking Correlations Using Multi-Dimensional Rotational {FFT} Generating Functions." In: *Bioinformatics* 24.17 (2008), pp. 1865–1873. DOI: 10.1093/bioinformatics/btn334. URL: http://dx.doi.org/10.1093/bioinformatics/btn334.

[217] David W. Ritchie and Sergei Grudinin. "Spherical Polar Fourier Assembly of Protein Complexes with Arbitrary Point Group Symmetry." In: *J. Appl. Crystallogr.* 49 (2016), pp. 158–167. ISSN: 16005767. DOI: 10.1107/S1600576715022931.

[218] Mette Ahrensback Roesgaard, Jeppe E Lundsgaard, Estella A Newcombe, Nina L Jacobsen, Francesco Pesce, Søren Lindemose, Andreas Prestel, Rasmus Hartmann-Petersen, Kresten Lindorff-Larsen, and Birthe B Kragelund. "Deciphering the alphabet of disorder – Glu and Asp act differently on local but not global properties." In: *bioRxiv* (2022).

[219] Peter W Rose, Andreas Prlić, Ali Altunkaya, Chunxiao Bi, Anthony R Bradley, Cole H Christie, Luigi Di Costanzo, Jose M Duarte, Shuchismita Dutta, Zukang Feng, et al. "The RCSB protein data bank: integrative view of protein, gene and 3D structural information." In: *Nucleic Acids Res* (2016), gkw1000.

[220] A. M. Roseman. "Docking structures of domains into maps from cryo-electron microscopy using local correlation." In: *Acta Crystallographica* D56 (2000), pp. 1332–1340.

[221] M. G. Rossmann. "The molecular replacement method." In: *Acta Crystallographica* A46 (1990), pp. 73–82.

[222] Dmitry Rykunov and András Fiser. "Effects of amino acid composition, finite size of proteins, and sparse statistics on distance-dependent statistical pair potentials." In: *Proteins: Structure, Function, and Bioinformatics* 67.3 (2007), pp. 559–568.

[223] D. Schneidman-Duhovny, Y. Inbar, R. Nussinov, and H. J. Wolfson. "PatchDock and SymmDock: servers for rigid and symmetric docking." In: *Nucleic Acids Research* 33 (2005), W363–W367.

[224] Dina Schneidman-Duhovny, Michal Hammel, and Andrej Sali. "FoXS: a web server for rapid computation and fitting of SAXS profiles." In: *Nucleic acids research* 38.suppl 2 (2010), W540–W544.

[225] Dina Schneidman-Duhovny, Ruth Nussinov, and Haim J. Wolfson. "Automatic Prediction of Protein Interactions with Large Scale Motion." In: *Proteins: Struct., Funct., Bioinf.* 69.4 (2007), pp. 764–773.

[226] Dina Schneidman-Duhovny, Michal Hammel, John A Tainer, and Andrej Sali. "Accurate SAXS profile computation and its assessment by contrast variation experiments." In: *Biophysical journal* 105.4 (2013), pp. 962–974.

[227] Gunnar F Schröder, Axel T Brunger, and Michael Levitt. "Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution." In: *Structure* 15.12 (2007), pp. 1630–1641.

[228] Gunnar F Schröder, Michael Levitt, and Axel T Brunger. "Super-resolution biomolecular crystallography with low-resolution data." In: *Nature* 464.7292 (2012), p. 1218.

[229] Georg E Schulz. "The dominance of symmetry in the evolution of homo-oligomeric proteins." In: *J Mol Biol* 395.4 (2010), pp. 834–843.

[230] Héctor Garcia Seisdedos. "Proteins Evolve on the Edge of Supramolecular Self-Assembly." In: *Biophys J* 112.3 (2017), 200a.

[231] Min-Yi Shen and Andrej Sali. "Statistical potential for assessment and prediction of protein structures." In: *Protein Science* 15.11 (2009), pp. 2507–2524.

[232] Edward SC Shih and Ming-Jing Hwang. "Alternative alignments from comparison of protein structures." In: *Proteins* 56.3 (2004), pp. 519–527.

[233] M.J. Sippl. "Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins." In: *Journal of Molecular Biology* 213.4 (1990), pp. 859–883.

[234] Guang Song and Robert L Jernigan. "An enhanced elastic network model to represent the motions of domain-swapped proteins." In: *Proteins: Struct., Funct., Bioinf.* 63.1 (2006), pp. 197–209.

[235] D C Sorensen. "Newton's Method with a Model Trust Region Modification." In: *SIAM Journal on Numerical Analysis* 19.2 (1982), pp. 409–426. DOI: 10.1137/0719026. URL: http://dx.doi.org/10.1137/0719026.

[236] Alessandro Spilotros and Dmitri I Svergun. "Advances in Small-and Wide-Angle X-ray Scattering SAXS and WAXS of Proteins." In: *Encyclopedia of Analytical Chemistry* (2014), pp. 1–34.

[237] V. Spiwok, P. Lipovova, and B. Kralova. "Metadynamics in essential coordinates: free energy simulation of conformational changes." In: *J. Phys. Chem. B* 111.12 (2007), pp. 3073–3076.

[238] M. Stepanova. "Dynamics of essential collective motions in proteins: theory." In: *Phys. Rev. E* 76.5 Pt 1 (2007), p. 051918.

[239] Kasper Stovgaard, Christian Andreetta, Jesper Ferkinghoff-Borg, and Thomas Hamelryck. "Calculation of accurate small angle X-ray scattering curves from coarse-grained protein models." In: *BMC bioinformatics* 11.1 (2010), p. 429.

[240] HEINRICH B Stuhrmann. "Interpretation of small-angle scattering functions of dilute solutions and gases. A representation of the structures related to a one-particle scattering function." In: *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 26.3 (1970), pp. 297–306.

[241] Heinrich B Stuhrmann. "Ein neues Verfahren zur Bestimmung der Oberflächenform und der inneren Struktur von gelösten globulären Proteinen aus Röntgenkleinwinkelmessungen." In: *Zeitschrift für Physikalische Chemie* 72.4_6 (1970), pp. 177–184.

[242] Karsten Suhre, Jorge Navaza, and Y-H Sanejouand. "NORMA: a tool for flexible fitting of high-resolution protein structures into low-resolution electron-microscopy-derived density maps." In: *Acta Crystallogr., Sect. D: Biol. Crystallogr.* 62.9 (2006), pp. 1098–1100.

[243] D Svergun, C Barberato, and MHJ Koch. "CRYSOL-a program to evaluate X-ray solution scattering of biological macromolecules from atomic coordinates." In: *Journal of applied crystallography* 28.6 (1995), pp. 768–773.

[244] DI Svergun. "Mathematical methods in small-angle scattering data analysis." In: *Journal of Applied Crystallography* 24.5 (1991), pp. 485–492.

[245] F. Tama and Y. H. Sanejouand. "Conformational change of proteins arising from normal mode calculations." In: *Protein Eng.* 14.1 (2001), pp. 1–6.

[246] Florence Tama, Osamu Miyashita, and Charles L Brooks III. "Flexible multi-scale fitting of atomic structures into low-resolution electron density maps with elastic network normal mode analysis." In: *J. Mol. Biol.* 337.4 (2004), pp. 985–999.

[247] Florence Tama, Osamu Miyashita, and Charles L Brooks III. "Normal Mode Based Flexible Fitting of High-Resolution Structure into Low-Resolution Experimental Data from Cryo-EM." In: *J. Struct. Biol.* 147.3 (2004), pp. 315–326. DOI: 10.1016/j.jsb.2004.03.002. URL: http://dx.doi.org/10.1016/j.jsb.2004.03.002.

[248] Florence Tama, Florent Xavier Gadea, Osni Marques, and Yves-Henri Sanejouand. "Building-Block Approach for Determining Low-Frequency Normal Modes of Macromolecules." In: *Proteins: Struct., Funct., Bioinf.* 41.1 (2000), pp. 1–7.

[249] Giorgio E Tamò, Luciano A Abriata, Giulia Fonti, and Matteo Dal Peraro. "Assessment of data-assisted prediction by inclusion of crosslinking/mass-spectrometry and small angle X-ray scattering data in the 12th Critical Assessment of protein Structure Prediction experiment." In: *Proteins: Structure, Function, and Bioinformatics* 86 (2018), pp. 215–227.

[250] Robert K-Z Tan, Batsal Devkota, and Stephen C Harvey. "YUP. SCX: coaxing atomic models into medium resolution electron density maps." In: *J. Struct. Biol.* 163.2 (2008), pp. 163–174.

[251] S. Tanaka and H.A. Scheraga. "Medium-and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins." In: *Macromolecules* 9.6 (1976), pp. 945–950.

[252] Mustafa Tekpinar. "Flexible fitting to cryo-electron microscopy maps with coarse-grained elastic network models." In: *Mol. Simulat.* (2018), pp. 1–9.

[253] F Emil Thomasen and Kresten Lindorff-Larsen. "Conformational ensembles of intrinsically disordered proteins and flexible multidomain proteins." In: *Biochemical Society Transactions* 50.1 (2022), pp. 541–554.

[254] F Emil Thomasen, Francesco Pesce, Mette Ahrensback Roesgaard, Giulio Tesei, and Kresten Lindorff-Larsen. "Improving the global dimensions of intrinsically disordered proteins in Martini 3." In: *bioRxiv* (2021).

[255] F Emil Thomasen, Francesco Pesce, Mette Ahrensback Roesgaard, Giulio Tesei, and Kresten Lindorff-Larsen. "Improving Martini 3 for Disordered and Multidomain Proteins." In: *Journal of Chemical Theory and Computation* 18.4 (2022), pp. 2033–2041.

[256] Monique M Tirion. "Large amplitude elastic motions in proteins from a single-parameter, atomic analysis." In: *Phys. Rev. Lett.* 77.9 (1996), p. 1905.

[257] L. Tong. "How to take advantage of non-crystallographic symmetry in molecular replacement: "locked" rotation and translation functions." In: *Acta Crystallographica D* D57 (2001), pp. 1383–1389.

[258] Chikashi Toyoshima, Shiho Iwasawa, Haruo Ogawa, Ayami Hirata, Junko Tsueda, and Giuseppe Inesi. "Crystal structures of the calcium pump and sarcolipin in the Mg 2+-bound E1 state." In: *Nature* 495.7440 (2013), pp. 260–264.

[259] Ahmet Uysal, Miaoqi Chu, Benjamin Stripe, Amod Timalsina, Sudeshna Chattopadhyay, Christian M Schlepütz, Tobin J Marks, and Pulak Dutta. "What x rays can tell us about the interfacial profile of water near hydrophobic surfaces." In: *Physical Review B* 88.3 (2013), p. 035431.

[260] Ilya A Vakser, Sergei Grudinin, Nathan W Jenkins, Petras J Kundrotas, and Eric J Deeds. "Docking-based long timescale simulation of cell-size protein systems at atomic resolution." In: *Proceedings of the National Academy of Sciences* 119.41 (2022), e2210249119.

[261] Erica Valentini, Alexey G Kikhney, Gianpietro Previtali, Cy M Jeffries, and Dmitri I Svergun. "SASBDB, a repository for biological small-angle scattering data." In: *Nucleic acids research* (2015), pp. D357–63.

[262] V. Vapnik. "Estimation of dependences based on empirical data." In: *Nauka* (1979).

[263] V. Vapnik. *The nature of statistical learning theory*. Springer, 1999.

[264] Vishwesh Venkatraman and David W. Ritchie. "Flexible protein docking refinement using pose-dependent normal mode analysis." In: *Proteins: Struct., Funct., Bioinf.* 80.9 (2012), pp. 2262–2274.

[265] Thom Vreven et al. "Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2." In: *J. Mol. Biol.* 427.19 (2015), pp. 3031–3041. ISSN: 1089-8638. DOI: 10.1016/j.jmb.2015.07.016.

[266] D Waasmaier and A Kirfel. "New analytical scattering-factor functions for free atoms and ions." In: *Acta Crystallographica Section A: Foundations of Crystallography* 51.3 (1995), pp. 416–431.

[267] Junmei Wang, Piotr Cieplak, and Peter A Kollman. "How Well Does a Restrained Electrostatic Potential (RESP) Model Perform in Calculating Conformational Energies of Organic and Biological Molecules?" In: *J. Comput. Chem.* 21.12 (2000), pp. 1049–1074.

[268] Max C Watson and Joseph E Curtis. "Rapid and accurate calculation of small-angle scattering profiles using the golden ratio." In: *Journal of Applied Crystallography* 46.4 (2013), pp. 1171–1177.

[269] Benjamin Webb and Andrej Sali. "Comparative Protein Structure Modeling Using MODELLER." In: *Curr Protoc Bioinformatics* 47 (2014), pp. 5.6.1–32. DOI: 10.1002/0471250953.bi0506s47.

[270] Dahlia R Weiss and Michael Levitt. "Can morphing methods predict intermediate structures?" In: *J. Mol. Biol.* 385.2 (2009), pp. 665–674.

[271] Robert Joseph Paton Williams. "the Conformation Properties of Proteins in Solution." In: *Biol. Rev.* 54.4 (1979), pp. 389–437.

[272] E Bright Wilson, J C Decius, and Paul C Cross. *Molecular Vibrations: The Theory of Infrared and Raman Spectra*. McGraw-Hill, 1955.

[273] Kurt Wüthrich. "The way to NMR structures of proteins." In: *Nature structural biology* 8.11 (2001), pp. 923–925.

[274] L. Yang, G. Song, and R. L. Jernigan. "How well can we understand large-scale protein motions using normal modes of elastic network models?" In: *Biophys. J.* 93.3 (2007), pp. 920–929.

[275] Sichun Yang, Sanghyun Park, Lee Makowski, and Benoît Roux. "A rapid coarse residue-based computational method for X-ray solution scattering characterization of protein folds and multiple conformational states of large protein complexes." In: *Biophysical journal* 96.11 (2009), pp. 4449–4463.

[276] Dingquan Yu, Grzegorz Chojnowski, Maria Rosenthal, and Jan Kosinski. "AlphaPulldown–a Python package for protein-protein interaction screens using AlphaFold-Multimer." In: *bioRxiv* (2022).

[277] Jian Zhang and Yang Zhang. "A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction." In: *PloS one* 5.10 (2010), e15386.

[278] Dmitrii Zhemchuzhnikov, Ilia Igashov, and Sergei Grudinin. "6DCNN with roto-translational convolution filters for volumetric data processing." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 4. 2022, pp. 4707–4715.

[279] Wenjun Zheng. "Accurate flexible fitting of high-resolution protein structures into cryo-electron microscopy maps using coarse-grained pseudo-energy minimization." In: *Biophys. J.* 100.2 (2011), pp. 478–488.

[280] Wenjun Zheng and Mustafa Tekpinar. "Accurate flexible fitting of high-resolution protein structures to small-angle x-ray scattering data using a coarse-grained model with implicit hydration shell." In: *Biophysical journal* 101.12 (2011), pp. 2981–2991.

[281] Ellen D Zhong, Tristan Bepler, Bonnie Berger, and Joseph H Davis. "CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks." In: *Nature methods* 18.2 (2021), pp. 176–185.

[282] Hongyi Zhou and Yaoqi Zhou. "Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction." In: *Protein Science* 11.11 (2009), pp. 2714–2726.

[283] Lei Zhou and Qinglian Liu. "Aligning experimental and theoretical anisotropic B-factors: water models, normal-mode analysis methods, and metrics." In: *J. Phys. Chem. B* 118.15 (2014), pp. 4069–4079.