NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

INTERDISCIPLINARY POSTGRADUATE PROGRAM
"INFORMATION TECHNOLOGIES IN MEDICINE AND BIOLOGY"

MSc THESIS

Structure tensor analysis on proteins: efficient feature extraction for heteromultimeric assembly prediction

Melivoia N. Rapti

Supervisors: **Ioannis Emiris,** Professor (NKUA)

Matteo Dal Peraro, Associate Professor (EPFL)

ATHENS

SEPTEMBER 2018

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
"ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗΝ ΙΑΤΡΙΚΗ ΚΑΙ ΤΗ ΒΙΟΛΟΓΙΑ"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάλυση δομικών τανυστών σε πρωτεΐνες: αποτελεσματική εξαγωγή χαρακτηριστικών για πρόβλεψη ετεροπολυμερών συστοιχιών

Μελίβοια Ν. Ράπτη

Επιβλέποντες: **Ιωάννης Εμίρης,** Καθηγητής (ΕΚΠΑ)

Matteo Dal Peraro, Αναπληρωτής Καθηγητής (EPFL)

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2018

MSc THESIS

Structure tensor analysis on proteins: efficient feature extraction for heteromultimeric assembly prediction

Melivoia N. Rapti

R.N.: ΠΙΒ0152

SUPERVISORS: **Ioannis Emiris,** Professor (NKUA)

Matteo Dal Peraro, Associate Professor (EPFL)

EXAMINATION COMMITEE: **Ioannis Emiris,** Professor (NKUA)

Matteo Dal Peraro, Associate Professor (EPFL)

Dimitrios Gunopulos, Professor (NKUA)

September 2018

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάλυση δομικών τανυστών σε πρωτεΐνες: αποτελεσματική εξαγωγή χαρακτηριστικών για πρόβλεψη ετεροπολυμερών συστοιχιών

Μελίβοια Ν. Ράπτη

Α.Μ.: ΠΙΒ0152

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Ιωάννης Εμίρης,** Καθηγητής (ΕΚΠΑ)

Matteo Dal Peraro, Αναπληρωτής Καθηγητής (EPFL)

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:

Ιωάννης Εμίρης, Καθηγητής (ΕΚΠΑ)

Matteo Dal Peraro, Αναπληρωτής Καθηγητής (EPFL)

Δημήτριος Γουνόπουλος, Καθηγητής (ΕΚΠΑ)

Σεπτέμβριος 2018

ABSTRACT

The knowledge of the shape, structure, and interactions of macromolecules, defines biology at the molecular level in atomic detail. Although knowing the architecture is an important step before reaching the knowledge of the function, it still is a challenging task. Current structure resolution techniques (X-ray Crystallography, cryo-EM, etc.), although quite successful, they fail to generalize well across different types of structures, since each one of these methods is designed for specific kinds of components. A way to combine experimental and computational data regardless of their resolution, is through Integrative Modeling (IM), which provides a comprehensive structural characterization of biomolecules. It gets as input (a) high resolution structures of the individual components composing the supramolecular complex, and (b) low-resolution envelopes of native assemblies, resulting in biologically relevant supramolecular assemblies consistent with the available set of experimental data. However, IM has limitations when it comes to heteromultimeric complexes, especially in the case of non-symmetric ones, where the heterogeneity increases the computational complexity. Most importantly, the individual components may adopt different conformations whether they are isolated or within their assembly. Very few methods exist to tackle this problem, and even fewer actually succeed; thus, a different way for characterizing and locating these components within their assembly, regardless of their different conformational states, is mandatory. In this work, we exploit the different aspects provided by the field of computer vision, and treat our biological problem as if it was a problem of object recognition. Specifically, we adopt the concept of localizing objects in a scene, and make use of local descriptors and the main steps of SIFT algorithm, for extracting distinctive features (local extrema) from images. Translated to our biological problem, we detect informative features (keypoints) in the atomic structures' density maps, so as to localize them within their macromolecular assembly. Our goal is to diminish the huge number of these extracted features, by specifically searching for corners, as these points remain stable regardless any rotation or change. We adopt the principles of Harris corner detector and expand them by using three-dimensional structure tensor analysis (STA). The significance lies in the fact that the eigenvalues and the corresponding eigenvectors of the structure tensor, describe the principal curvatures of the neighborhood around the local extrema. Based on the statistics of the eigenvalues' ratios, we apply multiple types of thresholding under different configurations, and benchmark the STA set of parameters on 54 different structures. For the evaluation of the parameters, we compare the extracted keypoints with a set that is known – from the already existing software – to lead to correct assembly prediction. Experimental results show the existence of parameter sets that remove almost all of the unstable keypoints (false positives), others that retain almost all of the stable ones (true positives), while others provide solutions that can balance the trade-off between these two. Finally, we verify that there are specific complexes (1Z5S, 2GC7) without a trustworthy density profile, since no solutions can be obtained for every resolution. The proposed method considerably speeds up the existing software by reducing the computational complexity – a key issue for heteromultimers, and is a general and accurate way for extracting localized features for correct assembly prediction, which can serve as a baseline for studying the dynamics of these keypoints under conformational changes.

SUBJECT AREA: Structural Biology, Biomolecular Modeling, Computer Vision

KEYWORDS: macromolecular structure, protein subunit localization, keypoint detection, Harris corner detection, extrema extraction

ΠΕΡΙΛΗΨΗ

Η γνώση του σχήματος, της δομής, και των αλληλεπιδράσεων των μακρομορίων, ορίζει τη βιολογία σε μοριακό επίπεδο σε λεπτομέρεια ατόμων. Παρόλο που η γνώση της αρχιτεκτονικής είναι ένα σημαντικό βήμα πριν την κατανόηση της λειτουργίας, εξακολουθεί να είναι μια δύσκολη διαδικασία. Οι τρέχουσες τεχνικές ανάλυσης δομής (X-ray Crystallography, cryo-EM, etc.), αν και αρκετά επιτυχείς, αδυνατούν να γενικεύσουν καλά σε διαφορετικούς τύπους δομών, καθώς κάθε μία από αυτές τις μεθόδους είναι σχεδιασμένη για συγκεκριμένους τύπους δομικών στοιχείων. Ένας τρόπος για να συνδυάσουμε τα πειραματικά με τα υπολογιστικά δεδομένα, ανεξάρτητα από την ανάλυσή τους, είναι μεσω του Integrative Modeling (IM), καθώς παρέχει έναν περιεκτικό χαρακτηρισμό της δομής των βιομορίων. Απαιτεί ως είσοδο (α) τις υψηλής ανάλυσης δομές των επιμέρους μονάδων που συνθέτουν το υπερμοριακό σύμπλεγμα, και (β) τους χαμηλής ανάλυσης φακέλους αυτών των συμπλεγμάτων, και μας παρέχει βιολογικά συσχετιζόμενες υπερμοριακές συστοιχίες, συνεπείς με το διαθέσιμο σύνολο των πειραματικών δεδομένων. Ωστόσο, το IM εμφανίζει κάποιες αδυναμίες όσον αφορά στα ετεροπολυμερικά σύμπλοκα, ειδικά στην περίπτωση των μη συμμετρικών, όπου η ετερογένεια αυξάνει την υπολογιστική πολυπλοκότητα. Το πιο σημαντικό είναι ότι οι επιμέρους μονάδες των συμπλόκων μπορεί να υιοθετούν διαφορετικές διαμορφώσεις ανάλογα με το αν είναι απομονωμένες ή μέσα στη συστοιχία τους. Συνεπώς, είναι αναγκαία η εύρεση ενός διαφορετικού τρόπου για τον χαρακτηρισμό και τον εντοπισμό αυτών των επιμέρους μονάδων εντός των συστοιχιών τους. Στην εργασία αυτή, εκμεταλλευόμαστε πτυχές του πεδίου της μηχανικής όρασης, και χειριζόμαστε το βιολογικό μας πρόβλημα σαν να ήταν πρόβλημα αναγνώρισης αντικειμένων. Συγκεκριμένα υιοθετούμε την έννοια του εντοπισμού αντικειμένων σε μια σκηνή, και χρησιμοποιούμε local descriptors και τα βασικά βήματα του αλγορίθμου SIFT για την εξαγωγή διακριτών χαρακτηριστικών (τοπικά ακρότατα) από εικόνες. Για το βιολογικό μας πρόβλημα, ανιχνεύουμε τα σημεία-κλειδιά (keypoints) των ατομικών δομών, ώστε να τις εντοπίσουμε μεσα στη μακρομοριακή τους συστοιχία. Στόχος μας είναι να μειώσουμε τον τεράστιο αριθμό αυτών των keypoints, αναζητώντας τις γωνίες, καθώς αυτά τα σημεία παραμένουν σταθερά ανεξάρτητα από οποιαδήποτε περιστροφή ή αλλαγή. Υιοθετούμε τις αρχές της μεθόδου ανίχνευσης γωνιών Harris, και τις επεκτείνουμε χρησιμοποιώντας μια 3-D ανάλυση δομικών τανυστών. Η σπουδαιότητά της έγκειται στο γεγονός ότι οι ιδιοτιμές και τα αντίστοιχα ιδιοδιανύσματα της δομής του τανυστή περιγράφουν τη βασική καμπυλότητα της δομής. Βασιζόμενοι στις στατιστικές των λόγων των ιδιοτιμών, εφαρμόζουμε πολλαπλούς τύπους κατωφλίωσης για διαφορετικές παραμέτρους, και δοκιμάζουμε αυτές τις παραμέτρους σε 54 διαφορετικές δομές. Για την αξιολόγηση των παραμέτρων, συγκρίνουμε τα υπολογισθέντα keypoints με ένα σύνολο για το οποίο γνωρίζουμε ότι επιτυγχάνει σωστή πρόβλεψη συστοιχιών. Τα πειραματικά αποτελέσματα δείχνουν την ύπαρξη παραμέτρων που αφαιρούν σχεδόν όλα τα ασταθή keypoints (false positives), παραμέτρων που διατηρούν σχεδόν όλα τα σταθερά (true positives), και παραμέτρων που δίνουν λύσεις εξισορροπώντας το trade-off μεταξύ των προηγούμενων δύο. Τέλος, επαληθεύουμε ότι υπάρχουν σύμπλοκα με αναξιόπιστο προφίλ πυκνότητας, καθώς δε βρίσκονται λύσεις για όλες τις αναλύσεις τους. Η μέθοδος που προτείνουμε είναι ένας γενικός, γρήγορος και ακριβής τρόπος για την εξαγωγή τοπικών χαρακτηριστικών για σωστή πρόβλεψη συστοιχίας, και μπορεί να χρησιμεύσει ως βασική γραμμή για τη μελέτη των δυναμικών αυτών των keypoints όταν υπόκεινται σε διαμορφωτικές αλλαγές.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Δομική Βιολογία, Βιομοριακή Μοντελοποίηση, Μηχανική Όραση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: μακρομοριακή δομή, εντοπισμός πρωτεϊνικών υπομονάδων, ανίχνευση σημείων-κλειδιών, ανίχνευση γωνιών Harris, εξαγωγή ακρότατων

CONTENTS

LIST OF IMAGES

LIST OF TABLES

PREFACE

The current Master Thesis was pursued from November 2017 until September 2018 in Lausanne, as a collaboration between the National and Kapodistrian university of Athens (NKUA) and École Polytechnique Fédérale de Lausanne (EPFL). It is a mandatory requirement for the graduation from the interdisciplinary postgraduate program Information Technologies in Medicine and Biology, of the Department of Informatics and Telecommunications of NKUA.

1. INTRODUCTION

1.1 Importance of the structure

The biological universe consists of two types of cells: prokaryotic and eukaryotic. In these cells, a whole different universe of macromolecules lies, defining with their structure and function, the function of the cell itself. Therefore, the architecture of biological macromolecules is critical for our understanding of their biological function. Knowledge of the shape, structure and interactions of these macromolecules defines biology at the molecular level, in atomic detail.

On a more practical note, protein 3D-structures are the basis for structure-based drug design [1]. One example of such drug is Imatinib - or Gleevec [2]. Imatinib is a medication used to treat cancer, designed specifically to target two types of Leukemia. Its advantage above all previous drugs for cancer is that it can differentiate between cancer cells and other tissues, without harming the latter. This was only succeeded by knowing and determining its structure.

1.2 Structure resolution techniques

Although knowing the architecture is an important step before reaching the knowledge of the function, it still is a challenging task. Current resolution techniques used to determine the structure, although quite successful, they fail to generalize well across different types of structures, since each one of these methods is designed for specific kinds of components. Resolution, in terms of protein structure determination, is a measure of the quality of the data that has been collected on the crystal containing the protein [3]. In other words, it is the distance that corresponds to the smallest observable feature in the diffraction pattern resulted after the X-rays have penetrated the protein crystal. Thus, the smallest this distance (computed in Angstrom) is, the higher the resolution will be.

X-ray Crystallography can determine structures of proteins that form diffractable crystals, but encounters difficulties when it comes to larger or more flexible proteins. Nuclear magnetic resonance (NMR) spectroscopy provides information on proteins in solution, rather than being restricted by a crystal (as in X-ray crystallography), and thus, can actually study the atomic structures of more flexible proteins. As a result, these two methods are complementary, as the characteristics of one fill the gaps of the other [4], and both result in high resolution structural models. On the other hand, cryo-electron microscopy (cryo-EM), cryo-electron tomography (cryoET) and small-angle X-ray scattering (SAXS) yield structures in lower resolution (>10 Å), but they can handle larger proteins. We will discuss cryo-EM in more detail in 1.5, as it is a method of significant interest for this work.

We can easily conclude then, that we cannot rely on a single technique to get atomic resolution. We would like to have a method that combines these techniques and all the knowledge we have at our disposal, in order to efficiently extract structural information from macromolecules.

1.3 Integrative Modeling

A way to combine experimental and computational data in order to obtain higher resolution information, is through Integrative Modeling (IM) [5], which provides a comprehensive structural characterization of biomolecules, by building models. It requires as input (a) high resolution structures of the individual components composing the supramolecu-

lar complex (i.e. structures resulted by X-ray Crystallography), (b) low-resolution envelopes of native assemblies (i.e. cryo-EM maps), (c) other information, such as stoichiometry from protein quantification, restraints from cross-linking, distance between the amino acids. After collecting data, IM chooses how to represent and evaluate the models, finds models with high score and analyzes them. It repeats this procedure until to converge to an ensemble of models that fit all the current information and is found to be satisfactory, according to the criteria that have been set [6]. On the output, IM results in biologically relevant supramolecular assemblies consistent with the available set of experimental data. The general features and procedures of IM are shown in Figure 1.

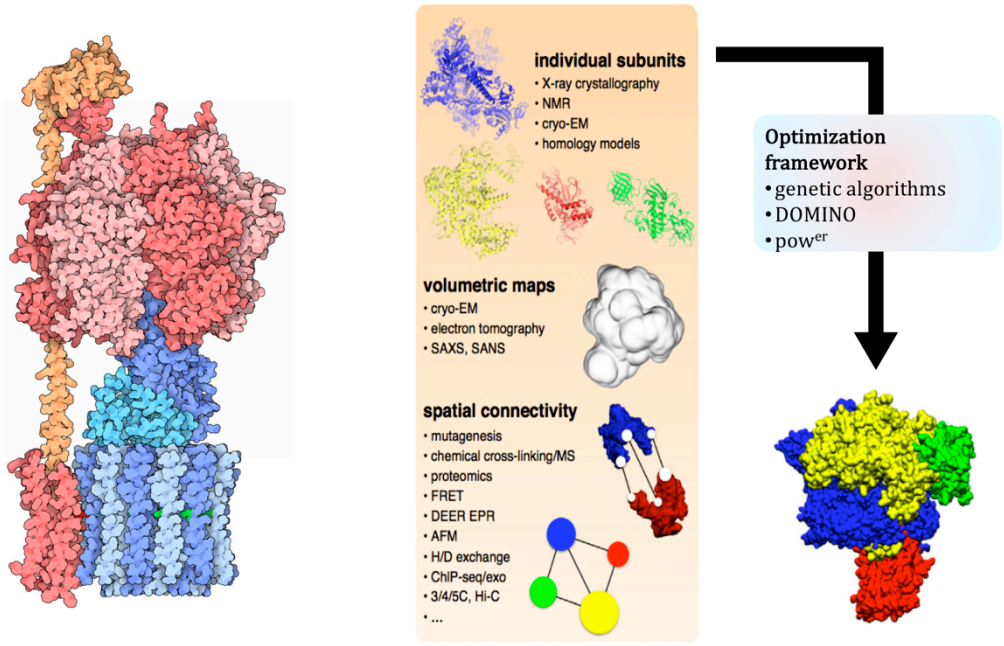


Figure 1: (a) A visual example of a heteromultimeric complex, (b) Integrative modeling strategies. The figure was originally published in [7].

1.4 Asymmetric heteromultimeric assemblies

Although Integrative Modeling seems to be the key to unlock the structural information of biological macromolecules, it has limitations when it comes to heteromultimeric complexes.

Heteromultimeric complexes can either be symmetric or non-symmetric, depending on the arrangement of their component(s). While current softwares exhibit success in the models' prediction for symmetric heteromultimeric complexes ([8] – [11]), non-symmetric cases are way more challenging. Symmetric assemblies are made of the repetition of a single subunit by sampling a four-dimensional search space of the three Eulerian angles α, β, γ - defining the protein orientation - and the radius of the symmetric assembly [12]. By contrast, the absence of any symmetry and geometry, leads to an increase in terms of computational cost, as the search space increases as well. In other words, one has to consider not only one, but all of the individual components of the asymmetric construct, along with their dimensions and constraints. Moreover, the highly dynamic nature of these structures coupled with their complexity, generates a significant heterogeneity. Most importantly, the individual components may adopt different confor-

mations whether they are isolated or within their assembly. An example of a symmetric and a non-symmetric construct is depicted in Figure 2.

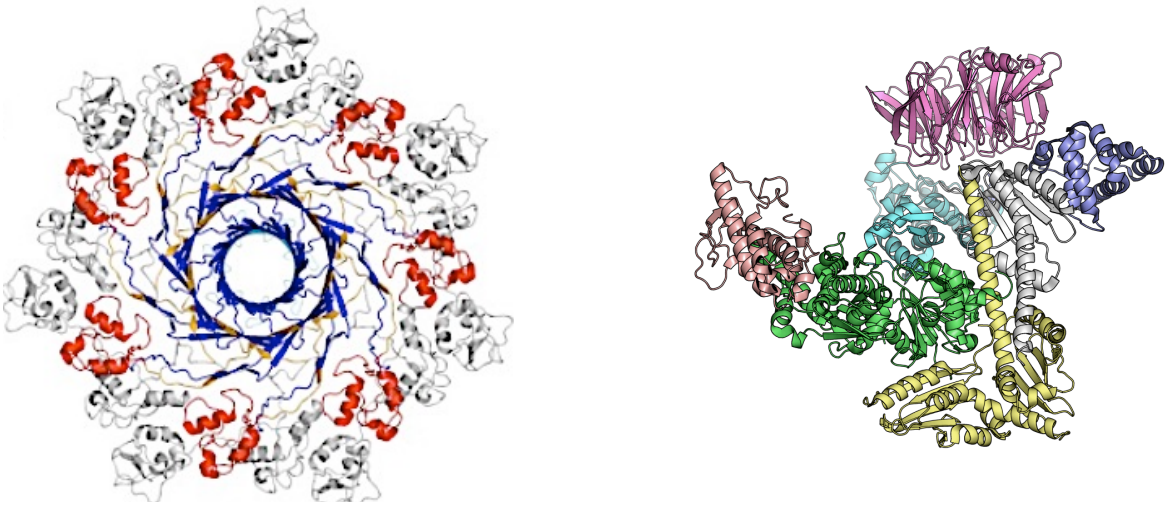


Figure 2: The challenge between (a) symmetric homomultimers - such a construct is aerolysin pore toxin [30], (b) asymmetric heteromultimers - such a construct is the crystal structure of Arp2/3 complex with bound ATP and calcium with pdb name: 1tyq

1.5 Cryo-Electron Microscopy

Cryo-electron microscopy has established itself as a mainstream technique to capture the structure of large macromolecular assemblies. The steps that are involved in determining molecular structures are: sample treatment, EM grid preparation, image acquisition, image processing [13], and they are also depicted in Figure 3. Cryo-EM results in three-dimensional grids – or density maps - consisting of voxels, each one having a nucleic density value. Current improvement at each one of these steps led cryo-EM to be able to reach higher resolution cryo-EM maps [14], [15] and to be considered as a potential important tool for drug discovery.

In more details, a cryo-EM experiment begins with a purified protein sample. Then the sample solution is deposited on the sample grid, and vitrification follows, in which the protein solution is cooled so rapidly that water molecules do not have time to crystallize; in this way, the sample is being protected from radiation damage as well. The sample is then screened for particle concentration – by particles we mean the 2D projections of the sample molecules –, distribution and orientation, with the use of a transmission electron microscope. Next, a series of images is acquired and two-dimensional classes – particle images representing the same view – are computationally extracted. Finally, the data is processed by reconstruction software yielding detailed 3D models of biological structures.

However, cryo-EM captures the native states of the molecules so that different conformational states are captured and can be found in the same data set, too. This leads to very heterogeneous data sets, and the corresponding need for 3D classification algorithms, able to differentiate between different states of the same molecule and different molecules.

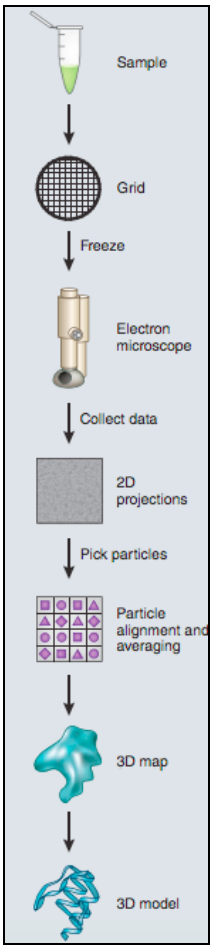


Figure 3: The overall single-particle cryo-EM workflow, from protein sample to 3D model. The figure and its title were originally published in [16].

Going back to Integrative Modeling concept, cryo-EM now offers a guide to localize the subunits within their assembly, individually rather than simultaneously, as it reduces the search space by fitting the atomic models into the density/cryo-EM maps. The fact though that we are dealing with highly dynamic structures, is still considered a problem that should be faced. A different way for characterizing and locating these components within their assembly, regardless of their different conformational states and their dynamics, is mandatory.

1.6 Outline

Considering the importance of the biological macromolecules' structure, as well as the limitations that Integrative Modeling faces when it comes to extract structural information from asymmetric heteromultimeric complexes, we use the method being proposed by the current software of the lab [17], for characterizing and localizing the assemblies' individual components. Heteromultimeric Assembly Prediction (hetAP) software exploits the field of computer vision and treats the aforementioned biological problem as if it was a problem of object recognition. Particularly, it uses the concept of localizing objects in a scene; if someone had to localize a specific car in a scene of a traffic road, then the analogy would be to localize an atomic structure within its assembly. In other words, it adopts the existing theory for 2D images, in the 3D space of protein structures. It is based on local descriptors and the main steps of SIFT algorithm, and

extracts density keypoints (local extrema) from the density maps of both the complex and its components. Roughly quoted, it does not consider the bigger conformational issue, but it rather focuses on local information. We will refer with more details to hetAP software in 2.2.

In this work, we will diminish the huge number of these extracted features, by specifically searching for corners, as these points should remain stable regardless of any rotation or change. To do so, we will adopt the principles of Harris corner detector and expand them by using three-dimensional structure tensor analysis, in order to define the principal curvatures of the neighborhood around the local extrema. The proposed method is a general, fast, and accurate way for extracting localized features for correct assembly prediction, which can serve as a baseline for studying the dynamics of these keypoints under conformational changes.

The rest of the dissertation is organized as follows; in Chapter 2, the principles of Local Descriptors and SIFT's algorithm, as applied in the field of Computer Vision, are presented. The Harris corner detection algorithm and its extension to 3D structure tensor analysis is provided in Chapter 3. In Chapter 4 we present the implementation of the proposed method for extracting localized features for heteromultimeric assembly prediction. Finally, Chapter 5 shows and analyzes our implementation's results, while the conclusions and future work are discussed in Chapter 6.

2. COMPUTER VISION

2.1 The SIFT algorithm

In the blink of an eye, SIFT [18] introduces two major stages of computation

1. Feature Extraction – Accurate Keypoint Detection and Localization
2. Descriptor Generation

We focus on the first step, which is about detecting the keypoints on the image (i.e. localized, distinctive points of interest) and evaluating their stability (i.e. their robustness against rotation, translation, or other image modifications). In short, the algorithm detects the local extrema (local maxima, local minima) of an image, and discards the "unstable" keypoints that usually lie on the edges and low contrast regions.

In more details, the algorithm executes the following 6 steps, with steps 1-5 belonging to the feature extraction stage, while the last step is the generation of the descriptor:

1. First, it creates a scale space with the corresponding octaves. Each octave starts with an image of specific scale; i.e. the first octave contains an image of double the size of the original, the second contains an image of the same size as the original, the third half of the original, the fourth quarter of the original etc. Each image in the octave is then progressively blurred by convolving it with a Gaussian kernel (Gaussian blurring); it blurs the first image, then blurs the result of the first blurring, then the result of the second blurring etc.
2. Then, from the scale space it generates another set of images using the Laplacian of Gaussians (LoG). However, it requires the second order derivatives, which is computationally expensive, and thus it approximates the LoG through the Difference of Gaussians (DoG); that is, the difference between two consecutive scales (blurring levels). Apart from the computational complexity, another advantage of the DoG images is that their detected extrema are scale invariant.
3. The next step is to coarsely locate the maxima and minima. This is done iteratively through each pixel, by checking its 3x3 neighbors. The check is done within the current image, and also the one above and below it (different scales/blurring levels). This way, a total 26 checks are made, and a point is marked as an approximated keypoint if it's the maximum or the minimum of all 26 neighbors. To obtain the true local maxima/minima, one has to obtain the subpixel values; they are generated by finding the extrema of the Taylor expansion of the image around the approximated keypoint. These subpixel values increase both the chances of matching and the stability of the algorithm.
4. The procedure described in step 3 results in a big number of obtained keypoints. However, some of them might lie along an edge, might don't have enough contrast. In both cases, they are not "useful" as features. To get rid of them, the algorithm follows an approach that is similar to the one used in the Harris detector [19] for removing edge features; remove low contrast features by simply checking their intensities. It computes a 2x2 Hessian matrix, H, at the location and scale of the keypoint, and uses its two eigenvalues' ratio – actually the ratio of the trace of H to the determinant of H – rather than their individual values. If α is the larger eigenvalue and β is the smaller eigenvalue of H, then the aforementioned ratio will be

$$\alpha = r\beta$$

And then,

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

A more detailed description of the Harris corner detector is analyzed in Chapter 3.

5. After step 4, the algorithm has detected stable, scale invariant keypoints. The next thing is to assign an orientation to each keypoint, so that it becomes also rotation invariant. The idea is to collect gradient directions and magnitudes around each keypoint, to figure out the most prominent orientation(s) in that region, and finally to assign the orientation(s) to the keypoint. For more details about this step, we refer the reader to read the official description of the algorithm in [18].
6. Finally, the algorithm generates a feature vector for every keypoint, which describes the keypoint in a unique way. For more details, we encourage the reader again to read the official description of the algorithm in [18].

An overview of the aforementioned 6 steps of SIFT algorithm is depicted in Figure 4

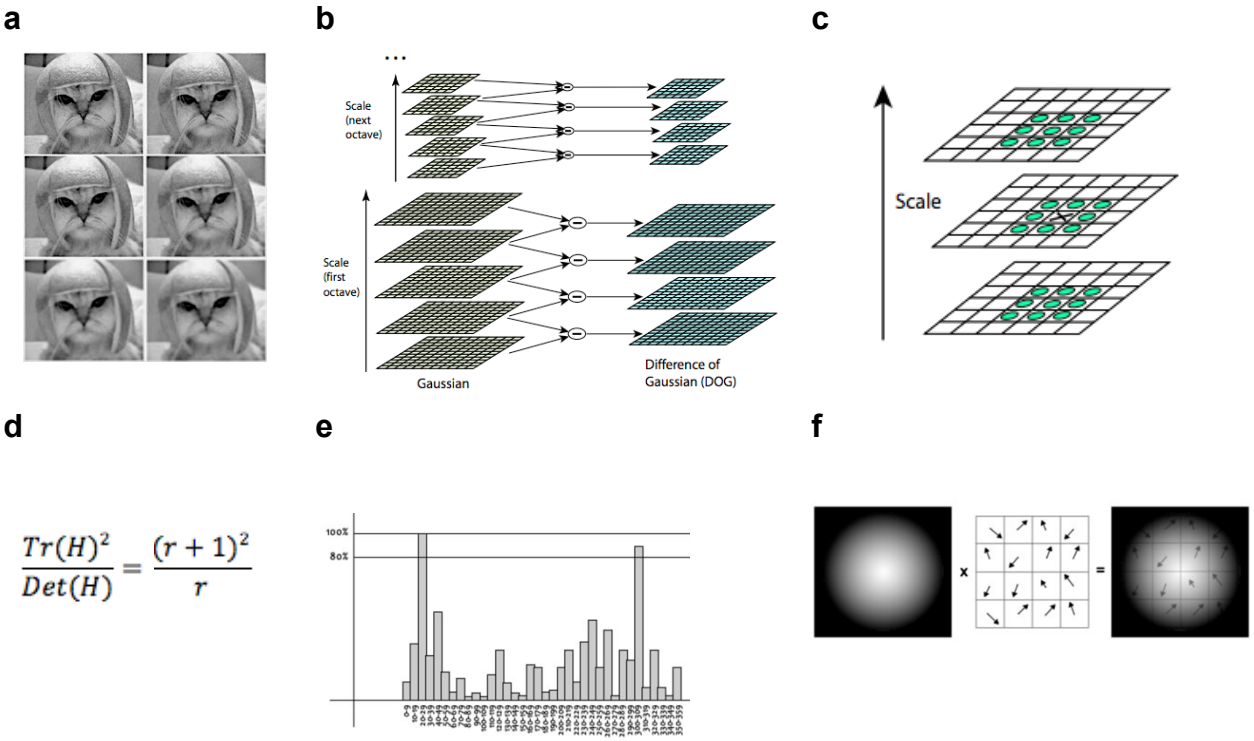


Figure 4: The 6 steps of the SIFT algorithm: (a) Scale space, (b) Difference of Gaussians, (c) Local Extrema extraction, (d) Low contrast features' removal, (e) Orientation assignment, (f) Descriptor generation

2.2 Applying SIFT's principals to macromolecular assemblies – hetAP

As we have already mentioned, SIFT algorithm is applied in the case of 2D images, in order to extract distinctive features that can be identified even under different views of an object or scene. Thus, it is designed for two-dimensional problems. HetAP expands these principals in 3D space, so as to deal with the three-dimensional data. Recall that the goal of the software is to localize the individual components (subunits) within their native heteromultimeric assembly (complex), in order to surpass the challenges that IM faces, when it comes to assemble all these highly dynamic subunits into their complex.

In more details, the software operates as follows:

1. It gets as input (a) the cryo-EM maps (point clouds) of the assemblies, and (b) the atomic structures of their components, resulted from experiments like X-ray crystallography, NMR etc.
2. Then, in order to be able to compare the complex and its subunits, it uses the pdb2vol tool from Situs package [20], which projects an atomic structure on a 3D grid (point cloud). In this way, now the density maps of the subunits can easily be compared with the cryo-EM maps of their complex. This tool also allows one to lower the resolution of an atomic structure to a user-specified value. HetAP specifies the values of resolution to be: 7, 10, 15, 20
3. As a next step, it applies (if needed – only for resolution 10, 15, 20) a Laplacian of Gaussian filtering to the complex and its subunits, in order to enhance the sharpness and the contrast of the edges.
4. Then, for both the complex and its subunits, the software extracts the local extrema on a 3x3x3 box around each candidate voxel, and thus finds their keypoints.
5. Once the keypoints are obtained, it assigns them the proper orientation, following the paradigm of SIFT algorithm, but in a very different way as now we do not have two but three dimensions and this make the problem more complex, and generates the local descriptors of the complex and its subunits.
6. Finally, for the localization of a subunit within its complex, hetAP takes every subunit iteratively and tries to "match" its descriptors with the descriptors of the complex in a greedy way (compare every local descriptor of each subunit, with every local descriptor of the complex). Through this matching procedure, the software is able to predict the assembly.

The basic steps of hetAP software are shown in Figure 5. HetAP is a complete software, which means that after implementing all these steps, it has solutions – keypoints that actually lead to correct assembly prediction. We will use this knowledge in order to make sure that our method, that will be introduced properly in Chapter 4, diminishes the number of keypoints being extracted and, in the same time, keep most of these solutions found by hetAP.

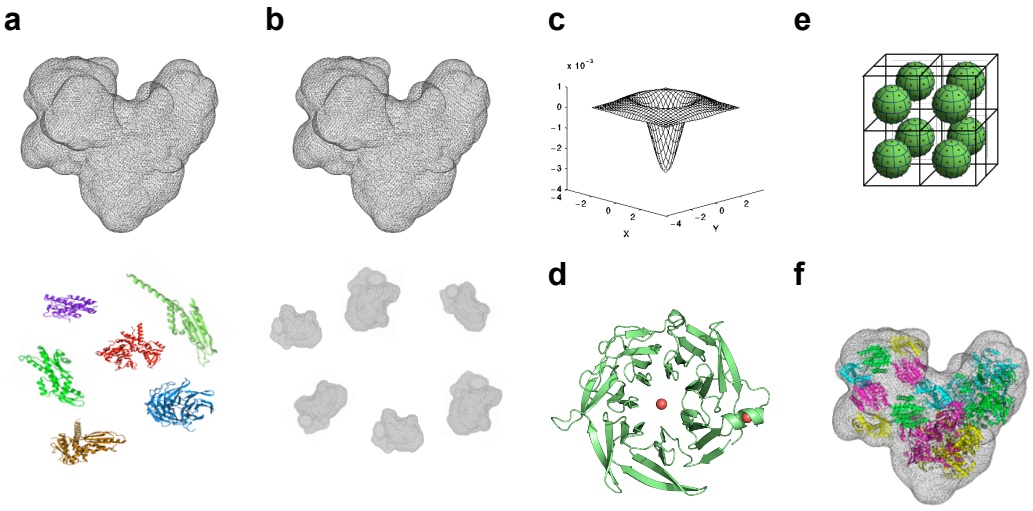


Figure 5: Main steps of the hetAP software. (a) input of the software: atomic structures of the individual components and cryo-EM map of their assembly, (b) density maps of the individual components (Situs package), (c) LoG filtering, (d) local extrema extraction, (e) descriptor generation around each local extremum, (d) matching of each component's descriptor with its corresponding descriptor from the assembly.

2.3 Motivation – Keypoint reduction

Although hetAP software is a very good tool for predicting the heteromultimeric assembly using localized information, it still has some limitations. These limitations relate to the fact that hetAP does not apply any filtering to the produced keypoints, which as a result lead to a large amount of extracted keypoints, many of which can give dubious, and even wrong, solutions. By consequence, the computational cost for the descriptor matching will be very high, as it is trying to match each keypoint descriptor of the subunit to each keypoint descriptor of the assembly.

Motivated by this main difficulty of hetAP, we focus on the 4th step of SIFT algorithm and aim to integrate into the software a filtering step, in order to reduce the amount of extracted keypoints, by removing the false positives and resulting in this way to significant speedup efficiency.

3. CORNERS AS KEYPOINTS – HARRIS CORNER DETECTION

As already mentioned in the previous chapter, our goal is to detect the most stable keypoints, and discard as many false positives – keypoints that in practice are not of interest – as possible. Recalling that a point-of-interest in two dimensions is a point that has a well-defined position, is ideally fast to compute, and can be robustly detected under conditions of different lighting, translation, rotation and other transforms. Several interest point detectors there exist, commonly using maxima and minima points, such as gradient peaks or corners. Most of them also apply a Gaussian filter first, in order to reduce the noise and detail of an image.

Moravec [21] is an early corner detection algorithm that tests each pixel of an image to see if a corner is present, by correlating the patch/window centered on the pixel with the surrounding – overlapping – patches of the neighboring pixels. The problems that Moravec algorithm faces are the noisy response because of a binary window function that it is using, and the fact that it considers only the smallest Sum of Squared Differences (SSD), where the SSD function calculates the correlation difference between two overlapping patches. The Harris and Stephens corner detector [19] provides significant improvements over the aforementioned method, that we will discuss in detail in 3.1. Other worth mentioning algorithms are Shi-Tomasi [22] corner detector, an optimization on the Harris method using only the minimum eigenvalues for discrimination, the SUSAN method [23], [24] that is dependent on segmenting image features based on local areas of similar brightness. Moreover, the Trajkovic and Hedley corner detector [25] and FAST detector [26], both based on SUSAN; for more details, we encourage the reader to read the corresponding bibliography.

We focus and base our method on the Harris & Stephens detector, since it can be extended to the three-dimensional space, and its principles are also applied in the 4th step of SIFT algorithm.

3.1 Harris Corner Detection

In general, whenever an object exists on an image, then the image should contain some edges, some corners, and some other regions that are mostly flat (i.e. background, surface of object etc.). In order to detect which points of the image correspond to one of these three regions, Harris detector algorithm captures the variations of the image. It considers a neighborhood *P* around a point, obtained by shifting a subregion in which the intensity of the gradient is studied to determine if the region around the reference point contains a corner. Based on that, corners can be defined as the regions within the image in which there are large variations in the intensity of the gradient in all directions, as depicted in Figure 6.

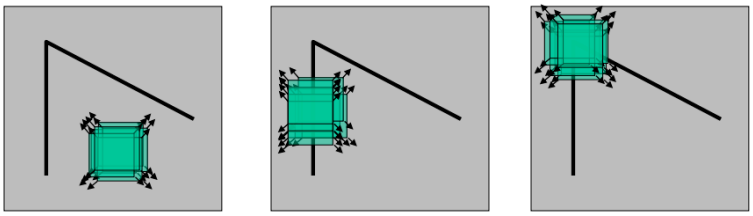


Figure 6: Left (flat): the gradient is zero among every direction. Center (edge): the gradient is constant only across the direction of the edge. Right (corner): the gradient changes among every direction.

To better understand the algorithm, let *I* be a 2D grayscale image. Consider taking an image patch over the area of a point (u, v) and shifting it by (x, y). Then, the weighted Sum of Squared Differences (SSD) between these two patches, denoted by S(x, y), is given by

$$S(x,y) = \sum_u \sum_v w(u,v)[I(u+x, v+y) - I(u,v)]^2$$

Using a Taylor expansion, then

$$I(u+x, v+y) \approx I(u,v) + I_x(u,v)x + I_y(u,v)y$$

where I_x, I_y are the partial derivatives of the image I.

This produces the approximation

$$S(x,y) \approx \sum_u \sum_v w(u,v)\left[I_x(u,v)x + I_y(u,v)y\right]^2$$

which can be written in a matrix form

$$S(x,y) \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix}$$

where *A* is the structure tensor

$$A = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x(u,v)^2 & I_x(u,v)I_y(u,v) \\ I_x(u,v)I_y(u,v) & I_y(u,v)^2 \end{bmatrix}$$

and *w(u, v)* denotes the type of window that slides over the image. If a gaussian window is used, then the response will be isotropic.

The next step of the algorithm is to define a score function for determining if a point belongs to a corner or not. The eigenvalues of the structure tensor can help determine the suitability of the window. This is done through the corner response calculation

$$R = \det(A) - k\left(trace(A)\right)^2 = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

where $k \in [0.04, 0.06]$.

All windows that have a score *R* greater than a certain value are corners. Observe that *R* depends only on the eigenvalues of *A*. Thus, the final step of the algorithm is to classify the points of the window using the obtained eigenvalues.

Since A is symmetric, we can visualize it as an ellipse with axis lengths determined by the eigenvalues, and orientation determined by *R*. In short, a big circle should correspond to a corner point, a smaller circle to a point of a flat region, and an ellipse to an edge point. The classification regions with respect to the eigenvalues and the shape of the ellipse are depicted in Figure 7.

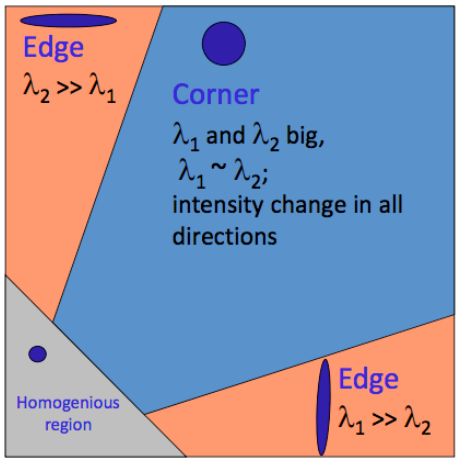


Figure 7: Classification regions and the corresponding ellipses

3.2 Corner detection using 3D structure tensor

However, in order to apply the 2D Harris corner detector to our problem, it has to be extended to the three-dimensional space. The authors in [27] extend the structure tensor to the 3D space, and do the corresponding analysis for extracting edge voxels from three-dimensional volumetric maps. Their motivation is the fact that they want a descriptive mechanism, able to represent the surroundings of mobile robots, as captured by their sensors. To do so, they apply a structure tensor operation to the voxel map, followed by a classification of the obtained eigenvalues, in order to remove voxels that are part of flat regions. This classification is done by thresholding both the magnitudes of the eigenvalues, as well as the ratios of the middle-to-largest and smallest-to-largest eigenvalues.

3.2.1 3-D Structure Tensor

Following the same notion as before, but with the difference that the pixels become voxels, the structure tensor becomes:

$$A = \sum_{(x,y,z)} w(x,y,z) \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_y I_x & I_y^2 & I_y I_z \\ I_z I_x & I_z I_y & I_z^2 \end{bmatrix}$$

Furthermore, in [27] they exploit the fact that the eigenvalues and their corresponding eigenvectors that result from this analysis, summarizes the distribution of the gradient within a neighborhood of a voxel *p*, defined by a Gaussian window *w*. In other words, the eigenvalues and the eigenvectors describe the curvature of a structure.

Thus, the classification of the voxels is now done based on the relative magnitude of the structure tensor eigenvalues as follows:

- A planar voxel will have one direction with a large gradient (normal to the plane), and two directions with small gradients. Thus, it will have one large and two small eigenvalues.
- A line or edge will be characterized by two directions of large gradient, and one direction of small gradient. Thus, it will have two large and one small eigenvalue.

- An isolated region in space (corner) will have large gradients in every direction. Thus, it will have three large eigenvalues.
- Similarly, for homogeneous regions, the gradient will be small in every direction. Thus, they will have three small eigenvalues.

In order to simplify the aforementioned classification rules, instead of focusing on the values of the eigenvalues independently, they choose to study the distribution of the middle and smallest eigenvalues, normalized by the largest. Following this approach, the rule for classifying the voxels into "edges", "planar", and "corners", can be easily obtained on the space generated by the distribution of the ratios, as shown in Figure 8.

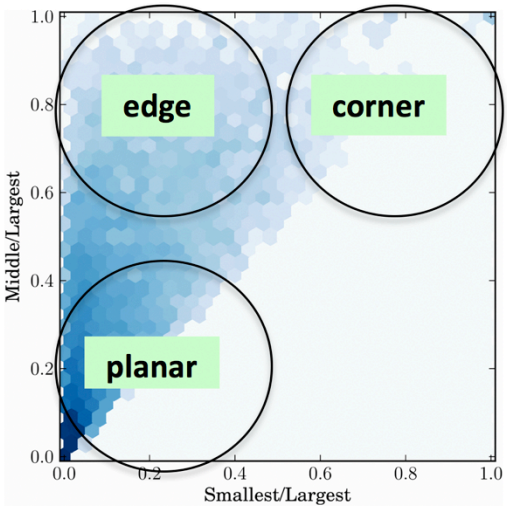


Figure 8: Different classification regions of voxels. The basic figure (2D histogram plot) was originally published in [27].

Motivated by the authors' approach, we adopt the 3D structure tensor analysis, and integrate it in hetAP, in order to detect the isolated regions of the 3D space (corners). Once the analysis is adopted, the next step that needs to be considered is the proper values of the corresponding thresholds for deciding if a given keypoint can be classified as a corner.

4. IMPLEMENTATION

In this Chapter we will analyze the proposed method, implemented in the hetAP software. Recall that our goal is to define a unique, global set of parameters for detecting the most stable keypoints and discard as much false positives as possible, which will increase the speedup of hetAP. First, we will show how we extend the structure tensor to the 3D domain, based on the approach described in [27]. Then, we will analyze the results obtained from the 3D structure tensor analysis, which, finally, will lead us to search for stable keypoints based on the statistics of the eigenvalues' ratios distribution.

4.1 Extension to 3D structure tensor

3D structure tensor, same as in the 2D case, is derived from the weighted sum of squared differences between shifted volume patches. In order to extend the tensor structure on the 3D space, we focus on the 4th step of hetAP, where the local extrema of the voxels are extracted. For a given complex and its subunits, we operate as follows:

1. We iterate on the data of the complex and the subunits.
2. For every voxel, we define a 3x3x3 neighborhood around it, and hetAP extracts the local maxima and the local minima of every neighborhood. The extracted extrema are considered as the keypoints of the given structure (complex, subunit).
3. Up to this point, we have the keypoints for a given protein structure. Then, for every one of these keypoints (x,y,z) we define again a NxNxN neighborhood around it. Each one of these neighborhoods that corresponds to different keypoints, is a subvolume of the original volume of the structure.
4. For every voxel x_p,y_p,z_p in each subvolume (including the keypoint x,y,z itself), we compute the gradient structure and generate a matrix of the three partial derivatives of the function I of each voxel

$$\begin{bmatrix} I_x^2 & I_xI_y & I_xI_z \\ I_yI_x & I_y^2 & I_yI_z \\ I_zI_x & I_zI_y & I_z^2 \end{bmatrix}$$

5. Then, again on every voxel x_p,y_p,z_p in the subvolume, we apply a 3D Gaussian weighting function

$$w(x,y,z) = \exp\left[-\left(\frac{(x-x_p)^2 + (y-y_p)^2 + (z-z_p)^2}{2\sigma^2}\right)\right]$$

6. Finally, the 3D structure tensor A is obtained by summing over the product of the gradient matrices with the Gaussian weighting functions

$$A = \sum_{(x,y,z)} w(x,y,z) \begin{bmatrix} I_x^2 & I_xI_y & I_xI_z \\ I_yI_x & I_y^2 & I_yI_z \\ I_zI_x & I_zI_y & I_z^2 \end{bmatrix}$$

Now that we have successfully generated the 3D structure tensor, we can proceed to the structure tensor analysis.

In Figure 9, we visualize how it actually looks like each aforementioned subvolume on the structure, consisting of the central voxel, the neighboring voxels and the direction of the gradient vectors.

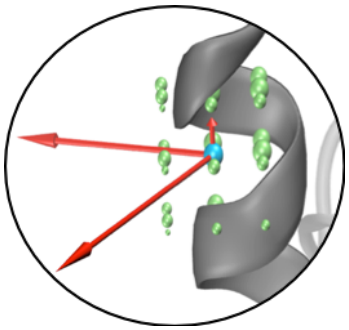


Figure 9: Visualization of a 3x3x3 neighborhood on the structure. The central keypoint *x,y,z* (in light blue) and its 26 direct neighboring voxels x_p,y_p,z_p (in light green) scaled by their density value. The red arrows show the direction of the gradient vectors, thus they correspond to each one of the three eigenvalues of the structure tensor. This is a case of an edge voxel as it has two directions of large gradient, and one direction of small gradient

4.2 3D structure tensor analysis: the individual eigenvalues case

Proceeding to the structure tensor analysis, what the study does first is to sort the three eigenvalues of each structure tensor, that corresponds to each keypoint, in order to characterize them as small, middle, and large. Then, we take the sorted eigenvalues and plot their distribution depending on the class they belong to. Recall that for a voxel to lie on an isolated region in space (corner), all three eigenvalues should be large.

The distribution of each eigenvalue class is depicted in Figure 10. The x-axis corresponds to the eigenvalues, while the y-axis to the value of the distribution, as computed by fitting a Normal distribution to the data using a Kernel Density Estimation – KDE. With black lines we denote the obtained eigenvalues, while with red we superimpose the eigenvalues of the keypoints that are known to lead to correct assembly prediction.

Since our goal is to select those keypoints that have large eigenvalues in all three directions, we are looking for a unique, global threshold that will keep only those keypoints that have eigenvalues above that threshold. However, we observe that from the distributions of Figure 10, it is not quite obvious where this threshold point should be placed. For this reason, we choose to study the distribution of the relative ratios of the eigenvalues, hoping that this will be more informative.

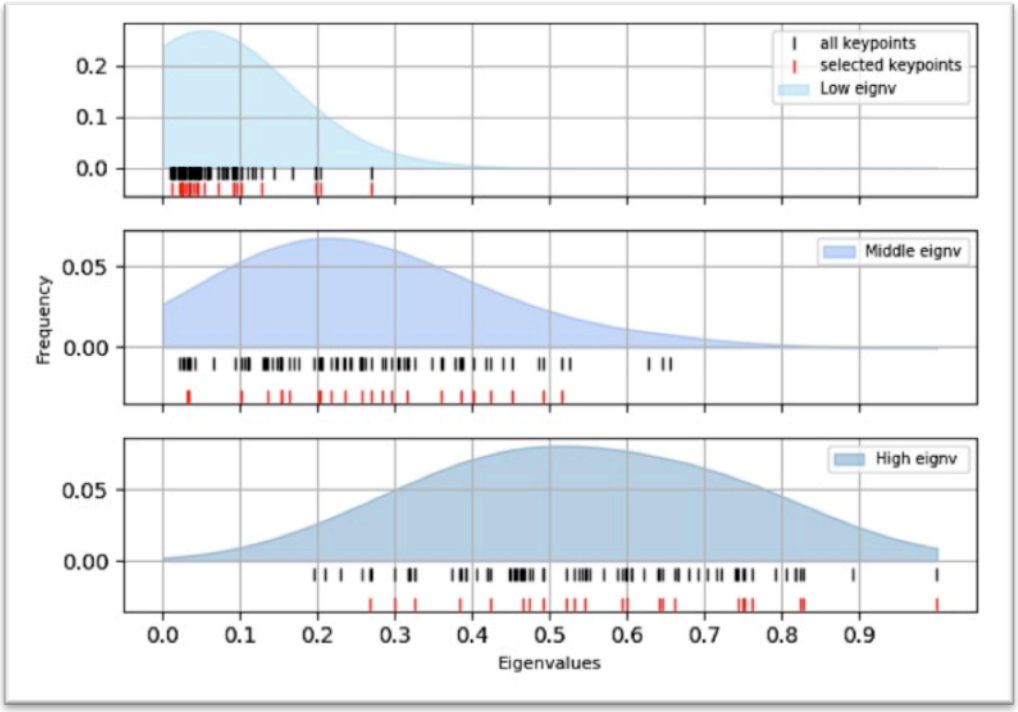


Figure 10: Distribution of the extracted keypoints' eigenvalues

4.3 3D structure tensor analysis: the normalized distributions case

Thus, instead of studying the distribution of each eigenvalue class individually, we now focus on the relative ratios of the eigenvalues. The idea is to count on the interactions between the three eigenvalues.

For our analysis, we define the following two types of ratios:

1. **Middle-to-Large / Small-to-Large**; that is the middle eigenvalue normalized by the large one, related to the small eigenvalue normalized by the large one. In other words, we look for cases were both middle and small eigenvalues are close to large ones, translated to three relatively large eigenvalues. From now on, we refer to this ratio as ML / SL.
2. **Small-to-Middle / Small-to-Large**; that is the small eigenvalue normalized by the middle one, related to the small eigenvalue normalized by the large on. In other words, we look for cases where the small eigenvalues are close to the middle ones, and at the same time the small to be close to the large ones. Thus, also the middle to be relatively close to large ones, resulting to three relatively high eigenvalues. From now on, we refer to this ratio as SM / SL.

We compute the aforementioned ratios of the eigenvalues, and again we apply a KDE on our data, in order to estimate their density. An example of the resulting plots for the two ratio types is depicted in Figure 11 and Figure 12 respectively. The green area corresponds to the KDE over the corresponding ratios. It is reasonable to say that, for both cases, we expect that the desired keypoints should present values that tend to be towards the upper-right corner of the plots, since both of the ratios should be close to 1.

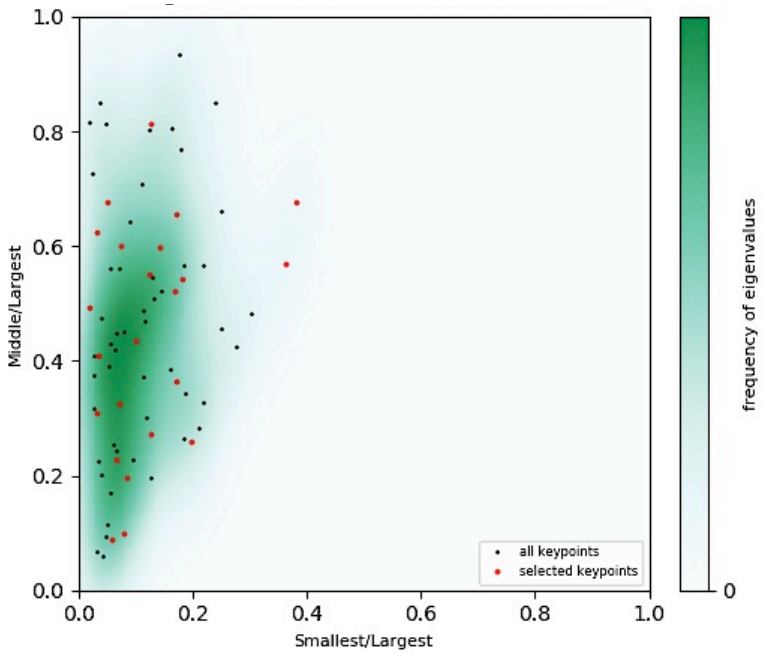


Figure 11: Density of the eigenvalue's ratios for ML / SL (1tyq_chainA, resolution=7, on a 3x3x3 neighborhood, and σ=2)

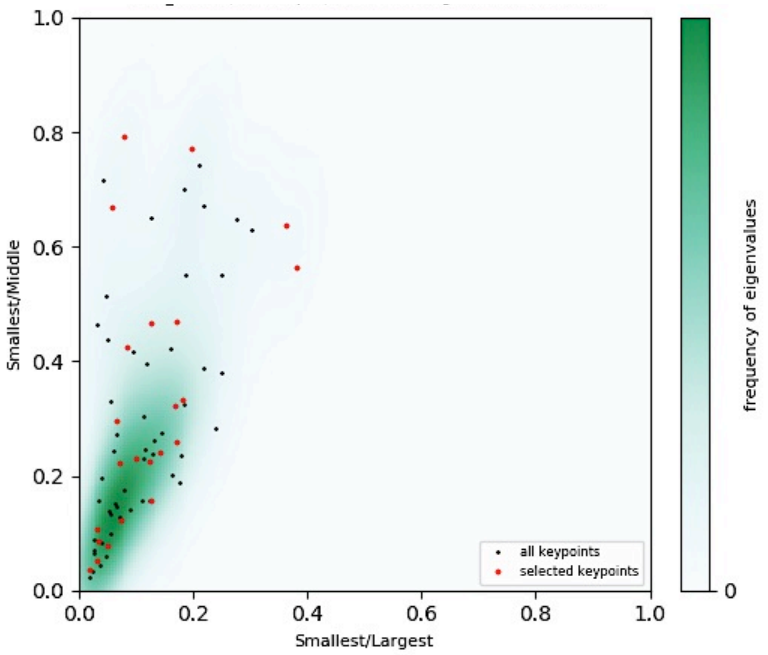


Figure 12: Density of the eigenvalue's ratios for SM / SL (1tyq_chainA, Resolution=7, on a 3x3x3 neighborhood, and σ=2)

Compared to the individual distribution case described in the previous section, we have a clearer image on how the three eigenvalues are related. Unfortunately, the problem of identifying the proper threshold point for keeping the desired keypoints still remains, since the position of the solutions (desired keypoints – known to lead to solution) is again mixed with many false positives.

4.4 Threshold selection

Thus, the next and most important step, is to choose a global threshold point based on the ratios of the eigenvalues we described before. It is not obvious how to select this point, since it highly depends on the protein's structure, the ratio type, and of course the parameters we use. For this reason, we should base the choice of the threshold on the statistics of the eigenvalue ratios. Since we fit a KDE on our data, we can exploit the statistics it provides, and specifically, we can make use of the covariance error ellipses.

4.4.1 Covariance Error Ellipses

A covariance error ellipse represents an iso-contour of the Gaussian distribution, and visualizes a confidence interval on a 2D space. The confidence interval refers to a region that contains a specific percentage of all samples drawn from a Gaussian distribution; if a set of measurements were repeated many times and a confidence interval calculated in the same way on each set of measurements, then a certain percentage of the time this interval would include the point representing the "true" values of the set of variables being estimated. The construction of an ellipse is based on the eigenvectors and the eigenvalues of the covariance matrix of a multivariate Gaussian distribution. For the case of two variables, the first principal component of the covariance matrix explains the "most variance", while the second one fits the errors produced by the first. Thus, for a given confidence interval, the scale *s* of the ellipse is given using the Chi-Square probability table, and the half major/minor axes lengths are obtained as

$$\sqrt{s\lambda_1}$$

$$\sqrt{s\lambda_2}$$

An example of the error ellipses under different confidence intervals is shown in Figure 13.

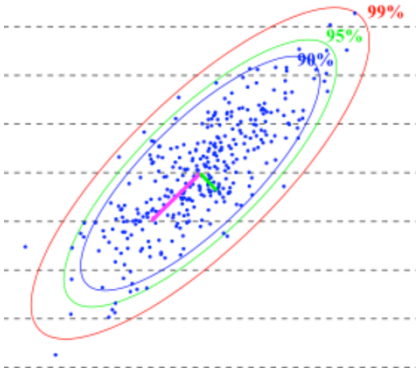


Figure 13: An example of error ellipses and the corresponding confidence intervals. The image is taken from [28].

In our implementation, we compute the covariance matrix of the KDE over the ratios, and extract the corresponding eigenvalues and eigenvectors. In general, 4 types of error ellipses can occur, depending on the directions of the eigenvectors. Based on each case, the threshold point is defined either by (a) the maximum value of the KDE or (b) the mean value of the eigenvalues' ratios. Then, we move this point along the direction of the eigenvector, in such a way to retain the values that tend to grow towards the up-

per right corner of the ratio plot – either in the direction of the major or the minor axis, dependent on the orientation of the ellipse. Obviously, since how far you can go along the axes of the ellipse depends on the specified confidence interval, the threshold point must be adjusted appropriately. In other words, two more parameters that we should consider for the computation of the threshold is whether we choose (a) or (b), and the confidence interval. The 4 different types of error ellipses, with the corresponding confidence intervals and the threshold points are depicted in Figure 14.

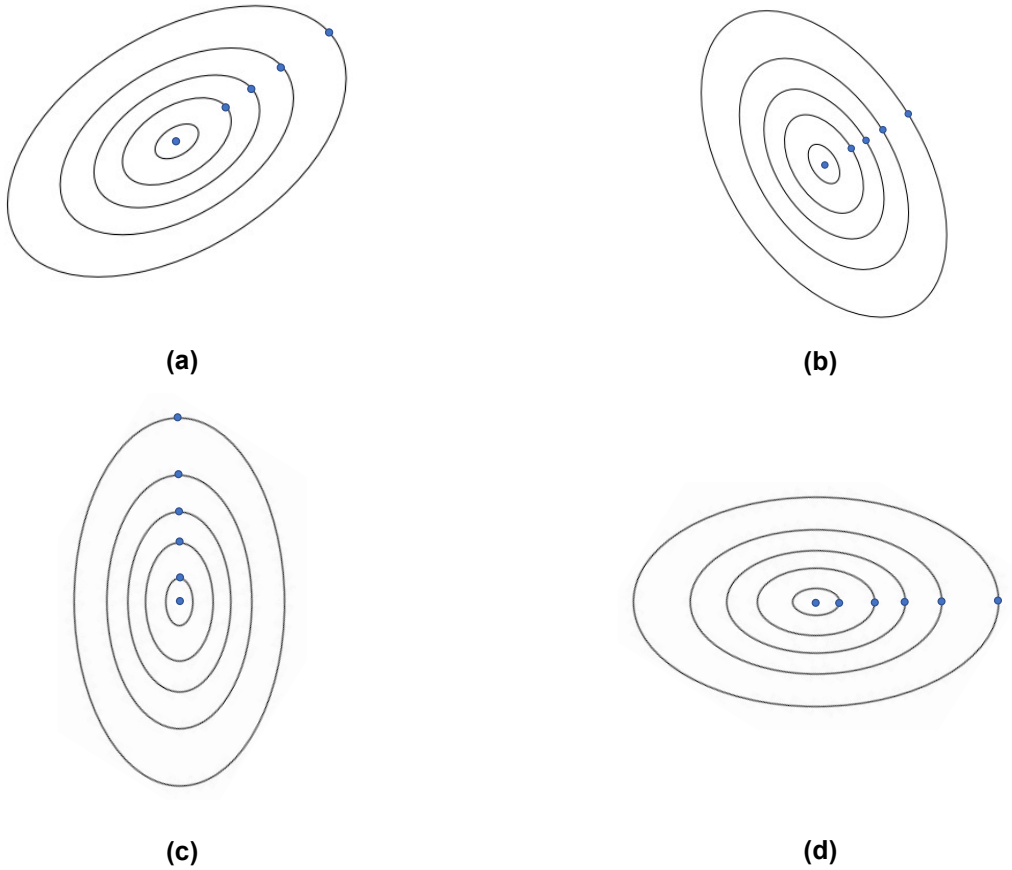


Figure 14: Different types of ellipses and threshold points in the direction of the KDE's covariance matrix's eigenvectors, that we chose for our method

4.4.2 Combining the resolutions

One important thing to note again here, is the resolution, which in structure determinations, is the distance corresponding to the smallest observable feature; if two objects are closer than this distance, they appear as one combined blob rather than two separate objects. In other words, higher numeric values of resolution mean poorer level of detail, and thus fewer local extrema.

Since the aforementioned error ellipses are based on the covariance matrix of the ratios, and thus on the ratio data themselves, which in turn are related to the extracted keypoints, then for lower resolutions we will not have enough points to fit the KDE. As a consequence, in some structures we don't have enough points to justify doing KDE on them.

To avoid this situation, we choose to reverse the procedure of our approach and determine the appropriate threshold points as follows:

1. For every structure individually, for every parameter combination (LoG, neighborhood size, sigma, eigenvalue ratio type, ellipse center) we extract the keypoints for each resolution independently, and we combine them to one big set of keypoints that were extracted. In this way, we manage to have the full picture of a structure under any resolution.

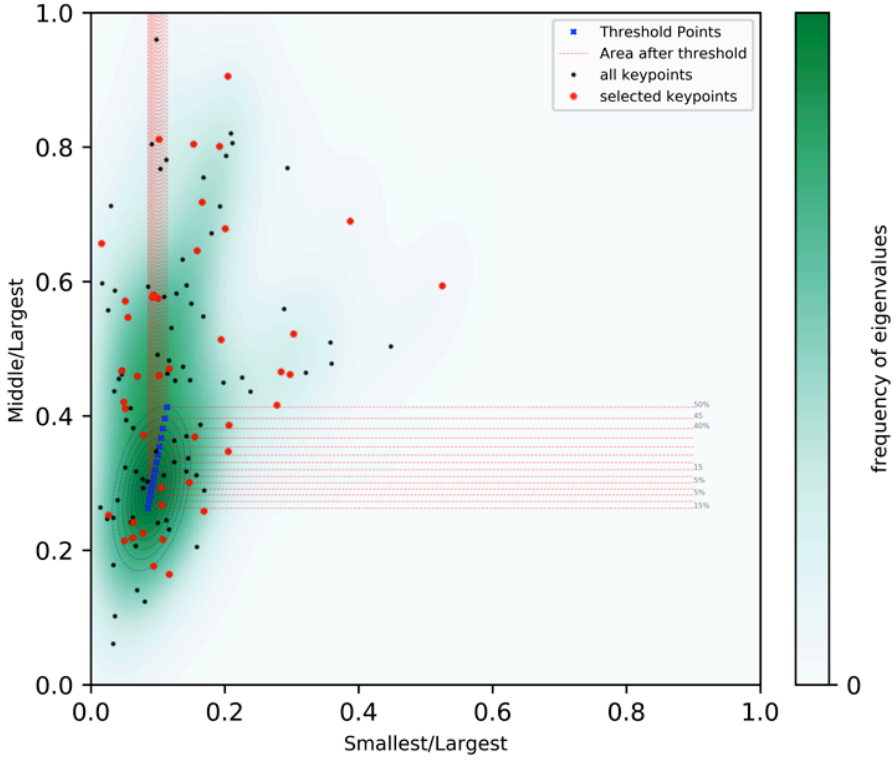
   The idea that we were based on to proceed in this way, apart from the keypoints' deficiency in some cases as mentioned before, is the fact that cryo-EM maps are usually 'composite' in terms of resolution. This means that in a single cryo-EM map, there are regions with higher and regions with lower resolution, depending on the quality of the map in these regions.

2. For these keypoints we fit the KDE on the ratios of their eigenvalues, and compute the corresponding ellipses for the different confidence intervals.

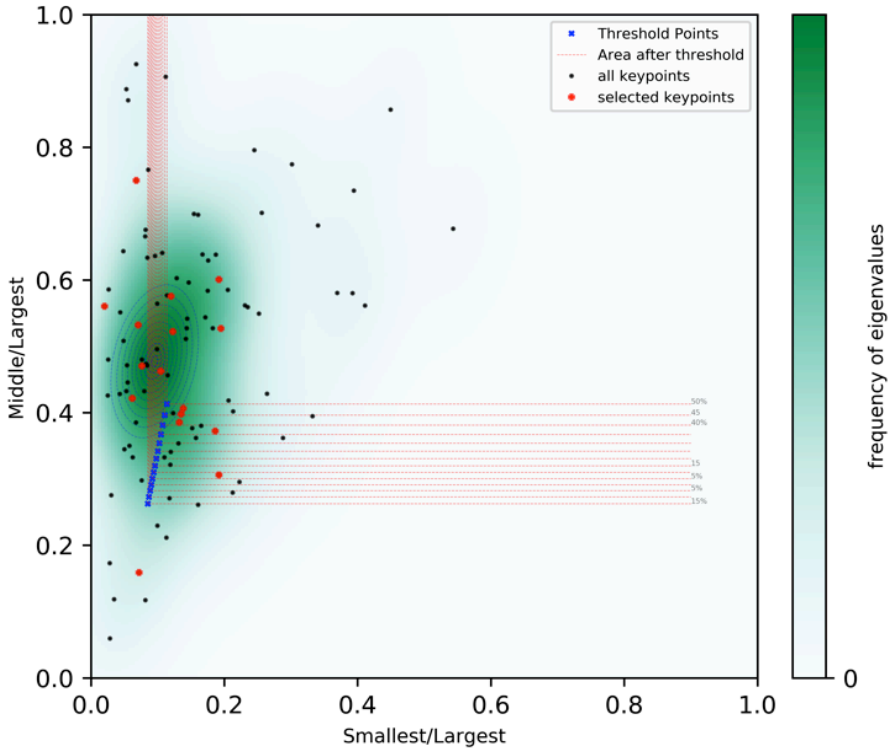
3. Based on the type of the ellipses and their center, we specify the threshold points (that correspond to specific confidence intervals and specific ellipse center), in order to re-apply them on each resolution independently.

An example of how the obtained threshold points (under every resolution) are applied to each individual resolution is depicted in Figure 15.

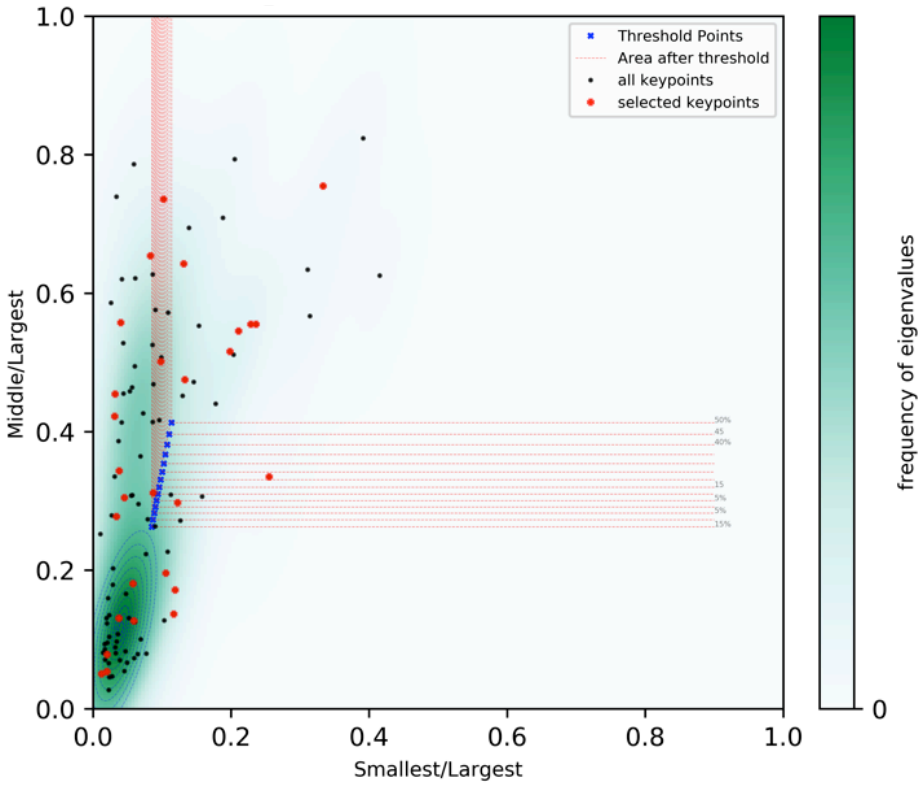
The aforementioned procedure, for a specific structure, and for a specific parameter combination, will result in specific threshold points. As a next step, we will set up a big experiment, where for every structure, for every parameter combination, we will apply these threshold points to each resolution independently, so that eventually we can select the parameter configuration that detects the most stable keypoints that lead to a more accurate prediction.



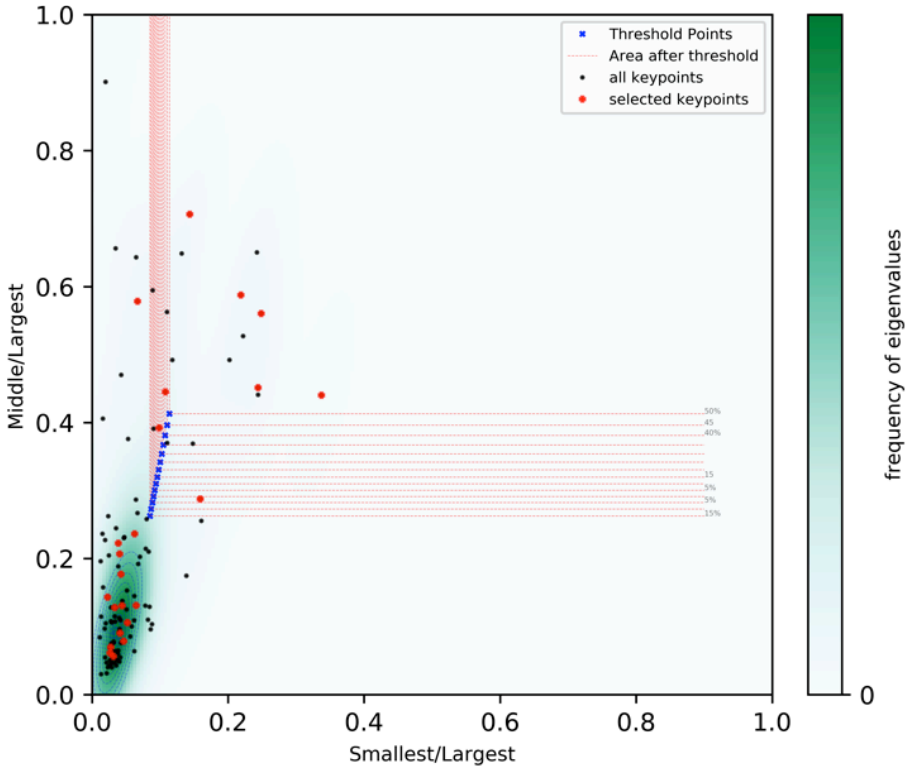
(a) Resolution 7



(b) Resolution 10



(c) Resolution 15



(d) Resolution 20

Figure 15: Applying on each individual resolution the threshold points that were obtained by combining all resolutions. The different threshold points are adjusted to the specified confidence intervals. The ellipses that are shown come from the ratios of the eigenvalues of the extracted keypoints after combining all resolutions.

5. EXPERIMENTAL RESULTS

5.1 Setup

We benchmark the proposed method for the complexes with pdb names {1CS4, 1E6V, 1GTE, 1TYQ, 1URZ, 1Z5S, 2BO9, 2GC7, 7CAT} and their corresponding subunits, as shown in Figure 16. Regarding the parameters, their type and their values are shown in Table 1. As for the experiment, first, for each structure and for each parameter combination, we extract the keypoints for each resolution, we combine these keypoints together, and we compute the threshold points on this combined set. Then, we repeat the same experiment, but now on each individual resolution, and we do not determine any new threshold points, but we apply the already obtained ones. In other words, we define the threshold points over all resolutions, and we apply them to individual resolutions afterwards.

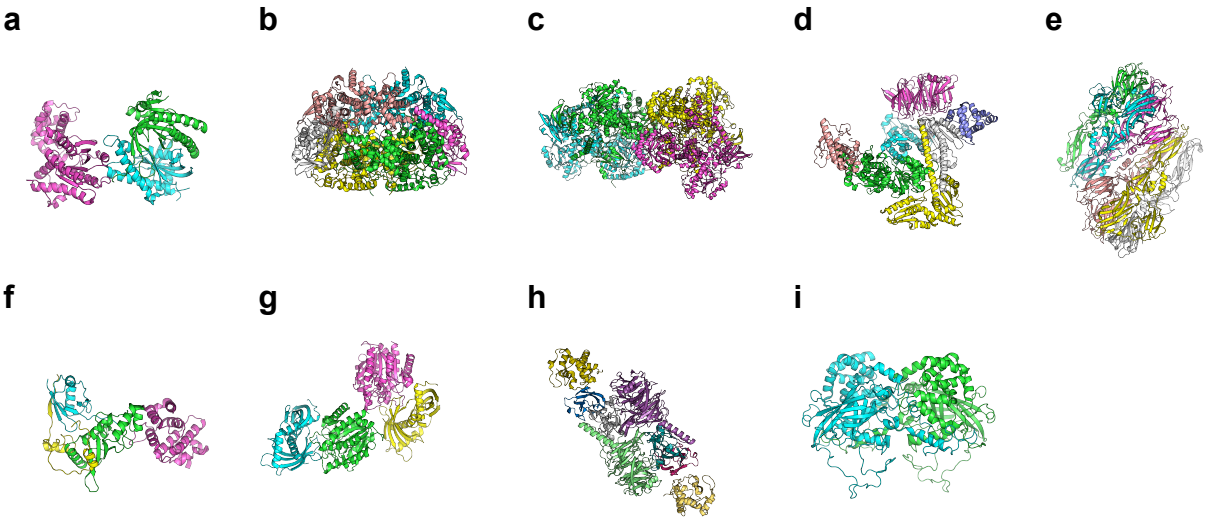


Figure 16: The protein assemblies we use to benchmark our method: (a)1cs4, (b)1e6v, (c)1gte, (d)1tyq, (e)1urz, (f)1z5s, (g)2bo9, (h)2gc7, (i)7cat

Table 1: The parameters and their corresponding values

Parameter	Parameter Value
Neighborhood size	• 3x3x3 • 5x5x5
Gaussian sigma (window)	• 1.0 • 1.5 • 2.0 • 2.5
Eigenvalues' ratio	• M/L-to-S/L • S/M-to-S/L
Confidence interval	• [5, 50]: direction of the eigenvector of the KDE's covariance matrix • 0: threshold point at the center of the ellipse • [-15, 5]: opposite direction of the eigenvector of the KDE's covariance matrix
Ellipse center	• Maximum value of Kernel • Mean value of the eigenvalues' ratios

Let the following be:

- N: the total number of local extrema.
- FP: the total number of local extrema that are false positives.
- TP: the total number of local extrema that are true positives.
- FP_T: the number of false positives that were thresholded
- TP_T: the number of true positives that were thresholded
- N_T: the number of total keypoints that were thresholded

Our goal is to keep only these sets of parameters that retain most of the keypoints that lead to correct assembly prediction, and discard most of the false positives, for every structure, and for every resolution. To determine this, we define three metrics that are based on the ratio of the keypoints that were thresholded:

$T_{TOT} = \frac{N_T}{N}$: the percentage of thresholded keypoints

$T_{FP} = \frac{N_{FP_T}}{FP}$: the percentage of thresholded false positives

$T_{TP} = \frac{N_{TP_T}}{TP}$: the percentage of thresholded true positives

It is obvious that higher T_{FP} and lower T_{TP} at the same time lead to a better solution. Based on these metrics, if *P* denotes the set of all the parameter combinations, we are looking for a solution to the problem:

$$Find\ p \in P\ such\ that: \begin{cases} \max\limits_{p} T_{FP} \\ TP > 0 \\ \min\limits_{p} T_{TP} \end{cases}$$

5.2 Results

5.2.1 Post-processing

Once we obtain the results, we apply a first post-processing step in order to remove some parameter combinations that do not provide solutions. This is done by determining for which parameters, all of the true positives were discarded, since the minimum requirement is a combination to provide at least one true positive. The reason is that even one true positive might be enough for hetAP to make a correct assembly prediction, due to the final step of the software, where it searches in a greedy way a perfect match for the descriptors. In addition, we apply a second post-processing step which has to deal with the resolutions. We keep only those parameter combinations that retain at least one true positive, for every applicable resolution. In this way, we ensure that regardless of the resolution, the remaining parameter sets will always provide solutions. Thus, the final number of parameter combinations is 415, from the initial 448.

5.2.2 General analysis

We first make a general analysis on the discarded keypoints.

1. For each combination of the Gaussian sigma, the neighborhood, the ratio type, the ellipse center and the confidence interval.
2. For all structures, regardless of the resolution
3. Collect T_{TOT}, T_{FP}, T_{TP}

As a result, for every possible parameter combination, we will have a collection of these three percentages, which per triples will correspond to each structure. The distribution of the minimum, the maximum, and the median values of the percentages of every parameter combination, for each collection is shown in Figure 17, Figure 18, and Figure 19 respectively.

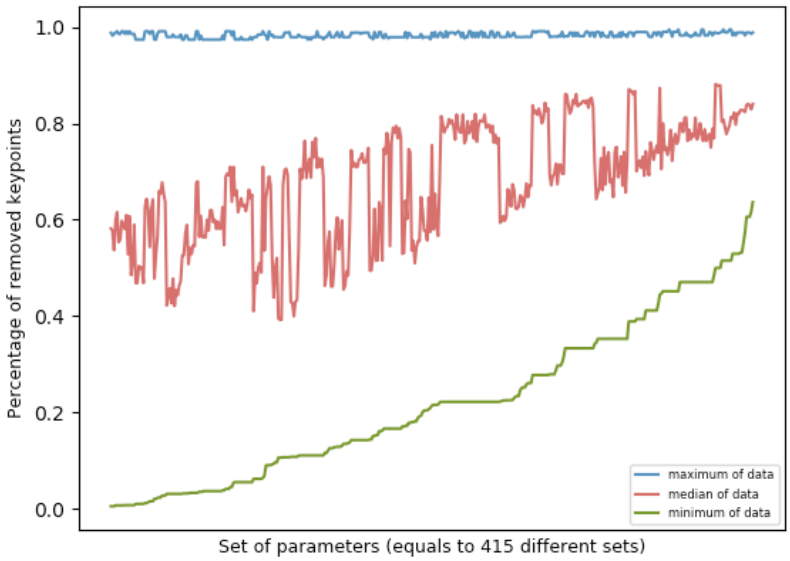


Figure 17: Minima, median and maxima values of every parameter combination, for the T_{TOT}

In Figure 17, we observe that the median values range between 40% and 80%, while the minimum values go up to 60%. This behavior indicates that our method filters out most of the computed keypoints, which was one of our basic goals; to reduce the huge amount of extracted keypoints. In other words, there are parameter combinations that discard at least 60% of the keypoints. However, although this is the overall behavior of our method, we need to also study how both the true and the false positives are handled.

Figure 18: Minima, median and maxima values of every parameter combination, for the T_{FP}

We continue our analysis for the case of the thresholded false positives. As shown in Figure 18, the median values are quite high, ranging from 40% to 85%, while the minimum value reaches again up to 60%. What is quite interesting is that for every parameter combination set (Gaussian sigma, neighborhood, ratio type, ellipse center, confidence interval), there exist structures, such that 100% of the false positives are discarded.

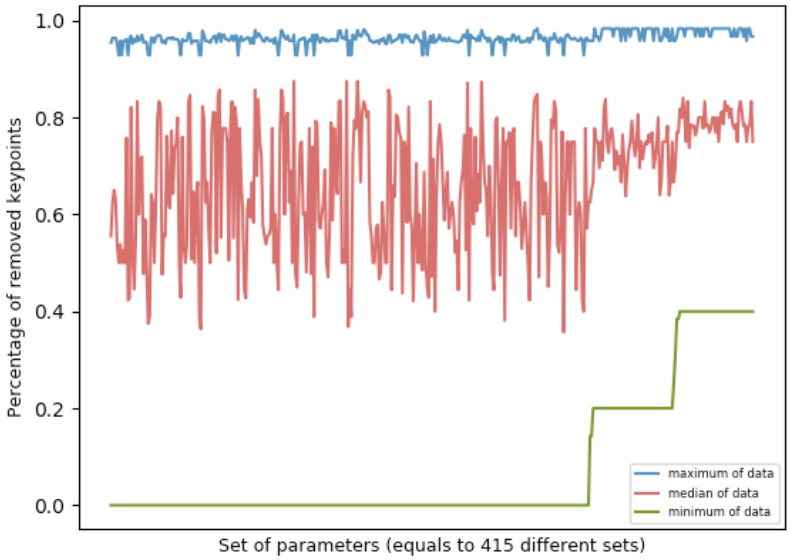


Figure 19: Minima, median and maxima values of every parameter combination, for the T_{TP}

We conclude our general analysis with the distribution of the thresholded true positives, as shown in Figure 19. The behavior is quite different compared to the thresholded false positives. Although the median values range from 40% to 80%, most of them are concentrated around 60%. What is significant and most important to note, is the fact that although the minimum might reach up to 40%, most of its regions are flat, with the biggest part being zero. Finally, we observe that due to the post-processing we mentioned before, we guarantee that the used parameters will always retain some true positives (maximum is never 100%).

From this general analysis, we can safely say that our method satisfies our goal, since indeed discards most of the false positives, and removes 60% of the true positives in average. Thus, our method removes a big number of keypoints, while still keeping solutions. As a next step, we need to go a bit deeper and try to specify the exact set of parameters that leads to the best solution.

5.2.3 Parameter-specific analysis

We now want to control the parameters in a way that will allow us to compare between the different combinations of the neighborhood, the ratio type and the ellipse center, in order to specify the best combination. Note that we do not consider the Gaussian sigma and the confidence interval, since these two parameters have a wider range of values that can take.

To do so, we will study the box-plots of the distributions of the minimum, maximum, and median percentages for all the thresholding cases (T_{TOT}, T_{FP}, T_{TP}), but we will now try to fix the parameters one at a time.

Figure 20 shows the box-plots for the T_{TOT} percentage. We first focus on the choice of the neighborhood. Although the difference between them is not quite significant, we choose arbitrarily the 3x3x3 neighborhood because it is faster to compute, for equal results.

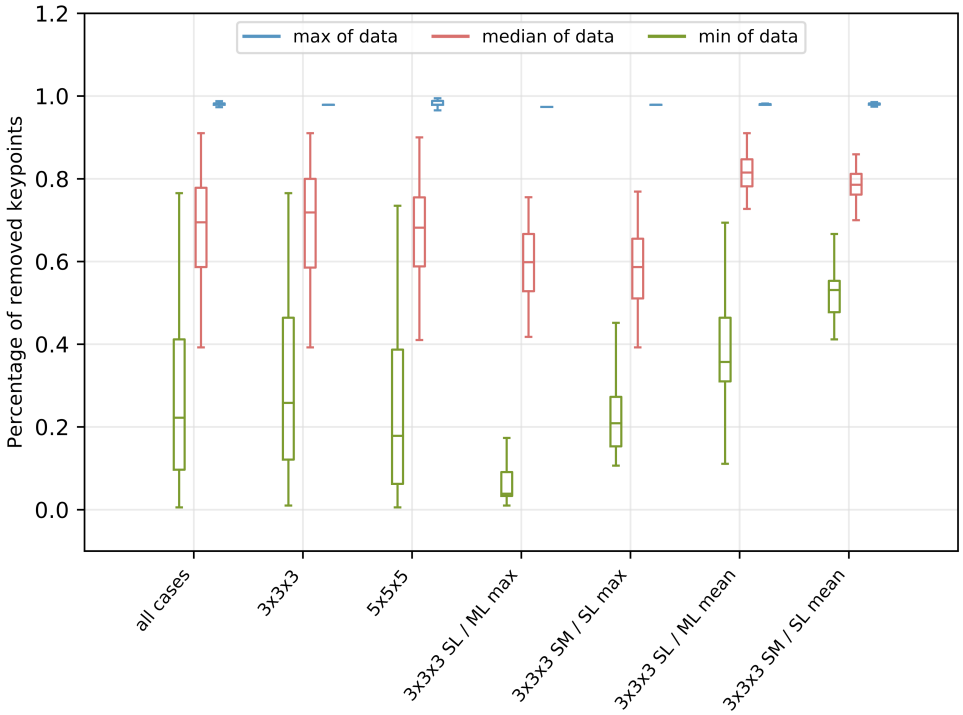


Figure 20: Minima, median and maxima values for the percentage of the discarded keypoints, under fixed parameter combinations

Concerning the other two parameters (ratio type and ellipse center), it is obvious that the choice of SM / SL and the mean value of the eigenvalues' ratios as the center of the ellipses, give a higher removal percentage, since it provides the highest minimum, while the median (~80%) is almost equal to the one obtained by the SL / ML. Thus, based on the values of T_{TOT}, the best parameter combination is:

- Neighborhood: 3x3x3
- Ratio type: SM / SL
- Ellipse center: Mean value of the eigenvalues' ratios

In order to ensure that this set is actually the best, we proceed to the box-plot of the thresholded false positives percentage, which is depicted in Figure 21. We can easily observe that the minimal removal of keypoints goes again up to 60%, and in some cases, we even have 100% of maximum removal. This means that there exist parameter sets that remove all the false positives, and retain only stable keypoints. Thus, the set that gives the best solutions for the T_{FP} is consistent with the best set of the T_{TOT}.

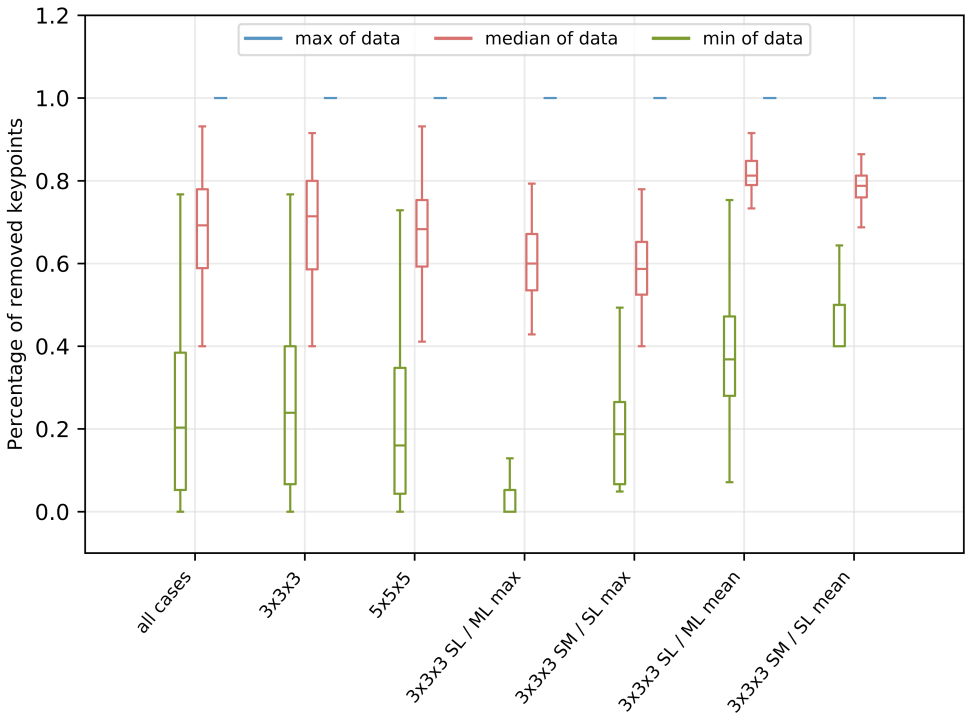


Figure 21: Minima, median and maxima values for the percentage of the discarded false positives, under fixed parameter combinations

Finally, we must study the behavior of this parameter set with respect to the percentage of true positives being removed, as shown in Figure 22. In this case, we would like parameters that keep most of the keypoints. The previous set of parameters is now the one discarding most of the true positives, while still keeping some of them (maximum removal is never 100%). Thus, the best set now seems to be the:

- Neighborhood: 3x3x3
- Raito type: SL / ML
- Ellipse center: Max value of KDE

If we go back to the previous plots, we can easily see that this exact set is one of the sets that keep most of the false positives, as well. In other words, there is not a unique set that maximizes the percentage of discarded false positives, and at the same time minimizes the percentage of discarded true positives. In fact, we have two different set of parameters that cause this trade-off, and the final step is to make a further comparison between them, in order to determine which one to select.

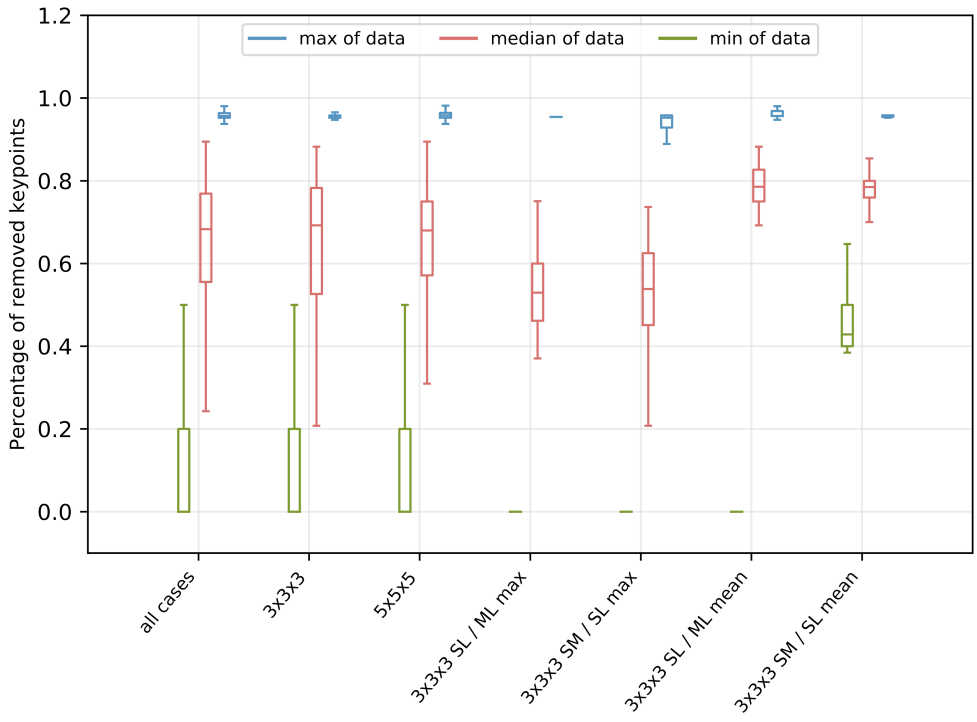


Figure 22: Minima, median and maxima values for the percentage of the discarded true positives, under fixed parameter combinations

5.2.4 Parameter comparison and selection

The parameter set {3x3x3, SM / SL, mean} results in high percentage of keypoints' removal for both false and true positives – 100% removal in some cases – while still having solutions. SM / SL in this case seems to filter out more keypoints, along with the choice of the mean value of the eigenvalues' ratios as the ellipse center, which is also more exclusive than taking the max value of the KDE.

On the other hand, the set {3x3x3, SL / ML, max} results in a lower percentage of keypoints' removal, even in the case of false positives. SL / ML seems to be more conservative in terms of discarding keypoints, which comes from the fact that the points in the ratio plots of the eigenvalues are distributed in a vertical way, and combined with the choice of threshold points on the ellipses, it does not allow an extreme thresholding. The max value of the KDE also supports this soft thresholding, as it is more inclusive than the mean value of the eigenvalues' ratios.

In both cases, we observe that the neighborhood of 3x3x3 results in better solutions, which leads us to the conclusion that it is more reasonable to consider the information that a smaller neighborhood provides, rather than considering more neighboring voxels' gradient information.

We decide that the first parameter set {3x3x3, SM / SL, mean} is the best for our case, as we efficiently diminish the number of the keypoints (false and true positives), while at the same time, manage to retain solutions. Recall that the minimum requirement we want is a combination to provide at least one true positive.

Finally, for this set, we wanted to check if there are confidence intervals that are more likely to appear. We observed that in most of the cases, the confidence interval is 35%,

and in general the value range is between 30% and 40%. We could not draw any conclusion about the Gaussian sigma choice.

5.2.5 Structural specificity

As an additional notice, we observed that regardless the parameters, there are three structures for which we cannot find solutions:

- 2GC7_chainB
- 2GC7_chainF
- 1Z5S

If we take a look at their secondary structures (Figure 23), 2GC7_chainB and 2GC7_chainF are mostly globular and of constant density, whereas 1Z5S is a single coil with no much secondary structure. We conclude that these are three cases with an untrustworthy density profile in general, and that we expect that the solutions found will be less or even zero.

a **b**

Figure 23: Secondary structures of the three aforementioned proteins: (a)2gc7 [B,F], (b)1z5s [A]

5.2.6 Computational complexity

Finally, we would like to provide a qualitative analysis on the effect our method in terms of computational complexity. To understand the order of the impact, we provide the following example:

Recall that hetAP takes every subunit iteratively and tries to "match" its descriptors with the descriptors of the complex in a greedy way, so as to localize the subunit within its complex. Thus, let us say that we have 10^2 subunit descriptors and 10^3 complex descriptors. Then, hetAP will have to apply 10^5 comparisons, in order to match the former with the latter. Now, that we have chosen the best parameter set, we have 80% keypoint removal, which means only $4x10^3$ comparisons to make. In other words, we can reduce the theoretical computational time in hetAP by 96%.

6. CONCLUSIONS AND FUTURE WORK

In this work, we explained the importance of the multimers' structure for the definition of their function. We focused on hetAP software's vulnerabilities, and integrated a method for filtering the extracted keypoints, in order to retain the most stable ones; that is the ones that will lead to correct assembly prediction. Our method extends the Harris corner detection to the 3D space, and uses a structure tensor analysis to discard the unwanted keypoints. We benchmark a set of different parameters, and specify the proper thresholds based on the statistics of the structure tensor, and select this parameter set that is more likely to lead to correct assembly prediction. The proposed method is accurate, computationally efficient, and generalizes across different protein structures, regardless of their resolution. However, we observed that some structures do not apply to our method, and we did a preliminary analysis on their secondary structure.

As a next step, we aim to study how all the extracted keypoints of our method hold over the conformational changes, by taking advantage of Molecular Dynamics. In more details, each system (atoms and molecules) evolves dynamically; it consists of trajectories defining the position of each atom or molecule, at a specific moment (momentum). However, the high-dimensionality of such systems requires ways to reduce the big number of the produced trajectories. Thus, we can make use of clustering methods so as to group the trajectories, and find one representative for each cluster. Then, for each representative we will extract and filter – with the proposed method – the keypoints, and we will compare them with the keypoints of the other representatives. These keypoints that are consistently present in each cluster - and that are already known to lead to correct assembly prediction – will be the ones that alone can characterize a specific structure, regardless of its different conformational states.

Furthermore, we will focus on the computation of parameters that are structure, and secondary structure specific (i.e. parameters for a structure that consist of β-sheets or loops, or for a coiled coil protein where α-helices are coiled together etc.). Moreover, current progress in cryo-EM, can determine structures in high resolution (<5 Å), resulting to an increment of the quality and the size of the data. As a result, the number of features that need to be considered for localizing the subunits into their macromolecular assemblies, becomes quite large. For this reason, it is of great importance to find methods for reducing this large number of features, in order to accomplish faster and more effective localization.

Finally, in the sense of a general machine learning framework, it is worth studying the adoption of machine learning techniques for extracting stable keypoints. We might be able to train a system with the data used in our method, where the corresponding targets (stable keypoints) will be based on the obtained results of our method, in order to predict/detect the most stable keypoints that might lead to correct assembly prediction. From a different point of view, another use of machine learning could be towards the parameter estimation; using the data and the results from our method, we can train a system for estimating the best set of parameters that should be plugged in our method, in order to lead to more stable keypoints.

ABBREVIATIONS – ACRONYMS

MD	Molecular Dynamics
IM	Integrative Modeling
STA	Structure tensor Analysis
cryo-EM	cryo-Electron Microscopy
cryo-ET	cryo-Electron Tomography
NMR	Nuclear Magnetic Resonance
SAXS	Small-Angle X-ray Scattering
hetAP	heteromultimeric Assembly Prediction
SSD	Sum of Squared Differences
SM	Small-to-Middle
SL	Small-to-Large
ML	Middle-to-Large
KDE	Kernel Density Estimation

REFERENCES

[1] Anderson A.C. The process of structure-based drug design. Chem. Biol. 2003;10:787–797. doi: 10.1016/j.chembiol.2003.09.002.

[2] Cowan-J SW, Fendrich G, Floersheimer A, Furet P, Liebetanz J, Rummel G, Rheinberger P, Centeleghe M, Fabbro D, Manley PW. Structural biology contributions to the discovery of drugs to treat chronic myelogenous leukaemia. Acta Crystallogr D Biol Crystallogr. 2007 Jan;63(Pt 1):80-93. Epub 2006 Dec 13.

[3] Protein Data Bank (PDB): https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/methods-for-determining-structure

[4] Brünger AT. X-ray crystallography and NMR reveal complementary views of structure and dynamics. Nat Struct Biol. 1997 Oct;4 Suppl:862-5.

[5] Russel D, Lasker K, Webb B, Velázquez-Muriel J, Tjioe E, Schneidman-Duhovny D, et al. Putting the pieces together: integrative modeling platform software for structure determination of macromolecular assemblies. PLoS Biol. 2012; 10(1):e1001244.

[6] Alber F, Förster F, Korkin D, Topf M, Sali A. Integrating diverse data for structure determination of macromolecular assemblies. Annu Rev Biochem. 2008;77:443–477.

[7] Tamo GE, Abriata LA, Dal Peraro M. The importance of dynamics in integrative modeling of supramolecular assemblies. Curr Opin Struct Biol. 2015 Apr;31:28-34.

[8] I. Andre, P. Bradley, C. Wang, D. Baker Prediction of the structure of symmetrical protein assemblies Proc. Natl. Acad. Sci., 104 (2007), pp. 17656-17661.

[9] F. Alber, S. Dokudovskaya, L.M. Veenhoff, W. Zhang, J. Kipper, D. Devos, A. Suprapto, O. Karni-Schmidt, R. Williams, B.T. Chait, et al. Determining the architectures of macromolecular assemblies Nature, 450 (2007), pp. 683-694.

[10] S.J. de Vries, A.D.J. van Dijk, M. Krzeminski, M. van Dijk, A. Thureau, V. Hsu, T. Wassenaar, A.M.J.J. Bonvin HADDOCK versus HADDOCK: new features and performance of HADDOCK2. 0 on the CAPRI targets Proteins, 69 (2007), pp. 726-733.

[11] Degiacomi MT, Dal Peraro M. Macromolecular Symmetric Assembly Prediction Using Swarm Intelligence Dynamic Modeling. Structure. 2013 Jul 2;21(7):1097-1106.

[12] Tamo GE, Maesani A, Trager S,Degiacomi MT, Floreano D, Dal Peraro M. Disentangling constraints using viability evolution principles in integrative modeling of macromolecular assemblies. Sci Rep.2017;7(1):235.

[13] Renaud JP, Chari A, Ciferri C, Liu WT, Rémigy HW, Stark H, Wiesmann C. Cryo-EM in drug discovery: achievements, limitations and prospects. Nat Rev Drug Discov. 2018 Jul;17(7):471-492.

[14] Kuhlbrandt, W. Biochemistry. The resolution revolution. Science 343, 1443–1444 (2014).

[15] Bai XC, McMullen G, Scheres HW. How cryo-EM is revolutionizing structural biology. Trends Bioche, Sciences. 2015;40:49–57.

[16] Doerr A. Single-particle cryo-electron microscopy. Nat Methods volume 13, page 23 (2016)

[17] S. Traeger, hetAP (software), Lab for Biomolecular Modeling (LBM), EPFL, 2016-2018.

[18] Lowe, David G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. 60 (2): 91–110.

[19] Chris Harris and Mike Stephens (1988). "A Combined Corner and Edge Detector". Alvey Vision Conference. 15.

[20] Wriggers W. Using Situs for the integration of multi-resolution structures. Biophys Rev. 2010 Feb; 2(1): 21–27.

[21] H. Moravec (1980), Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Tech Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute.

[22] J. Shi and C. Tomasi (1994), Good Features to Track. 9th IEEE Conference on Computer Vision and Pattern Recognition. Springer.

[23] S. M. Smith and J. M. Brady (May 1997). "SUSAN – a new approach to low level image processing". International Journal of Computer Vision. 23 (1): 45–78.

[24] S. M. Smith and J. M. Brady (January 1997), "Method for digitally processing images to determine the position of edges and/or corners therein for guidance of unmanned vehicle". UK Patent 2272285, Proprietor: Secretary of State for Defence, UK.

[25] M. Trajkovic and M. Hedley (1998). "Fast corner detection". Image and Vision Computing. 16 (2): 75–87.

[26] E. Rosten and T. Drummond (May 2006). "Machine learning for high-speed corner detection,". European Conference on Computer Vision.

[27] J. Ryde and J. A. Delmerico, Extracting edge voxels from 3d volumetric maps to reduce map size and accelerate mapping alignment, in Computer and Robot Vision (CRV), 2012 Ninth Conference on. IEEE, 2012, pp. 330–337.

[28] Vincent Spruyt, How to draw an error ellipse representing the covariance matrix, http://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/
[29] Pawel A Penczek, Analysis of Conformational Heterogeneity of Macromolecules in Cryo-Emectron Microscopy, lecture of the University of Texas-Houston Medical School, Department of Biochemistry.
[30] M. Podobnik, M. Kisovec, G. Anderluh. Molecular mechanism of pore formation by aerolysin-like proteins. Philos Trans R Soc Lond B Biol Sci. 2017 Aug 5;372(1726).