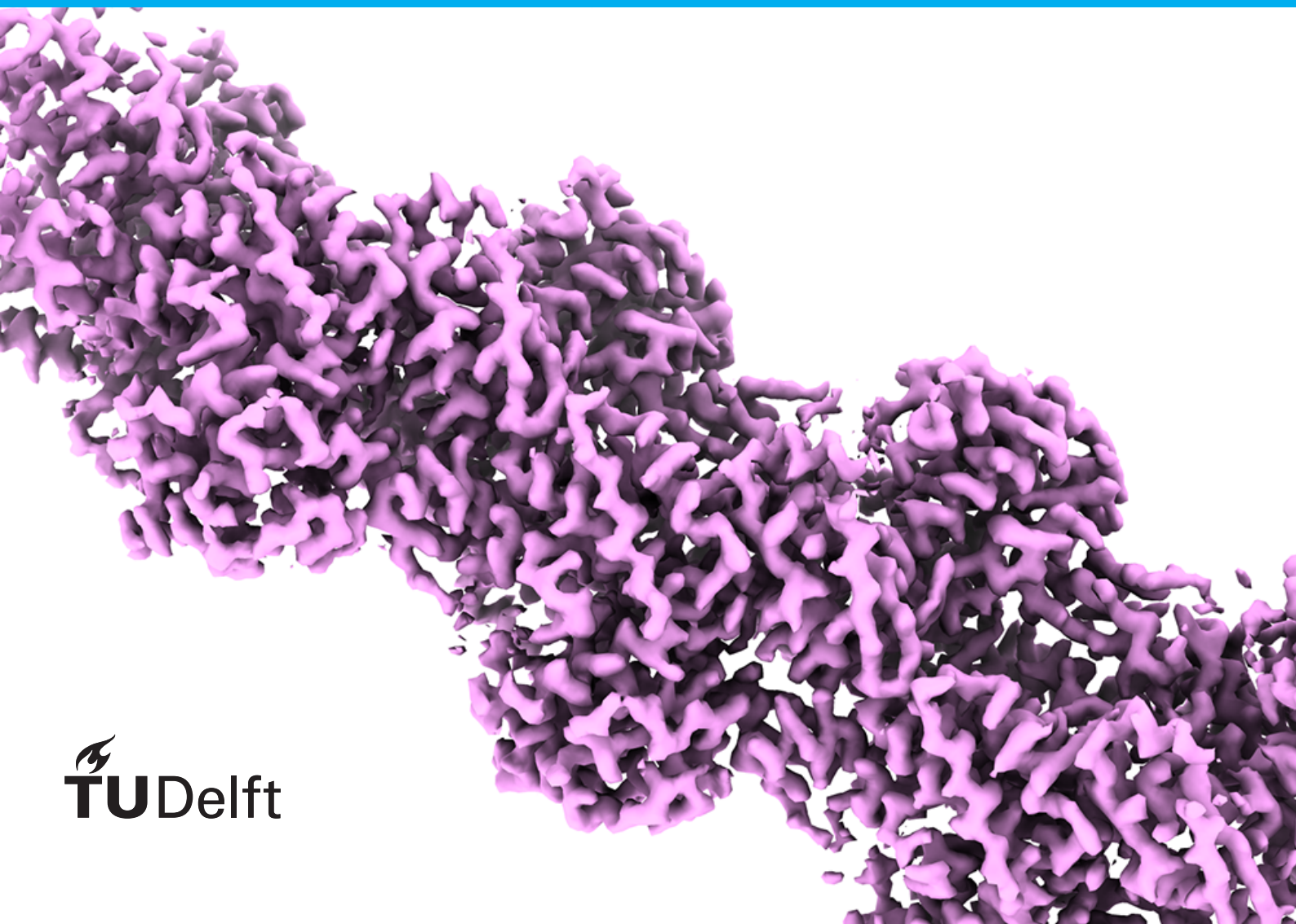
iLocScale

Cryo-EM map sharpening by learning scattering properties of macromolecules

I.S. Bot

Bachelor thesis
Applied Mathematics
& Applied Physics



TU Delft

iLocScale

Cryo-EM map sharpening by learning scattering properties of macromolecules

by

I.S. Bot

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Tuesday March 7, 2023 at 15:00 PM.

Student number: 5072689
Project duration: February 7, 2022 – March 7, 2023
Thesis committee:

Asst. Prof. dr. ir. A. Jakobi,	TU Delft, supervisor
Dr. ir. H. Kekkonen,	TU Delft, supervisor
Assoc. Prof. Dr. ir. J. Hoogenboom,	TU Delft, assessment committee (AP)
Dr. ir. Y. van Gennip,	TU Delft, assessment committee (AM)

This thesis is confidential and cannot be made public until February 28, 2023.

TU Delft

Preface

The research presented in this thesis concludes my journey for the Bachelor of Applied Mathematics (BSc) and Bachelor of Applied Physics (BSc) at the Delft University of Technology in Delft (TU Delft).

Over the course of this research, I have had the opportunity to learn from and work with some of the smartest people in the field of map-sharpening at a lab that welcomed me with open arms. I'm immensely thankful for their guidance, support and encouragement that has been invaluable in developing my understanding of this topic and teaching me both how to work with the cutting-edge in machine learning as well as important hard skills like coding as part of a team.

I would like to take this opportunity to thank Dr. ir. A. Jakobi and Dr.ir. H. Kekkonen for their supervision for this project. The members of the thesis committee for their feedback. Lastly I would like to especially thank ir. A. Bharadwaj for his help and guidance during this project.

I hope that the conclusion of this thesis will contribute to the ongoing development of map sharpening and inspire further research specifically in machine learning based map-sharpening.

I.S. Bot
Delft, February 28, 2023

Contents

1

Abstract

Cyo-EM is a powerful technique in biophysics to model macromolecules such as proteins or viruses with near-atomic resolution, by freezing a sample and passing a beam of electrons through the sample forming a 2D image of the sample's Coulomb potential. By recording tens of thousands of these images and using computational methods to form a 3D density map.

The interpretability of these cryo-EM maps decreases in high-resolution areas as a result of loss of contrast as a result of technical limitations in the current methods used for capturing cryo-EM maps as well as limitations imposed by the laws of physics. This results in a loss of contrast, by using map sharpening to increase these areas of low contrast locally, which is indicated by a decrease in intensity in the Fourier domain. It is possible to locally sharpen the cryo-EM map. As a result, interpretability can be improved. While global sharpening methods would result in oversharpening of some areas should be and undersharpening in others, using a local sharpening method can adjust for a variation in contrast within a map. This resulted in the development of LocScale[13]. Which is a local sharpening method using a (pseudo) atomic model.

The main drawback of using a local sharpening method which would require an atomic model is the fact that it is impractical in cases where there is no model available as well as requiring atomic model refinement for sharpening, which takes a lot of resources. This may, for example, be the case of a newly discovered protein. In such a case it may be desirable to make use of machine learning to sharpen the map for its effectiveness in new and unknown situations where there would be a lack of an atomic model. The main drawback of just using a machine learning-based approach for predicting the fully sharpened map is its black-box characteristic which makes it hard to understand what the program is doing and how its sharpening and if it may for example be over- or under-sharpening some regions. Therefore it would be desirable to use ML in a way where the black-box characteristic is minimized as well as solving the problem of hallucinations.

While other machine learning methods are able to be used to sharpen EM-maps. When decomposing the sharpened maps in Fourier space, it often becomes clear that not only the amplitudes but also the phases, instead of just the amplitudes of the EM-map have been changed. Resulting in hallucinations appearing, and ML sharpening methods are often implemented in such a way that the entire method is housed inside a black box, making it challenging to understand why certain parts of the macromolecule are sharpened.

Using a machine learning-based approach to predicting local radial profiles it is possible to limit the use of machine learning to only one step, the step for determining the radial profiles, compared to the model-based method, LocScale. And use all other steps which would normally be associated with LocScale, vastly reducing the use of ML. Since the radial profiles describe the amplitudes of the EM-map it is possible to change these while leaving the phases unchanged. Reducing the chance of hallucinations appearing to zero.

In this thesis, it is shown that it is possible to produce a machine learning method (named iLocScale) to effectively sharpen EM maps by predicting radial profiles. This resulted in a method which is almost as sharp as their model-based (target) counterpart (LocScale) in the modelled region and vastly outperforms LocScale in the unmodelled regions for sharpness. While also offering a large improvement in correlation coefficient for local resolution and B-factor between the iLocScale sharpened maps (0.3416) and unsharpened maps (0.0465, $\Delta = 0.3416$). Moreover, iLocScale vastly outperforms LocScale in unmodeled regions. It also takes approximately 75% less time.

2

Introduction

In January of 2023, this year has already been named the year of Artificial Intelligence (AI) where the question is not whether AI will impact our lives. But how it will be applied to impact our lives?

One of these areas is biophysics specifically, the shape and behaviour of proteins, which are one of the fundamental building blocks of life on earth and as vital to human life as water and oxygen. Forming both the building blocks of deoxyribonucleic acid (DNA), the code containing the instructions for the development and function of all organisms on earth. As well as viruses and more complex macroscopic body parts like muscles.

Since the function of a protein is for a large part influenced by the shape as well as the amino acids it's built from, making it critical to have an effective method of capturing both the shape and the components of proteins. Because it helps us understand ourselves better as well as it helps us understand the threats to our environment such as viruses better and how to produce effective medicine to protect ourselves from these dangers.

A cryogenic electron microscope (cryo-EM) is one of the methods used to capture the structure of proteins. By taking tens of thousands of pictures of a protein at a very cold temperature (which minimises movement from the protein). Resulting in many two-dimensional (2D) images showing the Coulomb potential of a 2D image of the protein. By capturing many of these 2D slices by postprocessing, stitching them back together and forming a three-dimensional (3D) map showcasing the entire protein with near-atomic resolution. The drawback of using this method is that the map may suffer from interpretability issues in the areas where contrast has been lost as a result of imperfect averaging and damage caused by the electrons among other factors. Mapsharpening is used to increase interpretability. One of the possible approaches is to increase the contrast that has been lost. This loss in contrast can be found when the map is decomposed in Fourier space. By increasing this contrast it is possible to increase the interpretability of the map and thus predict useful information about the functions of the protein.

The current methods for map sharpening, including methods like LocScale, often use atomic or pseudo-atomic models. Which makes them perfect for sharpening maps of which the atomic structure is already (partially) known. A major downside is that these methods are worse or incompatible for sharpening protein maps where the atomic structure is unknown. Or where the atomic structure has not been modelled. For example in cases where proteins contain lipid belts that are not modelled. Resulting in the suppression of the lipid belt using model-based sharpening methods even though these lipid belts do in reality exist. And are part of the captured EM-map.

Leveraging the strengths of machine learning (ML) it may be possible to use sharpening methods used for maps where the atomic structure is already (partially) known to train a model that would be able to sharpen protein maps in cases where the atomic structure is unknown. The main drawback of using machine learning to sharpen maps would be the black-box characteristic of machine learning making it hard to distinguish if the sharpened map forms a good estimate of the data represented. Which may result in inaccurate predictions. A method to decrease this black-box characteristic would be to make use of machine learning as little as possible and in a method where it would be easy to quantify how accurate the given predictions are.

Another risk when working with ML is its dependency on high-quality data. When it is trained on inaccurate data, the program might learn the relation between the questions and answers. But this might not be the desired relation. A clear example of this is when a program is given model-based sharpened maps as an

answer and tries to find the relation from the unsharpened maps. This would yield a method which would suppress the lipid belts like a model-based sharpening method would and only sharpens the modelled regions. As can be seen from the results of DeepEMhacer[7], an ML method using LocScale sharpened maps with muted lipid belts.

While other machine learning methods, like DeepEMhacer[7], often change the properties of the entire EM-map risking an output map where the phases have been affected. This could result in hallucinations appearing in the map that should not be present. By only editing the radial profiles (and thus only touching the amplitudes and leaving the phases unchanged) it might be possible to sharpen using machine learning without the risk of hallucinations appearing in the final sharpened map.

Decreasing this black-box characteristic may be possible by taking a pre-existing method and substituting the step which would require a (pseudo) atomic model using an ML-based approach. In this paper, we will look at substituting the step in LocScale which would require an atomic model where the atomic model would be refined to acquire the optimal radial profiles and B-factors. By using machine learning, LocScale uses an atomic model to predict local B-factors to model an optimal radial profile from the radial profile given by the unsharpened map. ML can be used to predict a radial profile normally produced using an atomic model, by using these radial profiles created by the atomic model as a target.

The core question of this research is: how can machine learning be introduced in a model-based sharpening method to produce a model-free sharpening method while leaving the phases intact?

Where this question has to be answered with regards to minimizing the number of steps where machine learning is required and where the model-based sharpening method is transformed into a model-free method using ML.

This thesis is structured such that in the Theory 3 chapter a basis will be built forming an explanation of the relevant background. Then in the Approach 4, the roadmap of this research will be explained and the criteria used for evaluating the results will be given. Then in the Results 5, the resulting parameters, theoretical capabilities as well as practical applications of the implementation will be shown. After which in the Discussions 6 the limitations of the proposed method will be brought to light ending in the Conclusion 7. Additional data can be found in the Appendix A.

In this thesis, grey EM maps refer to unsharpened maps. While yellow refers to LocScale sharpened maps and pink refers to iLocScale sharpened maps (with merged radial profiles from both linear and non-linear models). Green is used to referring to iLocScale using a non-linear model. While orange refers to iLocScale using a linear model.

This research has been carried out at Jakobi Lab part of the research department of Bionanoscience at the Delft University of Technology. Under the supervision of Arjen Jakobi and Hanne Kekkonen as well as the guidance of Alok Bharadwaj. This thesis forms the conclusion of both the AM3000 Bachelorproject and TWN3002-16 Bachelor Project AMP. For the BSc thesis in Applied Mathematics and BSc thesis in Applied Physics.

3

Theory

3.1. Cryogenic electron microscopy (cryo-EM)

Cryo-electron microscopy (cryo-EM) is an electron microscopy method that is widely used to study the structure of biological molecules like proteins at a near-atomic resolution. The entire pipeline of getting a 3D map and 3D model from a sample is illustrated in figure 3.1.

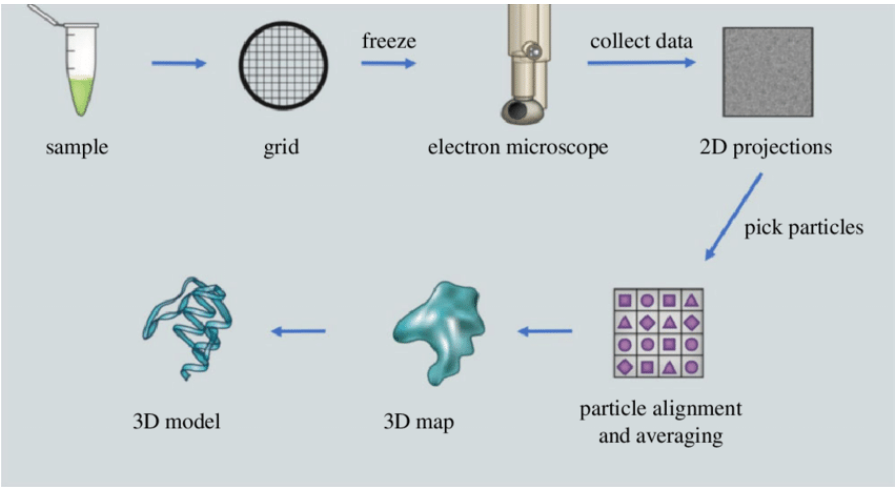


Figure 3.1: Schematic showcasing the cryo-EM pipeline (Hey 2020 [11]). First, a sample is placed on a grid, after which it is flash-frozen to preserve the protein's state. After this 2D images are collected using an electron microscope. The set of 2D images is processed to form a 3D map.

Its first step involves taking a sample and freezing it at a cold temperature thereby preserving the structure of the sample and protecting it from the impact of electrons later on. Then the sample is loaded into the electron microscope where electrons are used to image the Coulomb potential of the sample. Note that this results in a two-dimensional (2D) image, by taking 2D snapshots of the sample. By taking many of these images and stitching them together using post-processing methods to form a 3D map of the protein (the picking particles and particle alignment and averaging step in figure 3.1). This 3D map is made up of 3D pixels called voxels. Which provides information about the protein's structure by displaying the electron potential at each point. However, the 3D map generated using cryo-EM is not a perfect reflection of reality. As a result of a combination of factors like the physical limits of physics. Sample preparation and the accuracy of the microscope. Inaccuracies are introduced during the post-processing steps when going from thousands of 2D images to a 3D map or from averaging over multiple images at the same point. This results in a loss of contrast in the map. Which reduces its interpretability.

3.2. Fourier Transform

Since a Cryo-EM map is mathematically speaking just an image extended to 3D, it can be transformed using the same methods. In this thesis, the image is influenced by transformations in the Fourier domain. To get to the Fourier domain from a 3D image in real space with size $N \times N \times N$, the discrete Fourier transform in formula 3.1 can be used.

$$F(k,l,m) = \frac{1}{N^3} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(a,b,c) \exp\left[-i2\pi\left(\frac{ka}{N} + \frac{lb}{N} + \frac{mc}{N}\right)\right] \tag{3.1}$$

Where $f(a,b,c)$ represents the image in the spatial domain. Allowing for the image to be decomposed into sine waves described by given amplitudes and phases. An example of an image decomposed is shown in figure 3.2. The shown example is in 2 dimensions but this can easily be extended to 3 dimensions as would be the case with a protein map and as described in equation (3.1).

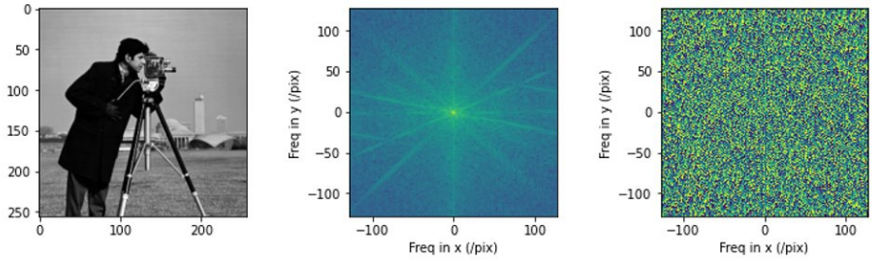


Figure 3.2: Example of decomposing a 2D image (left) into a sine waves and phases plotted in Fourier space, amplitudes of the sine waves are shown in the middle image. Phases shown in the right image. Sourced image[12].

Although equation (3.1) might look usable, it actually takes very many steps to calculate. As just a one-dimensional Fourier transform requiring $\mathcal{O}(N^2)$ operations (this is a result of the fact that $\vec{y} = F_N \vec{x}$ is used to denote the transformation from the time domain ($\vec{x}$, vector of length N) to the frequency component ($\vec{y}$, also a vector of length N) where the transformation is described by an $N \times N$ matrix F resulting in $\mathcal{O}(N^2)$ required operations). While a fast Fourier transform, like the Cooley-Turkey algorithm[5] which is the algorithm, used when making use of the fast Fourier transform in NumPy[9], a Python library (built primarily on C) is able to do it in just $\mathcal{O}(Nlog_2(N))$ operations. It leverages the simple, yet powerful observation that if we define $w_N^{ka} = \exp[-2i\pi(ka/N)]$, it would hold that $w_{2N}^{2ka} = w_N^{ka}$ as $w_{2N}^{2ka} = \exp[-i2\pi(2ka)/2N] = \exp[-i2\pi ka/N] = w_N^{ka}$. Instead of solving an $N \times N$ matrix, with the Cooley-Turkey algorithm, we split the problem into two new problems. An even and odd problem; $\vec{y}_e = F_{M,1}\vec{x}_e$ and $\vec{y}_o = F_{M,2}\vec{x}_o$ then set $M = N/2$ (in the case N is even, as it is easier to explain. In case of an odd length, it is possible to get an even length by appending 0 and removing it upon completion of the algorithm), so now the operation takes $\mathcal{O}((N/2)^2)$ for each problem or $\mathcal{O}(2(N/2)^2) = \mathcal{O}(N^2/2)$ for both. So by doing this simple step the number of computations is halved. Note that we can repeat this step, resulting in four problems with total size $\mathcal{O}(4(N/4)^2) = \mathcal{O}(N^2/4)$. Now if we repeat this until we're at the minimum problem size, the problem will be reduced to a level where we're left with N problems all of size 1. Thus leaving a problem where solving all of these equations would take $\mathcal{O}(N)$. Note that after doing this we have to use all of these individual equations to solve our original equation. This is done by using $y_N = y_N^e + w_N^{ka} y_N^O$ for $N = 0, 1, ..., M-1$ (giving the first half of the components) and $y_{N+M} = y_N^e - w_N^{ka} y_N^O$ (giving the second half of the components), for each iteration. Until the original problem is solved. The structure created when going back to the original problem from the individual ones is shown in figure 3.3. Which shows that it takes $\log_2(N)$ steps to go back to the original problem. As the individual problems don't only need to be solved, but also to be put back in place to solve the original problem adding $\log_2(N)$ steps to the complexity of the problem reduces the original problem from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log_2(N))$ steps.

3.2.1. Phase information

When decomposing a map in Fourier space it is described by both amplitudes as well as phases. Boosting and muting different amplitudes will highlight different structures, allowing for signal to be boosted and noise to be reduced (or the other way around in the case of an ineffective method). Improving interpretability but not changing any of the captured information. While changing the phases would result in the creation of new information which does not exist. While many sharpening techniques like DeepEMhacer[7] change the entire

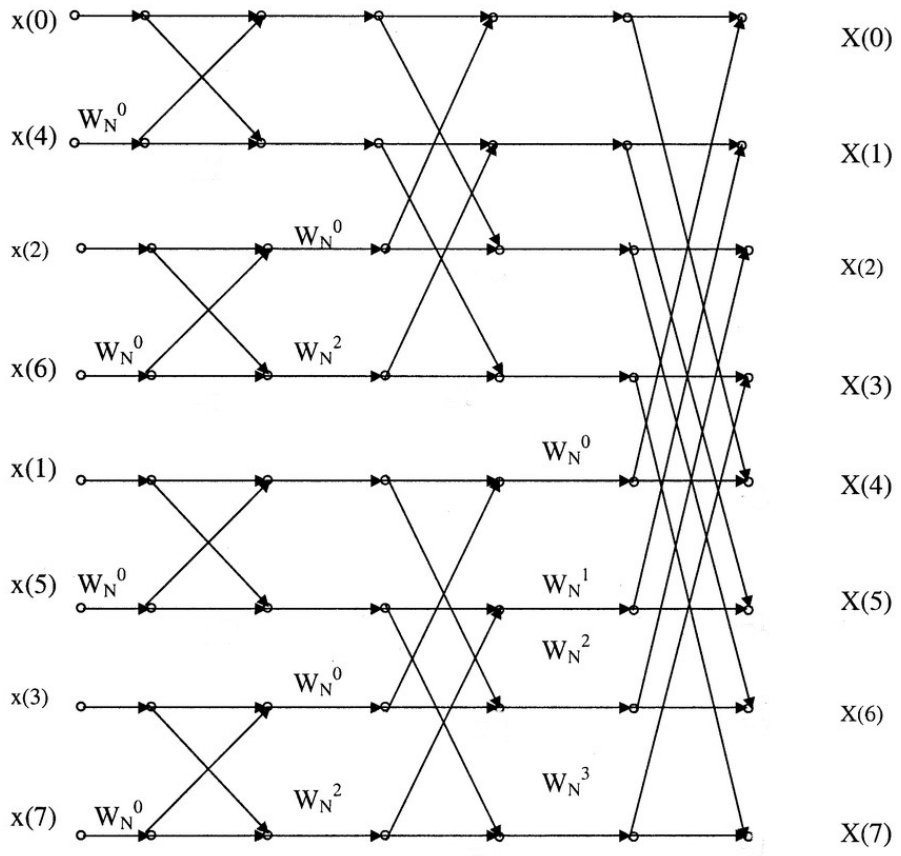


Figure 3.3: For N=8 there are $\log_2(N) = \log_2(8) = 3$ steps required to get back from the individual solutions to the original Fourier transform. Image is a modified version of an image from Maharatna 2004[15]

structure of the EM-map, thus including the phases. Meaning that it may result in a sharpened map where non-existent information is added, which is referred to as hallucinating. Where the proposed implementation of using machine learning for the prediction of radial profiles thus only touching the amplitudes and keeping the phases untouched removes the potential risk for hallucination appearing in the sharpened map. In short, the phases contain the information and the amplitudes are used to determine what parts of this information to display. Both allow for highlighting the features by increasing the amplitude of signal as well as allowing for the removal of noise by decreasing its amplitude.

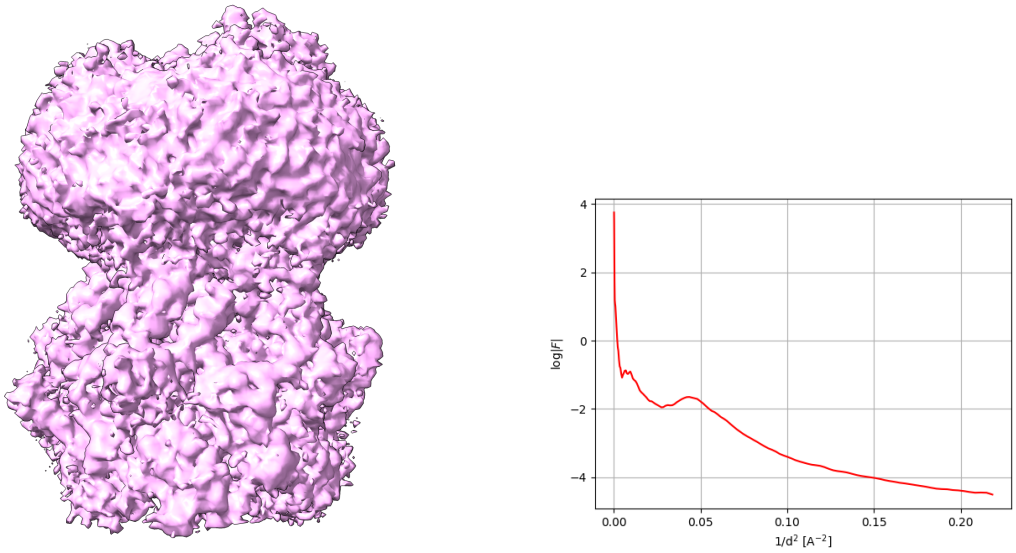
3.2.2. Radial profiles

The upside of describing an image in Fourier space is that it is possible to highlight or reduce certain features, by influencing the image in the Fourier domain. One of the ways to improve interoperability is by changing the radial profile. This is determined by taking the centre in Fourier space and calculating the average intensity at a distance from the centre (this distance is referred to as a frequency since it's a distance d from the centre in the shape of a circle). This can be done using equation (3.2) where the average intensity is taken at a distance r from the centre.

$$F(r) = \frac{3}{4\pi\left[(\delta r + r)^3 - r^3\right]} \int_0^{\pi} \int_0^{2\pi} \int_r^{r+\delta r} I(\hat{r}, \theta, d\phi) d\hat{r} d\theta d\phi \tag{3.2}$$

In equation (3.2), $I(r,\theta)$ denotes the intensity at a distance r and angle θ in the Fourier domain. Which is integrated over a thin slice of a circle of size δr. Which is divided by the area to get the average based on the size of the slice; $\pi\left[(r+\delta r)^2 - r^2\right]$. By taking the average intensity of a circle for a certain frequency $f = 1/d$, it is possible to plot this intensity as a function of frequency. An example is shown in figure 3.4. Note that the intensity on the y-axis is commonly plotted on a logarithmic scale with base e. And on the x-axis, it is plotted as a function of $1/d^2$, since this helps visualise the B-factor which will be explained in section 3.3.

By producing such a radial profile it is possible to analyze the frequency content of a map. And by changing the intensity in the Fourier domain it is possible to illustrate the structures and properties of a protein better. Both by highlighting structures more clearly as well as by minimizing the impact of noise. In figure 3.4b the bump at 0.04Å^{-2} is a result of the secondary structures, like β-sheets and α-helices present in the EM-map shown in figure 3.4a.



(a) EMDB 0282 sharpened using iLocScale.

(b) Radial profile from the entire EM-map. Note the bump around 0.05Å^{-2}. As a result of secondary structures in the EM-map.

Figure 3.4: Figure illustrating the sharpened EM-map of 0282 and its radial profile taken over the entire EM-map.

Instead of calculating the radial profile over the entire EM-map, it is also possible to determine the profile locally. This can be done by taking windows of a specified size and calculating the radial profile.

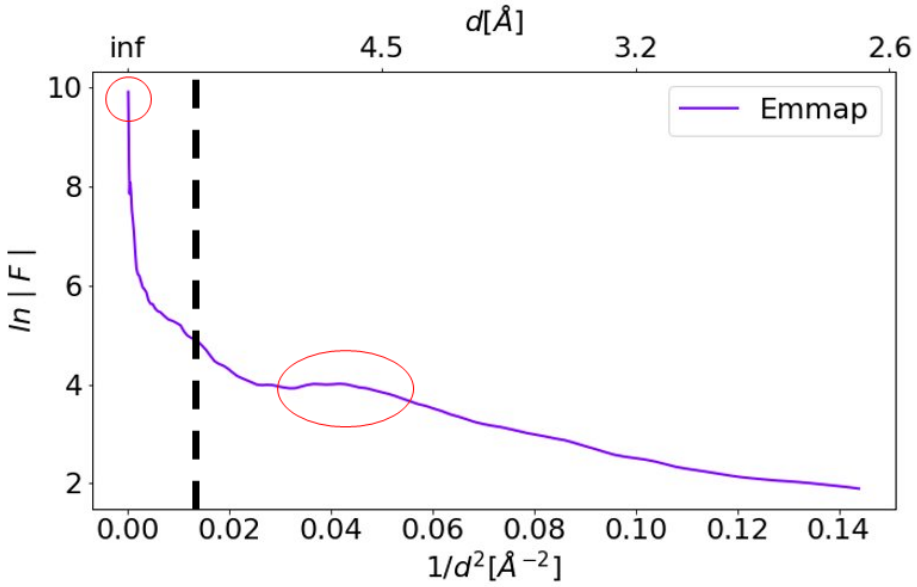


Figure 3.5: Example image of a radial profile plot. The bottom x-axis shows $1/d^2$ or f^2 where d is the distance from the centre of the image in Fourier space [Å^{-2}]. The top x-axis shows the resolution. Y-axis indicates the natural log of the radially averaged intensity observed in Fourier space.

In the radial profile in figure 3.5 there are two regions a Guinier and Wilson region[17]. A transition between this region takes place at a cutoff k_c. As the x-axis is normally shown as a function of $1/d^2[\text{Å}^{-2}]$ re-

sulting in the Guinier region referring to the left area with respect to the cutoff, in the graph where $d > 10$Å and the Wilson region refers to the right region where $d < 10$Å, showcased in the top x-axis. The difference is that in the Guinier region, the protein is modelled as a continuous blob at a fixed volume. While in the Wilson region, the protein is modelled as a random bag of atoms. Which results in the radial profile being mathematically described differently in each region. Note that this cutoff of $k_c = 0.1$Å^{-1} is an approximation that has been used during this research. The physically more accurate value could be determined using $k_c = 0.30Mw^{-1/12}$Å^{-1}[17]. Where Mw is the molecular weight in MDa. Which for most proteins results in $k_c \approx 0.1$Å^{-1}. The big difference in the way the radial profile behaves is seen in the (left) Guinier region in that the fall-off is quadratic as a function of $|F(f)|$. While in the (right) Wilson region the fall-off becomes linear as a function of $\log|F(f^2)|$. The first point in the Guinier region is also heavily dependent on the number of atoms in the cube used to determine the radial profile. While the shape of the hump in the Wilson region is highly influenced by secondary structures, like β-sheets and α-helices.

3.2.3. Synthetic radial profile

By using the mathematical representation of the radial profiles it is possible to model synthetic radial profiles which are based on a combination of both a high-frequency and low-frequency part. The high-frequency part can be calculated using equation (3.3).

$$|F(f)|^2 \approx N|\hat{f}(f)|^2 \cdot \exp\left[-\frac{\beta f^2}{4}\right] \tag{3.3}$$

Where the intensity can be solved for and plotted as shown in figure 3.5 forming the high-frequency part of the total radial profile. In equation (3.3) as N is the number of atoms and $|\hat{f}(f)|$ the atomic form factor (a factor dependent on the scattering properties of the atom). While $|\hat{f}(f)|$ does depend on the frequency, it will always be the same for both an unsharp and sharpened EM-map. Therefore when comparing the intensity for the high-frequency part of a sharpened and unsharpened map the only difference is in the way it exponentially decays. When shown as $\log|F(f)|$ will become linear when plotted as a function of f^2, β is the B-factor for which more explanation will follow in section 3.3. This results in a difference in a linear coefficient (β) and by replacing this coefficient it is mathematically possible at higher frequencies to switch between sharpened and less sharp radial profiles by boosting these frequencies which would normally decay.

$$|F(f)| = c - f^2\pi^2\Lambda \tag{3.4}$$

Where c is a constant dependent on amplitude at 0 frequency; being related to the number of atoms. Although this constant will also be influenced as a result of external factors like in the case of loss of contrast it is independent of frequency. Λ represents the moment of inertia and is thus dependent on the shape and mass of the macromolecule and also independent of frequency. Resulting in the low-frequency part only being dependent on the square of the frequency. Lastly both the high and low-frequency parts have to merge to form one radial profile.

3.2.4. Merging radial profiles

When producing a synthetic radial profile is made by taking two profiles, one describing the high-frequency component and one describing the low-frequency part. Or when merging two radial profiles in a similar fashion it can be done by having a merged profile equal to $w_1 *$ low-frequency radial profile + $w_2 *$ high-frequency radial profile, where w_1 and w_2 are weights. The weights are given in equation (3.5)

$$\begin{aligned} w_1 &= \frac{1}{1 + exp\left[k * (d_{cutoff} - d)\right]} \\ w_2 &= 1 - w_1 \end{aligned} \tag{3.5}$$

Where d_{cutoff} depends on the number of atoms in the given window describing the radial profile. $d = 1/f$, where f is an array containing the frequencies corresponding to the discrete points of the radial profile and k, is a smoothing parameter to control the transition region of the two profiles dependent on the B-factor and calculated using $k = 0.0045 \cdot \beta$. In the implementation in LocScale d_{cutoff} has been set to 7Å as it forms a good approximation since the cutoff would be between 6-8Å for most windows of size 25 × 25 × 25Å, (the previously mentioned distance of 10Å would be for an entire macromolecule, unlike the local windows used in LocScale. Figure 3.6 showcases how the weights of each radial profile would be distributed. Where 1 would represent the merged radial profile consisting solely of the low-frequency profile and 0 would imply that the merged radial profile only contains the high-frequency profile.

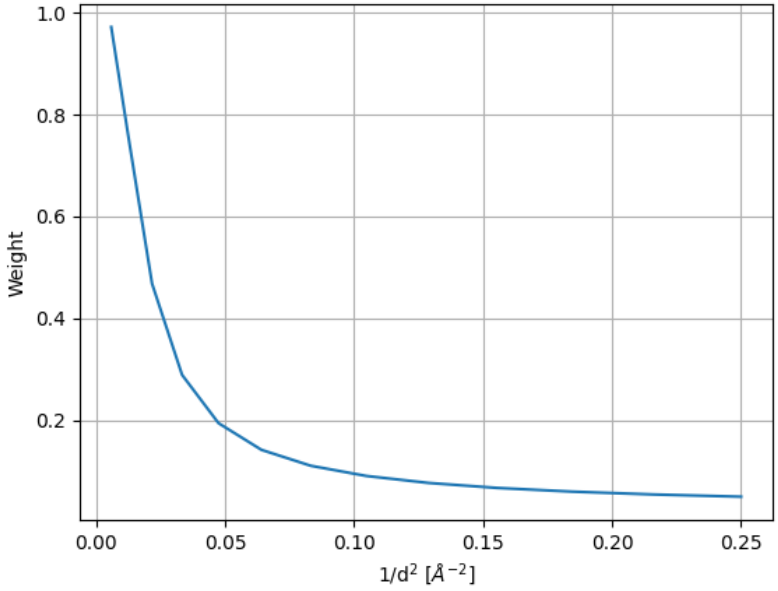


Figure 3.6: Plot showcasing the weights of different profiles. w_1 of equation 3.5 plotted for EMDB 0282 with a B-factor of 120.5 and d_{cutoff} of 7Å.

3.3. B-factor

The B-factor is a term used for the rate at which the intensity (or signal strength) decays in the Wilson region[17]. It indicates the rate at which contrast decays. Visually it is represented by the positive value of the slope at which the signal decays in figure 3.5 when fitting a straight line thru the data in the Wilson region. Note that since the plot is using a logarithmic scale the linear correlation of the slope is described by an exponential function in the form of $C\exp(-\beta\lambda)$, where C is a constant and λ is dependent on the frequency at which the intensity is measured. The value β is referred to as the B-factor. As more flexible regions may move more during the data-gathering process of Cryo-EM the resulting map in these regions will have a higher B-factor. While fewer mobile parts will have a lower B-factor. Therefore the B-factor does not only mathematically represent some slope in the Fourier domain. It also physically represents the properties of the protein.

In general, increasing the decaying signal to get a B-factor closer to zero would yield more intensity. It could be compared with brightening an image taken at night. But just like what would happen when brightening an image captured at night, recklessly increasing the intensity would also boost the appearance of noise. Therefore it is in practice not desirable to simply increase the overall intensity such that beta would equal zero. Brightening the whole map such that beta would equal zero is less effective than it seems due to two main problems. Firstly, increasing the global map such that the new B-factor would equal zero may result in some already sharp parts of the map being over-sharpened, while some blurry areas of the map will still be under-sharpened. Secondly, it is often not the case that the optimal B-factor equals exactly 0. In the case of LocScale, the first issue is resolved by instead of looking at the whole map and changing its B-factor. The B-factor is determined for small areas and these windows and determining the B-factor, therefore, decreasing the risk of oversharpening. The second issue is solved by LocScale as a result of fitting the atomic model to the unsharpened map during the process of Servalcat refinement. Thus resulting in a B-factor that is more consistent with the expected structure of the protein. Figure 3.7 (c) demonstrates the effects of different B-factors applied to the same structure. Giving a clear illustration of how a higher B-factor would lead to a more blurred image. While a lower B-factor would lead to more prominent sidechains. While figure 3.7 demonstrates the effect of different B-factors on the associated radial profile.

The B-factor relates to how certain the location of an atom is, a high B-factor would imply there is much uncertainty in the location of an atom while a low B-factor indicates a higher level of confidence in the atom's location. Which results in flexible regions of the protein having higher B-factors and worse resolution. And less flexible regions do the opposite, this correlation between the two can be noticed. And in the case of a

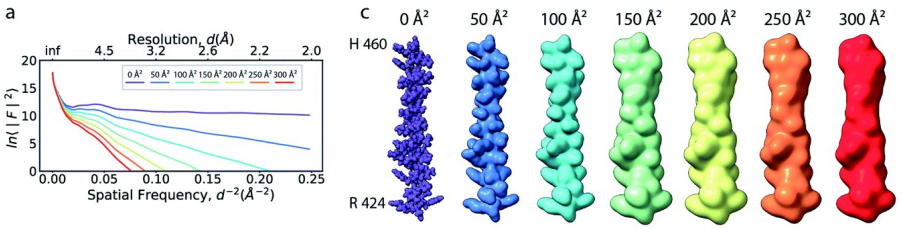


Figure 3.7: Figure (a) showcasing the radial profile associated with α-helices (figure (b)) at different B-factors.[3]

sharpened map, this correlation will in general increase which can be seen when the correlation between local resolution and B-factor increases. As the local resolution is determined by taking two half maps only containing half the information of the EM-map and looking at the correlation in Fourier space as explained in section 3.6.

As the goal of map-sharpening is creating a map that is as close to possible to the perfect map. This is unknown, it is not possible to directly compare a map to the perfect map and conclude the quality of the generated map. But it is possible to see if a map is sharper by comparing it to the unsharpened map as well as concluding if it may be more optimal by looking at the correlation between the B-factors and local resolution. As these tend to be more correlated in a more optimal map. Forming the most important metric in determining if the map is more optimal.

3.4. Map sharpening

Using cryo-EM it is possible to create three-dimensional maps of particles, but at high resolutions, these maps suffer from loss of contrast[16] which would make interpreting these maps highly difficult. To make the interpretation of the data easier it is vital to remove as much noise as possible from the map and increase the actual details captured to make them more prominent. The easiest way to suppress noise is by using a mask which mutes voxels which contain noise. Two popular masking options are an FDR-mask [2] or by using an atomic model mask, which masks based on knowledge of where the voxels containing parts of the protein are situated. One of the methods available for map sharpening is the model-based method of LocScale[13]. Model-based methods like LocScale first require a model of the structure of the protein. Based on known information about its structure. In the specific case of LocScale, it is based on using an atomic model (PDB) and fitting it to the observed unsharpened map to later scale the intensity at each voxel to the most optimal value according to the fit.

But as it requires an atomic model (PDB) it is hard to use when there is no model available or where it is incomplete. Since the atomic model is only truly required in one step of the sharpening process, by replacing this step using machine learning (ML) it may also be possible to use LocScale in situations where no model is available or for the sharpening of lipid belts since they are often unmodelled and thus suppressed when sharpened using LocScale. The LocScale pipeline in simplified form is shown in figure 3.8. The atomic model is refined using Servalcat[18]. By refining the atomic model, the atomic model is fit to the unsharpened map. Resulting in radial profiles which are determined to be the most optimal for the gathered data.

3.5. Masking

Since a map will often contain a combination of the actual captured protein, and noise, removing the noise makes it easier to see the protein and its shape. To remove noise a mask is applied where every voxel is multiplied either by a 1 in case the voxel contains a part of the signal or 0 in case the voxel contains noise. The two main types of masks used for this thesis are the atomic model mask and an FDR-mask. The atomic model mask is based on an atomic model. Where the location of the atoms is identified and these voxels are kept as signal. The great advantage is that whatever remains is guaranteed to contain signal. The main strength of the atomic model mask is also its biggest drawback. As a result of removing all unmodelled areas, it will also remove all unmodelled areas that are part of the protein. This for example can be seen in the case of lipid belts which will also be removed. The FDR-mask, or false discovery rate mask, is a mask based on confidence. Where a threshold is set and based, often at 99%. Resulting in a 99% certainty that the voxel is not noise, resulting in almost all signal getting thru and a little bit of noise. This is done by taking four small windows which are likely to only contain noise as illustrated in figure 3.9. As highlighted by the red boxes in figure 3.9, these windows are purposefully chosen to be on the sides as it is unlikely for the protein to be

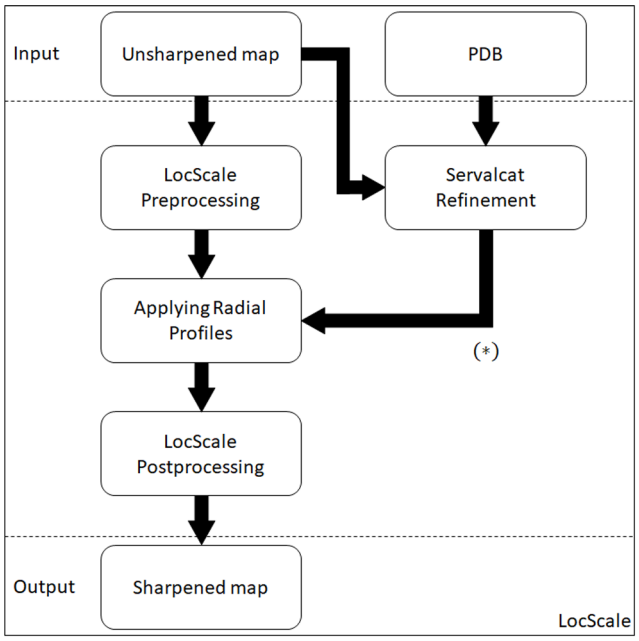


Figure 3.8: LocScale Pipeline: using the unsharpened map and PDB as input. The PDB gets refined using Servalcat [18]. Where the radial profiles from the refinement are used as the theoretical most optimal profiles. By replacing (*) with an ML-based method the dependency on a PDB can be removed.

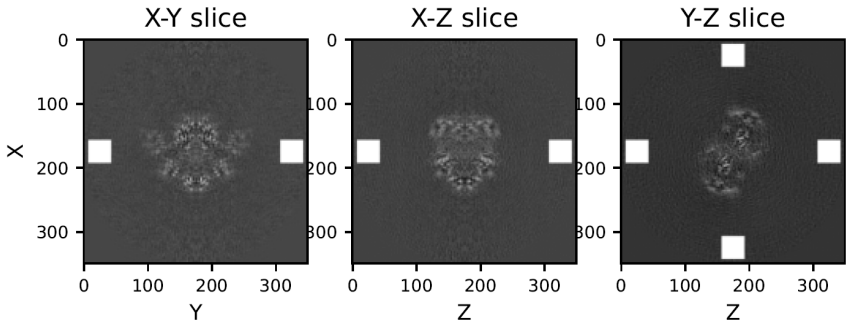


Figure 3.9: Showcasing the four windows on the sides chosen for fdr-masking. While the protein is clearly visible in the center, the taken boxes on the sides do not contain any parts of the protein. Allowing for windows purely containing noise.

situated on the sides. By taking the intensities of the voxels in these windows a cutoff intensity is calculated such that 1% (or another desired percentage) of the voxels with the highest intensity in these noise-containing windows get thru. The main drawback of using an FDR-mask is that it will still let noise thru. As a result of the threshold, approximately 1% (or whatever value is used as the threshold) is expected to get thru. But since the volume of the model is much larger than the actual subject itself. Much of the resulting output is also noise. For example, if the protein only occupies 10% of the volume. From the other 90% of noise 1% will get thru. This is a significant amount at 9% of the data being noise in case the full 10% of the volume occupied by the protein also gets thru. Resulting in a situation where almost half of the voxels in the masked image are noise. Although the intensity of the protein itself is in general still far greater than that of the noisy areas insuring viewing the protein with even a small threshold would already get rid of the remaining noise.

3.6. Map resolution and local resolution

In the case of a cryo-EM map, the resolution refers to the amount of detail that can be differentiated in the map (and is thus different from the voxel size). This resolution is a factor highly influenced by the quality of the microscope as well as the set of images used to produce the map. In general, a better microscope with a larger number of images will yield a higher resolution. While at lower resolutions it may only be possible

to denote the shape of the protein, while at higher resolutions (a lower number) more details will start to appear like secondary structures such as α-helices or even sidechains. The resolution of a map is determined by looking at the Fourier Shell Correlation (FSC)[8]. To calculate the FSC of an EM-map it is first required to split this EM-map into two half-maps, which are maps reconstructed independently using only half of the experimental data.

Making it possible to calculate the FSC value, which lays between 0 and 1 and denotes how well these maps correlate in Fourier space as a function of radius (yielding the resolution at the value for which $FSC(r) = 0.143$[16]). The value can be determined using equation (3.6).

$$FSC(r) = \frac{\sum_{r_i \in r} F_1(r_i) \cdot F_2(r_i)^*}{\sqrt{\sum_{r_i \in r} |F_1(r_i)|^2 \cdot \sum_{r_i \in r} |F_2(r_i)|^2}} \tag{3.6}$$

Where r denotes a certain frequency and, F_1 equals the complex structure factor for the first map and F_2 for the second map. Note that $F_2(r_i)^*$ refers to the complex conjugate of $F_2(r_i)$.

Where local resolution differs from the resolution of the entire map. Is in that local resolution is used to denote the resolution in different regions of the map. As the resolution can vary within a single map. This is caused by the fact that the 2D images used to construct the 3D map may have some images of higher quality resulting in a higher resolution in the areas described by these higher-resolution 2D images. As the map is constructed from multiple images describing the same region. More flexible regions, which may move more during the process of capturing multiple images can be expected to have a worse resolution. To determine the local resolution it is possible to use the same formula, equation (3.6). Allowing for voxel-specific local resolution. This is done by taking a window around the voxel and using the previously mentioned equation to determine the local resolution of this window which will be assigned to this specific voxel. Now by repeating this for all voxels it is possible to create a map detailing the local resolution for each voxel.

3.7. Machine Learning

In machine learning, a given architecture will try and learn to make as accurate as possible estimations from a given input and compare these to the output. A loss function is used to quantitatively describe how accurate its estimate is while learning it is trying to minimize this loss. This is done by using an optimizer, optimising the weight of each neuron in a network trying to scale a neuron's response. While the way a neuron specifically responds is dependent on its activation function, where a linear activation function would result in a network closely resembling a linear regression model. While non-linear activation functions would enable the network to learn more complex non-linear relationships.

3.7.1. Loss-function

The loss function is used to quantitatively conclude how well the network is behaving. Where a lower value would indicate that the network is working well, while a higher value is less optimal. By choosing a specific loss function, the way how the network would optimize changes. As for predicting radial profiles, when trying to get as accurate of a result as possible for the whole profile it would be practical to use a mean squared error (MSE) as a loss function. To calculate the loss equation (3.7) is used.

$$MSE(\theta, \hat{T}) = \frac{1}{n} \sum_{i=1}^{n} |\theta_i - \hat{T}_i|^2 \tag{3.7}$$

Here θ is the answer profile, while $\hat{T}$ is the estimator the network predicts for a given profile. By looking at how much of the prediction is and squaring this error for each item i in the dataset of length n it is possible to derive the mean. Which is the value the network will try to minimize.

3.7.2. Optimizer

To minimize the loss value an optimizer is used. This optimizer makes use of statistical gradient descent to find the lowest value. For this thesis, **Ada**ptive **M**oment Estimation (Adam)[14] has been used, an optimization method build on the basis of stochastic gradient descent (SGD)[4]. For its ability to quickly and effectively converge to an optimum in a low number of epochs enabling quickly iterating the network's parameters. Adam works by first initializing the network with random weights w at iteration 0 ($t = 0$) with $m_w^0 = v_w^0 = 0$. Adam then calculates the gradient of the loss function with respect to the weights. The weights are then updated in the direction of steepest descent as it is expected to yield lower loss, the step size is based

on the learning rate (lr), initially the learning rate (lr) is set thus dictating the size of the first step. Then what is referred to as the first moment (m_w) and second moment (v_w) are calculated for the gradient of each weight. Note that the bias-corrected moments ($\hat{m}_w$, $\hat{v}_w$) are used for actually updating the weights to compensate for the bias in m_w and v_w as they are biased to zero. The rate of this exponential decay is set with β_1 for the first moment and β_2 for the second moment. After which the new learning rate is determined by dividing the first moment by the root of the second and a small value ϵ to prevent dividing by zero errors. Then Adam updates the weights in the direction of the steepest descent by removing the scaled gradient from the current weights. This is then repeated. this is mathematically described using equations (3.7.2). The difference between SGD and Adam is that Adam uses an exponentially decaying average of past gradients, while SGD does not. This allows Adam to dynamically adjust the learning rate resulting in an optimization algorithm which is more resistant to noisy gradients than SGD.

$$\begin{cases} m_w^{(t+1)} = \beta_1 m_w^{(t)} + (1-\beta_1)\nabla_w L^{(t)} \\ v_w^{(t+1)} = \beta_2 v_w^{(t)} + (1-\beta_2)\left(\nabla_w L^{(t)}\right)^2 \\ lr^{(t+1)} = lr^{(t)} \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \\ \hat{m}_w = \frac{m_w^{(t+1)}}{1-\beta_1^t} \\ \hat{v}_w = \frac{v_w^{(t+1)}}{1-\beta_2^t} \\ w^{(t+1)} = w^{(t)} - lr^{(t)} \frac{\hat{m}_w}{\sqrt{\hat{v}_w}+\epsilon} \end{cases}$$

(3.8)

The following parameters were used for the adam optimizer: an initial learning rate (lr) of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 * 10^{-7}$.

3.7.3. Regularization

One of the major risks involved with machine learning is overfitting. This occurs when the model tries to capture the noise in the dataset and include it in the model. Such a model is too complex for what it is trying to model. Which results in noise in the training data being included in the model. One of the ways of noticing a model is overfitting is when the training loss is low while validation loss or testing loss is high. As the training dataset is perfectly fit inside the model, including the noise, resulting in poor generalization. Without any provocative measures, the model will include the noise in the model as it will result in a lower loss. One of the methods to prevent overfitting is by adding L1 or L2 regularization, which penalizes the production of a complex model thus reducing the risk of overfitting. L1 regularization adds a penalty proportional to the absolute value of the model weights, as shown in equation (3.9). While L2 regularization adds a penalty proportional to the square of the model weights, (3.10). Where the parameters λ_1, λ_2 determine how strongly the model's complexity is penalized. L1 regularization can be introduced by adding a factor $\lambda_1 \sum_{j=1}^{p} |w_j|$ to the loss function as shown in equation (3.9), note that w in this case refers to the weights of the model, which dictates how the output of a neuron will be scaled and thus the influence of a given neuron on the input of the next layer.

$$loss = MSE + \lambda_1 \sum_{j=1}^{p} |w_j| \tag{3.9}$$

The advantage of using L1 regularization is that weights of unimportant features, which tend to be small, are sent towards zero. Thus removing them from the model, and reducing the model's complexity and thus the risk of overfitting. While adding L2 to the loss would introduce $\lambda_2 \sum_{j=1}^{p} w_j^2$ as shown in equation (3.10).

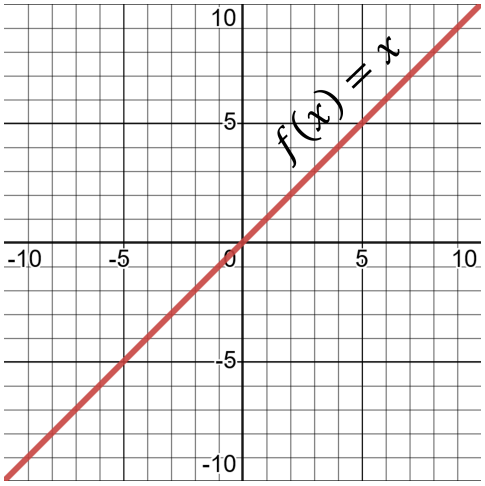
$$loss = MSE + \lambda_2 \sum_{j=1}^{p} w_j^2 \tag{3.10}$$

The upside of using L2 regularization is that squaring each weight results in the larger weights being penalized more than the smaller ones. Resulting in a more smoothed-out model, where the weights are brought more closely together, resulting in less variation in the weights and reducing the impact of individual features. In

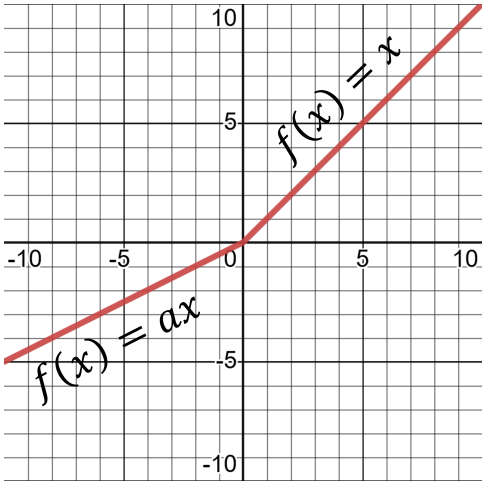
both equations (3.9) and (3.10), w_j represents weight j of the model, the loss is the resulting value by taking the MSE and adding L1 or L2 regularization and λ_1 and λ_2 are their associated hyperparameters which are set.

3.7.4. Activation function

The activation function of a layer determines whether a neuron should be activated or not and what type of transformation would occur as a result of the neuron. When predicting the high-frequency part of the radial profile, as we're expecting a linear transformation, similar to applying a new B-factor as seen in section 3.3, it is logical to make use of a linear activation function. But this does not completely encompass the full radial profile as a linear response would not correctly model the lower-frequency area of the radial profile. Thus logically asking the question: what is the effect of using different activation functions on the network's accuracy?



(a) Graphical depiction of the behaviour of a neuron with a linear activation function.



(b) Graphical depiction of a neuron's response with a PReLU activation function. The plot is shown using $a = 1/2$.

3.7.5. Linear activation function

As the name would suggest, a linear activation function interacts in a linear manner where each is activated and is in this fashion very similar to a linear model. The neurons' response based on the input is shown in figure 3.10a, also given by $f(x) = x$. Resulting in a neuron with a linear response, where the output would be equal to the input multiplied by a weight set by Adam. Since protein maps can be sharpened by applying a new B-factor, resulting in a linear transformation to the higher-frequency area of the radial profile, a model with linear activation functions would be a logical choice for accurately modelling the high-frequency part of the radial profile.

3.7.6. PReLU activation function

A **p**erametric **re**ctified **l**inear **u**nit[10] activation function consists of a linear activation function for positive inputs and a perametric linear activation function in the negative domain. Which can be summed up by equation (3.11).

$$f(x) = \begin{cases} x, & x \geq 0 \\ ax, & x \leq 0 \end{cases} \tag{3.11}$$

The response of the PReLU activation function is shown in figure 3.10b. Although it may look similar to a linear activation function, it is not the same in case $a \neq 1$. In this case, it will also not be linear, allowing the network to learn more complex, non-linear relationships. For example in the case of $a = -1$ the neuron would transform a given input in a similar was as $f(x) = |x|$ result in an activation equaling the absolute value of the input as any input value will be mapped towards their positive corresponding value. Since a linear function would have to suffice the following two properties: $\forall x, y \in \mathbf{R}, c \in \mathbf{R} : \mathcal{L}(cx) = c\mathcal{L}(x), \mathcal{L}(x+y) = \mathcal{L}(x) + \mathcal{L}(y)$. As the function in equation (3.11) can be written as $f(x) = x[H(x) + aH(-x)]$, where $H(x)$ is the unit step function and a is a constant set by the network. The first property; $\mathcal{L}(cx) = c\mathcal{L}(x)$ since $f(cx) =$

$cx[H(x)+aH(-x)] = cf(x)$ Equation (3.11) does not satisfy the second property as $f(x+y) = (x+y)H(x+y)+a(x+y)H(-x-y) = x\left[H(x+y)+aH(-x-y)\right]+y\left[H(x+y)+aH(-x-y)\right]$ which in general is not equal to $f(x)+f(y) = x[H(x)+aH(-x)]+y\left[H(y)+aH(-y)\right]$. For example given $a \neq 1$ we have $x = 1, y = -1 \Rightarrow f(x)+f(y) = 1-a \neq f(x+y) = 1+a-1-a = 0$. Therefore the second property is not met in case $a \neq 1$ and thus equation (3.11) is not linear in general.

Everything described in section 4.4 comes together in a neuron which has a weighted input: $w_1x_1+...+w_nx_n$ where n is the number of neurons in the previous layer. After which a neuron-specific bias term is added, to change when (or how) the activation function is activated. Since in this thesis, only a linear and PReLU activation function have been used, the model is always activated in case $a \neq 0$ and the bias term only affects how the neuron responds. The neuron's response is given by $f(w_1x_1+...+w_nx_n+b)$, where f is a set activation function.

4

Approach

4.1. Concept

The main goal of using a machine learning (ML) approach to map sharpening is to leverage the strength of machine learning in its ability to make predictions in new situations. Although there are also risks associated with the use of machine learning. One of which is the black-box characteristic of machine learning, making it hard to understand why and how the result came to be and even more important if it is accurate (which may result in oversharpening for example). Thus the optimal application of machine learning is one where its strength can be maximized while minimizing the black-box characteristic and its associated risks. The primary way of minimizing this effect is by minimizing the number of steps in which machine learning is used as it will minimize the chance of the program returning nonsensical predictions. As well as it being easier to figure out if the predictions are nonsensical in case there is just one step using machine learning.

The strength of machine learning lies in its ability to make predictions in new situations. In the case of LocScale[13], which requires prior knowledge, it is best to apply machine learning in this step where LocScale requires prior knowledge. Since using LocScale would normally require an atomic model for predicting the local B-factors. By doing so, the hope is that it may improve the capabilities of LocScale in environments lacking an atomic model or unmodelled regions.

By using LocScale as a basis and just replacing the step where machine learning would be better suited to enable a LocScale-based approach to map sharpening without an atomic model, a process as shown in figure 4.1 can be used, requiring the unsharpened map as its only input (compared to classical LocScale as shown in figure 3.8); using machine learning to generate a radial profile from the captured data and then continue the process of LocScale using the predicted radial profiles, removing LocScales dependency on an atomic model.

As shown in figure 4.1 the only alteration between the ML model-free-based LocScale approach and the classical model-based LocScale approach is the method by which the local B-factors are determined.

4.2. Data gathering

The dataset in this thesis is the same as the dataset used for the creation of Emmernet[6]. Both because of continuity as both theses are written as part of Jakobi lab and it offers a good basis for comparing both methods as well as the selection criteria offering a well-rounded dataset. The dataset used for Emmernet is based on the dataset used for DeepEMhacer [7] with some additional quality checks. Manually inspecting the unsharpened maps resulted in 6 maps being discarded from the dataset. While checking the FDR confidence maps resulted in the removal of another 16 maps. Resulting in a final dataset consisting of 129 maps compared to the original 151 maps used for DeepEMhacer. From this resulting dataset, one additional map has been removed (EMDB 4531) as there was no local resolution map in the dataset.

Lastly due to problems in gathering the radial profiles, 5 additional EMDBs have effectively been removed from the dataset (0490, 0492, 4907, 8911, 20449). Resulting in 4 fewer EMDBs in the dataset and one less in the validation dataset. Resulting in a dataset consisting of 84 training EM-maps, 15 validation EM-maps and 13 testing EM-maps.

The selected dataset is then split into training, validation and testing EM-maps. Insuring the network will never see a radial profile from a testing map before the testing phase itself. The split is again the same as used for Emmernet. This split has been chosen such that proteins with different properties are in each dataset.

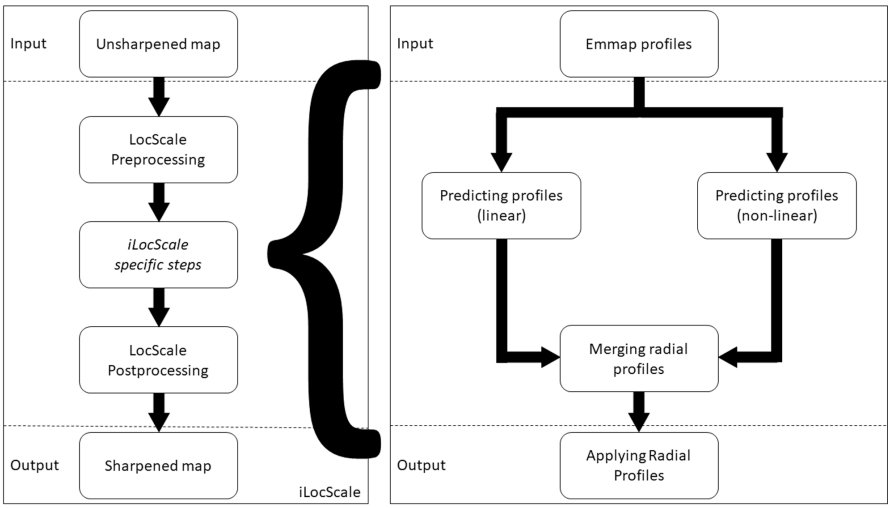


Figure 4.1: iLocScale pipeline: the dependency on a PDB has been removed. While also adding iLocScale-specific steps. By using the radial profiles of the EM-map window to predict a radial profile using two independent models and later merging these profiles back together, leveraging the strengths of using a model based on linear activation functions in the high-frequency areas while using a model consisting of both PReLU and linear activation functions for the low-frequency area. Having merged these radial profiles, applying the merged radial profiles allows iLocScale to continue with the postprocessing steps in a similar fashion to classical LocScale.

Thus both the training, validation and testing datasets contain proteins with for example lipid belts or bad resolution. Appendix IA.1 shows in which set each EMDB has been placed.

4.3. Pre-processing

A few preparatory steps must be applied to the maps before being used. These pre-processing steps will yield the input for the neural network (NN). A few steps must be taken to help the NN's interpretability of the data. Firstly the EM-maps will be loaded at a certain pixel size dependent on the data-gathering process. These maps will be resampled to a new pixel size of 1Å. After which the map will be standardized such that the average intensity equals 0 with a standard deviation of 0.1. Then this new standardized and resampled EM-map will be chunked into small cubes of 25Å (or 25 pixels) in each dimension. Since it has been shown that working with cubes of the size $20-30$Å is optimal for a NN[13]. After this an atomic model mask (a mask compiled from the PDB by looking at which voxels contain atoms and which voxels are supposed to be empty) is applied to remove cubes that contain noise. The choice for using an atomic model mask over an FDR-mask is based on the fact that an FDR-mask uses a confidence interval to determine whether a voxel contains noise or signal. So even using an FDR-mask at a 0.99 threshold would still let 1% of noise thru. This is disastrous as this noise may not be a part of the PDB and thus yield an answer radial profile correlating to zeros. This would not only mess with the effectiveness of the network in predicting radial profiles since it would also have to question if the profile is part of the atomic model. But also yield a problem since we're interested in taking the log of the radial profile, which would not be defined as the profile containing zeros.

From the leftover cubes, the radial profile is determined and given to the network.

4.4. Machine learning

4.4.1. Training the model

The model is trained on a dataset (X) which contains the radial profiles from the training dataset which have been derived using the previously described pre-processing steps from the train EM-maps. To produce the answer dataset (Y) for the radial profile from the unsharpened map.

The radial profile for the refined PDB is taken and the same cubes radial profile is used as the answer. Both the X and Y are randomly shuffled. These steps are also used to gather the input and answers for the validation and test datasets. The maps used as answers

To produce the radial profiles used for training (X) the previously mentioned pre-processing steps are applied to an unsharpened map of which an atomic model is available. These radial profiles in which the

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 13)                182

 dense_1 (Dense)             (None, 13)                182

 dense_2 (Dense)             (None, 13)                182

 dense_3 (Dense)             (None, 13)                182

 dense_4 (Dense)             (None, 13)                182

=================================================================
Total params: 910
Trainable params: 910
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.2: Summary of the model used. Linear activation layers. Using radial profiles of shape 13.

pre-processing step outputs are taken and randomized by random shuffling and split in training, validation and testing datasets yielding a matrix containing vectors where each vector describes the radial profile of a cube. To gather the answers (Y) both the atomic model (PDB) and unsharpened maps are used. By taking the atomic model and performing refinement using the unsharpened map in Servalcat [18] after which the refined atomic model map will be upsampled to match the pre-processing steps; upsampling to 1Å and standardizing using a mean of 0 and standard deviation of 0.1 as well as cubing into $25 \times 25 \times 25$Å cubes. After which the radial profile of these cubes is determined which will also undergo the same shuffling and splitting process such that the answer Y will match the correct training vector X. The model is trained using the radial profiles provided by the pre-processing step (X) and the target radial profiles (Y) provided by the (atomic-model) model maps as used as one of the inputs for LocScale. The architecture used for the model is shown in figure 4.2.

4.4.2. Using synthetic profiles

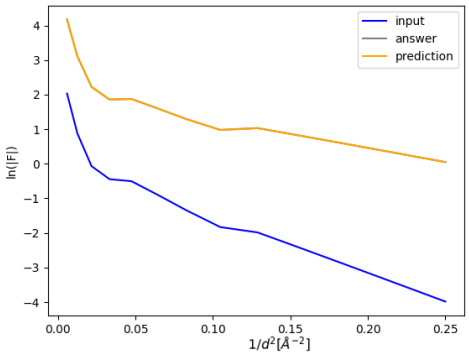
By training the model on synthetic profiles with a defined linear relation between the sharpened and unsharpened profiles it is possible to prove if the model is able to learn this relation. Besides that, it is possible to get an indication of how many epochs are approximately required to yield a well-rounded model. Lastly and most importantly by adding noise to the radial profiles, it is possible to see how the model behaves as a result of noise. To do this a set of 50 thousand synthetic profiles have been generated with between 200 and 1000 atoms (n) in a cube (of size 25Å) with a uniform distribution. Besides this fact, the B-factor (β) of the EM-map is uniformly distributed between a B-factor of 50 and 250. The reference B-factor used to produce the (Y) radial profile, is half the B-factor of the EM-map forming the radial profile used for (X). With all other properties, like the number of atoms staying the same.

To make it more challenging for the network, noise is added to the profile. This is done by taking 13 unique values from a standard normal distribution. And adding these to the reference profile. By doing this the synthetic profiles become more akin to reality as the limitations of cryo-EM microscopy and post-processing methods will also induce random errors. And thus it is also desirable for the model to be able to function correctly even though the inputted data is not perfect.

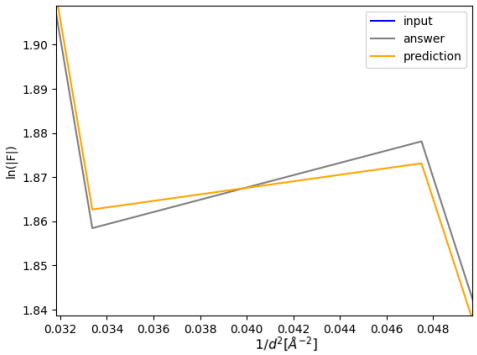
These two sets of radial profiles are then fed into the model.

In figure 4.3 it can be clearly seen that the model is able to almost perfectly predict the relationship between the EM-profile and reference profile. As can be expected with a perfect relationship.

Figure 4.4, gathered by taking the predicted values and removing the true value, reaffirms that the magnitude of the error is relatively small in comparison to the radial profile itself for all profiles. But it demonstrates



(a) Full-scale illustration of simulated radial profiles, answer profile in orange, EM-map profile in blue and target profile in grey.



(b) Zoomed in to highlight the accuracy of the prediction. The target profile is in grey and the prediction is in orange.

Figure 4.3: Showcasing the accuracy of the predicted from the synthetic profile proves the ability of the network to very accurately predict a relationship. The second image showcases the error as it is not visible in the first image due to the predicted line fully covering the line representing the answer. This test has been done in a system without any noise added to the radial profile.

the network's difficulty with predicting the Guinier region as well as the higher frequency parts of the Wilson region, as a function of frequency.

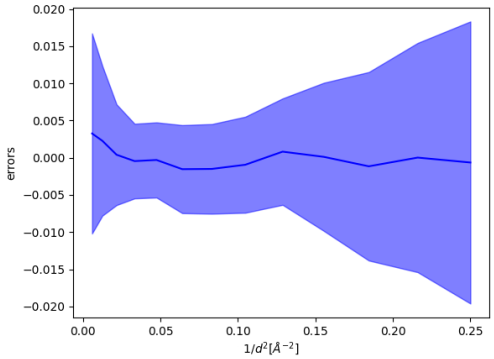
While comparing it to the same figure for synthetic profiles where random noise has been introduced (picked from a normal distribution with $\mu = 0$, $\sigma = 1$) as should be expected the error will increase as the network uses synthetic profiles with randomly added errors. It is also important to note that a random error will not induce any bias in the ability of the network to predict. The added error is purposefully large such that the errors in figure 4.4a are negligible in figure 4.4b. Thus showcasing the effect of a random error at different frequencies. Illustrating the network's sensitivity to errors within the 0.01Å and 0.04Å frequency range. When random noise is added to the model the model becomes worse, as seen in figure 4.4b. This clearly insinuates that random errors in real data as a result of (pre-)processing and physical limitations of the microscope are inversely correlated to the network's ability to sharpen. It also shows, as previously noted, how it disproportionally affects the lower frequency regions more.

Figure 4.5 shows that the loss converges quickly (in about 10 epochs or so). Suggesting many epochs may not be necessary for achieving a good model.

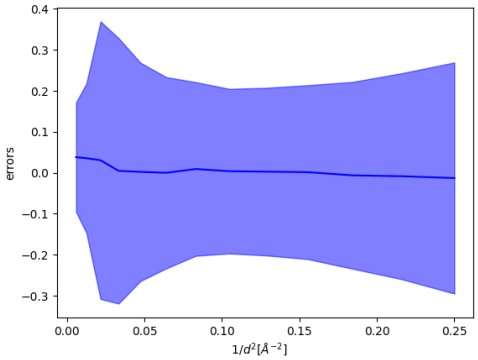
4.4.3. Hyperparameter tweaking

To tweak the hyperparameters of the model Talos[1] has been used. The options for the hyperparameters and which hyperparameters have been tweaked are shown in table 4.1.

As the number of combinations for networks grows quickly by adding more options and running the network takes 5-30 minutes to train for each option, making it impossible to brute force every possibility using a gridsearch method for finding the most optimal hyperparameters as there are almost 750 million possible combinations. Therefore the method of random search was used for finding the optimal hyperparameters.



(a) Error showcased synthetic profiles (n=50.000) with no added errors to the input radial profiles. The mean is depicted by a dark line and one standard deviation is shown in the shaded blue area. Note the order of magnitude to be relatively small.



(b) Error showcased synthetic profiles (n=50.000) with an added normally distributed error ($\mu = 0, \sigma = 1$) to the input radial profiles. The mean is depicted by a dark line and one standard deviation is shown in the shaded blue area.

Figure 4.4: Showcasing the accuracy of the predicted from the synthetic profile proves the ability of the network to very accurately predict a relationship. The second image showcases the error as it is not visible in the first image due to the predicted line fully covering the line representing the answer. Errors are attained by taking the predicted value and removing the true value.

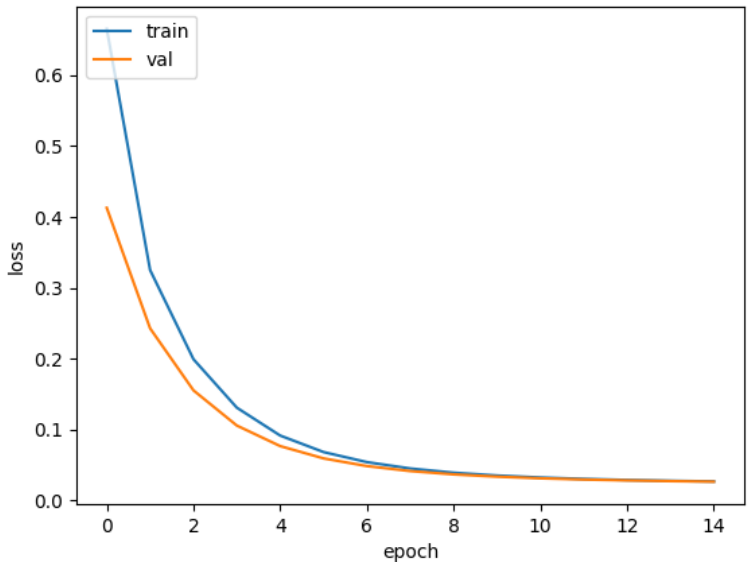


Figure 4.5: Loss as a function of epochs. Showcasing MSE loss function on a dataset of 50 thousand synthetically generated radial profiles with relation 0.5 and no random noise added.

Specifically the quantum randomness method of Talos.

The optimal parameters found when using hyperparameter tweaking are a learning rate of 0.01, 15 epochs, a batch size of 3, using 13 neurons for the hidden layers, no dropout and 0.01 for both the λ_1 and λ_2 regularizers.

4.4.4. Using the model

By reconstructing the model and inputting the radial profile the model will predict an estimate for an optimal radial profile. It is also possible to input a set of radial profiles (as an array for example) in which case the array will have shape $(n, 13)$ where n is the number of profiles. This will output an array of the same shape describing the optimal profiles. It is more efficient to predict an entire array at once over using a loop by using individual predictions.

The model has been implemented in LocScale to replace the step where normally the EM-map radial profiles would be scaled to the PDB profiles. And replaced such that the EM-map profiles would be used to first predict the optimal profiles and then to be scaled against these profiles. After which the normal LocScale

Table 4.1: Hyperparameters tweaked using Talos. Values are used to tweak the parameters. Choices denoted using [option 1, option 2, ..., option n]. For intervals options denoted using (lower bound, upper bound, stepsize)

Hyperparameter	Values
Learning rate	[0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1, 0.2, 0.5]
Epochs	[5, 10, 15, 20]
Batch size	[3, 6, 12, 24]
Hidden layers	(13, 400, 2)
Dropout	(0, 0.3, 0.1)
L1	(0, 0.1, 0.001)
L2	(0, 0.1, 0.001)

procedure would continue.

4.5. Network evaluation

To determine whether or not the method is effective it is required to both validate using quantitative and qualitative methods for the network there are several quantitative methods of evaluating its capabilities.

Radial profile comparison

The easiest method of quantitatively determining the model's effectiveness in learning radial profiles is by looking at the behaviour of the loss function in the test dataset. By predicting the radial profile in the test dataset and calculating the MSE between the predicted radial profile and answer profile and comparing this to the MSE of the unsharpened profile and answer profile forming a quantitative basis of the model's effectiveness. Sadly this does not give information about the bias of the network, as the MSE contains both the variation and bias, requiring a method which does display the bias as a separate variable. For analysing the bias it is required to do the same analysis but instead of taking the MSE, it is possible to just take the mean of the entire set of unsharpened or predicted profiles and compare it to the mean radial profile of the target.

B-factor comparison

It is relatively easy to determine the accuracy of the network's ability to predict the slope. This can be done by comparing the B-factor of the predicted model to the B-factor of the theoretical profile that has been used to train the model. And fit a line using linear regression to determine both how accurate it is, if the model is biased or if there is a lot of variation (which can be spotted in case the correlation is low).

Error analysis

By plotting the error, by taking the difference between the predicted profiles and the target profiles it is possible to determine at which frequencies the model is effective.

4.6. Evaluating sharpened maps

For the sharpened maps it is possible to determine their quality using some quantitative metrics. But it is also highly important to qualitatively judge whether the map's interpretability has improved compared to the unsharpened (or other sharpened maps).

4.6.1. Visual inspection

The method for qualitatively determining the method's effectiveness in sharpening maps is by visually comparing the sharpened map to the unsharpened map to see if the interpretability has improved.

In the case of an α-helix for example a more sharp result would accentuate the side chains. While a less sharp version would not only have less prominent side chains. The helix part would also look more like a blob. An example has been shown in figure 4.6 an α-helix is shown twice clearly demonstrating the difference between a sharp and less sharp version. sec A sharper map will tend to have more prominent sidechains which would not be visible or harder to see in the unsharpened map. While also sporting a skinnier α-helix compared to an unsharpened more blurry α-helix.

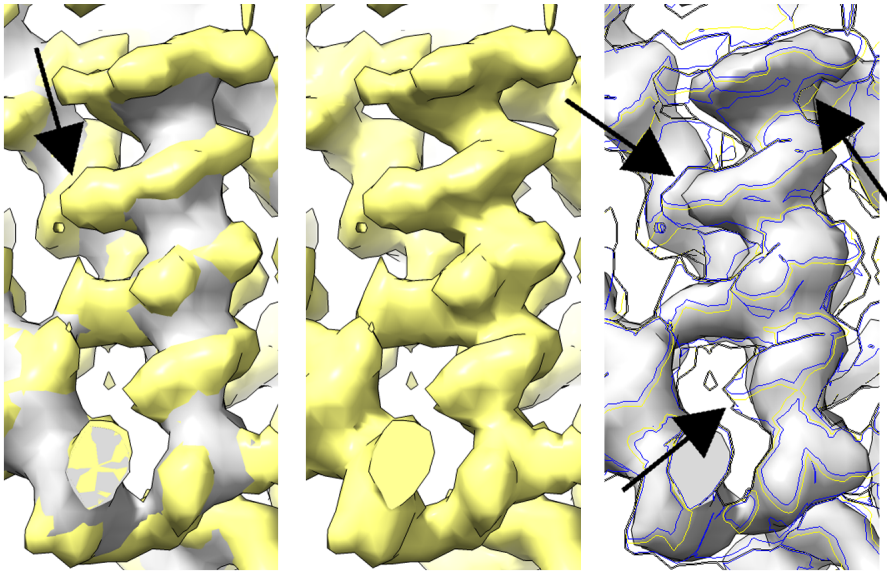


Figure 4.6: Example comparison between sharpened and unsharpened maps: EMDB 4997 sharpened using traditional LocScale (yellow) superimposed on the unsharpened EM-map (grey) in the most left image. Middle image showcasing the LocScale sharpened map with prominent side chains and a skinny α-helix. The right image shows the unsharpened map with a more blurred α-helix and less prominent sidechains. The edges of the LocScale sharpened map are superimposed on the unsharpened map, showcasing the difference in prominence in sidechains (left upper and left bottom arrow in the right image) as well as the more blurry characteristic of the α-helix with the arrow in the top right pointing towards a part of the helix that is more skinny on the superimposed LocScale map.

4.6.2. Local FSC resolution and B-factor correlation

From literature[3] it follows that there is a correlation between the local FSC resolution and the B-factor. By fitting the correlation of the FSC resolution and the B-factor from the predicted sharpened map compared to the profiles of the unsharpened map an improvement in the correlation can be used to argue that the sharpening method results in a sharpened map which more closely resembles reality.

4.6.3. Realspace cross-correlation

Another method of quantifying the quality of a sharpened map is by calculating the cross-correlation between the sharpened map and an atomic model map simulated at zero B-factor. Where the atomic model map is a simulated map derived from the PDB. When this map is simulated at a B-factor of zero it would represent the theoretical best possible map. Note that this does not mean that it is the best possible sharpened map from the given data. Just the best possible map associated with the physical structure of the macromolecule itself. Thus the correlation will only equal one in the case of a theoretical map (which is not possible in practice as a result of imperfections in gathering the data). But having a higher Realspace cross-correlation does indicate that a map is of higher quality.

4.6.4. FDR-masking and Atomic model mask comparison

When using FDR-masking as a result of issues with the dataset would be left with unusable radial profiles which clearly did not correspond to parts which should be left inside the protein. More about this can be read in the Discussion section denoting the limitations of the dataset. Luckily since it is quite easy to remove these radial profiles it is possible to do a fair assessment of both the capabilities of the network trained on an FDR-masked dataset as well as a network trained on an atomic model mask dataset. Giving a slide edge to the FDR-based dataset (likely caused by the dataset being far larger in size). Resulting in a loss of 0.3396 versus 0.5963 and validation loss of 0.7111 versus 0.4837 and a test loss of 0.6152 versus 0.4379 for the FDR and atomic model masked datasets respectively. Although this would indicate using an atomic model mask-based dataset would result in a worse network, when the trained network is actually implemented in iLocScale. It shows slightly better results as displayed in figure 4.7. Figure 4.7 showing a small but existent increase in sidechains. Making AMM-trained iLocScale slightly superior to FDR-trained iLocScale. Correlating the B-factors and local resolution over all voxels in the atomic model mask yielded a marginally better correlation when using the model trained on the radial profiles in the atomic model mask of 0.1273 against 0.1207 (Δ =

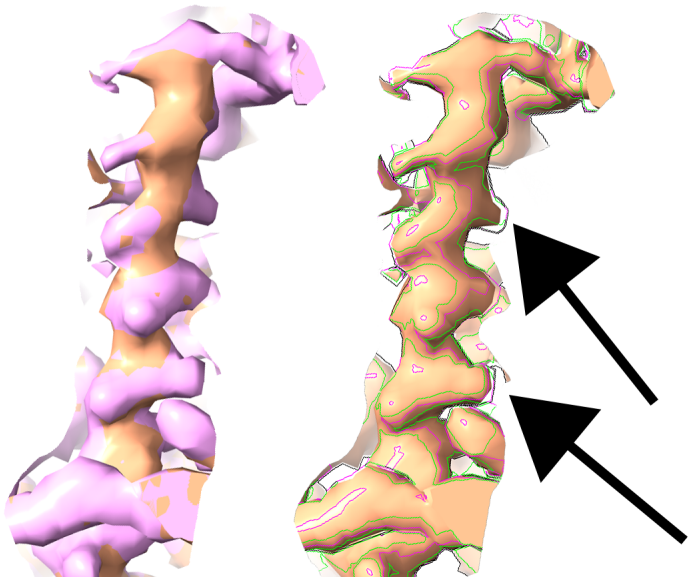


Figure 4.7: EMDB 4997 α-helix showcased for atomic model mask trained iLocScale (pink) and FDR trained iLocScale (orange). Left image shows a superposition of both images. Right image showing FDR-trained iLocScale with AMM-trained iLocScale contour superimposed. Arrows highlighting areas of the helices where AMM-trained iLocScale features the sidechains more prominently.

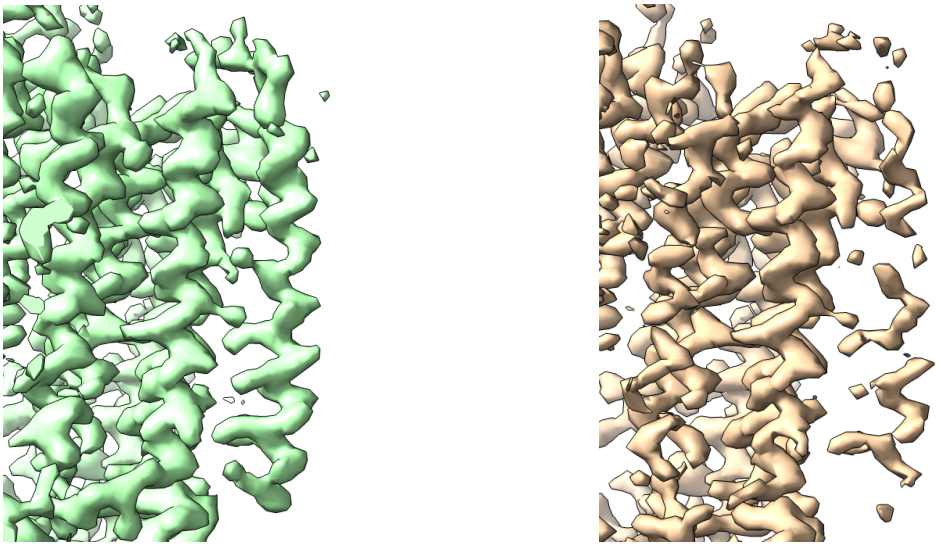
0.0066) for the model trained on FDR radial profiles. The difference increases when using an FDR-mask to 0.0393. Where the average correlation coefficients equal 0.3274, and 0.2881 for the atomic model mask and FDR-trained networks respectively. The table containing all results can be found in Appendix A.2. Showcased in table A.4 and A.5. As the model trained on the radial profiles gathered using the atomic model mask, yields better results it will be used in all further results and comparisons mentioned.

5

Results

5.1. Activation function comparison

The main upside of using a linear activation function is the fact that it mathematically describes the physical reality. Where a B-factor and linear function are used to produce a sharpened radial profile from an unsharp one. This description fits well inside the Wilson region while failing in the Guinier region where the relationship is quadratic. And especially since the first point is very much dependent on the number of atoms, estimating this from a previous radial profile is not accurately possible by using a linear relation. Thus by replacing the first two layers for PReLU[10] layers it is possible to not only lower the overall loss. But specifically, lower the loss in the Guinier region yielding better connectivity in the sharpened protein map. As shown in the comparison made in figure 5.1.



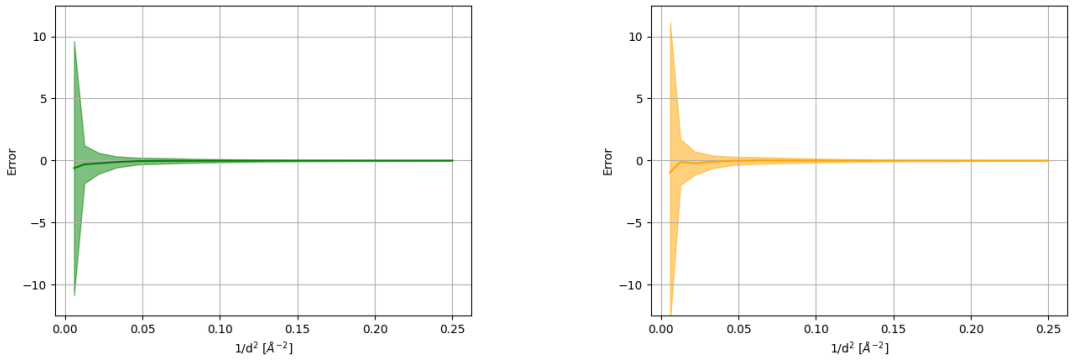
(a) iLocScale sharpened map using both PReLU and linear activation functions

(b) Figure showing iLocScale just using linear activation functions.

Figure 5.1: Comparison of using both PReLU and linear activation functions, figure 5.1a. And only using linear activation functions, figure 5.1b. Demonstrating that using only linear activation functions leads to a lack of connectivity at in lower-resolution areas.

Even though the loss overall does improve, it mainly improves in the Guinier region as suggested by the increase in connectivity and actually worsens a bit in the Wilson region. This is shown by the fact that the maps sharpened using the combination of activation functions are a little less sharp than maps sharpened solely using linear activation functions in their densely connected hidden layers.

This is also numerically confirmed when looking at the error in the predictions as a function of frequency as shown in figure 5.2. Showcasing how the non-linear model has much lower errors for the low-frequency area.



(a) Error from the non-linear model predictions as a function of frequency.

(b) Error from the linear model predictions as a function of frequency.

Figure 5.2: Comparing the errors between the non-linear (left) model and linear (right) model. Demonstrating how the non-linear model has a higher accuracy at a lower frequency. The shade shows one standard deviation.

Although the error does improve. It still remains quite high, this is a result of the non-linear model not perfectly fitting the theoretical shape of the radial profile. Thus optimizing the result will still leave errors larger than required. This can be solved by using a custom loss function which may better describe this region as mentioned in section 6.4.1. Since it is almost impossible to compare the errors in the higher-frequency domain. Table 5.1 has been added to show the mean of the ratio between the error of both the radial profile predictions. Where a value less than 1 implies that the non-linear model is more accurate at a given frequency, while a value larger than 1 implies that the linear model performs better.

Frequency ($1/d$[Å^{-1}])	0.0769	0.1122	0.1474	0.1827	0.2179	0.2532	0.2885
Ratio	0.1522	0.8151	1.6302	3.5653	2.1144	1.7264	1.1404

Frequency ($1/d$[Å^{-1}])	0.3237	0.359	0.3942	0.4295	0.4647	0.5
Ratio	1.0642	0.8539	1.0693	1.3943	1.3135	1.614

Table 5.1: Table showcasing the accuracy of the linear and non-linear model at a given frequency. The ratio has been calculated by taking $E\left[\frac{error\ non-linear\ profile}{error\ linear\ profile}\right]$ at a given frequency.

Where table 5.1 shows how at a low frequency the error for the non-linear profiles is much smaller than those for the linear profile (as indicated by a ratio of 0.1522 and 0.8151 for the first two frequency bins). But also what might appear strange at first sight at a frequency of 0.359 where the non-linear model also outperforms the linear model. This might be a result of secondary structures, adding some non-linearity in the high-frequency domain. As both functions have their own strengths and weaknesses in different regions resulting in an optimal solution where both models are used.

5.2. Merging multiple predictions

From section 5.1 it is clear that the two proposed models with different activation functions have their own advantages and disadvantages. Logically hypothesising a more optimal map consisting of the low-frequency part of the model using both PReLU and linear activation layers while using the high-frequency part of the model solely consisting of linear activation functions. This can be done by using the method described in section 3.2.4. By implementing this method in iLocScale it is possible to leverage both the increased connectivity and sharpness of the individual models as shown in figure 5.3. Making use of two models instead of one results in the prediction step taking place twice as well as introducing another step of merging the profiles. Resulting in approximately 20 to 25% more time required for sharpening the entire test dataset.

5.3. Application in iLocScale

In iLocScale by using a merged predicted radial profile, a vast improvement in overall sharpness can be observed in comparison to the unsharpened maps, without any loss in connectivity. In the following chapters,

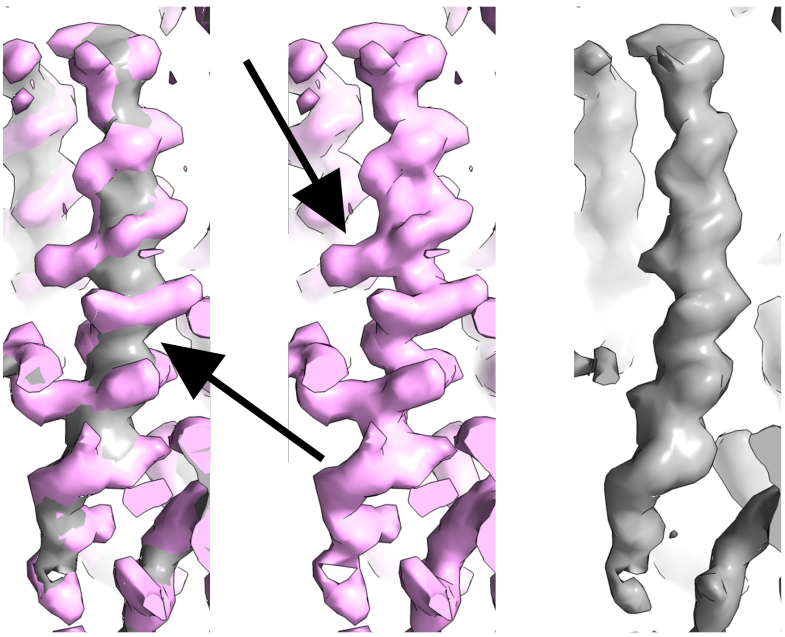


Figure 5.3: α-helix of 4997 showcasing a comparison between iLocScale (pink) trained on the radial profiles gathered using the atomic model mask and the unsharpened map (grey). Showing more prominent sidechains as well as being less blurry.

the different methods of evaluating iLocScale both quantitatively as well as qualitatively are shown.

5.3.1. Comparing test proteins

The sharpened maps are visually slightly worse than LocScale in the modeled regions. But still way more interpretable than the unsharpened maps previously shown in figure 5.3. A comparison between the LocScale and iLocScale sharpened maps is shown in figure 5.4, showcasing a modelled region. Where LocScale is slightly sharper than iLocScale. While figure 5.5 shows a comparison between iLocScale and LocScale for a protein with unmodelled regions at the end. Showcasing how iLocScale would sharpen these regions while normal LocScale would mute this area.

Another example showcasing the difference between iLocScale and traditional LocScale can be observed in figure 5.6. Showcasing how the lipidbelt around EMDB 9610 will clearly be shown when sharpened using iLocScale while also being removed when using LocScale.

5.3.2. Correlation to local resolution

As literature suggests that an increase in a correlation between the B-factor and local resolution. Would represent a map more accurately showcasing the physical characteristics of the protein. Makes looking at the changes in correlation a great measure of highlighting if a map is a more or less accurate representation of reality. In figure 5.7 the B-factor of 0311 is plotted against its local resolution, for a sample ($n = 500$) of the voxels included in the fdr filter. Clearly demonstrating an increase in correlation for both iLocScale and LocScale compared to the unsharpened map. An example of iLocScales' B-factors and correlation have been plotted in figure 5.9 and figure 5.10. Showcasing how the iLocScale sharpened map for EMDB 20220 has higher B-factors on the outside of the molecule where the resolution is also worse. While having lower B-factors in the more interior areas of the molecule where the resolution is better.

As image 5.7 would suggest there is a strong correlation to be found between the iLocScale and LocScale B-factors. While the unsharpened B-factors seem to be unrelated. This is confirmed by figure 5.8. Where for each EM-map in the test dataset a sample of 1000 iLocScale B-factors is plotted and compared to the LocScale

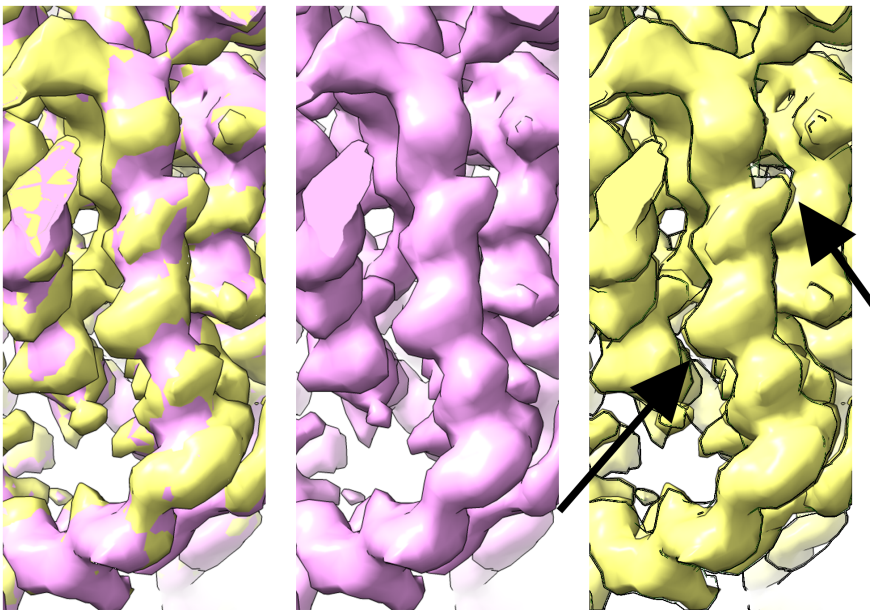


Figure 5.4: Comparison between iLocScale (pink, middle image) and classical LocScale (yellow, right image) in a modelled α-helix of EMDB 4997. iLocScale edges are traced over the LocScale image. Showing that LocScale is slightly sharper.

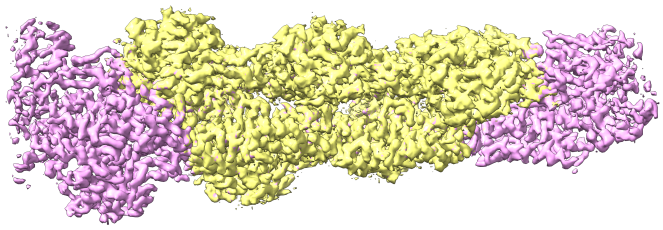
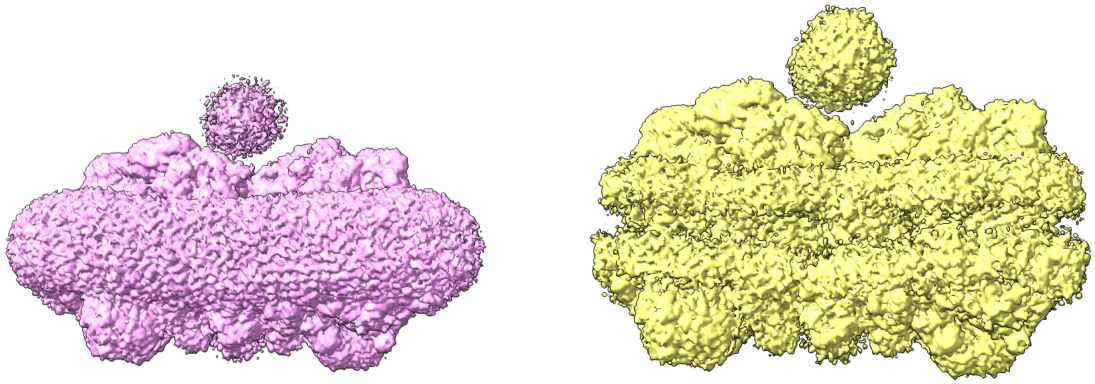


Figure 5.5: Comparison between iLocScale (pink) and classical LocScale (yellow) for EMDB 10365. Where the ends are unmodelled and thus have been completely removed in traditional LocScale.



(a) EMDB 9610 sharpened using iLocScale. Clearly showing the lipid belt around the middle of the structure.

(b) EMDB 9610 sharpened using LocScale, where the lipidbelt in the middle has been partially removed as it is an unmodelled region.

Figure 5.6: Visual comparison between iLocScale (left) and LocScale (right) sharpened maps of EMDB 9610. Highlighting how iLocScale handles unmodelled regions without a problem.

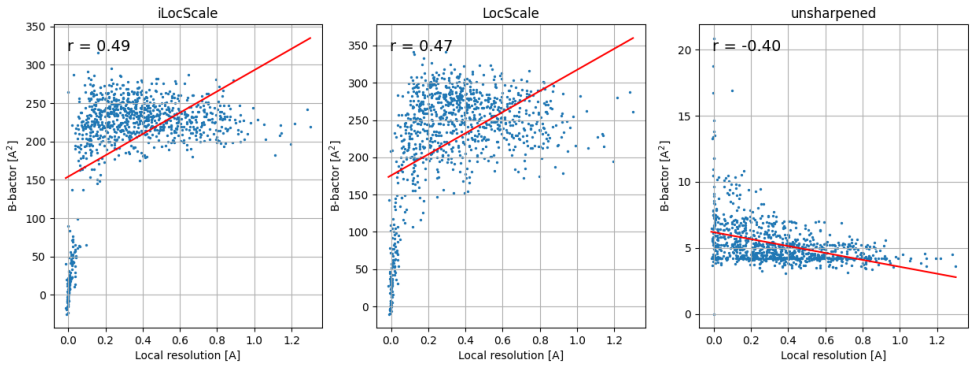


Figure 5.7: From left to right: iLocScale, LocScale and the unsharpened map. Showcasing the increase in correlation between the B-factor and local resolution for both iLocScale (0.49, $\Delta = 0.89$) and LocScale (0.47, $\Delta = 0.87$) over the unsharpened map (0.40). For EMDB 0311,

B-factor, as the LocScale radial profiles have been used as targets for training purposes. In general, the correlation seems very good. Except for 4141 where it seems to be off, this might be caused by its low resolution (of 6.7Å). More details for each EM-map in the test dataset on the correlation of B-factor comparison to the target have been given in Appendix A.3.

In all tested cases the correlation improves when using the AMM version of iLocScale compared to the unsharpened map. While the average correlation coefficient for all maps and their voxels in the FDR-mask improves from 0.0465 to 0.3416 ($\Delta = 0.2951$, Δ refers to the difference compared to iLocScale unless otherwise noted). Even forming a drastic improvement over classical LocScale (0.1429, $\Delta = 0.1987$). It is expected that the correlation coefficients compared to LocScale would improve as LocScale mutes unmodelled areas. The average performance in modelled areas is almost similar on average, when looking at the average correlation coefficients for the voxels in the atomic model mask insuring the areas are modelled the difference between iLocScale and LocScale decreases with the averages for iLocScale and LocScale equaling 0.1482 and 0.1215 ($\Delta = 0.0267$). Although the difference is highly dependent on the specific EM-map, with traditional LocScale being vastly superior for 282 ($\Delta = -0.1531$), 311 ($\Delta = -0.1818$), 560 ($\Delta = -0.2868$), 10365 ($\Delta = -0.5939$), losing out to iLocScale for 4571 ($\Delta = 0.5321$), 4997 ($\Delta = 0.1690$), 7127 ($\Delta = 0.1859$), 8702 ($\Delta = 0.1972$). With a relatively small difference in other test maps. For all correlation coefficients see table A.4 and table A.5 in Appendix A.2.

The differences for both versions of iLocScale, traditional LocScale and the unsharpened maps for both the voxels in the FDR and atomic model mask are shown in table 5.2.

AMM v. fdr trained		AMM v. locscale		AMM v. unsharpened		locscale v. unsharpened	
AMM	FDR	AMM	FDR	AMM	FDR	AMM	FDR
0.0490	0.0845	0.0268	0.1987	0.1800	0.2951	0.0465	0.0965

Table 5.2: Table showing the correlation coefficients of the B-factor and local resolution for all voxels in each EM-map averaged over all EM-maps, rounded to 4 numbers. FDR refers to the AMM and FDR trained refers to the implementations of iLocScale trained on either the atomic model mask or FDR radial profiles.

As shown in table 5.2. iLocScale beats LocScale overall in the FDR region as these may contain unmodelled parts of the protein. Except for 0311 and 8702 being the only cases where LocScale shows a significant improvement in correlation coefficients over iLocScale.

Appendix A.2 contains more information on the correlation coefficients showcasing how the B-factors correlate to the local resolution for each EMDB in the test dataset.

5.4. Realspace cross-correlation

The real space cross-correlation showcases at which level the map is similar to the atomic model map. These results have been shown in table 5.3.

From table 5.3 it follows that the RSCC is slightly better with a mean difference of 0.003, for iLocScale compared to normal LocScale. iLocScale vastly outperforms the unsharpened maps in terms of RSCC, with a mean difference of 0.0668. And also slightly outperforms EMmernet MBfa, on average with 0.0097 Although

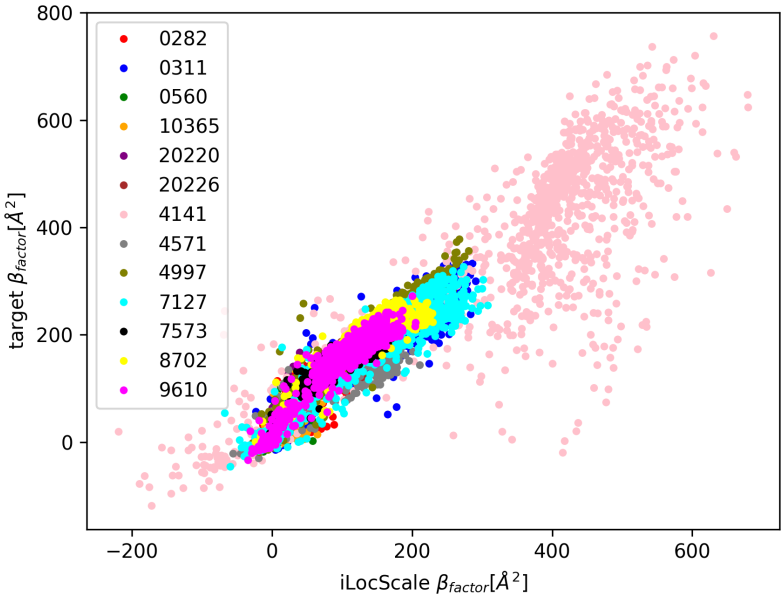
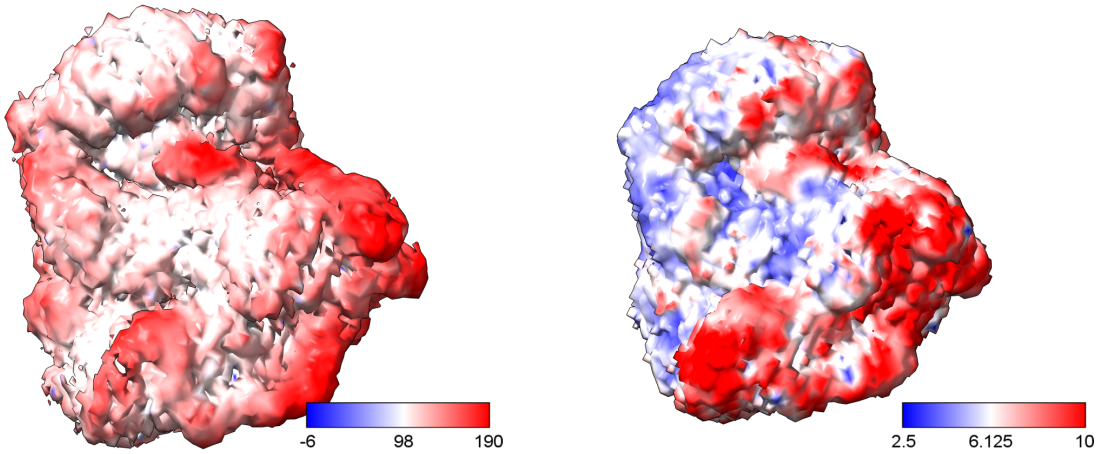


Figure 5.8: iLocScale B-factors taken from each map ($n = 1000$). And plotted against the target (LocScale B-factors, y-axis).

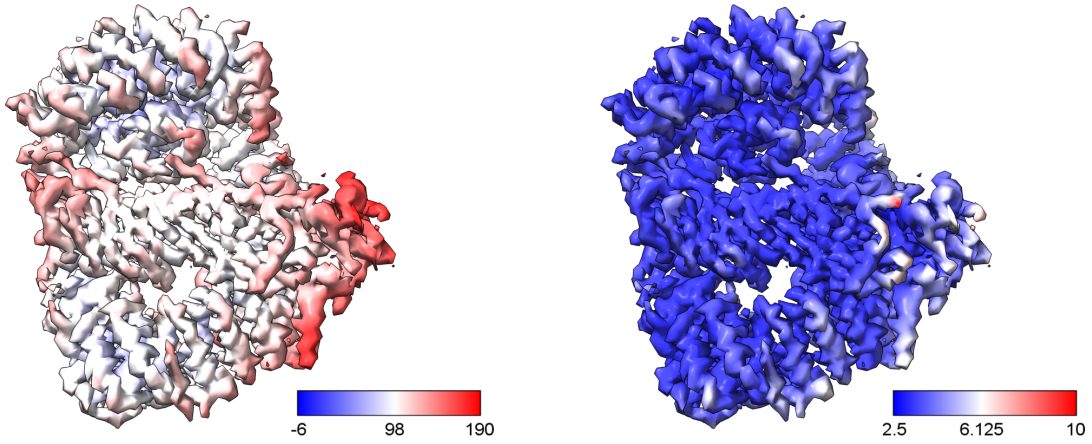


(a) iLocScale sharpened map coloured using B-factor surface colours.

(b) iLocScale sharpened map coloured using local resolution surface colours.

Figure 5.9: Visual representation of B-factors and local resolution. Outside of the molecule containing higher B-factor and lower resolution areas as well as matching variations with the bottom and right having higher B-factors/resolution while the middle and top left have better resolution and lower B-factors.

the RSCC was taken with respect to the atomic model map thus it will only include modelled areas. It is still able to show how iLocScale is almost as sharp as LocScale in these modelled regions with the average difference between iLocScale and LocScale is almost 1/25th of the difference between iLocScale and the unsharpened map in terms of RSCC. In all maps, except for EMDB 4141 the RSCC increases significantly compared to the unsharpened map.



(a) iLocScale sharpened map coloured using B-factor surface colours.

(b) iLocScale sharpened map coloured using local resolution surface colours.

Figure 5.10: Inside of the molecule, both being at a higher resolution as well as relatively lower B-factor compared to figure 5.9.

EMDB	ilocscale	locscale	Emmernet MBfa	Unsharp	iloc v. loc	iloc v. Em Mbfa	iLoc v unsharp
282	0.5873	0.591	0.5777	0.5248	-0.0037	0.0096	0.0625
311	0.3812	0.3925	0.3759	0.3511	-0.0113	0.0053	0.0301
560	0.681	0.6896	0.6798	0.608	-0.0086	0.0012	0.073
10365	0.6259	0.624	0.5978	0.5737	0.0019	0.0281	0.0522
20220	0.5835	0.53	0.5838	0.5023	0.0535	-0.0003	0.0812
20226	0.5434	0.4992	0.5599	0.4482	0.0442	-0.0165	0.0952
4141	0.3679	0.3801	0.3947	0.3673	-0.0122	-0.0268	0.0006
4571	0.6118	0.6142	0.607	0.5737	-0.0024	0.0048	0.0381
4997	0.4787	0.486	0.4455	0.3973	-0.0073	0.0332	0.0814
7127	0.4715	0.4778	0.4518	0.4219	-0.0063	0.0197	0.0496
7573	0.5187	0.5264	0.4996	0.4276	-0.0077	0.0191	0.0911
8702	0.4241	0.388	0.3974	0.328	0.0361	0.0267	0.0961
9610	0.5525	0.5897	0.5301	0.4355	-0.0372	0.0224	0.117
AVG	0.5252	0.5222	0.5155	0.4584	0.003	0.0097	0.0668

Table 5.3: Table displaying real space cross-correlation for the test EM-maps with an atomic model map simulated at zero B-factor.

6

Discussion

iLocScale has been shown to produce maps which are much sharper than their original counterpart. As well as outperforms LocScale in unmodeled regions and is better than LocScale in computing time as it is able to skip the slow step of refining a PDB. However, iLocScale appears to slightly underperform LocScale in the modeled regions based on visual inspection of the maps and RSCC analysis. This is also to be expected, as it is using LocScale maps as target maps. The only potential way of creating maps sharper than LocScale is by producing a model based on better radial profiles (which seems unlikely). Or by combining it with another method to produce a more effective process.

6.1. Risks involving Machine Learning

One of the primary risks of the used approach is that even though the risks of using machine learning have been minimized there may still be a chance that the resulting predictions could be a result of overfitting and thus may not yield the most accurate results. Although by reducing the number of epochs and adding regularizers, in practice removing overfitting from this implementation of iLocScale. When building upon these findings, there is a risk of overfitting returning and potentially decreasing the effectiveness of a newly developed model. There are also other risks with using machine learning, namely the fact that the way the model will behave is still not as predictable as a traditional method like LocScale.

6.2. Limitations of the dataset

The quality of the dataset has a significant impact on the resulting accuracy of a trained network, which is a major disadvantage of using machine learning. When creating the training, validation and testing dataset there were many radialprofiles included after FDR-filtering which would be of low quality, this problem did not exist while using the atomic model mask. But using an atomic model mask has its own drawback since it would filter out way more cubes than an FDR-mask would, leaving a smaller dataset. These bad radial profiles from the fdr dataset look like the example radial profile shown in figure 6.1.

From these profiles, it becomes clear at first glance that these do not look like normal radial profiles. This problem was quite common when using the fdr dataset and might be caused by cubes coming from regions where there is a single atom existent in this cube (also explaining why the first point would be 0 (as $ln(1) = 0$, indicating the existence of 1 atom in this cube), but nothing else. Still allows the cube to get thru the FDR filter (as it technically contains signal) and also results in every other point being sent towards 0 (in a non-logarithmic scale). This also results in a very low value in a logarithmic-based scale as can be seen in figure 6.1. This would yield very high loss values, as the model will mainly focus on predicting these errors since being off when guessing whether something will almost randomly be sent towards either -6 or -12 (in a log plot this would be equivalent to sending the other values to zero in non-logarithmic scale) would result in a large loss when taking the mean squared error of this difference. While optimizing the exact shape of a radial profile becomes insignificant since the order of scale is so much smaller in the differences between the theoretical, unsharpened and predicted radial profiles. The difference between a sharpened and unsharpened radial profile tends to be less than 1. Thus the effect of learning how to sharpen profiles is outweighed by the model's desire to accurately predict to which negative value the profiles are sent to as it would have significantly more effect on the MSE.

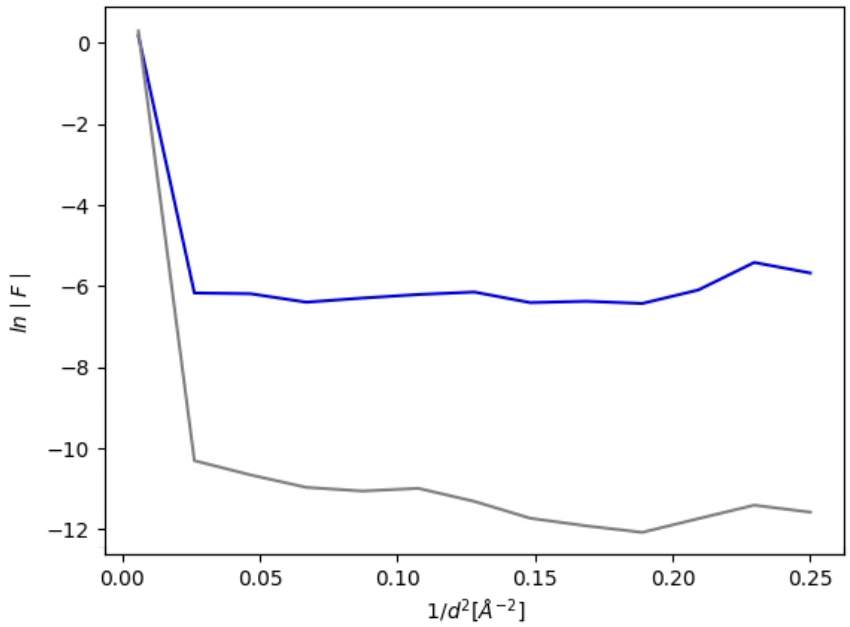


Figure 6.1: Example plot showing problematic radial profiles, which are included in the FDR-masking-based dataset. On the y-axis, the radial intensity is in logarithmic scale. And on the x-axis, the point of the radial profile is numbered. Clearly showing that only the first point (x=0), has a value of 0 (thus corresponding to one atom in the box) while all the other values are incredibly low.

To check if this issue also existed in iLocScale, a binarized output map has been produced using the same filter, which is almost empty. Suggesting that this issue does not exist in the implementation in the pipeline of iLocScale when the windows are taken using the (in LocScale and iLocScale implemented methods, only when using the pre-processing steps as described in the Approach section). and is purely part of the training, validation and testing dataset. Finding the cause of this problem would require further research as the highpass filter is practically able to solve the problem. This can be implemented by insuring all points in the radial profile are at least -6. By removing the radial profile which contains any point below -6. As seen in chapter 5, using an atomic model mask yields better results even with a relatively small dataset compared to when using FDR-masking, although only slightly. This may be caused by the exclusion of these previously mentioned radial profiles. Or because using FDR-masking also includes radial profiles of unmodelled regions where the target profile is of lower quality.

6.3. Combining multiple radial profiles

As shown in chapter 5 Results there are two options for loss functions proposed, both coming with their own advantages and disadvantages. A model based purely on linear activation functions will result in a more physically accurate prediction in the high-frequency areas of the radial profile. Resulting in more accurate B-factors as well as a sharper over map. Compared to using both PReLU and linear activation functions. Although doing so will result in a decrease in connectivity. For example, resulting in disconnected α-helices. One of the potential solutions to this problem would be by changing the implementation in LocScale such that both models would be used. In such a case both profiles would be merged together using the high-frequency parts of the model with the linear activation functions. While the low-frequency parts of the model use both PReLU and linear activation functions is used.

Although this would increase the overall runtime of LocScale by a significant margin the step of predicting the optimal radial profiles would have to take place twice. Even though it would still result in an approximate decrease of 75% in runtime compared to traditional LocScale.

6.4. Outlook

6.4.1. Custom activation functions

A potentially better solution than merging multiple radial profiles would be using a custom activation function. Designed to fit both the quadratic part in order to the linear part perfectly. This can be done by using a function in the form of equation (6.1).

$$f(x) = \begin{cases} \log(c - dx), & x < 0 \\ ax + b, & x > 0 \end{cases} \tag{6.1}$$

Where a, b, c, d are such that the function would work well for all frequencies. And are tweaked by the model. How such an activation function would behave has been shown in figure 6.2. As the shown image has quite a sharp bend, it might be best to also make use of weights to smooth out this transition in a similar fashion as explained in section 3.2.4.

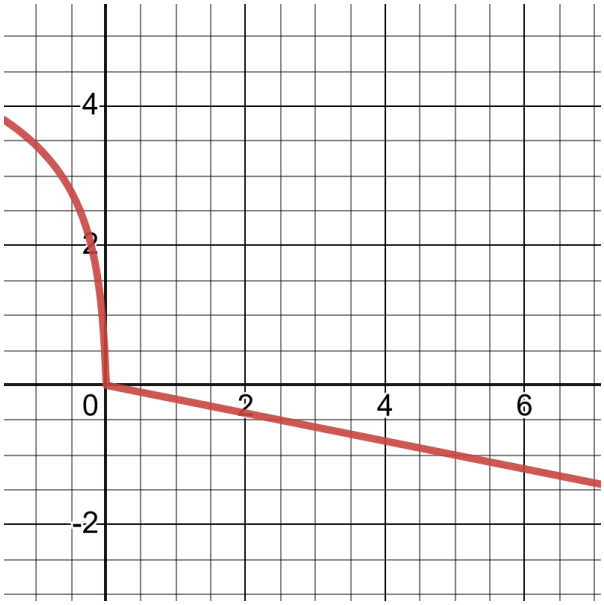


Figure 6.2: Graph showcasing how the custom activation function would behave. $log(c - dx)H(-x) + [ax + b]H(x)$, illustrated using $a = -0.2$, $b = 0$, $c = 1$, $d = 30$. With $\log|F|$ on the y-axis and $1/d^2$ on the x-axis.

While using custom activation functions is outside the scope of this research. It may very well be the best option for improving iLocScale in a sense of combining both the better sharpening characteristics of using linear activation functions as well as the increase of connectivity by giving a better estimate for the low-frequency domain of the radial profile. It might also improve the speed of iLocScale as it would cut the number of predictions required in half. Although this step takes relatively little time and it would only yield marginal benefits. What might result in a major time reduction is removing the step required for applying the radial profiles. The resulting reduction could yield a reduction for running the entire test set thru iLocScale from 3 hours and 30-40 minutes to about 3 hours.

6.5. Merging iLocScale and LocScale

Since LocScale is slightly sharper in the modelled areas but fails in the unmodeled areas. It may be possible to create a method both utilizing LocScale for the modeled areas and iLocScale for the unmodelled areas as it is almost as sharp as LocScale for the best possible results. Although this would still be a slow method as the PDB will still need refinement. It would yield a slightly more sharp result and might be of higher quality than the model-free version of LocScale.

7

Conclusion

It is possible to make use of machine learning to remove the dependencies of LocScale on an atomic model. By predicting the radial profiles from unsharpened radial profiles, keeping the phases untouched.

iLocScale works by combining two different models; one model is based on linear activation functions and is used for predicting the high-frequency domain of the radial profile, leveraging the linear models' ability to produce the highest level of sharpness. While the lower-frequency domain is predicted using a network consisting of both PReLU and linear activation functions, resulting in a better prediction, especially at zero frequency resulting in improved connectivity compared to using a linear model. The linear and non-linear predicted radial profiles are merged together around the point of 7Å using multiplication with weighted functions insuring the final merged profile is continuous.

iLocScale offers a vast improvement over the unsharpened map shown both in much more sharp maps with more prominent sidechains. As well as an improved correlation between the B-factors and local resolution, suggesting that they form a better representation of reality. With an average difference in the correlation coefficient of 0.2951 over the entire test dataset.

The modelled regions of LocScale are visually slightly sharper than iLocScale although they both result in an almost similar average correlation coefficient (0.1482 versus 0.1215). Although LocScale mutes the unmodelled regions, iLocScale handles them correctly sharpening them in the same manner as modelled regions. Resulting in iLocScale outperforming conventional LocScale in these regions. For the modelled regions, the real space cross-correlation of LocScale and iLocScale is similar with a difference of just 0.0030 in favour of iLocScale. While iLocScale vastly outperforms the unsharpened maps by an average difference in RSCC of 0.0668.

iLocScale works independently of an atomic model. Thus resulting in the ability to skip the slow and tedious refinement process which would be required in traditional LocScale. Results in lowering the processing time from about 45 minutes to 10 minutes.

In the future, it might be possible to improve iLocScale by using a custom loss function as it might result in a better representation of reality. The radial profile consists both of a quadratic part in the low-frequency domain and in a linear part in the high-frequency domain. Resulting in an activation function which describes the radial profile physically perfectly and thus makes it possible to remove the current methods dependency on two networks in favour of a single network.

It might also be possible to implement iLocScale and LocScale together forming a method where LocScale is used in the regions where an atomic model is available, utilizing its slight edge in sharpness over iLocScale. While using iLocScale for the unmodelled regions.

A

Appendix

A.1. Appendix I

The dataset has been split into a training, validation and testing dataset. The EMDBs and PDBs can be found in table A.1, table A.2 and table A.3 for the validation, testing and training EMDBs respectively.

EMDB	PDB	Resolution
0193	6hcg	4.3
0257	6hra	3.7
0264	6hs7	4.6
0499	6nsk	2.7
10401	6t8h	3.77
20849	6uqk	3.77
4611	6qp6	3.2
4646	6qvb	4.34
4733	6r69	3.65
4789	6rb9	3.2
7133	6bqv	3.1
7882	6dg7	3.32
8069	5i08	4.04
9112	6mgv	3.1
9298	6mzc	4.5
9374	6nhv	3.5

Table A.1: Table showcasing the EMDBs (n=15) and their associated PDB as well as resolution used for validating the model.

EMDB	PDB	Resolution
0282	6huo	3.26
0311	6hz5	4.2
0560	6nzu	3.2
10365	6t23	3.1
20220	6oxl	3.5
20226	6p07	3.2
3545	5mqf	5.9
4141	5m1s	6.7
4571	6qk7	3.3
4997	6rtc	3.96
7127	6bpq	4.1
7573	6crv	3.2
8702	5vkq	3.55
9610	6adq	3.5

Table A.2: Table showcasing the EMDBs (n=13) and their associated PDB as well as resolution used for testing the model. As well as used for gathering iLocScale sharpened maps.

EMDB	EMDB	PDB
0026	6gl7	6.3
0038	6gml	3.2
0071	6gve	3.9
0093	6gyn	3.4
0094	6gyo	3.4
0132	6h3c	3.9
0234	6hjn	3.3
0408	6nbd	3.2
0415	6nbq	3.1
4288	6fo2	4.4
0452	6nmi	3.7
0567	6o0h	3.67
0589	6nmi	3.7
0592	6o1m	3.15
0665	6oa9	3.9
0776	6ku9	2.67
10049	6rx4	3.3
10069	6s01	3.2
10100	6s5t	4.15
10105	6s6t	4.1
10106	6s6u	3.5
10273	6sof	4.3
10279	6sp2	3.33
10324	6swe	3.1
10333	6swy	3.2
10418	6t9n	2.96
10534	6tni	3.4
10585	6ttu	3.7
10595	6tut	3.25
10617	6xt9	3.8
20145	6oo4	3.3
20146	6oo5	4.2
20189	6osy	4.3
20234	6p19	3.8
20249	6p4h	3.2
20254	6p5a	3.6
20259	6p62	3.57
20270	6p7v	4
20271	6p7w	4.1
20521	6pxm	2.1
20986	6v0b	4.1
21012	6v1i	3.8
21107	6v8o	3.07
21144	6vbu	3.1
21391	6vv5	3.5
3661	5no2	5.16
3662	5no3	5.16
3802	5of4	4.4
3885	6el1	6.1
3908	6eoj	3.55
4032	5lc5	4.35
4073	5lmn	3.55
4074	5lmo	4.3
4079	5lmt	4.15
4148	5m3m	4
4162	6ezo	4.1
4192	6f6w	3.81
4214	6fai	3.4
4241	6fe8	4.1
4272	6fki	4.3
4401	6i2x	3.35
4404	6i3m	3.93
4429	6i84	4.4
4588	6qm5	3.6
4589	6qm6	3.7
4593	6qma	3.7
4728	6r5k	4.8
4746	6r7x	3.47
4759	6r8f	3.8
4888	6ric	2.8
4889	6rid	2.9
4890	6rie	3.1
4917	6rla	3.9
4918	6rlb	4.5
4941	6rn3	4
7009	6ave	3.7
7090	6bf6	6.5
7335	6c24	3.5
8911	6dt0	3.7
8958	6e1n	3.7
8960	6e1p	3.7
9258	6muw	3.6
9931	6k7g	3.3
9941	6k7m	2.95
9695	6iok	3.64

Table A.3: Table showcasing the EMDBs (n=84) and their associated PDB as well as resolution used for training the model.

A.2. Appendix II

Correlation coefficients of the B-factor and local resolution for all voxels in each EM-map. Rounded to 4 numbers. Using the atomic model mask.

method / EMDB	iLocScale atomic	iLocScale fdr	locscale	unsharpened
282	0.0475	0.0334	0.2006	-0.0516
311	0.1365	0.1055	0.3183	-0.1226
560	-0.1169	-0.1484	0.1699	-0.1094
10365	-0.1973	-0.2415	0.3966	-0.1986
20220	0.4098	0.3543	0.2936	0.2052
20226	0.0719	0.0043	0.0066	-0.1974
4141	0.0495	0.035	-0.0187	-0.0387
4571	0.2825	0.2578	-0.2496	0.1172
4997	0.2325	0.2078	0.0635	0.0872
7127	0.4517	0.4275	0.2658	0.2174
7573	0.2906	0.1968	0.1406	-0.0388
8702	0.3434	0.2124	0.1462	-0.284
9610	-0.0745	-0.1553	-0.1542	0.0013
AVG	0.1482	0.0992	0.1215	-0.0318

Table A.4: Table showing the correlation coefficients of the B-factor and local resolution for all voxels in each EM-map. Rounded to 4 numbers. Using voxels included in the atomic model mask. iLocScale atomic refers to the implementation of iLocScale with a model trained on the radial profiles from the atomic model mask. While iLocScale FDR refers to the model trained using the radial profiles from the FDR mask.

Correlation coefficients of the B-factor and local resolution for all voxels in each EM-map. Rounded to 4 numbers. Using the FDR mask.

method / EMDB	iLocScale atomic	iLocScale fdr	locscale	unsharpened
282	-0.1654	-0.2487	-0.1824	-0.1216
311	0.6006	0.5173	0.7803	0.2284
560	-0.1621	-0.3006	-0.3023	-0.5539
10365	0.3818	0.3309	0.3594	0.3592
20220	0.4233	0.3544	0.3921	0.3233
20226	0.3186	0.2907	0.3258	0.2362
4141	-0.156	-0.2736	-0.3457	-0.381
4571	0.4095	0.3592	0.4812	0.1309
4997	0.5719	0.491	0.1052	-0.0534
7127	0.6171	0.5032	0.3775	0.2099
7573	0.5802	0.4546	-0.4074	-0.2183
8702	0.3032	0.2201	0.5268	0.0865
9610	0.7185	0.6445	-0.2524	0.3581
AVG	0.3416	0.2572	0.1429	0.0465

Table A.5: Table showing the correlation coefficients of the B-factor and local resolution for all voxels in each EM-map. Rounded to 4 numbers. Using the voxels included in the fdr mask. iLocScale atomic refers to the implementation of iLocScale with a model trained on the radial profiles from the atomic model mask. While iLocScale FDR refers to the model trained using the radial profiles from the FDR mask.

A.3. Appendix III

Showcase of the B-factors from the iLocScale map and the unsharpened map compared to the target B-factors. For each individual EMDB in the test dataset as well as the correlation. Every plot has been produced by taking 1000 voxels from the B-factor map picked inside the atomic model mask.

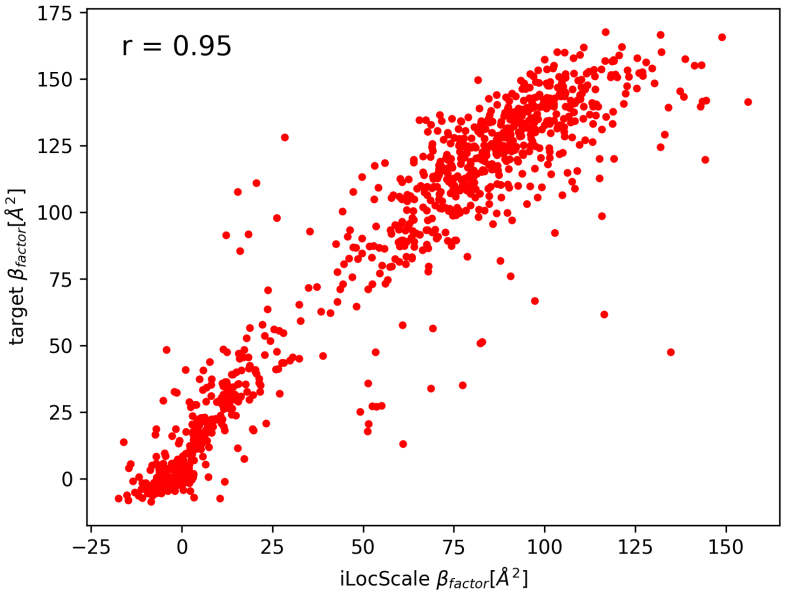


Figure A.1: For EMDB 0282 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

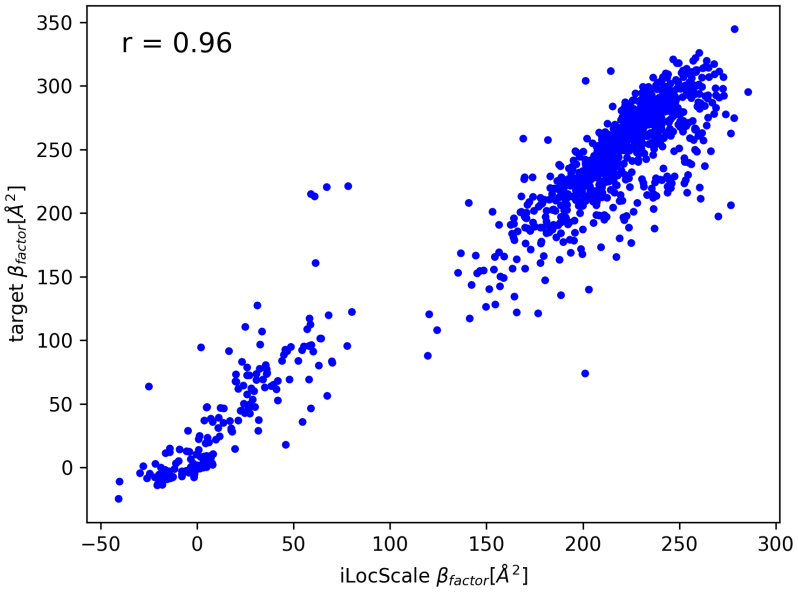


Figure A.2: For EMDB 0311 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

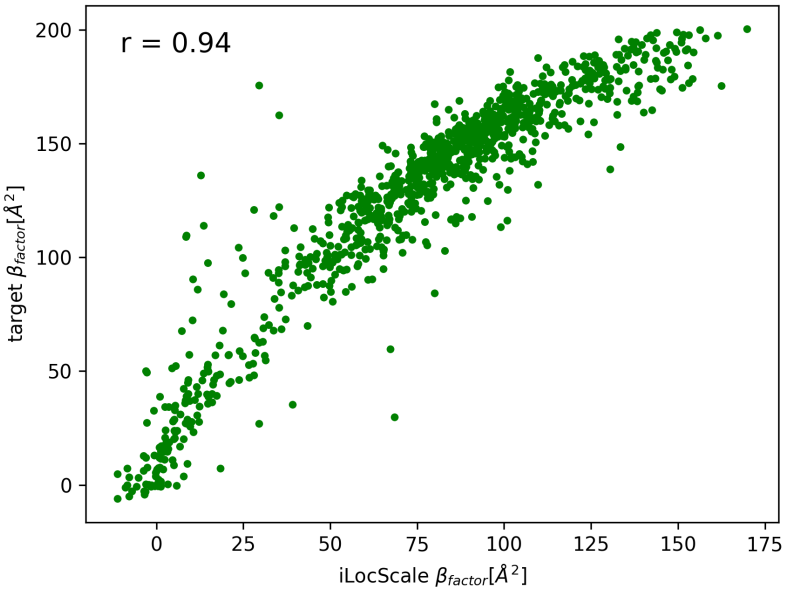


Figure A.3: For EMDB 0560 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

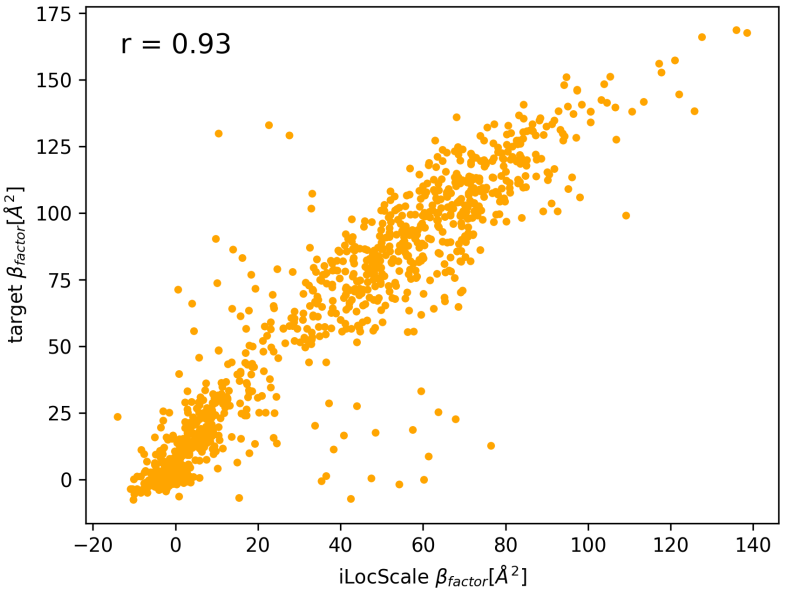


Figure A.4: For EMDB 10365 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

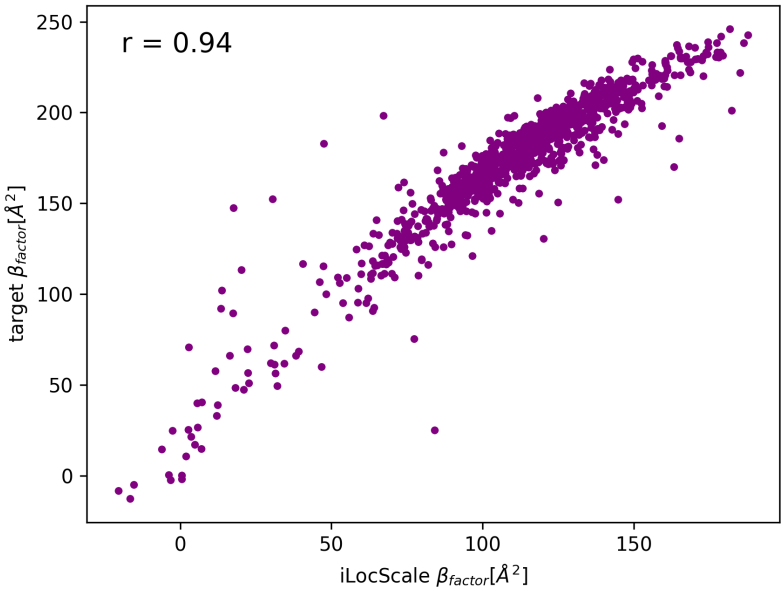


Figure A.5: For EMDB 20220 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

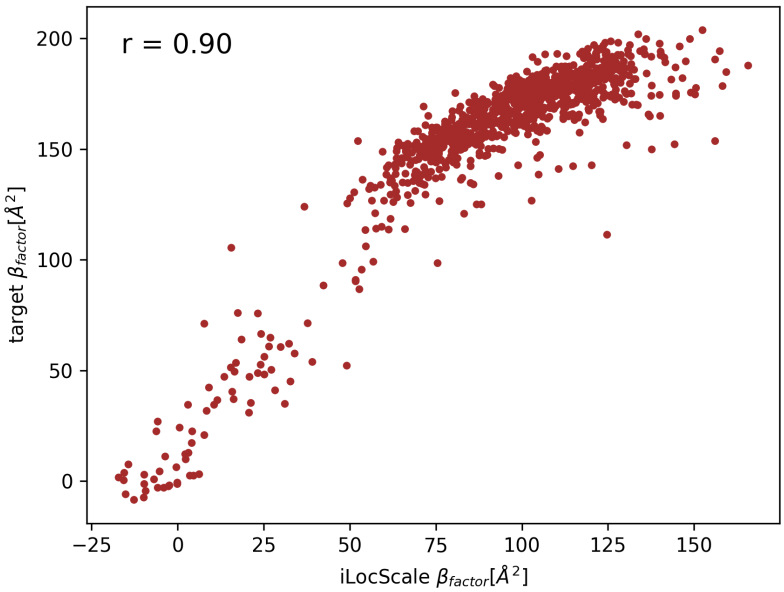


Figure A.6: For EMDB 20226 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

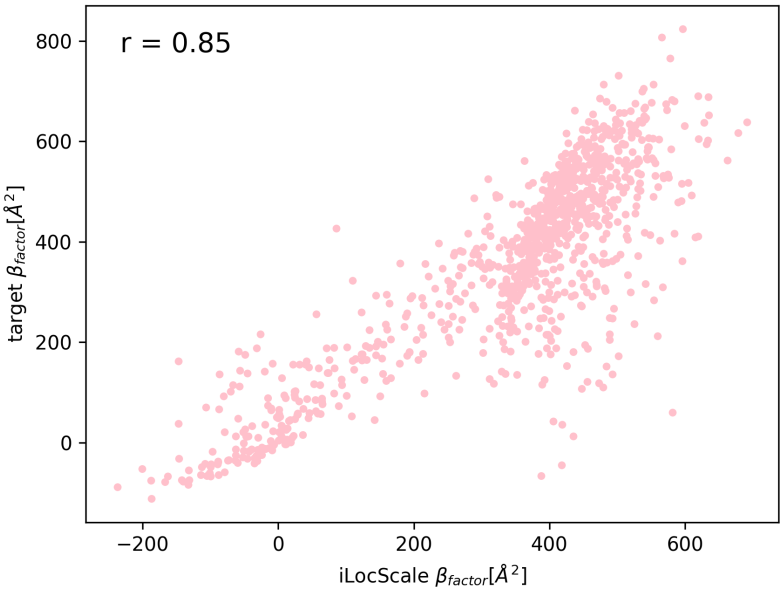


Figure A.7: For EMDB 4141 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

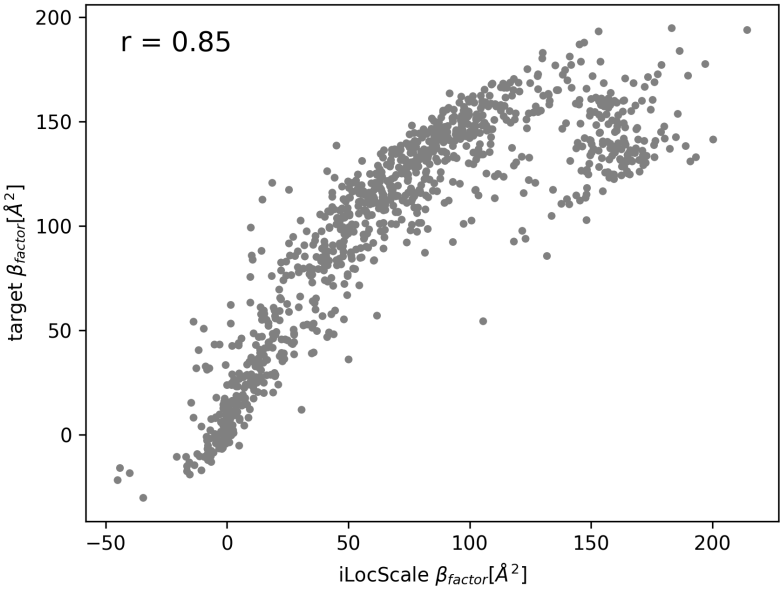


Figure A.8: For EMDB 4571 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

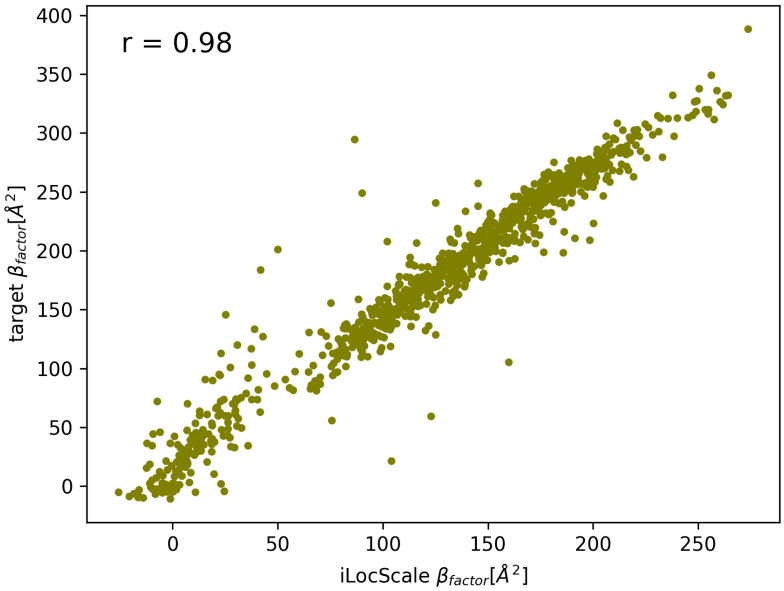


Figure A.9: For EMDB 4997 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compare to the B-factors of the target (LocScale).

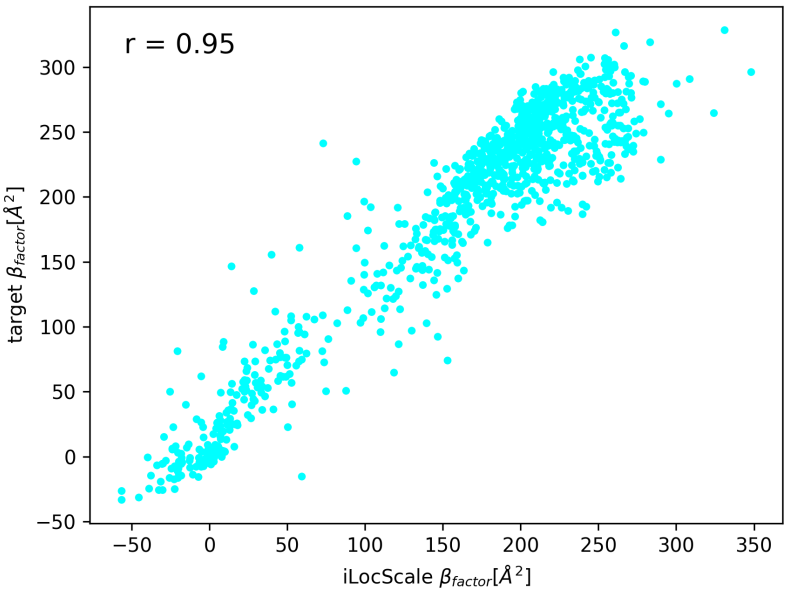


Figure A.10: For EMDB 7127 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

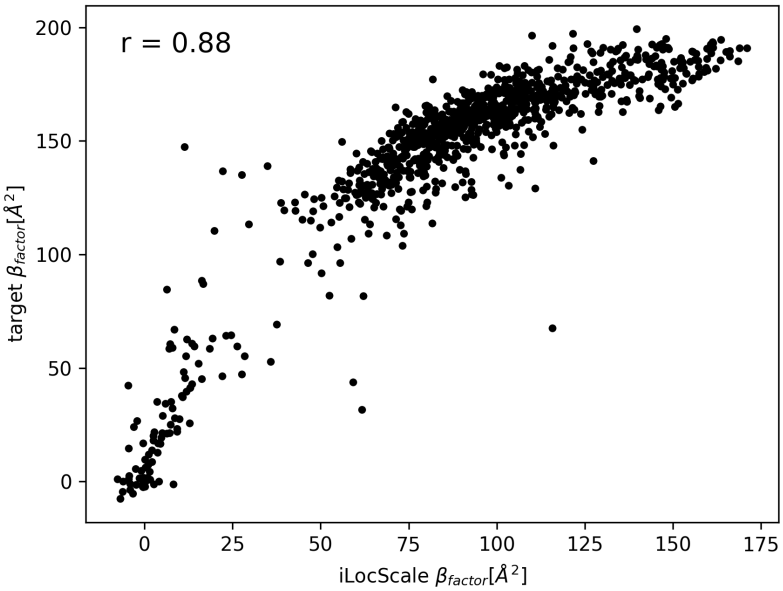


Figure A.11: For EMDB 7573 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compare to the B-factors of the target (LocScale).

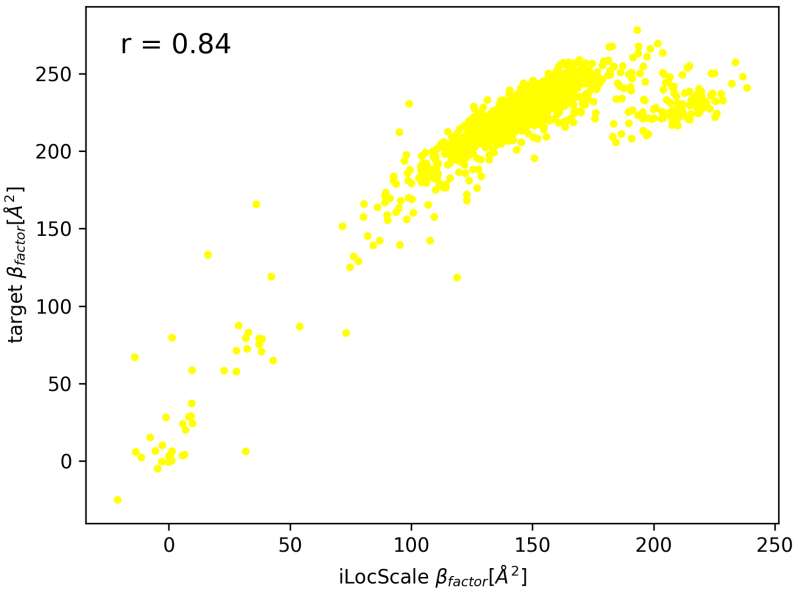


Figure A.12: For EMDB 8702 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

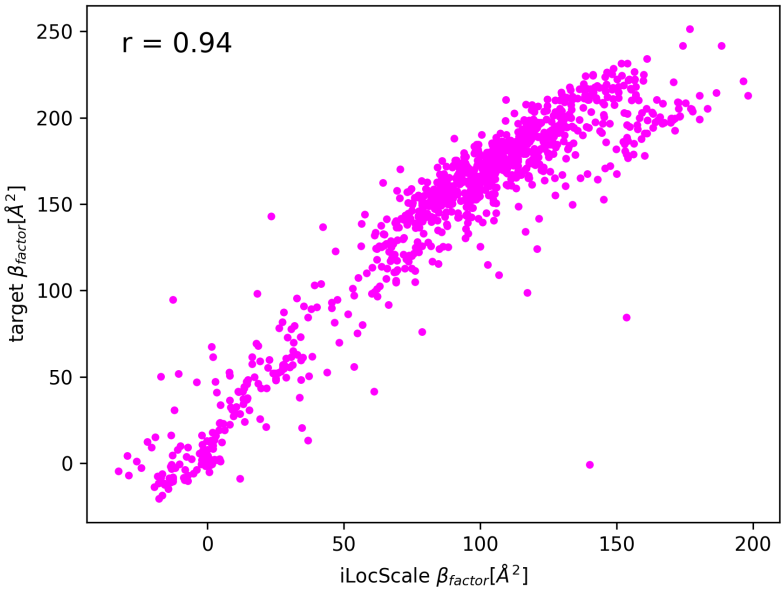


Figure A.13: For EMDB 9610 the scatter plot showcasing how a sample (n=1000) B-factor of the iLocScale map compares to the B-factors of the target (LocScale).

Bibliography

[1] Autonomio Talos [Computer software], 2020. http://github.com/autonomio/talos.

[2] Maximilian Beckers, Arjen J Jakobi, and Carsten Sachse. Thresholding of cryo-em density maps by false discovery rate control. *IUCrJ*, 6(1):18–33, 2019.

[3] Alok Bharadwaj and Arjen J Jakobi. Electron scattering properties of biological macromolecules and their use for cryo-em map sharpening. *Faraday Discussions*, 240:168–183, 2022.

[4] Léon Bottou. Stochastic gradient descent tricks. *Neural Networks: Tricks of the Trade: Second Edition*, pages 421–436, 2012.

[5] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[6] Reinier de Bruin. Catching the invisible with emmernet. 2023.

[7] Ruben Sanchez Garcia, Josue Gomez-Blanco, Ana Cuervo, Jose Maria Carazo, Carlos Oscar S Sorzano, and Javier Vargas. Deepemhacer: a deep learning solution for cryo-em volume post-processing. 2020.

[8] George Harauz and Marin van Heel. Exact filters for general geometry three dimensional reconstruction. *Optik.*, 73(4):146–156, 1986.

[9] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585 (7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[11] Tony Hey, Keith Butler, Sam Jackson, and Jeyarajan Thiyagalingam. Machine learning and big scientific data. *Philosophical Transactions of the Royal Society A*, 378(2166):20190054, 2020.

[12] Arjen Jakobi, Alok Bharadwaj, Maarten Joosten, and Stephan Huber. High resolution imaging (nb4020), 2023.

[13] Arjen J Jakobi, Matthias Wilmanns, and Carsten Sachse. Model-based local density sharpening of cryo-em maps. *Elife*, 6:e27131, 2017.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Koushik Maharatna, Eckhard Grass, and Ulrich Jagdhold. A 64-point fourier transform chip for high-speed wireless lan application using ofdm. *IEEE Journal of Solid-State Circuits*, 39(3):484–493, 2004.

[16] Peter B Rosenthal and Richard Henderson. Optimal determination of particle orientation, absolute hand, and contrast loss in single-particle electron cryomicroscopy. *Journal of molecular biology*, 333 (4):721–745, 2003.

[17] Amit Singer. Wilson statistics: derivation, generalization and applications to electron cryomicroscopy. *Acta Crystallographica Section A: Foundations and Advances*, 77(5), 2021.

[18] Keitaro Yamashita, Colin M Palmer, Tom Burnley, and Garib N Murshudov. Cryo-em single-particle structure refinement and map calculation using servalcat. *Acta Crystallographica Section D: Structural Biology*, 77(10), 2021.