Towards automated crystallographic structure refinement with a differentiable pipeline

Minhuan Li  
John A. Paulson School of Engineering and Applied Sciences  
Harvard University  
Cambridge, MA 02138  
minhuanli@g.harvard.edu

Doeke R. Hekstra  
Department of Molecular & Cellular Biology  
John A. Paulson School of Engineering and Applied Sciences  
Harvard University  
Cambridge, MA 02138  
doeke_hekstra@harvard.edu

Abstract

The lack of interfaces between crystallographic data and machine learning methods prevents the application of modern deep learning frameworks to crystal structure determination. Here we present SFcalculator, a differentiable pipeline to generate crystallographic observables (structure factors) from atomistic molecular structures and a bulk solvent model. This calculator fills the gap between the long-established crystallography field and state-of-the-art deep learning algorithms. We discuss the correctness and performance of SFcalculator by comparing with the current most-used tool Phenix. Finally, we demonstrate with an initial try that it enables automated structure refinement in a well-regularized latent space defined by a deep generative model, providing a principled way to impose prior knowledge. We believe this tool paves the way towards fully automated structure refinement and a possible end-to-end model, which is crucial for the next generation of high-throughput diffraction experiments.

1 Introduction

X-ray crystallographic structure refinement is the process of achieving agreement between an atomic model of a structure and the experimental data (structure factor amplitudes)[1]. This often is a complex procedure involving multiple strategies of model parameterization and optimization, which can possibly take days or weeks. Over the past few decades, significant progress on refinement methods has been made, featuring the availability and constant improvement of highly automated model building tools and streamlined software suites like CCP4[2], Phenix[3] and Coot[4]. Although efforts have been made to achieve the automated refinement[5], for most cases it is still inevitable to introduce human insight by manual model building. The development of bright new X-ray sources at synchrotrons and X-ray Free Electron Lasers (XFELs) and ongoing robotization are driving rapid increases in experimental throughput and now enable, for example, the collection of thousands of data sets in crystallographic drug fragment screens. Automated structure determination pipelines will be essential for efficient use of these new capabilities.

On the other hand, structure refinement can also be understood as a sampling problem of macromolecular structures constrained by both crystallographic data and physical priors. In addition
to traditional simulation-based conformational sampling methods, 3D structure sampling is under intense study using carefully-designed generative models\cite{6,7,8} in the deep learning field, and with notable successes \cite{9,10,11,12}. When it comes to the data-constrained sampling, some researchers have successfully constructed neural network models, like CryoDRGN\cite{13} and Cryo-VAE\cite{14}, to reconstruct protein structures from Cryo-EM images. However, to our knowledge, such work is still missing for crystallographic data. One reason is the lack of interfaces connecting crystallography data and modern machine learning methods. We will present our SFcalculator in Section \ref{sec:2} to fill the gap. The other reason lies in an intrinsic property of crystallography: the x-ray beam scattering reveals the Fourier transform of the model’s electron density, but only the magnitude (or amplitude) of the complex number is recorded. This is the "phase problem"\cite{15}: the phase of the complex number is lost. So more structural information is missing in crystallography compared than Cryo-EM, which will consequently require more prior knowledge. How to impose the prior makes the data-constrained sampling in crystallography hard. Currently most tools adopt geometric constraints\cite{16}, coming from geometry statistics of high quality data. An alternative approach is to use the potential energy calculated from a molecular mechanics force field as a more principled restraint\cite{17,18}, but still have some performance issue at the moment. In Section \ref{sec:3} we will propose a framework based on a Boltzmann Generator model\cite{6} as an alternative way to incorporate physical prior knowledge. Together, these two parts constitute an initial try towards machine-learning-assisted automated structure refinement in crystallography.

2 A Differentiable SFcalculator

Scattering relates to the Fourier transform of sample’s electron density map. Generally, both macromolecular atoms and solvent background will contribute to the process, and the solvent correction plays an significant role in the model building\cite{19}. The total structure factor can be defined as:

$$F\text{\textsubscript{model}} = F\text{\textsubscript{protein}} + F\text{\textsubscript{solvent}}$$ (1)

which consists of two parts, $F\text{\textsubscript{protein}}$ quantifies the contribution from protein atoms, and $F\text{\textsubscript{solvent}}$ serves for the solvent correction. The solvent contribution is often approximated using a so-called bulk-solvent mask correction\cite{20}, in which the space outside of the protein atoms is treated as uniformly distributed electron density, represented by a “solvent mask”. The relative contributions of protein and solvent are then balanced by a linear scale and an anisotropic factor $B\text{\textsubscript{sol}}$:

$$F\text{\textsubscript{model}} = k\text{\textsubscript{total}}(F\text{\textsubscript{protein}} + k\text{\textsubscript{solvent}}\exp(-\frac{B\text{\textsubscript{sol}}s^2}{4})F\text{\textsubscript{mask}})$$ (2)

Model quality in crystallography is usually measured by R factors:

$$R = \frac{\sum||F_{\text{obs}}| - |F_{\text{model}}||}{\sum|F_{\text{model}}|}$$ (3)

where $F_{\text{obs}}$ denotes the amplitudes observed from experiments and $F_{\text{model}}$ denotes the structure factor calculated from the model. R factors usually come with subscripts “work” and “free”, indicating training and validation set separately.

2.1 The Protein Contribution $F\text{\textsubscript{protein}}$

For the part of the structure with an explicit atomic representation, there are two ways to calculate $F\text{\textsubscript{protein}}$:

- Direct method: As the Fourier transform is a linear transformation, one can directly sum contribution from all atoms\cite{21}:

$$F\text{\textsubscript{protein}}(\vec{h}) = \sum_{G} \sum_{j} O_{j} \cdot f_{\vec{h},j} \cdot \text{DWF}(\vec{h}) \cdot \exp\left[2\pi i \vec{h} \cdot \left(\vec{R}_{G}\vec{x}_{j} + \vec{T}_{G}\right)\right]$$ (4)

where $G$ is the index of symmetry operations (appearing as rotation matrix $\vec{R}_{G}$ and translation vector $\vec{T}_{G}$ given the space group; $j$ is the atom index, $O_{j}$ is occupancy and $\vec{x}_{j}$ is the
Figure 1: Comparison between SFcalculator and Phenix. a) Statistics of pearson correlation between magnitude of $F_{protein}$ calculated by our SFcalculator and Phenix.fmodel. 907 PDB data used here. b) Statistics of pearson correlation between magnitude of $F_{mask}$ calculated by our SFcalculator and Phenix.fmodel. 907 PDB data used here. c) Visualization of real space solvent masks from two methods. PDBid: 3tsv. The whole unit cell is discretized into grid with size [54, 72, 90], here is the slice of $z = 20$. The two maps look generally similar but differ in high resolution details. The Pearson correlation between the two real space map is 0.83.

Fractional coordinates of atom $j$; $\vec{h}$ is the miller index, $f_{\vec{h},j}$ is the gaussian approximation for atomic scattering factor for the atom type of atom $j$, and DWF($\vec{h}$) is the debye-waller factor.

- FFT-based method: In this approach, one first calculates the overall electron density from the atom coordinates, followed by a discrete Fourier transform into reciprocal space. In this approach, the continuous integral will be replaced with discrete summation and the atomic density is usually truncated within a sphere.

Most current tools (like Phenix) by default use the FFT-based method, because the direct method scales with $N(\text{atoms}) \times N(\text{HKL})$ and can be slow for large $N(\text{atoms})$ and $N(\text{atoms})$. However, here we adopt the direct-method in our differentiable SFcalculator for two reasons:

1. Equation 4 is intrinsically differentiable, while in the FFT-based method, truncation around atom density requires extra care to realize differentiability.
2. With carefully vectorized codes and GPU acceleration, the direct-method can be very faster than both FFT-based and direct methods running on CPU by Phenix. See computation time comparison in Figure 2(a).
To validate our implementation of eq. 4, we computed the correlation between $F_{\text{protein}}$ from our SFcalculator and the Phenix FFT-based method. As shown in Figure 1(a), the correlation coefficient remains high across the resolution range. This is a convincing evidence that our $F_{\text{protein}}$ is correct.

2.2 Bulk Solvent Correction $F_{\text{mask}}$

A differentiable calculation of the solvent mask part is not trivial. Currently, the most popular approach is the probe-shrink method introduced in CNS[22, 23]. It uses van der Waals (or similar) atomic radii $r$ and two parameters $r_{\text{probe}}$ and $r_{\text{shrink}}$:

1. Discretize the real space unit cell into grid. Mark the region within radius $r + r_{\text{probe}}$ of each atom as non-solvent.
2. Shrink the non-solvent area by $r_{\text{shrink}}$. This step will eliminate small solvent islands.

The above approach involves non-differentiable operations like rounding to discretize. Here, we describe a differentiable way to approximate a solvent mask:

1. Do a discrete FFT on the Fprotein we obtained above, resulting in a protein electron density map in real space:

$$\text{protein map} = \text{FFT3d}(\text{Fprotein})$$

2. Define an electron density cutoff by the solvent volume percentage.

$$\text{cutoff} = \text{percentile}(\text{protein map}, \text{solvent percentage})$$

3. Apply a sigmoid function to set the mask in high-density regions to approximately 0 and low-density regions to 1:

$$\text{mask map} = \text{sigmoid}((\text{cutoff} - \text{protein map}) \times \text{scale})$$

usually choose scale as 50 to make the mask more binary.
4. Do an inverse discrete FFT on the mask map to output $F_{\text{mask}}$.

$$F_{\text{mask}} = \text{iFFT3d}(\text{mask map})$$
Figure 3: Framework of automated refinement in latent space. We pre-train a deep generative model with physics prior knowledge, encoding a good initial model into the latent space, then do optimization with the experimental likelihood loss.

The pipeline to calculate $F_{\text{mask}}$ above is fully differentiable and fast. However, our results differs from the Phenix calculation at high resolution, as shown in Figure 1(b). The correlation of $F_{\text{mask}}$ is high at low resolution but quickly decays when it goes to the resolution < 6 Angstrom. This is also clear from visualization of the solvent mask in Figure 1(c). The two maps look generally the same but their high-resolution details are different. However, the contribution of bulk solvent is significant at low resolution (> 8 Angstrom) and becomes weaker at middle and high resolution [20]. As shown in Figure 2(b), this results in slightly worse R factors. However, as shown in the next section, the optimization trajectories demonstrate that this does not preclude optimization.

3 Boltzmann-Generator-Based Refinement

In this section, we present an initial try to connect our SF calculator with deep generative models and do automated structure refinement. As shown in Figure 3, the general framework consists of a pre-trained generative model, combined with SF calculator to connect the atomic model to experimental data. We chose the Boltzmann generator [6] as our first such generative model as its exact likelihood computation enables two resources of physical knowledge during training, one is the likelihood of training data, the other is the potential energy in the real space.

To generate a set of conformations for training of the Boltzmann generator, we ran a single 100ns molecular dynamics (MD) simulation (step size equals 2fs, typically takes around 2-5 hours with OpenMM [24]) started from an AlphaFold [10] prediction generated via ColabFold [25]. We then used these MD conformations along with self-generated samples evaluated by the force field to train the Boltzmann Generator as described by [6]. As shown in Figure 4(a), after training, samples drawn from the latent prior of the Boltzmann generator correspond to structures with energy close to the MD samples. The merit of this approach is that sufficiently close to the origin of this latent space points are likely to correspond to physically reasonable structure. In addition, the lossless encoding of the Boltzmann Generator ensures that we can start from a good initial guess.

By connecting the pre-trained generative model to the SF calculator, we can navigate in the latent space by minimizing the crystallographic loss:

$$T_{\text{data}} = \sum_h \left( \frac{F_{\text{obs}}^h - |F_{\text{model}}^h|}{\sigma_F^h} \right)^2$$

where $h$ is the miller index in the working set, $\sigma$ is the observed standard deviation. To illustrate the progress of structure refinement, we show an example of optimization trajectory in Figure 4(b) with model system PDB 3tsv. Initial model is the still AlphaFold2 prediction. As a benchmark, we used the PDB deposited model (PDB ID 3tsv, previously refined in Phenix) stripped small molecules and explicit waters to make sure the force field could work. As the objective loss went down, the R factors in the test set as a measurement of model quality was also going better, resulting in a model comparable to the benchmark.
4 Conclusion

In this work, we presented a differentiable SFcalculator to allow direct optimization of atomic models against crystallographic data using machine learning methods. And we show its usefulness by connecting it to a pre-trained Boltzmann Generator and perform an automated refinement. We believe this tool could fill the gap between crystallography field and the fast evolving deep learning field, especially generative molecular models, and pave the path towards better data-constrained sampling models.

5 Code Availability

SFcalculator is available in Tensorflow2, Pytorch and Jax on Github: [https://github.com/Hekstra-Lab/SFcalculator](https://github.com/Hekstra-Lab/SFcalculator)

6 Acknowledgements

The authors thank Kevin Dalton for helpful conversations and Harrison Wang for initial tests of the solvent mask. This work was supported by an NIH Director’s New Innovator Award (DP2-GM141000) to D.R.H. Funding to support this research was provided for M.H.L. by the Eric and Wendy Schmidt Center.

References


