UNIVERSITY CEU - SAN PABLO

POLYTECHNIC SCHOOL

BIOMEDICAL ENGINEERING DEGREE

BACHELOR THESIS

Image processing algorithms for the determination of the optical aberrations o an electron microscope

Author: Federico Pedro de Isidro Gómez
Director: Carlos Óscar Sánchez Sorzano

July 2019

UNIVERSIDAD SAN PABLO-CEU
ESCUELA POLITÉCNICA SUPERIOR
División de Ingeniería

Datos del alumno

NOMBRE:

Datos del Trabajo

TÍTULO DEL PROYECTO:

Tribunal calificador

PRESIDENTE:	FDO.:
SECRETARIO:	FDO.:
VOCAL:	FDO.:

Reunido este tribunal el _______/_______/_______, acuerda otorgar al Trabajo Fin de Grado presentado por Don___________________________________la calificación de ___________________.

ACKNOWLEDGMENTS

“All that is gold does not glitter,

not all those who wander are lost”

John Ronald Reuel Tolkien

ABSTRACT

Electron microscopy (EM) is one of the state-of-the-art techniques in imaging technologies and, more in relation with this project, in structural biology. The Contrast Transfer Function (CTF) characterizes the imaging system of the microscope. The proper calculation of the CTF allows a posterior correction of the acquired image.

In the first part of this project some corrections of an already developed CTF estimation algorithm are presented. For this purpose, the algorithm has been dissected and corrections have been performed in some steps that resulted in a significant improvement of the CTF estimates. The main corrections were focused in the selection criteria of the CTF estimation and the defocus calculation of the micrograph.

Once all the corrections were implemented, the presence of false positive estimations of the CTF decreased a 95.76% from all the micrographs processed.

After observing the impact of a correct calculation of the defocus of the micrograph, the second part of the project consisted in the development of a deep neural network that allows the correct calculation of the defocus.

This new algorithm improves the defocus estimation presenting a mean absolute error of 504.1992 from all the micrographs processed.

RESUMEN

La microscopía electrónica (EM) es una de las técnicas mejor establecidas en la obtención de información estructural de macromoléculas biológicas. La función de transferencia de contraste (CTF) permite la caracterización del sistema de imagen del microscopio. La correcta estimación de la CTF permite la corrección posterior de las imágenes obtenidas.

En la primera parte de este proyecto se presentan algunas correcciones en el algoritmo, ya existente, de estimación de la CTF. Para ello, dicho algoritmo se ha diseccionado y se han aplicado correcciones en aquellos pasos que han resultado en una mejora significativa de las estimaciones. Las principales correcciones se concentran en los criterios de selección de la estimación y el cálculo del desenfoque de la micrografía. Este procedimiento no consiste en el desarrollo de un nuevo algoritmo si no en la corrección de uno ya existente.

Una vez implementadas las pertinentes correcciones, la presencia de falsos positivos en las estimaciones de la CTF se han reducido un 95.76% entre todas las micrografías procesadas.

Tras observar la importancia de un correcto cálculo del desenfoque de la micrografía, la segunda parte del proyecto consistió en el desarrollo de una red neuronal profunda capaz de calcular correctamente el desenfoque.

Este nuevo algoritmo permitió el cálculo del desenfoque con un error medio absoluto de 504.1992 entre todas las micrografías procesadas.

INDEX

FIGURE INDEX

TABLE INDEX

1 INTRODUCTION

In this section a general overview of the project is presented. The main structure is introduced and the basis that supports and give sense to the development of the project are stablished, in consideration with the background and objectives set.

1.1 Motivation, background and context

Imaging techniques are one of the most prolific and developed fields in engineering, even more in the case of medical imaging. The last decades have supposed a revolution for imaging techniques and image processing, allowing to address new domains and contributing with new solutions to different fields, offering a new powerful source of detailed and specific information. Among all the different imaging techniques, this thesis is focused on electron microscopy of single particles.

Electron microscopy is one of the latest and most disruptive techniques in image processing and its usage has been tested against several challenges. In order to accomplish the development of this technique a broad implication of several fields is needed. Electron microscopy belongs to the set of image techniques that require a comprehensive post processing of the images obtained.

The development of electron microscopy has meant a new contribution to the collection of imaging techniques. This technique can push further the resolution limit that can be obtained by the rest of the available techniques, even more after the resolution revolution. The resolution revolution comprehends the significant increment of the maximum resolution of the technique after the development of computational methods and a robust theoretical background.

This increment in the resolution allows the study of new structures and complexes unable to be analyzed before, having special applications in different fields from biology to material science. This project has been developed in the Biocomputing unit of the Natl. Center of Biotechnology, CSIC (CNB), focused on the study of macromolecular structures.

This technique has several other applications outside the study of macromolecular complexes. Still within the biological research field, this technique is also used in nanotechnology and nanomaterials, physiology, histology, toxicology and drug research. Also, this technique is also involved in other research fields as material science and it is widely used in several industries as pharmaceutical, chemical industry, forensics or semiconductor design.

The work developed in this thesis has a direct implication in this post processing of the raw image obtained by the microscope, more in detail, in the estimation of the contrast transfer function (CTF). The CTF mathematically describes how the aberrations introduced by the microscope affect the image, and its relevance comes from the fact that this step is one of the first tasks in the post processing of the raw image.

The first part of the project consisted in the application of some corrections in the already developed algorithm for the CTF estimation included in the software package Xmipp (X-windows based microscopy image processing package) [1,2]. The corrections performed in this algorithm are based mainly in two different features: the defocus calculation of the micrograph and the selection criteria for the quality of the estimation.

All the presented methods and algorithms are included in the software framework Scipion, developed by the Biocomputing Unit of the CNB. The source code is open and free access.

Once all corrections were performed in the CTF, a significative decrement in the estimation failure was achieved. Also, and according to the actual literature [3], some conclusions were obtained about the significance of the different parameters according to the CTF estimation, realizing of the fundamental relevance of a correct calculation of the micrograph defocus. Due to this, the second part of the thesis consisted in the development of a robust algorithm able to provide a better calculation of the defocus. For this purpose, a deep learning estimation algorithm has been developed.

Deep learning and artificial intelligence techniques are useful tools for problem solving. Deep learning architectures can produce a prediction in the output of a system from an input information, as long as the collected training data is enough both in quality and quantity. The deep neural network architecture fits nicely with the aim of this project since, from all the information contained in an image set, only a real value (the defocus) must be estimated.

One of the typical drawbacks in the usage of artificial intelligence techniques, if not the most frequent, is the construction of a database with enough information able to properly train the developed network. To achieve this objective, thousands of micrographs have been analyzed, estimating their CTF's and properly discarding the ones that did not fulfil the requirements for the database. Once enough data had been collected the network was trained and the defocus was estimated.

The final defocus estimation tool is included in the CTF estimation protocol allowing a different procedure for the defocus obtention and improving the previous results obtained by this protocol.

1.2 Objectives

As has been exposed in the previous introduction, electron microscopy technique has undergone a prodigious progress for the imaging research field.

Considering the wide range of design implications and fields of application of this technique, it has become necessary to narrow the work line of the project. Here are presented the main objectives established in this project:

1. Make a comprehensive study of the already developed algorithms, analyzing its functioning and identifying the main sources of possible failures.
2. Apply the pertinent corrections necessary for a significant improvement in the results obtained by the algorithm, performing a thorough study of them.
3. Development of a database able to train a deep learning network in order to predict the defocus of a contrast transfer function.
4. Design a deep leaning network able to predict properly the defocus of a contrast transfer function once it is properly trained.
5. Compare the results obtained from both methods, implementing the usage of the deep neural network or not, and establishing a robust solution for the contrast transfer function protocol.

1.3 Document Structure

This document is structured in seven different sections:

In this first section the main topic of the project is presented, including the main motivation to accomplish it, its background and the context that surrounds it. Also, the main objectives and the structure of the project are introduced.

In the second section the actual state-of-the-art of the electron microscopy technique is introduced. For this the first part of this section a brief historical introduction of the technique is presented to subsequently give way to the main physical and computational fundamentals of the electron microscopy technique and the single particle analysis.

In the third section the material and methods involving this project are presented. This is the first section where the project is separated in two different branches considering the two main parts of the project: the CTF estimation correction and the development of the deep learning algorithm for the defocus estimation of the micrograph. For each branch in this section the performed implementations are presented, including their description and theoretical implications.

In the fourth section the results obtained from the implementations explained in the last chapter are presented. As in the previous section the results are presented in two different paragraphs one per each branch in which the project consists.

In the fifth section it is presented a discussion about the results obtained from the previous section. In this discussion the results will be explained in context for a better understanding of the implications of this project. As in the two previous sections, two different paragraphs are included for each branch on the thesis.

In the last section some final conclusions are exposed, summarizing and highlighting the main ideas showed in this project.

There is a last bibliographic appendix in which all the references used during the elaboration of the project and thesis are referenced.

2 STATE OF THE ART

2.1 History

The history of electron microscopy began in 1926, when Busch stated the first steps in the development of electron optics. Busch studied the trajectories of charged particles in axially symmetric electric and magnetic fields, observing that such fields could act as a particle lens, respecting the foundations of geometrical electron optics. Some years earlier in 1923, de Broglie introduced the concepts of corpuscular waves, associating a wavelength with charged particles. this implied the beginning of the electron optics. [4]

Although these previous developments, the most spread date is the year 1931 when the first electron microscope was built by Ernst Ruska, he was awarded for it with the Nobel prize in 1986 [5]. Ruska and his research team were disappointed learning that even with electrons, the wavelength would limit the resolution, but they found that, using de Broglie equation, electron wavelength were almost five orders of magnitude smaller than the wavelength of light used in optical microscopy. In 1932 Knoll and Ruska tried to estimate the resolution limit formula, obtaining a theoretical limit of 0.22 nm at an acceleration voltage of 75 kV. This value was experimentally reached only 40 years later, after many technical limits were overcome although there was a theoretical base for it [4].

In 1938, two prototypes of an electron microscope came into operation after Siemens and Halske developed a new Ultramicroscopy laboratory, reaching a resolution of 7nm. In this same year, Hiller and Prebus constructed a new transmission electron microscope (TEM) using magnetic lenses, reaching a magnification of 40.000X and a resolution of 6 nm. [6]

On the other hand, in 1935 Knoll built a first approach to the scanning microscope with a resolution limit of 100 µm [4], leading to the construction of the first scanning electron microscope (SEM) by von Ardenne in 1938 [6]. Also, von Ardenne clearly described the theoretical principle underlying the scanning microscope, it is known it today. Some years later, in 1942, Zworykin described and developed a new scanning electron microscope and showed that secondary electrons provided topographic contrast. This new device reached a resolution of 50 nm, still considered low compared to the performance obtained by transmission electron microscopes [4].

In the following years, the electron microscopy techniques were developed through several countries as France, USA, Switzerland or Sweden, increasing gradually their resolution. With this increment of resolution new applications were available, most of them focused on biology and materials.

In 1948 Oatley began to build an SEM based on the Zwoeykin's and, in 1963, Pease and Nixon combine the recent improvements in a single instrument and included the new detector designed by Everhart and Thorley three years earlier. This research program culminates with the accomplishment of the first commercial design for a scanning electron microscope in 1965. [4]

In these same years, from 1948 to 1953, an important development outside of electron microscopy had an enormous implication on it, the invention of the ultramicrotome. The arrival of the microtome made possible to cut very thin sections of biological samples improving the quality of the micrographs and reducing the need of microscopes with a higher voltage for analyzing thicker samples. [6]

In 1956 Smith showed that signal processing could be used to improve micrographs, becoming the first post processing implementation in electron microscopy [4], and in 1965, Hanszen introduces the notion of contrast transfer function into electronic optics, culminating with its visualization in 1966 by Thon. [6]

In the decade of 1960's a new line of research was established in the electron microscopy technique: the electron tomography. Electron tomography used previous tools and concepts in its development established years before. The mathematical concepts of tomography to reconstruct a 3D volume from a set of 2D projections was first proposed by Radon in 1917, although this early mathematical foundation was not practically applied for 3D reconstruction until 1956, in astronomy, and in the field of medicine in 1961, for medical imaging using X-rays. [8]

In 1968, electron microscopy and image processing overcame a new defeat obtaining the first three-dimensional reconstruction by Rosier and Klug, specifically from a virus. Soon after, Crowther had placed the theoretical footing. In this same year, the first attempts to process scanning microscope images were performed. [6]

In the year 1970 a disruptive event took place in the electron microscopy timeline since the first full study about the mechanism of scanning transmission electron microscopy (STEM) image formation was released. Using this same technology Crewe presented images of single atoms obtained by forming the ratio of the elastic and inelastic images. This contrast mechanism is known as “Z-contrast”, since the ratio of scattering cross section is approximately proportional to the atomic number Z. In the same year, new studies rose the concern on the radiation damage on the specimen due to the electron bombardment, especially for biological samples. [6] This critical problem is counterbalanced by the high resolution attainable with the electron microscope [7].

In 1972, a new milestone was reached for electron microscopy after the development of real-time, direct-view stereoscopic scanning electron microscopy, enabling microdissection and 3D video recording. In the same year, the scanning transmission electron microscopy is used for molecular weight determination by Engel. Two years later, and due to the improvement of the techniques and the worries on the radiation damage, interest in low voltage scanning electron microscopy revived. [6]

Apart from the improvement in biological research and the cumulation of new information, in 1976, crystallographic information was extracted from convergent-diffraction beam patterns. This work line was taken the posterior years and in 1980, new crystallographic information was extracted from convergent beam electron diffraction patterns with a large angular view. [6]

In 1981 a new breakthrough was achieved in the electron microscopy technique by Dubochet and McDowall with the introduction of the cryo-technique. This technique was designed in order to improve the specimen protection from the previous solutions, based in the amelioration by dehydration or water-substitution methods and the mitigation by stain with heavy metals. With this technique the biological specimen is embedded in water or a buffer solution by rapidly freezing to temperature of liquid nitrogen (90ºK), allowing the study of the sample in its native state without the introduction of preservation artifacts. [7]

For several years, the computer-controlled microscopy was developed, reaching a robust stable point by the year 1989. One of the most relevant contributions in the development was made by Koster, although several teams worked in the performance of this technique. In this same year an intensive documentation explaining the advantages in the combination of magnetic and electrostatic lenses is released by Frosien, Plies and Anger. Also, the year 1994 supposed the beginning of a new branch in this field in tele-microscopy. [6]

After almost 70 years of development and several new solutions contributed to different field, in the year 1997 the relevance of electron microscopy was broadly recognized and several historians in science became interested in covering the trajectory of this technique. [6]

In the last years of the 1990's decade, the implementation of a lens-aberration corrector finally achieved an improvement in resolution, reaching 1.4 Å per pixel resolution. [8]

In the 21st century new improvements have been performed both in electron microscopy and electron tomography. In the electron microscopy filed new improvements in the device and the post processing image have triggered the ability to image atoms models at 0.5 Å per pixel resolution [8]. One of the most significant advances for cryo-electron microscopy was the development of a new family of detectors, the direct detector devices (DDD) or direct electron detectors (DED), by J. Frank and R. Henderson.

In the electron tomography field, the most disruptive improvement has been performed in the last recent years. Due to the high resolution obtained by electron microscopy and its close result to the limit resolution, the electron tomography is known as the future approach for high 3D resolution structures identification in many fields. Several new solutions in instrumentation, computational power, and reconstruction algorithms have improved the outcome of this technique. Example of promising directions in both hard and soft materials include high resolution tomography of nanoparticles, tomographic studies on single particles, the structural determination of very large biological complexes, and the structure of whole cells in a native environment. [8]

It is also worth mentioning that in the last recent years it has been an improvement in cryo-electron microscopy for structural biology based on their techniques to achieve 3D imaging: single particle electron microscopy (cryo-EM), cryo-electron tomography (cryo-ET), cryo-electron crystallography (cryo-EC), and electron diffraction (ED). As it is mentioned in the previous paragraph, cryo-electron tomography is the best 3D imaging method that can image cells and its organelles in their native state at molecular resolution. [7]

2.2 Fundamentals of electron microscopy

Whether considering electromagnetic waves or particles, a radiation microscope consists of a device able to get a magnified image of a sample employing any kind of energy which fulfills the duality wave-particle. From now on, the exposition is focused on the electron microscope. [5]

Modern microscopes are constituted by several optical elements. As an initial brief introduction of the working system of a microscope, the first element required is a source of radiation together with a collimator or condenser lens that work as an illumination system, lightning the sample. Then, the radiation, after interacting with the sample, will be detected by an imaging system composed by an objective and an ocular. This imaging system will capture the scattered radiation by the sample and record an image.

Between the sample and the detector, two groups of lenses are disposed. The first group are the objective lenses, which are the closest elements to the sample inside the microscope and in charge of the collection of the radiation scattered by the sample, projecting and magnifying it. Finally, the second group of lenses, the ocular system, produces a new magnification of the radiation to improve the image received by the detector. A general overview of the system can be visualized in Figure 1. [5]

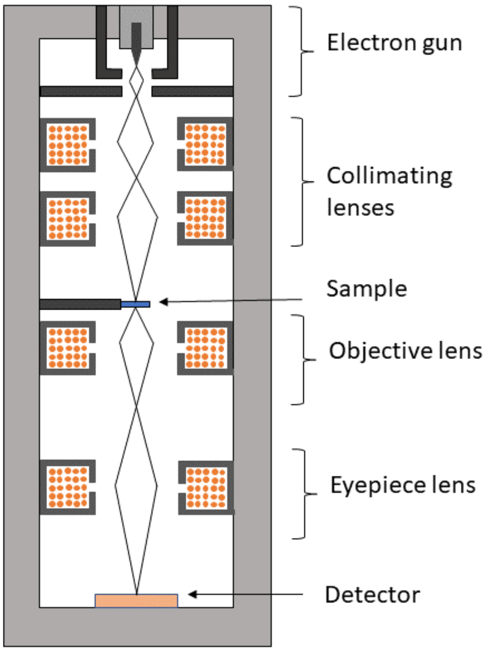


Figure 1: Scheme of the electron microscope including all its components. [5]

Two principal approaches can be employed to address a sustainable theoretical framework to describe the microscope functioning. The first one relies on the geometrical optics: operating with this approach the wavelength and the energy of the radiation can be neglected after a comparison with the dimensions and energy of the electron elements. This approach leads to a corpuscular treatment of the radiation and the obtention of the image is treated as impacts of the particles (the electrons) onto the detector. The second approach relies on the electromagnetic theory or physical optics: operating with this approach the wavelength of the radiation is comparable to the elements involved in the system, but the energy of the radiation is not able to interact due to the small energy that it presents. [5]

Following the geometric approach for the study and design of a microscopes, some limitations are established. Geometrical optics defines an imaging system as perfect (or stigmatic), always it fulfills the Maxwell conditions of Maxwell optical theorems:

1. There exists a relation one-to-one between object and image (the image of a single point is a single point).
2. There is a similarity relationship between object and image being the magnification the similarity ratio.
3. If the object is on a plane, the image lies on another plane. [5]

These conditions are the fundamental constraints for the design of an image system, and they are limiting the correct functioning of the system. The two main issues that leads to the rupture of these laws are the aberrations and the diffractions. Aberrations are any deviations that leads to an imperfect image, and it depends on the image systems. Diffraction is an intrinsic property of radiation due to the interaction of the radiation with the matter, which is unavoidable. Both phenomena are common for all optical systems. Under these conditions it is always fulfilled that the image obtained from a point-object is always a point. [5]

Before proceeding with a more accurate description of the elements of an electron microscope, some analysis about the use of electrons must be performed since it is a differential point of this technique. The use of electrons increases the diffraction limit in comparison to the optical microscope limit, but, on the other hand, it presents some drawbacks from an experimental point of view:

1. There are no natural electron sources, implicating the design of an electron gun able to provide enough energy to accelerate electrons towards the sample without interacting with the rest of the system.
2. Due to the strong interaction electrons-matter the system needs to work in vacuum and must be supplied with a source of enough energetic electrons.
3. The sample can be damaged by the electron radiation, ionizing, heating and degrading it. To solve this issue, radiation must be as low as possible, considering the compromise with contrast and hence with resolution.
4. Focusing electrons suppose a challenge due to it requires of the correct operation of a complex system of electron lenses. Also the high energy presented by the electrons makes even more difficult their correct focusing.

5. The high radiation also damages the sensors and detection system, reducing the lifetime of these devices and compromising the contrast and quality of the obtained images. [5]

Although the drawbacks presented, the solutions proportioned and the correct development of the technique, including its exceptional results, compensates them.

Once the use of electrons for the image conformation is clarified, a clear description of the different components of the microscope will be presented. The first element to be analyzed is the electron gun. Every imaging system requires an illuminating source in order to produce an image, being in the case of this technology a source of electrons. Unfortunately, this illuminating system must fulfill some restrictions, basically a high energy throughput and stability of the radiation. This high energy rate of the radiation will assure the electron transmission through the sample under a weak interaction or even if there is none. The other stability constraint is the most critical one since it involves three experimental challenges: monochromaticity, coherence and collimation. [5]

Monochromaticity implies that no fluctuation of energy appears in the energy beam, or from another point of view, that the wavelength of the electrons radiated is maintained constant. Coherence assures that a known phase radiation is maintained through all the radiation, both temporal and spatial. Respectively, the temporal coherence is presented by a beam if the phase difference is defined properly between two instants of time, meanwhile, in the case of the spatial coherence, the phase difference must be well defined between two different emitters. Finally, collimation implies a distribution of the electrons emitted from the gun as parallel as possible, in order to avoid the divergence of electrons. [5]

The functioning of an electron gun consists in a material, typically a metal, that under certain conditions emits electrons. For this purpose, the electrons of the material must lay on its surface in order to be emitted and not to form an internal electric field generating currents when it is under certain conditions of temperature and electric field. This is the reason why it should be a metal material. [5]

Also, the pattern of the metal emitter is relevant since the electric field and therefore the electron density, is higher in the sharpened regions under an electric field. Therefore, the shape of the electron gun must be as sharpened as possible, maximizing the charge density and the electric field. This also has come positive implications in the phase difference of the emitted front wave. [5]

The structure of an electron gun is composed of three main components: the cathode o filament, the Wehnelt cylinder or grid, and the anode. The cathode will be contained inside the Wehnelt, a metal hollow cylinder with a hole in its basis. Then, a high differential voltage is applied between the cathode and the anode, guiding the electron current by modifying the potential of the Wehnelt cylinder. This difference of potential allows to focus the electrons beam, increasing the directionality. [5] All these elements can be visualized in Figure 2.

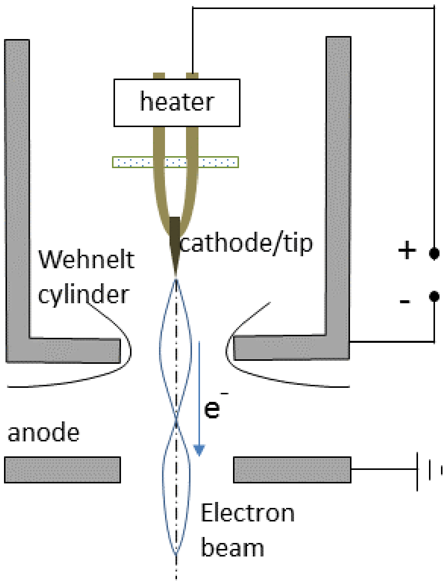


Figure 2: Scheme of an electron gun including all its components. [5]

Although the previous exposition properly explains the general functioning of an electron gun, there exist different types of electron guns which are going to be briefly exposed:

1. Thermionic electron gun: this kind of electron gun is based in the increment of the temperature of the tip of metal to achieve the electrons

emission. With this technique the electrons acquire enough thermal energy to overcome the potential barrier and be emitted. Only few metals can be used for this technique since high temperatures have to be reached without melting.

2. Field electron emission gun: this kind of electron guns is supported by the fact, although the electrons ought to overcome the potential barrier to leave the tip, due to tunneling effect, the electron has a chance to be out the tip when the energy is lower that the working energy. For this, the electron gun supports the tunneling effect by immersing the tip in a strong electric field.
3. Schottky emission guns: modern electron guns are supported by this effect which can be considered as a combination of both thermionic and field emission effect. For this combination the basis is to heat the cathode reducing the potential barrier, aiding the emission field to emit the electron by reducing the potential barrier. [5] In Figure 3 energetic profile function of this phenomena is showed.

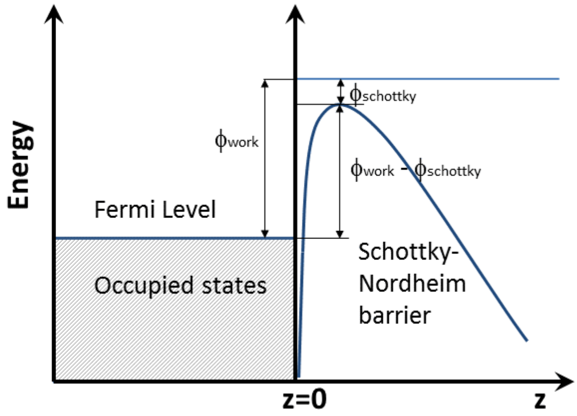


Figure 3: Work function of the Schottky-Norheim effect. [5]

Once the electron guns have been explained, it will be presented the next main core elements of al electron microscope, the magnetic lenses. A lens can be defined as every element that makes a collimated radiation beam converge towards a fixed point. In the case of the light optics, a variation of the refractive index is caused in the material in which the radiation is propagated. Since the electron microscope works in vacuum, implying there is no method for a material medium change, another mechanism for electron focusing must be applied. [5]

The main difference between electromagnetic waves and electrons is the mass and the electric charge, which builds a perfect scenario for the application of the Lorentz equation:

$$\boldsymbol{F} = e[\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}]$$

which relates the force suffered by a particle with charge e and speed $\boldsymbol{v}$ when it is introduced in an electromagnetic field composed by the electric field $\boldsymbol{E}$ and the magnetic $\boldsymbol{B}$. [5]

As the Lorentz equation proposes, electron trajectories can be modified wisely by the application of electromagnetic fields, which carries to the constitution of two different types of lenses: electrostatic and magnetic. Since the magnetic lenses are the most common used in the field only its basis will be exposed. [5]

Between all the different categories of magnetic lenses, only the ones that presents a rotationally symmetric shape will be suitable for this technique. This constraint is crucial since any other shape would not respect the constraints in the shape of the magnetic field generated, which must be also rotationally symmetric. In fact, the intensity of the magnetic field generated should depend on the distance to the center of the lens, presenting a smooth decay as it moves away from the center of the lens, presenting in this point its maximum. [5]

However, the previous explanation is slightly incorrect since, if a constant magnetic field is considered, an electron passing along the $\boldsymbol{z}$ axis, would not experiment any deviation, meanwhile an electron passing at a given distance from the $\boldsymbol{z}$ axis will suffer a Lorentz force to be focused that increases accordingly to its distance to the $\boldsymbol{z}$ axis. In order to correct this behavior a radial magnetic field is also needed to correctly focus the electrons beam. In order to generate this radial field several coils encapsulated in an iron box are used. This iron box is used in order to produce some confinement of the field and it presents a small hole increasing its intensity in the aperture area. This disposition is going to ensure that an intense enough field is achieved in order to produce a significant change in the electron's trajectory. [5]

The expected electron trajectory through the magnetic field is expected to be an helicoidal movement. With the proper arrangement of the magnetic fields the electron beam is properly focused. [5] A schematic of the disposition of the difference elements of an electromagnetic lens and the magnetic field associated to it is exposed in Figure 4.

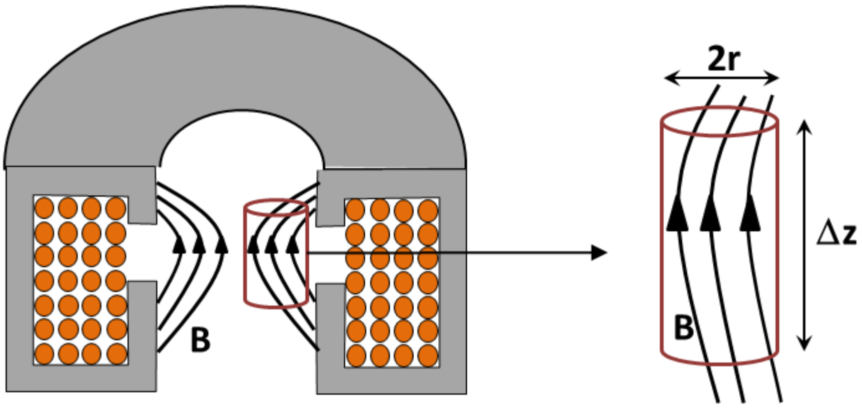


Figure 4: Schematic of a magnetic lenses and its magnetic field associated. [5]

After exposing the design and functioning of the electromagnetic lenses a brief mention must be done about aberrations. Considering an aberration as every deviation of the ideal behavior of an optical system, any failure in the function of a lenses suppose a source of aberrations. An ideal electromagnetic lens produces a spherical wave front form the electron beam, although some deviations in the wave front are inevitable despite the design efforts. Also, it must be considered that the weak interaction of the illuminating beam with the sample is typical of this technique, being the induction of aberrations especially critical in the image formation process. Although there are several procedures that can be applied to avoid this process, the most extended one in the field is the introduction of some defocus that will increase the contrast of the obtained image. [5]

Once the illuminating source of the sample, the electron beam, has been properly exposed, the next important component is the sample and its correct preparation. For the proper colocation of the sample a metallic grid is placed inside the microscope. Each hole is composed by a deposited carbon grating in which the sample will be place after the pertinent biochemist processes, depending on the nature of the run study. The main problem is that after this biochemist process, the result is an aqueous solution, incompatible with the vacuum generated inside the microscope. To solve this problem, different treatments can be applied in order to solidify the sample, leading to two different modalities of electron microscopy:

1. Negative stain: for this technique a small concentration of uranyl acetate is diluted in the sample and, after setting the sample on the carbon grid and removing the excess of solution for a plane shape, the uranyl dries and wraps the sample. Due to the high-density difference between the uranyl and the sample the contrast on the image is increased. The main drawback of this technique is that the obtained image rises information of the uranyl-biological complex interface, so high-resolution information is lost. Also, in case the uranyl does not cover properly the hole complex, as in large complexes, the information of the non-covered part will be also lost. As it can be seen in Figure 5, the images obtained looks like white particles over a black background. Although this technique has been substituted by the vitreous ice technique, it is a useful first step to obtain particle low resolution structures.
2. Vitreous ice: this technique consists in freezing the sample the quickest as possible in order to obtain a non-crystalline structure, known as vitreous ice. The speed in the freezing process is fundamental since it will preserve the complexes in their native state and keeping the high-resolution information. As a drawback, it must be considered that the images will present low contrast although not only the interface will be shown. As it can be seen in Figure 5, the images obtained looks like black particles over a white background. The use of this technique has supposed a revolution in the cryo-electron microscopy field in the recent years. [5]

In Figure 5 can be seen a schematic showing the main differences between both sample preparation techniques and the resulting image.

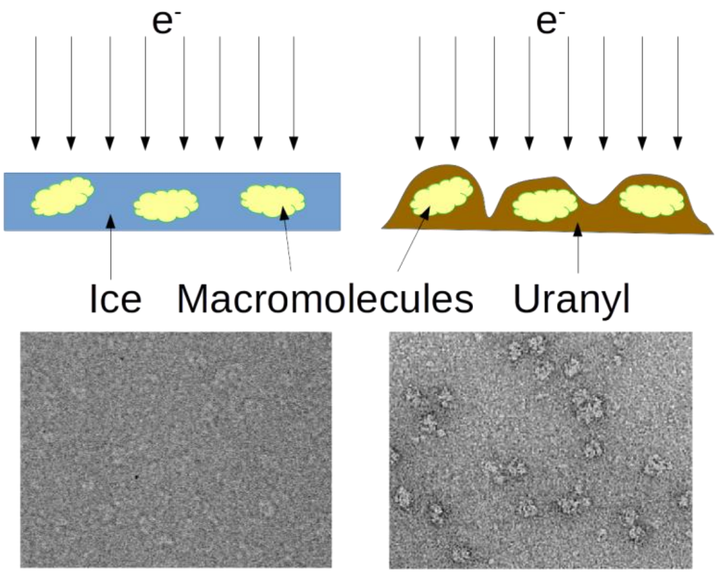


Figure 5: Schematic of the sample preparation and micrograph obtained from both vitreous preparation (left) and negative stain (right). [5]

Once the sample is properly prepared the next step to take place in the imaging processing is the image formation. Unlike the electron lenses able to treat the electron microscope from the geometrical optics, to study the image formation process it is necessary analyze this process from the deep electron-matter interaction framework. [5]

When electron-matter interaction is considered in real systems, Maxwell conditions are broken due to aberrations and the diffraction limit. Then, under these circumstances, the image of a point-object is no longer a point, and instead it will be a spot, known as point spread function (PSF). Since electrons only impact in a single position, the spatial distribution of the beam is obtained on the screen after several electrons collide with the detector. [5]

Imaging detectors or imaging sensors register the intensity pattern obtained from the electron beam after interacting with the sample. The detector design is as important as good optical system for a correct image formation. [5]

In the beginning of this technique, the detector consisted on a sensitive plate to radiation. This design was not very convenient since the information had to be extracted from the film by inspection of the detector and digitized posteriorly. Nowadays, this technique is obsolete, and it has been substituted by digital sensors. These are several different techniques for the design of a digital sensors, that have been evolved during the last 40 years. [5]

The main advantage accomplished by the digital sensors is its automatic digitalization of the detected signal, in comparison with the photographic detectors, which are an “analogic” technology that requires from a posterior digitalization of the image processing of the signal. Due to this, the automatic processing of the detected micrographs can be done at once, without requiring the substitution of the detector every time a micrograph is captured. [5]

In the early years two different technologies arise for the design of digital detectors: charge coupled device (CCD) detectors and complementary metal-oxide-semiconductor (CMOS) sensors. Since then, the improvements where focused on the improvement of the signal detection, increasing their efficiency and reducing the noise detection. After some years of research, a new generation of detectors was developed: the direct detector devices (DDD) or direct electron detectors (DED). The main advantage in this new generation of detectors was the measurement of the radiation without needing the conversion of the electron beam into light, using a phosphor layer with the only backward that it introduces some blurring in the detected image. [5]

Inside these different technologies, several families of detectors have been developed with different characteristics. For a good differentiation of the capabilities of different sensors, some characterizations are needed from these. Although there are some external factors that affects in its functioning this characterization is possible, since the resolution is not the only parameter to consider. Here are exposed some of the most relevant:

1. Picture element size: a sensor is basically a composition of a matrix of identical picture elements (or pixels). Each pixel works as a single detector of radiation, but it must be considered that not the whole pixels’ surface works as an effective radiation detector area.

2. Resolution. Coarsely, resolution can be defined as the minimum difference that can be detected by a sensor, or on the case of an image detector the highest spatial difference that can be detected. Due to this, resolution is measured as the number of different pixels that form the output image, considering the actual size of the detector. Hence, the digital detector introduces a discretization of the image, determined by the sampling rate given by its resolution. These last two concepts are clarified in Figure 6.

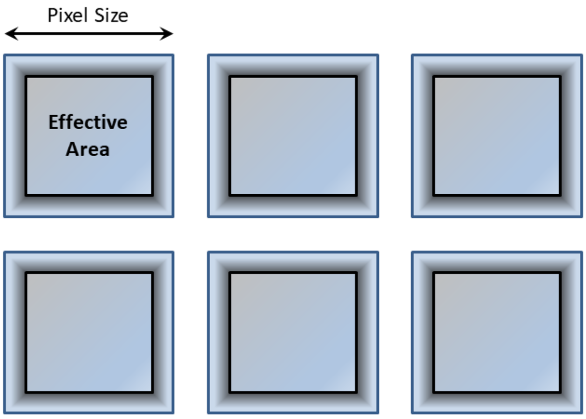


Figure 6: Schematic of a sensor with resolution 2 × 3 exposing the effective area of each pixel. [5]

3. Analog to digital converter. The radiation intensity detected by each pixel should be digitized, in order to be posteriorly processed as discrete values. For this, a specific number of bits are used to express the value, being the sensitivity or precision (minimal change that can be expressed) of the value directly proportional to the number of bits used.

4. Dynamic range. This feature is defined as the ratio of the maximum signal intensity and the root mean square of the noise, measured in decibels.

5. Gain. This feature is defined as the ratio between the recorder signal intensity and the real or theoretical signal intensity. Although a perfect sensor should present a gain equal to one, this is not feasible, appearing some signal losses in the detection process.

6. Detective quantum efficiency. This feature works as a measure of how the signal is preserved while the measurement process. Although an ideal detector should keep the signal to noise ratio (SNR), real detectors might introduce some errors in the

signal detected due to the conversion process, decreasing the quality of the output image. [5]

2.3 Fundamentals of single particle analysis

Electron microscopy technique has been used intensively in the elucidation of macromolecular complexes structures. These structures have been analyzed thanks to the posterior processing of the images obtained from the microscope, employing a set of image processing techniques known as single particle analysis (SPA). The main objective of this procedure is to obtain a 3D structure, or an atomic model if possible, of the macromolecular complex under analysis. In this section the whole workflow of the SPA is exposed, analyzing the tasks performed in each step. [5]

Before starting this exposition of the different steps that constitute the workflow, some assumptions that SPA makes to perform its analysis must be mentioned:

1. Homogeneous sample. Assumes that all, or almost all, the macromolecular complexes observed are identical but disposed in different orientations.
2. Projection assumption. The images obtained (micrographs) are considered a projection of the sample under a determine amplification of the microscope. [5]

The combined effect of both assumptions establishes a scaled identity condition for the complexes observed in the micrograph, which sets a solid base for the reconstruction process. It is necessary to mention that the homogeneous condition is weaker than the projection assumption. [5]

Before dawning into the SPA workflow, a new theoretical concept critical in the macromolecular complexes reconstruction must be exposed, the central slice theorem. Each projection of the complexes in the micrograph (particle) is analyzed, obtaining its angular orientation, forming the projection sphere for each possible orientation of the particle. From this projection sphere the central slice theorem is defined as follows:

Given a 3D structure defined by a function $S(x, y, z)$ and $P_\alpha(S)$ a projection of the structure under certain direction α, where P denotes a projection operator along, α, then,

$$D_\alpha \, [\mathcal{F}(S)] = \mathcal{F}(P_\alpha(S))$$

where D_α is the plane defined by the normal vector with direction α that passes through the origin of the Fourier space. In other words, the Fourier transform of the particle image projections fills the Fourier space in an orientation given by the angular projection of the particle. Filling the Fourier space with several projections of the same particle, the volume can be reconstructed by the inverse Fourier transform. For a better explanation Figure 7 shows a schematic with the relationships of the projections and its associated component in the Fourier space, and the reconstruction of the final volume. [5]

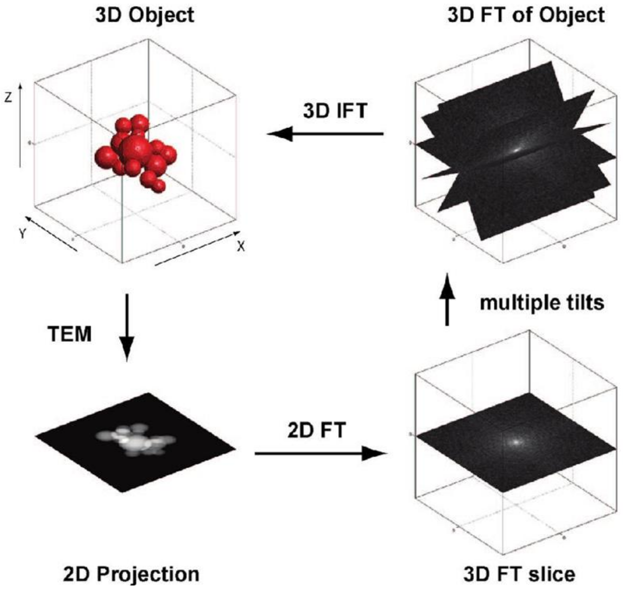


Figure 7: Demonstrative scheme of the central slice theorem. [5]

Even though the implications of the central slice theorem in the structure reconstruction of macromolecular complexes, due to the ignorance of the particle orientation in the micrograph, the reconstruction process is not straight forward. The reconstruction problem is addressed through the SPA analysis, composed of several steps some of them facing remarkable problems. A schematic including the different steps involved in the SPA is shown in Figure 8. [5]

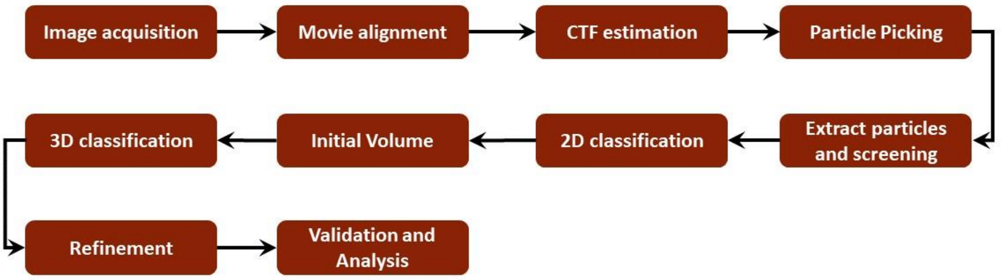


Figure 8: Schematic presenting the different steps that compose the SPA. [5]

In the first step of the workflow takes place the image acquisition process. Although this process is more related with the microscope functioning, some differentiation must be done between two different systems of image acquisition: movies and micrographs.

The structural information of the observed complexes is given by the contrast of them with the ice, being possible to increment this contract with a higher electron beam, acquiring a single micrograph. But another possible approach is to take several images with a lower electron beam intensity, acquiring a stack of movies instead. A comparison of both methodologies is shown in Figure 9. [5]

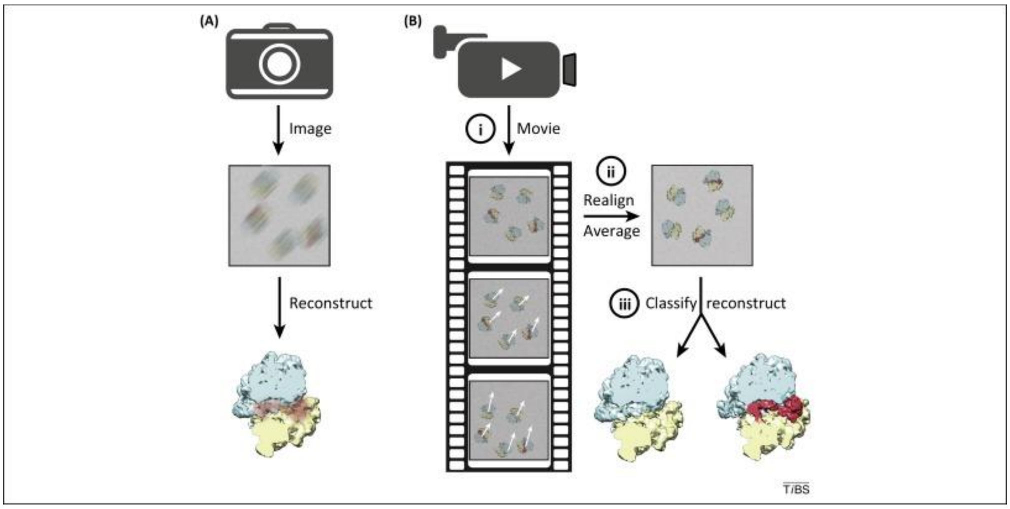


Figure 9: Schematic comparing both acquisition techniques: single-image micrograph (A) and movies recording (B). [5]

In case of obtaining a single micrograph, no post processing is needed, but in case a stack of movies is obtained, some operations must be performed. The first phenomena that must be taken under consideration is that the electron beam is going to produce some movement in the radiated particles consequence of the electron-particle interaction. This interaction is affecting its position for the next acquisition in the movie stack. Also, it must be considered that the contrast presented in each frame of the movie stack is very low, more compared with a single micrograph, presenting a very low SNR. [5]

To process the stack of movies some alignment must be performed between the different frames. This step of the workflow is known as motion correction or movie alignment. The task performed is to correct the movement of the particles form one frame to the next one in the movie stack. This process can be performed both globally, correcting the whole frame in the same direction and magnitude, or locally, allowing different corrections for each particle present in the frame. [5]

After the motion correction takes place, the obtained image present a significantly higher SNR and a higher contrast due to the combination of all the frames. Also, consider that the output of the procedure is also known as micrograph although the procedure behind it differs from the micrograph obtention procedure. [5]

Once the movie alignment process is finished (or in case the input image was a single micrograph image) the next step to take place is the contrast transfer function (CTF) estimation. As it is explained in the previous section CTF models the imaging performance of the microscope and, due to the imaging conditions changes for each acquisition, this process must be performed for each obtained image. The CTF quantifies the effect and variation of the obtained image from the ideal one, allowing the correction from aberrations and increasing the SNR of the posterior reconstruction. [5]

For the CTF estimation it is necessary to explore the Fourier transform of the acquired image. The acquired image is the convolution product of the CTF function, a sinusoidal function in the plane, and the Coulomb potential, procedure equivalent to a multiplication in the Fourier space. Once both elements have been defined, the most common approach begins with the power spectrum density (PSD) calculation (the squared modulus of the image Fourier transform). In the PSD some fringes, known as Thon rings, can be observed as in Figure 10. These fringes establish the base for the CTF estimation since the zeros present in the PSD coincide with the zeros present in the CTF, allowing the estimation of the behavior of the sinusoidal signal and hence the CTF. [5]

There exist several methods for the CTF estimation. Concerning to this project, the method used in the ctf_estimation belonging to the software package Xmipp.

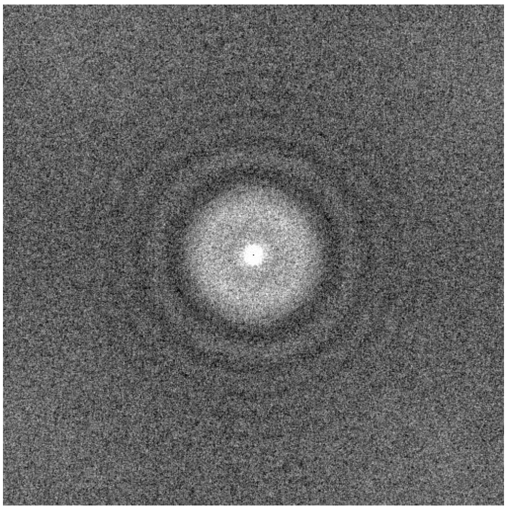


Figure 10: Calculated PSD from a micrograph, presenting clearly its Thon rings. [5]

The next step in the SPA workflow is the particle picking. In this process the particles present in the micrograph are selected, delimited and cropped, grating the proper identification of the particles and differentiating them from the ice despite noise. In this process the first raw data to perform a reconstruction is obtained, making this a dedicated step in the whole workflow. [5]

Although the element/pattern identification problem has been broadly solved in image processing, the low SNR presented in the micrographs makes this a hard problem to solve. In order to confront this problem several methods have been developed for this purpose specific for this field, based on very different solutions: template matching, edge detection, intensity comparison, texture-based methods, or neural networks, nonetheless some methods exploit several of these solutions. For a coarser classification, several categories are proposed differentiated by its usage:

1. Manual methods. This first kind of picking requires the completed identification of the particles that are going to be cropped. For this, the user must identify by eye view all the particles and select them manually. The main

drawback of this techniques is mainly the extensive amount of time required and the possible biasing of the results due to subjectivity of the user.

2. Semi-automatic methods. This approach is a combination that requires some previous interaction with the user before to start the automatic process. Although several techniques are applied in the usage of these methods, the essential functioning consist in the generation of some template from the input information of the user for a posterior identification of the particles present in the micrograph. For this identification, a template is generated from the data introduced by the used, and by means of correlation or other comparison process, elements like the template generated are founded in the micrograph. More complicated approaches have been developed in this category, allowing different templates generation, template-leaning techniques, and other techniques outside the template-matching techniques.

3. Automatic methods. This kind of algorithms are the last approach in the state-of-the-art, designed in order to avoid the biasing by human interaction. Although there exist some algorithms that automatically selects the template, most of the automatic methods make use of convolutional neural networks (CNN) based on deep learning techniques. These techniques receive a prior training to posteriorly differentiate particles from the background. [5]

Once particles are selected, they are extracted conforming a stack of particles. It must be considered that picking methods usually present false positive elements as artifacts, wrongly picked particles or even noise.

To avoid the misinformation present in the produced stack, the particles are presented sorted with some criteria, usually some similarity measure or SNR, in order to make easier some discarding of particles if needed. An example of an already picked micrograph is presented in Figure 11. [5]

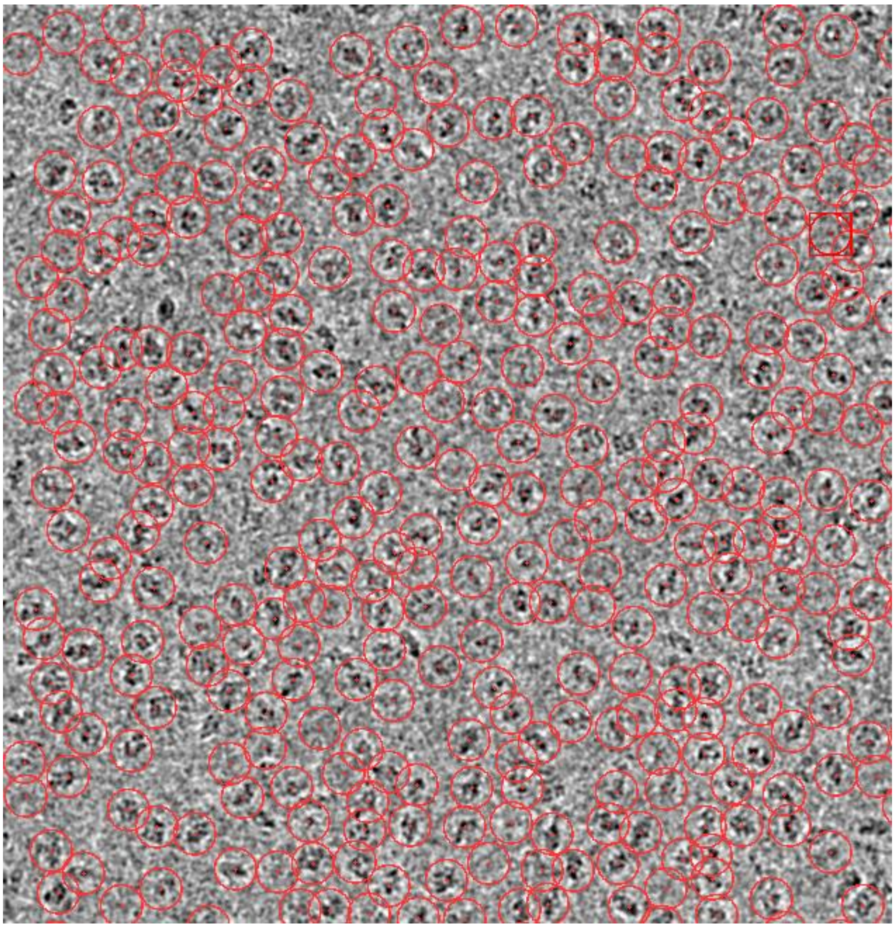


Figure 11: A beta-galactosidase micrograph with all its particles picked. [5]

Once the particles are extracted, the workflow can be continued to the 2D classification of particles. With the particle extraction concluded, the number of elements in the stack goes from several thousand to a few millions for a common SPA. Some drawbacks arose due to these magnitudes are:

1. Although the number of picked particles, the SNR of each particle is very low, which impedes a correct angular orientation estimation.
2. The exalted number of particles increases the computational effort significantly for the 3D reconstruction of the structure.
3. Apart from the homogeneous sample consideration, several artifact and other undesired structures are picked, breaking this condition and the projection assumption. Due to this some pruning of the particles should be performed. [5]

To address these three issues, the 2D classification process takes place. In this process the particles are grouped in different classes, being each of these classes defined as a subset of particles that presents similar characteristics under certain error tolerance. From each class it can be obtained a class average or class representative, result of a wise-mean averaging of the particles that compose the class. Obviously, the averaging procedure includes a previous alignment of the particles belonging to the class, being here where the wise behavior of the averaging comes from. [5]

This class averaging simplifies significantly the problem and instead dealing the elevated number of particles only a few images, the class averages, have to be considered. Furthermore, the averaging process increases the SNR of the class representatives compared with the unclassified particles, obtaining a high detailed image. Also, all the picking structures that are not corresponding to a particle class, as artifacts or empty particles, will be clustered in the same class making easier the pruning process. Finally, the classes construction solves other side problems as the existence of preferred directions, bad angular covering of the projection sphere or some other kinds of heterogeneity in the sample. [5] An example of the output of this procedure is shown in Figure 12.

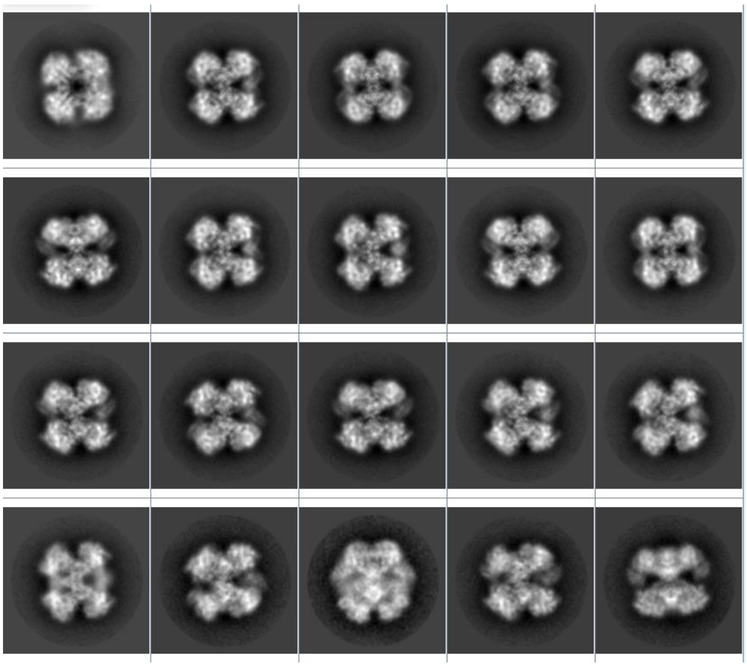


Figure 12: Set of classes obtained after the picking process. Each presented image is the class representative or average from each set of particles. [5]

For the set of classes creation there are several algorithms able to separate all the picked particles in different classes. The first algorithm is the multivariate statistical analysis (MSA), whose strategy is performing a dimensionality reduction of the data, removing the information that lacks sense for the classification and makes the process harder. In a simpler way, if an image is considered as a 3D object, this algorithm is going to focus on the “shadow” of the image, reducing the information contained in a 2D object and hence its complexity. These “shadows” are the best approximation of the original 3D object and the comparison between them is a much simpler process. Another different approach is the multi-reference classification (MRC), where an initial class representative is selected, and the rest of the particles are classified according to the similarity with them. As logical, the final classes are sensitive to the initial representatives selected, although there exist some methodologies to correct this behavior. [5]

Once the class representatives are calculated, enough information is available to start with the reconstruction of the initial volume. Due to the problems solved coming from the use of class averages, like SNR or angular assignment, it is possible to use these representatives to reconstruct a first coarse estimation of the 3D volume complex. This first volume presents a very low resolution with no high frequency features, being refined in the post processing through several iterations. Withal, a proper estimation of the initial volume is fundamental in order to obtain a posterior accurate high-resolution reconstruction. Poor or mistaken initial maps introduce some biasing in the structure that, in best case scenario, will slow the convergence to the final refined map. [5]

In this step the central slice theorem, previously introduced, is being used. There are several algorithms in the literature that address this problem. One of the approaches is based on the consideration of the classes as several projections of the macromolecular complex, reducing the angular assignment calculation as a common lines identification problem between all the projections. Unfortunately, these methods do not rise robust estimations of the common lines, mistaking some of them. [5]

Alternatively, some algorithms are based on statistical approaches for the optimization of the alignment variables. An example of these procedure is shown in figure 13. These algorithms propose a starting point base on a random angular assignment, obtaining a structureless ball, and through an iterative process of optimization, evolve the ball to a more realistic map more compatible with the classes. The main drawback of this method is that, due to its iterative nature and depending on the initial random ball, the reconstruction process can get stuck in some local minima. Other methodologies consider random reconstruction maximizing the number of projections compatibles with the reconstruction to posteriorly select the optimal one after a posterior refinement. [5]

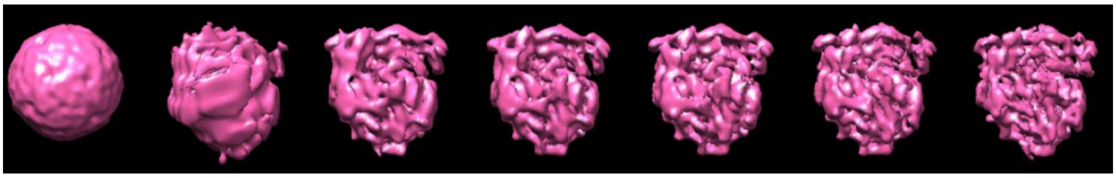


Figure 13: Evolution of the refinement reconstruction process from the ball shape to the resulting initial volume. [5]

In the uncommon case in which these methods fail in the calculation of al initial volume, an alternative solution is to use negative stain particles for the initial volume reconstruction. Due to its high contrast, this approach allows the use of other more robust reconstruction methods, as random conical tilt (RCT) or orthogonal tilt reconstruction (OTR), that should lead to a better initial volume. This new initial volume can be used for a posterior refinement employing cryo-EM data, allowing high resolution structures to be reconstructed. Although this method requires from a double sample preparation, it is obsolete given that the initial reconstruction problem has been addressed with the methods introduced before. [5]

Now, once the initial volume is constructed and a first approach to the real structure is defined, some problems arise due to the rupture of the homogeneity assumption. These problems appear since not all the particles used for reconstructions are different projections of the same macromolecular complex, presenting some heterogeneity. The heterogeneity of the sample can be classified between conformational heterogeneity: considering that macromolecular complex is not rigid and present certain flexibility, and structural heterogeneity: some proteins, despite the purification process, present non neglectable differences in their structure. [5]

Due to these two heterogeneity phenomena, if a refinement process is accomplished, it is necessary to classify all the sets of particles in groups belonging to a set of projections of the same macromolecular complex. This is the 3D classification step. [5]

Although a complete resolution of the heterogeneity problem is not feasible for all the cases, there exist many methods able to solve it under certain conditions. One solution is the definition of a phase space in which the different possible conformations of the macromolecular complex are considered. Alternatively, several statistical approaches can be considered, as maximum likelihood, Bayesian marginalization algorithms, or principal component analysis. [5]

Once the 3D classification is performed and it is possible to guarantee that the set of particles projections for reconstruction is homogeneous, the elucidation of a high-resolution map can be started, known as refinement. Under this process, the refinement starts from an initial volume with no high resolution resolved structures and, through an iteration process, the converge in a detailed 3D structure that presents high frequency features is reached. Through the refinement process, a continuous estimation of the wellness of the reconstructed model is performed, comparing the difference or distance between the images and the projections of the constructed structure. To perform this comparison several strategies can be followed: maximum likelihood, maximum *a posteriori*, or projection comparison. [5]

The maximum likelihood approach allows to reduce this task to an optimization process where each experimental image obtained is considered as a projection of the reconstructed structure from many directions, assigning a different probability to each of them. Then, the aim is to collapse these probabilities reducing the number of possible directions through an iterative process. On the other hand, maximum a posteriori method penalize some possible orientations considered in the previous method by introducing previous information of the structure before going through the iterative process. Finally, projection matching splits the maximum likelihood approach in two different tasks simpler to solve: firstly, the particles get assigned angles considering the initial map and, secondly, a new structure is reconstructed considering these angles, measuring its viability compared with the experimental images. [5] In Figure 14 the result of a refinement process is shown.

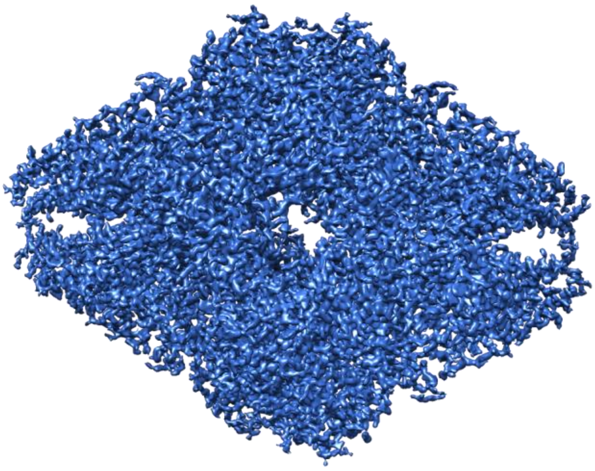


Figure 14: High resolution structure obtained after the completion of the refinement process. [5]

Although the reconstruction procedure is finished, and the main goal of the SPA is achieved, some final procedures must be done according to the validations and analysis of the final structure obtained. The validation is an even more necessary process in SPA since the very noisy images accompanied by the possible biasing coming from the user decision can affect the final reconstruction, even more if it is considered that the algorithms could elucidate wrong structures even if the user only takes correct decisions. [5]

To this purpose quantitative methods are going to be used in other to check the wellness of the reconstruction. A first approach consists in the comparison of the reconstructed structure with the results obtained by other techniques as x-ray crystallography or nuclear magnetic resonance (NMR), although this suppose an increment in the facilities and budget needed. An easier approach consists in the compotation with similar structures already resolved. Other approach would be based in the compatibility of the reconstructed structure with the particles it comes from. One of these methods is based on the geometrical constraint introduced by the tilt angles, aligning the tilt pairs with the reconstructed map to determine the robustness of the reconstruction. [5]

Particularly, many reconstructed structures suffer from overfitting, so some method is needed to its detection. To do this a subset of the particles is substituted by random noisy particles and the new reconstruction is compared with the original one with the use of the Fourier shell correlation (FSC). The FSC is a quality tool able to detect the overfitting between both reconstructions. The last group of methods are based on the validation of the alienability, considering each particle should be a projection of the reconstructed map. Then, the angular assignment of the particles should cast the most probable directions of each particles close between them, in contrast to the noisy image. [5]

Once the map has been properly validated, some measure of the quality must be done, expressing the spatial reliability of the reconstruction. Resolution, as me minimal observable change, arises as the most reasonable option although there is no consensus for a universal definition of this concept. Due to the high noise present in cryo-EM data, resolution is oriented as measure to distinguish the signal and noise at different frequencies. Several metrics arises as plausible options for this measurement: FSC, differential phase residual (DPR), or the spectral signal to noise ratio (SSNR). Currently the most spread metric is the FSC, defined in the Fourier space as a measure of the cross correlation of two maps at different frequencies, although in this case the FSC measures the self-consistency of the reconstructed map rather than the quality. [5]

3 MATERIALS AND METHODS

After the exposition of the physical basis of electron microscopy and the computational post processing of the images obtained, the new developed contributions are introduced. For this purpose, this chapter is divided in two different sections. Firstly, including all the corrections performed in the CTF estimation protocol, and secondly, including the development of the deep neural network able to calculate the defocus of a micrograph.

3.1 CTF estimation corrections

In this first section the main corrections performed in the CTF estimation algorithm from the Xmipp package are presented. Before examining these corrections, it is mandatory to introduce the estimation algorithm and its functioning. It is important to clarify that in this first part of the developed project no new workflow design is developed.

The design of this algorithm is detailed in [3], hence only a general overview of its function is explained. For this, a special attention is paid in the modules in which corrections has been introduced, in order to achieve a better understanding of them. As it has been introduced in the previous section, the estimation of the CTF requires a previous step for the calculation of the PSD, form which some features will be extracted. Once the algorithm estimates the PSD the feature extraction starts, which is focused in the detection of its sinusoidal behavior.

The disposition of the PSD, several concentric fringes, encourage its characterization by the calculation of its radial average, reducing its dimensionality and hence the computational complexity of the posterior calculations.

Also, it is relevant to be aware that if the concentric circles are deformed to an ellipsoidal shape, the radial averaging of the signal would be spoilt. This ellipsoidal shape in the PSD fringes implies a defocus variation depending on the angular direction, leading to an astigmatic image. Anyway, these astigmatic images, although it is still possible to obtain a correct estimation of their CTF, should be discarded.

Once the radial average of the PSD is calculated, the lower and upper bound are set in order to capture the oscillatory sinusoidal behavior that this signal presents. These two boundaries confine the signal and allow the calculation of the parameters that characterize it.

In Figure 15 an image of the PSD signal with its upper and lower boundaries is shown.

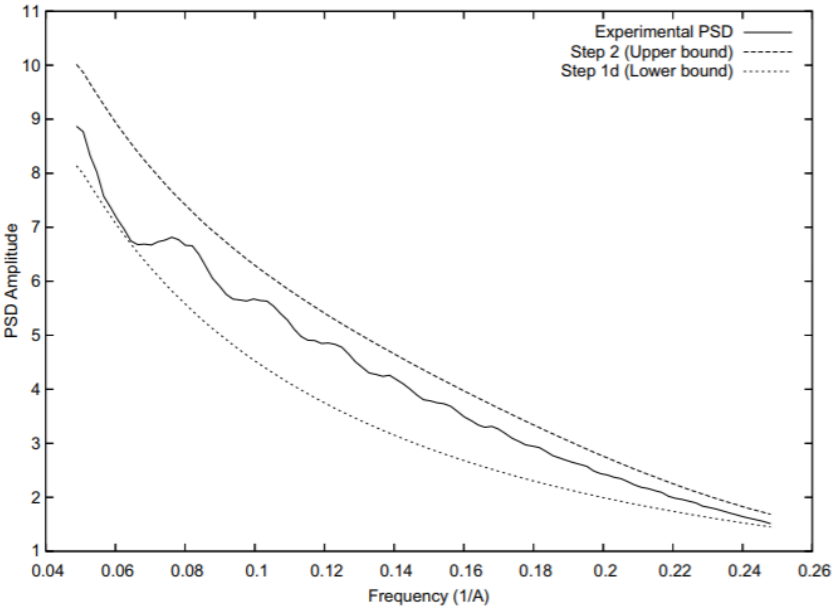


Figure 15: Radial average of the experimental PSD and the theoretical PSD lower and upper bounds. [3]

Once the radial average is confined into the boundaries, it is possible to start the calculation and posterior refinement of the characterization parameters to finally obtain a robust estimation of the CTF.

In Figure 16 an image of the result obtained after the correct parametrization of the CTF is displayed compared with the PSD.

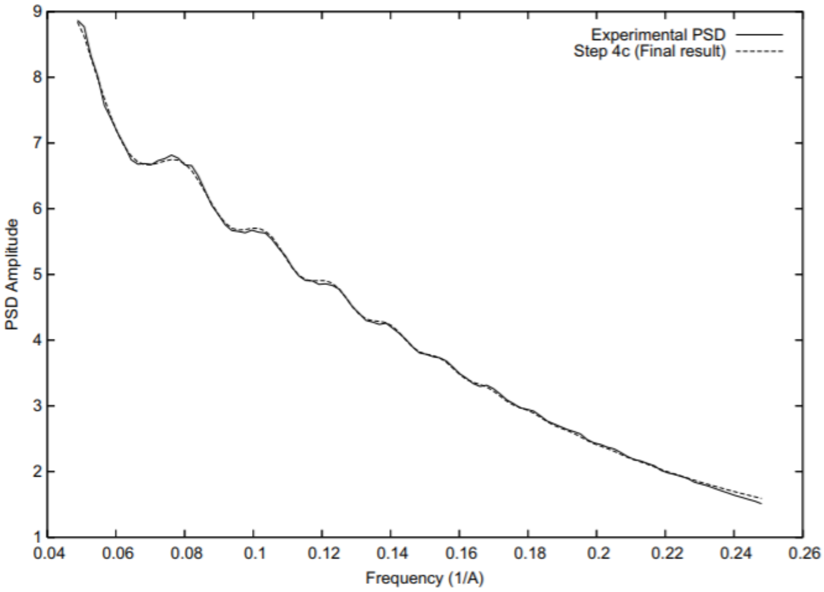


Figure 16: Radial average of the experimental PSD and the theoretical PSD. [3]

From these methods and other studies of the literature as [9], it is possible to observe the relevance of the defocus calculation to a proper estimation of the CTF. The defocus of the micrograph is characterized by four different parameters, necessary for a proper characterization in this dimensionality. But nevertheless, once the software solution of the protocol is implemented, in the first step of the defocus calculation, its characterization is performed by a single number, given by the following formula:

$$Defocus = 2 \cdot K_0 \frac{(2 \cdot T_S)^2}{\lambda}$$

where K_0 is defined as the index (or position) of the maximum of the Fourier transform of the radial average function respect to the square of the frequency, T_S is defined as the sampling rate of the micrograph, and λ is defined as the electron wavelength.

The first corrections implemented are applied over this function and its maximum value calculation. As it can be seen from the defocus calculation formula, its value only depends on three parameters, from which only the K_0, maximum index of the Fourier transform of the radial average function respect to the square of the frequency, requires to be calculated. The other two parameters, the sampling rate and electron wavelength, are given from the acquisition procedure of the micrograph. Due to this, the correct calculation of the maximum value of the PSD radial average is fundamental for the correct calculation of the defocus.

Until now the problematic is very clear and, although calculating the maximum of a signal could present some complexity in some cases, the signal is completely defined before starting the calculations and it is not wide extensive. Then, the problematic is not the actual calculation of the maximum, but the behavior that the PSD radial average present itself.

Although the performed averaging of all the points of the PSD that are at a same distance to the center, there are some behaviors that can introduce some artifacts in the resulting signal. These artifacts can lead to a spoilage in estimation of the maximum of the signal and hence in the defocus estimation. In general, these artifacts have no implication in the maximum on the function but in case they have, the defocus cannot be properly calculated.

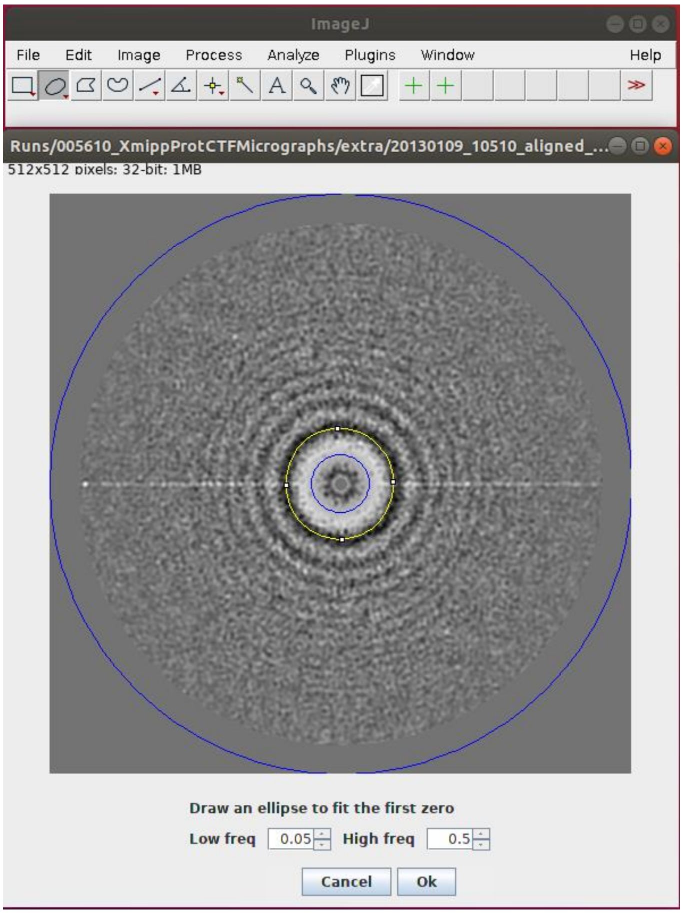
It is important to be aware that a failure in the estimation of the CTF suppose the discarding of the micrograph from the whole workflow, removing valuable information in case the quality of the image was acceptable. To prevent this to happen, some corrections have been performed in the algorithm in order to calculate correctly the maximum of the function.

Before applying any correction and due to none of the signals of interest are part of the output of the protocol, some debug code has been added. This debug code allows a posterior plotting of the three signals of interest: the background radial average signal of the PSD (psdbackground), an interpolation to obtain the PSD respect to the squared frequency (psd_exp_radial2), and the Fourier transform of the previous signal (amplitud).

Apart from the signal, some other useful information were included as an output of the debugging code as the minimum index form which the maximum of the signal is searched, every new maximum index observed in the signal, the final index obtained after the maximum index calculation and the calculated defocus. With all this new debug information, trying to diagnose any malfunctioning of the algorithm is possible.

Once the different signals could be displayed, interesting information were obtained from bad estimated CTFs. The first behavior that was checked is the correct calculation of the maximum value in the PSD radial average signal. Then, it became clear that the calculated defocus at the index at which the maximum value of the signal was presented was not corresponding with the actual defocus presented in the micrograph.

This checking is possible thanks to a CTF estimation tool integrated in the Scipion framework form the software package Xmipp. This tool allows the manual setting of the first fringe of the PSD, allowing the correct calculation of the defocus presented by this specific micrograph. Although this tool it is not useful for high throughput processing, it has supposed a critical aid for the proper understanding of the behavior of the algorithm and to apply the corrections needed. The interface of this tool is shown in Figure 17.


ImageJ
File
Edit
Image
Process
Analyze
Plugins
Window
Help
Runs/005610_XmippProtCTFMicrographs/extra/20130109_10510_aligned_...
512x512 pixels: 32-bit: 1MB

Draw an ellipse to fit the first zero

Low freq 0.05 High freq 0.5

Figure 17: Xmipp interface for manual estimation of the CTF. The inner and outer blue circles set the range in which the algorithm looks for a maximum and the yellow circle should indicate the first zero in the PSD.

With the usage of this tool, new ground truth information can be obtained and, after some comparisons of the behavior of the algorithm with its ideal one, the first corrections were applied. The first approach for a better estimation of the CTF was to apply some corrections in the Fourier transform of the radial average function respect to the square of the frequency signal, in order to improve the maximum detection.

From the ground truth information, it is possible to check that, the index at which the maximum is located in the signal do not correspond with the index from which the correct defocus is calculated for some micrographs. In order to deal with the irregularities that are causing this behavior, some preprocessing of the signal is performed before the maximum index calculation.

The first approach was to apply some outliers treatment of the sample, in order to avoid some noise or other phenomena that were introducing atypical values in the signal that could affect its radial average profile. The processing performed was very standard. Every value that present a difference with the mean of the signal higher than three times the standard deviation is detected. In order to not introduce artificial high frequency behavior in the signal, the values detected were cropped to three times the standard deviation plus the mean (the maximum allowed).

This outlier correction is accompanied with a subtle correction in the maximum calculation. Instead of following the straightforward process of searching for the maximum value of the signal and take its index, some conditions were applied to a better detection of the local maximums. These conditions will assure that the maximums detected are placed in a peak of the function and no at the top of a decaying behavior.

The combination of both corrections, the removal of outliers and the local maximum searching, shows an especially robust behavior against profiles that present high energy at low frequencies. In these cases, the maximum peaks obtained in the low frequencies range are corrected allowing the detection of the local maxima in the profile in some cases.

This procedure has shown a significant improvement in the defocus calculation and therefore in the CTF estimation. In the figure 18 it is shown the improvement obtained by this correction. In the example presented it can be shown the profile from which the maximum index is calculated, and the estimation obtained before and after applying the outliers corrections to the protocol.

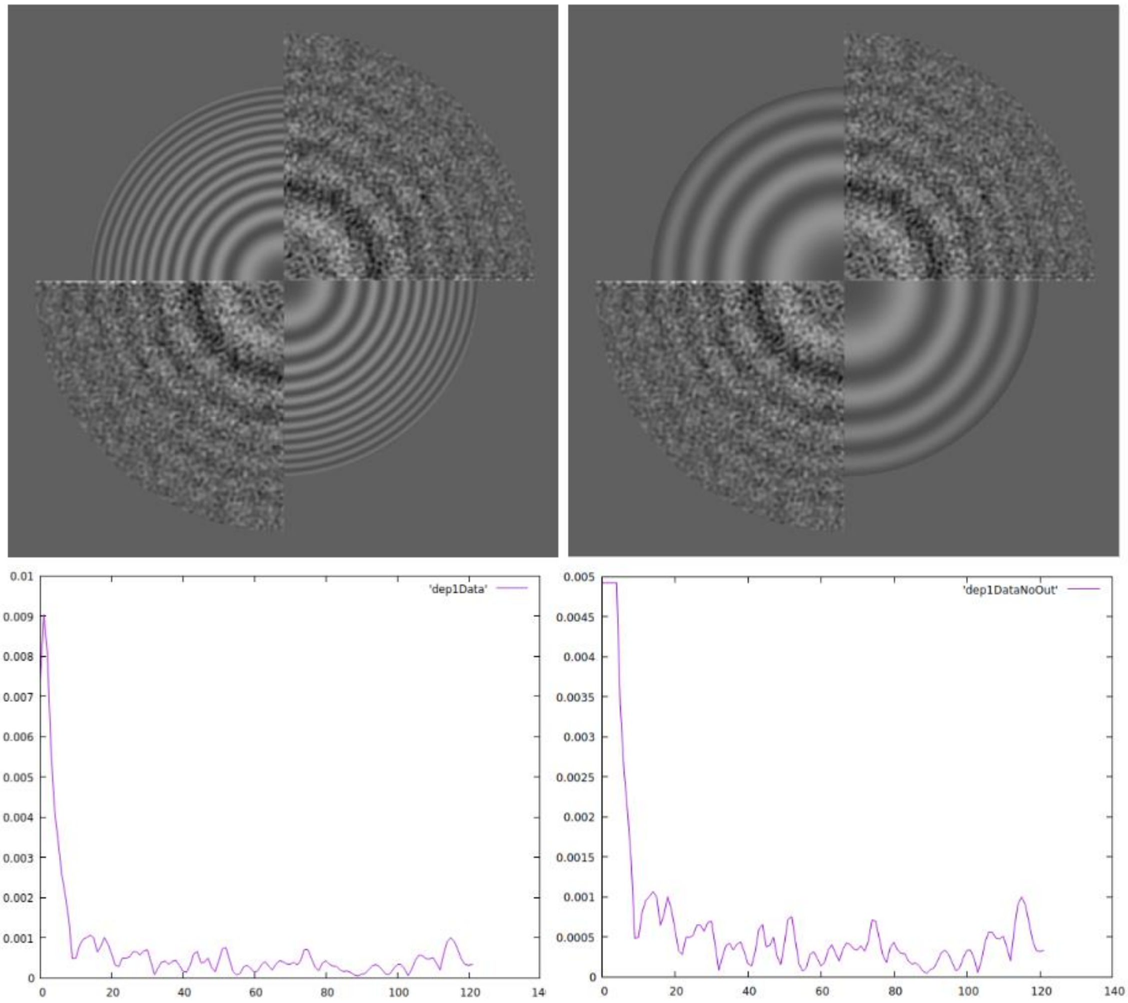


Figure 18: PSD and CTF estimation in quadrant disposition without the application of the outliers correction (top left), PSD and CTF estimation in quadrant disposition with the application of the outliers correction (top right), Fourier transform of the interpolation of the PSD average without the application of the outliers correction (bottom left), and Fourier transform of the interpolation of the PSD average with the application of the outliers correction (bottom right).

Once this first correction was applied with success the next correction was performed, this time in the PSD background radial average signal. As in the previous correction, the intention was to limit the disruptive information in the PSD radial average that could introduce artifacts that difficult the detection of its maximum value. With this purpose, a high pass filter was implemented in the algorithm in order to remove the high frequencies of the background radial average signal.

The filter used for this purpose was an implementation of one of the filters implemented in the libraries of the MATLAB software. Unlike in the previous correction, the results arose by this approach was not successful, spoiling the results obtained by the estimator. Due to this, the correction finally was not implemented. In figure 19 the results obtained by this correction are show. In the schematic presented the PSD background radial average and the estimation associates to it with and without the use of the high pass filter correction can be observed.

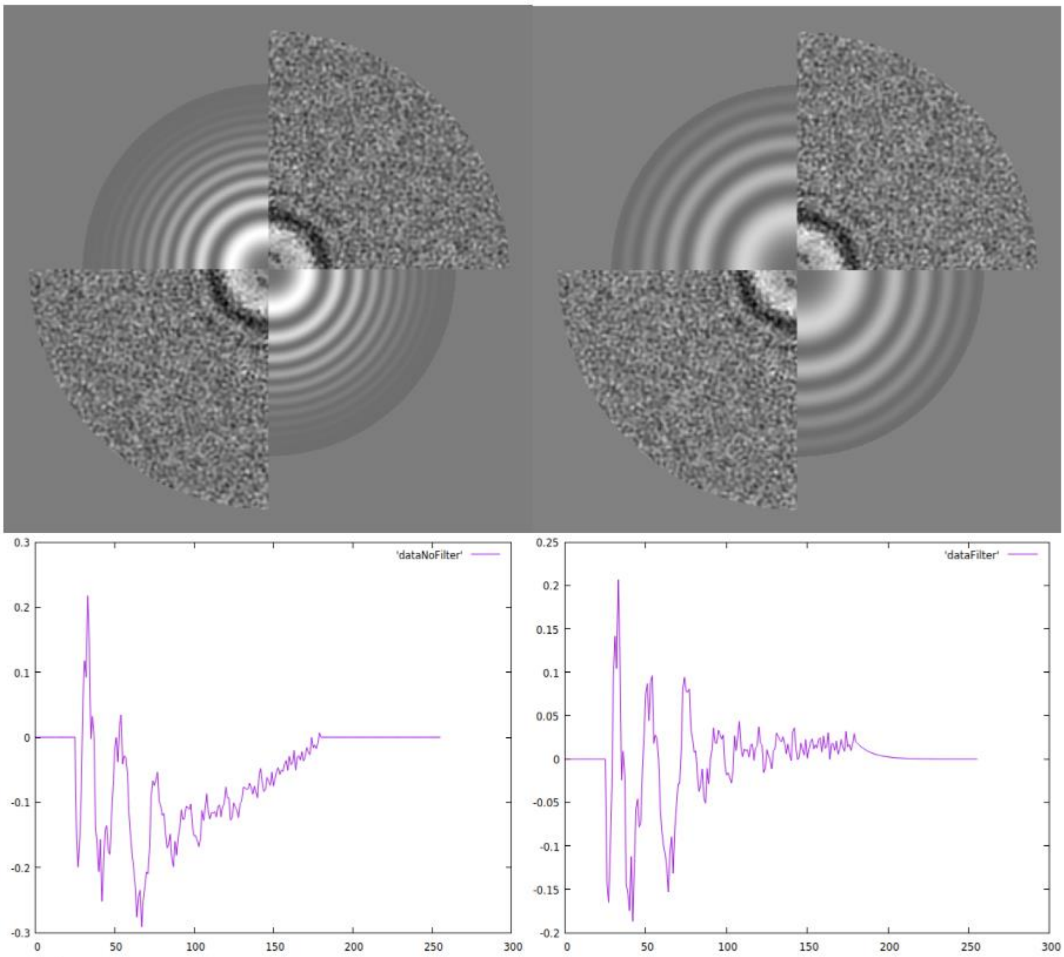


Figure 19: PSD and CTF estimation in quadrant disposition without the application of a high-pass filter (top left), PSD and CTF estimation in quadrant disposition with the application of a high-pass filter (top right), PSD background radial average without the application of a high pass filter (bottom left) and PSD background radial average with the application of a high pass filter (bottom right).

After these two corrections were applied the results obtained have improved although the rejection rate of good micrograph was still high. In other to continue with the correction process, new changes in the algorithm were performed although not related with the previous part of the procedure. The next approach to a better implementation of the algorithm was the correction of the downsampling procedure of the micrographs.

Before analyzing the downsampling correction, it must be understood the implications that this procedure has in the PSD. A reduction in the sampling rate suppose that the maximum frequencies that can be sampled are also reduced, given by the Nyquist-Shannon theorem:

$$f_s \geq 2 \cdot f_{max}$$

Also, performing a downsampling of the micrograph is going to have some visual implications in its PSD. A reduction in the sampling rate suppose and expansion of the signal in the Fourier space, losing high-frequency information. Graphically, bigger rings will be shown when the sampling rate decreases. In figure 20, it can be seen the PSD of the same micrograph under three different sampling rates.

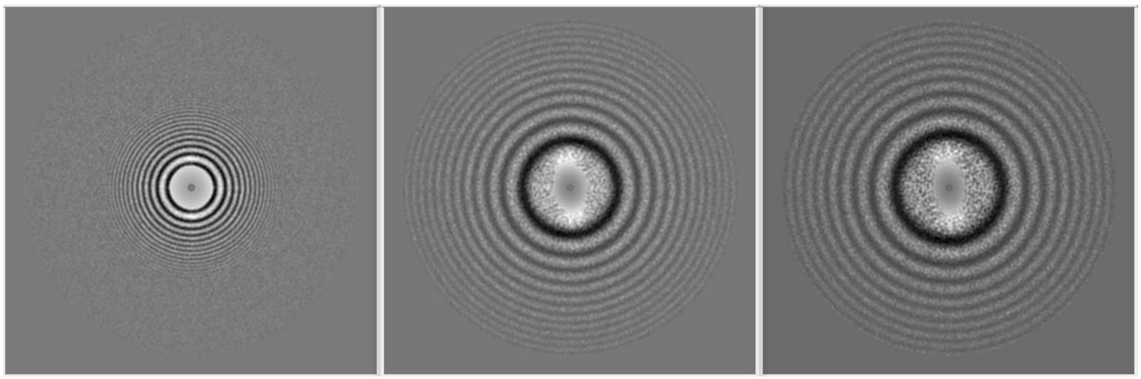


Figure 20: PSD of the same micrograph under three different sampling rates: 1 Å/px (left), 2 Å /px (center), and 3 Å /px (right).

Although, a decrement in the sampling rate have undesired effects in several applications, it is a useful tool for the CTF estimation and, specially, for the maximum index detection in the defocus calculation process. The reason for this is that, depending on the micrograph, a different disposition of the fringes can lead to a better detection of the maxima in the Fourier transform of the radial average function respect to the square of the frequency. Whether the fringes are very concentrated in the center or very expanded and only a few of them are shown, obtaining a correct estimation is harder. Due to this, several possible downsampling factors can be used until the estimation is correctly performed. This will increment the number of CTF properly estimated, although also the computational cost.

To implement this solution each micrograph is processed several times. For this, if the CTF is not correctly estimated, a new downsampling factor is applied and the process is repeated. This iterative process will be repeated until all the considered downsampling factors stored in a downsampling factor list are tried or the CTF is properly estimated. Now, the difficulty comes from the confection of the downsampling factor list.

First, it is important to mention that the target sampling is calculated considering the actual sampling rate of the processed micrograph, so there is no need of any specification process depending on sampling rate presented by the micrograph. Different approaches have been tried in the confection of this list. The original approach was that, given an initial downsampling factor, in case the CTF estimation fails, the original downsampling factor plus and minus one would be added to the downsampling factor list and they will be posteriorly tried for the CTF estimation. A posterior approach set a couple of fixed values, but this still presented improvable results.

Due to this approach did not arise the best results, the solution was to fix some new values that from experience arise better results. For this, two different downsampling are applied in order to obtain target sampling rates of 1.75 Å/px and 2.75 Å/px, to posteriorly repeat the estimation with the original micrograph with no downsampling applied (or applying a downsampling factor of 1). These values arose an improvement in the obtained results, being more robust against atypical sampling rates and simplifying the posterior analysis and correction.

Once the correction concerning the downsampling procedure of the micrographs is completed there is a last source of errors that must be corrected, the criteria for the goodness of an estimation. These criteria are composed by several features of the PSD and the CTF estimation that must be fulfilled in order to classify the estimation performed by the algorithm as a good estimation. These are the features that compose the criteria:

- Critical iceness (ctfCritIceness): This attribute gives a measure of the amount of ice that is present in the micrograph. An excessive presence of ice could imply the discarding of the micrograph. In detail, this attribute calculates the ratio between the PSD downsampled to 3 Å/px and 4 Å/px.

The vitreous ice can be detected as a thicker whitish fringe in the PSD. In case the ice present in the micrograph is crystallin, brilliant white conjugated spots appear in the PSD. Although before any presence of ice the micrograph should be discarded, crystallin ice is always harder to detect. In figure 21 is it possible to observe two micrographs both presenting these two kinds of ice.

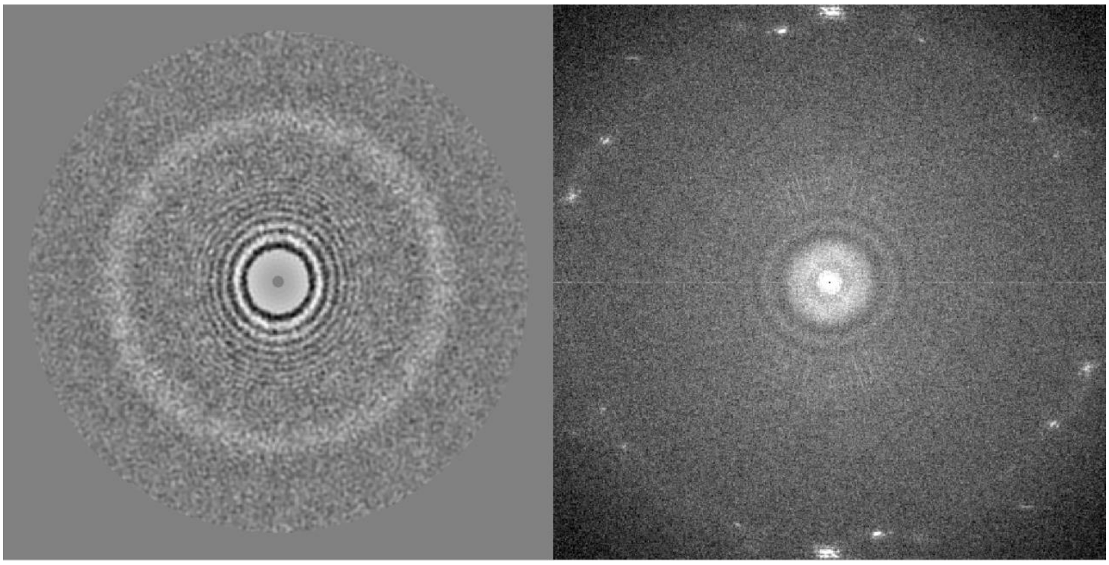


Figure 21: PSD of two different micrographs presenting vitreous ice (left) and crystalline ice (right).

- Correlation of the PSD at 90º (ctfPsdCorr90). This attribute measures the correlation (or similarity) between the PSD and itself rotated 90º. This is a strong measure against corrupted PSD, since it will discard both astigmatic PSD, where the defocus varies depending on the direction and fringes presents astigmatic behaviors, and very noisy PSD, where the fringes of the PSD are not well defined.

- Position of the first cero (ctfFirstZero). This attribute measures the radius of the first fringe presented by the PSD. If the radius of the first zero is very big the estimation will be discarded.
- Maximum critical frequency (ctfCritMaxFreq). This attribute gives a measure of the frequency at which the envelope signal of the PSD decays 100 times respect to the maximum value.
- Directional ratio of the first zero radius (ctfCritfirstZeroRatio). This attribute expresses the ratio between the radius of the first zero observed in two perpendicular directions, in order to detect astigmatic behaviors.
- First minimum and first zero ratio (ctfCritFirstMinFirstZeroRatio). This attribute expresses the ratio between the first minimum observed and the first zero.
- First and third zero correlation (ctfCritCorr13). This attribute measures the correlation between the radial profile of first and the third zero.
- Critical CTF margin (ctfCritCtfMargin). This attribute calculates the ratio between the variance of the radial profile of the first zero and the second maximum, giving a measure of how the second maximum is observed compared with the noise present in the zero.
- Validation of a non-astigmatic estimated CTF profiles (ctfCritNonAstigmaticValidity). These attributes compare the sine profiles in two perpendicular directions of the PSD and calculates the frequency at which they are in opposite phase.
- Characterization of the background PSD (ctfBgGaussianSigmaU). This attribute gives a measure of the decaying behavior characteristic of the PSD background, indicating gaussian width in one of the axes.

From the previous list it can be observed that, although all them appear as a homogeneous group, there are two different types of criteria: those applied to the PSD and those applied to the CTF estimation performed by the protocol.

From all these criteria it is possible to characterize properly the results obtained by the algorithm. Through a procedure of trial and error the boundaries values given to each attribute has been checked and slightly corrected for some of them. These corrections will improve the selection and discarding of the calculated estimations. In Table 1 the previous values for all the criteria and the new set ones are presented.

Table 1 Criteria values comparison before and after the corrections were applied.

Criteria	Before	After
ctfCritIceness	c>1	c>1.03
ctfPsdCorr90	Not applied	c<0.1
ctfFirstZero	c<5	c<5
ctfCritMaxFreq	c>20	c<0.9 U c>20
ctfCritfirstZeroRatio	c>1.1	c>1.1
ctfCritFirstMinFirstZeroRatio	c>10	c>10
ctfCritCorr13	c<0	c<0.27
ctfCritCtfMargin	c<0	c<1
ctfCritNonAstigmaticValidity	c<0 U c>25	c<0 U c>6.5
ctfBgGaussianSigmaU	c<1000	Not applied

From these two different criteria groups, one of the approaches performed in order to increase the efficiency of the protocol was to apply only once the PSD calculation criteria. Under this approach, if the PSD did not fulfill these criteria then the processed micrograph would be immediately discarded.

It is important to understand that a bad established criterion could spoil a good estimation algorithm. The establishment of the values for each criterion is a delicate process. Too restrictive values could lead to the loss of good micrographs and, on the other hand, if the values are not restrictive enough, micrographs that do not present enough quality could be included in the processing workflow.

Before exposing the next correction, it is important to understand the importance of these selection procedure and its relationship with the rest of the corrections performed and the functioning of the protocol.

As it has been exposed previously, once the defocus calculation procedure is fixed, then several estimations over the same micrograph will be performed at different resolution through the different downsampling factors. But this downsampling factor list depends also in the correct adjustment of the selection criteria. If the CTF estimation is incorrect, it must be detected by the criteria in order to discard it and, after the application of the next downsampling factor of the list, the CTF estimation may be properly detected.

Following the previous exposition, a robust defocus calculation algorithm accompanied by a coherent downsampling factor list could be useless under an incorrect set of selection criteria.

The next correction performed in the protocol was related with the manual downsampling procedure. If this option is selected, no downsampling factor list is generated and instead only one downsampling factor is tried, introduced by the user. Apart from this difference, the rest of the protocol behaves exactly as explained before.

The manual setting of the downsampling factor could arise some issues if the value introduced is smaller than the unit. Although this value would not make sense in principle, it may have it for some specific applications. In case this happens, the previous procedure was performing the sampling transformation in the Fourier space. This implies that, in case of downsampling factors smaller than one, the transformation of the signal will introduce some new “generated” or “invented” information that was not contained in the original image, introducing some artifacts.

In case the described procedure is followed some meaningless results are obtained. To solve this problem a different transformation is performed, avoiding the Fourier space. For this new transformation, a resize of the image is performed in real space until the desired resolution is obtained.

A comparison between the different results obtained after and before the correction are shown in the Figure 22. Be aware that neither of the two PSD profiles shown are suitable for performing a CTF estimation.

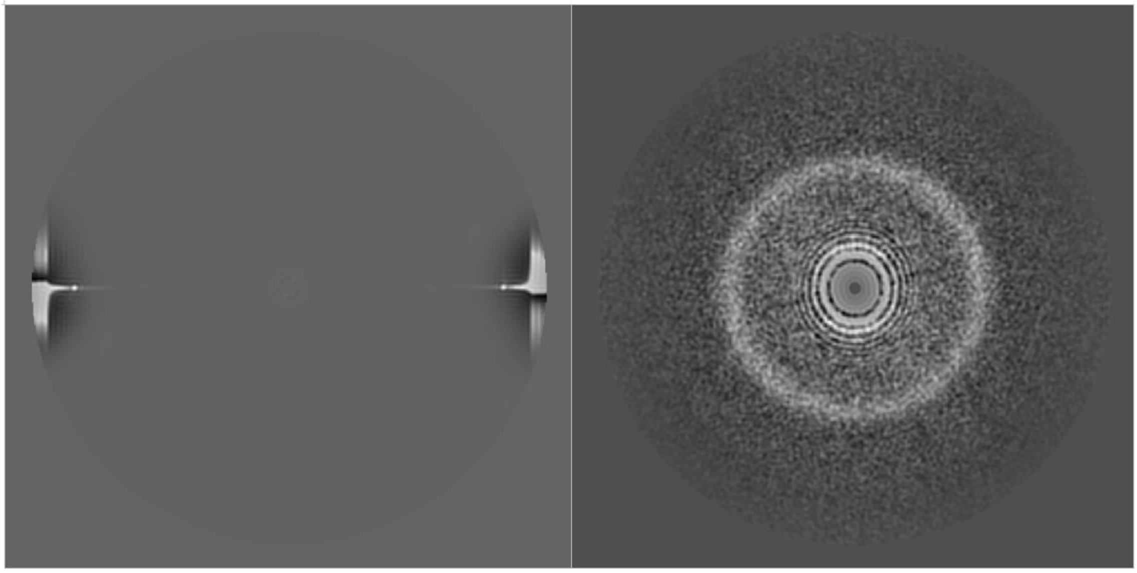


Figure 22: Comparison of the results obtained before (left) and after (right) the downsampling correction for downsampling factors smaller than 1 was applied.

The last correction performed it was not over a strict issue on imaging processing but in how the data is treated. A it has been exposed before, this CTF estimation protocol is integrated in Xmipp software package which, at the same time, is integrated in the Scipion framework as a plugin. Due to both software have their own metadata files for the data structures they work with, parsing processes are needed in order to work with the same data in both sides.

Some of these parsing processes where not correctly performed and for some of wrong estimated CTFs, the metadata associated to these estimations were mistreated. Once this problem was fixed and the parsing process of the metadata belonging to the conflictive images were properly performed, this issue did not appear anymore.

Finally, although this is not exactly a correction, a new functionality has been added, the amplitude contrast refinement. This new implementation appears as a new option for the user that allows him to apply this new procedure during the execution of the protocol. The amplitude contrast refinement process consists in the correction of the CTF through a change in the amplitude ponderation between the different frequencies that form the image.

In detail, under an ideal assumption, the whole image should be transmitted and properly detected. In this case the amplitude of the sinusoidal behavior of the CTF would be equal to one for all the frequencies. Under this assumption all the frequencies that compose the image should be transmitted without suffering any losses.

Outside the ideal case scenario, the behavior observed differs from the previous exposition. In a real case scenario, there is always some losses in the transmission, being the amplitude of the CTF smaller than one. Now, the behavior why amplitude contrast refinement is needed comes from the fact that this amplitude factor works as a function of the frequency. Due to this, the amplitude factor is unevenly distributed between the frequencies that compose the image, increasing the losses for high frequencies. To correct this behavior, the amplitude contrast refinement process evens the amplitude factor between the different frequencies that form the image, trying to avoid decompensations in its spectral domain.

This correction did not present any improvement in the obtained results from all the datasets in which it has been tested. Despite this, the functionality is maintained in the protocol because of two different reasons. Firstly, because under some conditions that were not present in the test datasets it could improve the obtained results and, secondly, because since it is an optional behavior, there is no reason for its application if it is not desired.

The impact of these corrections in the results obtained from the protocol are analyzed in the next chapter.

3.2 Deep learning network for defocus estimation

In this section of the material and methods chapter, the second developed part of the project is introduced. The project task performed consists in the development of a deep learning network able to predict the defocus of a micrograph from its PSD. Before presenting the details of the developed system, few theoretical concepts are introduced to help with the explanation.

Deep neural networks are a specific kind of artificial neural networks, characterized by an intensive number of different layers in its structure. This extensive architecture allows the development of a very robust design able to model very complex dynamic systems. Artificial neural networks are a new paradigm in computational systems inspired in the knowledge obtained from neuroscience. The main idea that these systems follow is to replicate the behavior of the human brain, creating an abstraction for a neuron and, by the interconnection of several of them, form a network. A schematic showing the behavior of a single neuron is shown in Figure 23.

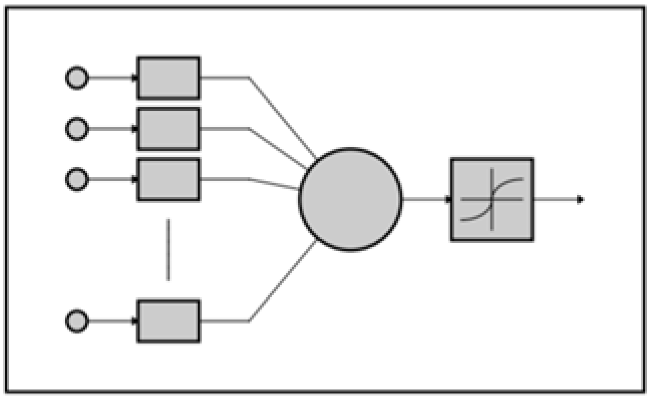


Figure 23: Schematic of a single neuron from a deep learning network. From the figure it can be identified the inputs of the neuron, the neuron itself, and the response generated from these inputs. [10]

Once the concept of the neuron is defined the next step is the wise interconnection of several of these elements forming a network. Although any disposition of the network is possible, the most typical approaches are disposed in several interconnected layers.

The only difference between a neutral network and a deep learning network is in the number of layers that compose the network, although there is not an established limit to consider a network from one group or the other.

A schematic of a simple neural network compared to a deep neural network is shown in Figure 24.

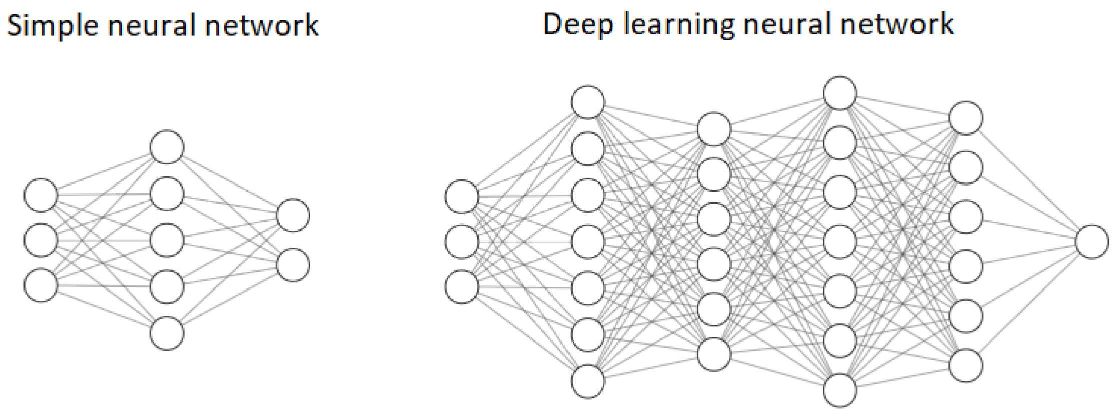


Figure 24: Schematic of a comparison between a simple neural network (left) and a deep learning neural network (right)

Once the theoretical concepts are clarified there are several tasks that must be solved in order to develop a functional deep learning network.

The first task to solve in order to develop any machine learning solution is the confection of a robust and intensive enough training database in other to have a high throughput source of information able to make the network learn. For this purpose, several sets of micrographs have been processed in order to accumulate enough information.

In order to meet the criteria for a good design of a deep learning database some features were taking in consideration. First, the source of information is going to be limited to a specific kind of electron microscope working at a specific voltage acceleration, precisely at 300kV. The reason for this, as it has been explained in the previous sections, is that the defocus is related with the acceleration voltage of the electron and its wavelength. The other limitation introduced and in the aim of looking for some homogeneity on the data was on the resolution of the training images. For this, a manual sampling was introduce depending on the sampling rate of the micrographs in order to assure that all the output PSD was sampled at the same frequency.

Once the basic limitations of the data collected for the database were established, the next step is to find enough different resources of information in order to assemble enough micrographs to train the network. For this purpose, several datasets were analyzed constituting an input of more than 10,000 micrographs.

Now the constraints for the database were established and the source of information was available, it is possible to start the processing process. In order to build a more robust training database some extra steps are going to be taked.

In principle only a PSD image vector with its associated resulting defocus is needed but, for a more robust implementation, three different PSD are going to be calculated at three different sampling rates: 1 Å/px, 2 Å/px, and 3 Å/px. The only problematic behavior of this solution is from where to take calculated defocus from the three different resolutions.

First, only the images from which the defocus is correctly calculated can be taken which, in spite of the corrections performed and presented in the previous section, decreases significantly the number of available PSD. The positive part of this condition is that, from the three different calculations of each PSD at each resolution, only one correct estimation is needed since the other two associated images are used to reinforce the learning. Then, only the PSD is needed, not a correct estimation of the CTF. This increments significantly the amount of data available to train the network.

Once every set of micrographs is processed the sets of three PSD at the three different sampling rates are obtained. A script in Python programming language has been written in order to prepare the database of PSD and its associated defocus. As mentioned in the previous section, different sampling rates can improve or deteriorate the results of the CTF estimation algorithm. Due to this, the number of good estimated CTFs will not match between the three different resolution groups.

There are different ways to address this problem. The optimal solution would consist in checking each one of the three groups and removing all the wrong estimations of the CTF. After that, the next step would be checking the good estimations obtained from each group and importing to the database all of them with its associated defocus and the other two PSD at different sampling rates, without coincidences in order to not duplicate information.

The previous approach implies a lot of validation code and, after observing the results obtained in each of the three groups, the decision taken was to not follow it. The reason for this is that the results obtained between the different groups are very uncompensated. Usually one of the groups presents a significantly higher number of good estimations with a very low failure rate in the estimation. Also, it must be considered that always a significant part of the discarded estimations is due to the quality of the micrograph and not because of the estimation itself.

Also, the success rate in the estimation of CTFs between the different resolution groups is compared. From the processed datasets it has been observed that the number of well estimated CTF from the two groups with the lowest success rate that are not included in the highest success rate group is low. Due to this, picking only the group with the best estimations leaves an insignificant amount of well estimated CTFs unselected from the other groups.

From the previous reasoning a much simpler approach has been taken, giving place to the following database confection procedure. The first simplification performed comes from the selection of the source of information. In this case only the groups from each triad which present the greatest number of good estimations is going to be considered. After these groups were selected by the user, the protocol is going to remove all the incorrect estimations and the defocus from the good ones is going to be saved in a metadata text file generated. This metadata file is going to have five different fields for each element:

- ID: a unique identifier for each element of the database.
- Defocus: the correctly calculated defocus value taken from the selected group.
- Kv: the acceleration voltage at which the microscope took the images.
- Subset: the resolution group selected from which the information associated to the estimation is taken.
- File: the file path where PSD used in the correct estimation of the CTF is stored. The file path is slightly change, including the subset group in the name of the file for an easier posterior management of the data.

Once the metadata.txt file is generated, the management of the database is much simpler. Some lines of this file are shown below:

```
 ID      DEFOCUS   kV  SUBSET   FILE
0000000   4206    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130111_10950_aligned_psdAt_1.xmp
0000001   6345    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130110_10652_aligned_psdAt_1.xmp
0000002  12453    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130111_11018_aligned_psdAt_1.xmp
0000003   7736    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130110_10742_aligned_psdAt_1.xmp
0000004   3246    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130110_10657_aligned_psdAt_1.xmp
0000005   7667    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130110_10646_aligned_psdAt_1.xmp
0000006  10516    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130110_10669_aligned_psdAt_1.xmp
0000007  15415    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130113_11301_aligned_psdAt_1.xmp
0000008  15528    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130109_10510_aligned_psdAt_1.xmp
0000009   8338    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130110_10672_aligned_psdAt_1.xmp
0000010   7280    300    1    /home/fede/Documents/DeepDefocus/TrainStack/20130109_10623_aligned_psdAt_1.xmp
...
```

From this file, the rest of the information can be imported easily to the database. Once the groups form which the estimation information is going to be taken are included in the database and hence in the metadata file, the other two groups associated to it can be imported. For this, the same protocol can be used but indicating with a flag that the estimation data is already included in the database. Then the protocol will check the matching files between the selected group and the files already present in the database and will import them, after indicating its subset group in the file name.

In order to reduce the hard disk traffic and accelerate the deep learning network training, an extra step is performed when the database is constructed. With the usage of a new developed Python script, the content of the database is saved employing a data structure easier to load a read by the network using the data structure classes from the Python software package NumPy.

Once this procedure is done for every analyzed group it is possible to start with the design of the network. For an easier way of designing the network, the software package Tensorflow has been used in conjunction with the library Keras, which gives a more comfortable interface to interact with these designment tools.

The main lines of the design are easy to follow since the constructing process consist mainly in the consecutive addition of several layers. The first one added is an input layer, which accepts the images into the network, and, after the propagation of the input data through the network, the defocus value is predicted at the output. Once the model is complete and constructed it, can be trained feeding it with the images and the defocus calculation associated to them.

The network is fed with a 3D data structure which consist in several blocks (one per each defocus estimated), consisting each block of a 3D element with the size of the PSD (512x512 pixels) and 3 overlapping layers for each resolution provided of the image. The defocus values are saved in a regular float vector.

Once the data is ready to feed the network it is necessary to establish the structure of the network that is going to be trained. The power and robustness of the Keras software package is that every network is constructed through the addition of several predefined layers that can be characterized with some feature parameters. This universal interface allows the design of very different networks using the same tools. Here, the main kind of layers used for the confection of the networks are exposed:

- Input(shape, name): the input layer is the first one added to the model in which the shape of the data coming into the network is specified. The rest of the layers will do automatically the shape inference, so this is the only one which needs explicit information about the shape of the input data. Attributes:
    - Shape: the dimensions of the input data of each element which inputs the network. In the case of this project the shape of the incoming data is three images of 512x512 pixels each (512,512,3).
    - Name: the given name of the layer. In the case of this project the given name is “input”.
- Conv2D(filters, kernelSize=(x,x), activation): in this layer a convolutional kernel is created and convolved with the input to produce a tensor of outputs, indicating its number and shape. In other words, this layer is going to convolve the input images with a set of filters from which some features are captured. Attributes:
    - Filters: the number of different filters applied to the image. In the case of this project the number of filters will vary depending on the approach taken and the layer location inside the model.
    - kernelSize: the dimension of the moving window used to capture the features of the images. In the case of this project the size of the of filters will vary depending on the approach taken and the layer location inside the model.

  - Activation: specification of the activation function to use for the filters. Different kinds of activations have implications in results obtained in the features extraction process. In the case of this project the activation is equal to “relu”, rectified linear unit.

- BatchNotmalization(): this layer is used in order to normalize the input data through a mathematical transformation of the input image by adjusting and scaling the activations. The reason behind applying this layer is to improve the speed, performance and stability of the network by reducing the covariance shift. The covariance shift is defined as the change from the input distribution to the internal layers of the network due to small changes from the input that get amplified down the network. Reducing the covariance shift implies a faster convergence of the network.

- MaxPooling2D(pool_size=(x,x)): this layer applies the maximum pooling operation for spatial data. This operation is defined as a wise-behavioral downsampling procedure in which, inside the filter specified, the algorithm will keep the maximum value and discard the others. This filter will move through the image with a specific stride (differential) constructing the output image. A schematic of this process is shown in Figure 25. Attributes:
  - pool_size: this parameter indicates the size of the filter in which the maximum pooling process will be applied. In the case of this project the size of the maximum pooling window will vary depending on the approach taken and the layer location inside the model.
  - stride: this parameter specifies the differential displacement of the maximum pooling window. Although it is not specified in this project, by default it is equal to the pool_size attribute, in order to avoid window overlapping.

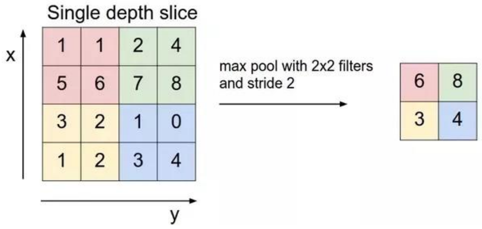


Figure 25: Schematic of a 2x2 filter and same stride maximum pooling procedure for an 4x4 pixel image.

- Dropout(rate): the dropout procedure consists in randomly setting a fraction of the input units to 0 during the training process. This application is done in order to avoid the overfitting of the network to the input data. Attributes:
    - Rate: faction of the data that will be dropped to zero. In the case of this project this rate will be set to 0.2 (20% of the input data).
- Flatten(): the procedure followed in this step is very simple. The shape of the data is prepared in order to fit the input of the dense layer of the neural network. For this the data undergo a flattening process allowing a sequential feeding of the network.
- Dense(units, name, activation): this layer consist in a regular densely-connected neural network layer. All the previous steps were performed in other to a proper preparation of the data and feature extraction, but once they are completed, the data generated is ready to feed the dense layer of the deep neural network. This layer includes the deep learning network and the output generator. Attributes:
    - Units: this parameter indicates the dimensionality of the output space. In case of this project this number depends on where this dense layer is located inside the network. If this layer is placed at the end of the network this value is equal to 1, since this design is just for predicting the defocus value.
    - Name: the given name to the network. In case of the project the given name is "output" if it is the last layer or "dense" if it is not.
    - Activation: specification of the activation function to use for the filters. Different kinds of activations have implications in results obtained in the learning process. In the case of this project the activation is equal to "linear" if it is at the output of the network and depend on the specific approach for any other case.

Once the construction procedure of the networks is clarified is easier to explain the designment procedure. But first, it is necessary to introduce the concept of network parameters and which are their implications in the functioning of the network.

These network parameters are going to characterize the model obtained after the training of the network. If it is true that not all the parameters are trainable, but most of them are, and specific changes on each of them lead to a different behavior of the network and the model obtained as a whole. Due to this, the more parameters the network has the more refined behavior will be possible to model, although the more information is needed for the training process, being aware also of avoiding overfitting.

Once the network is designed and the training data is ready it is possible to start training the network. To do this, the training dataset is divided in batches. A batch is a fixed size group of training elements and, in the training process, the network is fed with every batch one following the next. Once each batch has been propagated through the whole network, the weights are recalculated, and the next batch is introduced

Also, the training process is divided in epochs. An epoch consists in an iteration of the training process in which the network has been fed with the whole training dataset once. This process can be iterated until the network converges or the number or set epoch for the training process is completed.

If training process last too many epochs and the size of the data and the network architecture is not compensated, the network can suffer from overfitting. Overfitting is a statistical phenomenon in which, due to an intensive training process, the output model is too closely fitted to the specific data with which it has been trained.

In the development of the different models designed in the project, the number of epochs used by the default is 100. Although it is an elevated number, the argumentation for this is to be sure that no behavior is lost due to a lack of epochs, although it increases the training time. With the training design, although the network does not converge, a comprehensive study of the characterization of the learning process is achieved.

But, during this training process, some measure of the wellness of the estimations performed by the network is used. Although at the end of the training process the next followed step is the testing process, some inner testing during the training is needed to check how the network is behaving meanwhile.

To accomplish this, before starting the training process some amount of the training data is removed before the construction of the batches. These data will be used as a validation group or batch and, at the end of the epoch, the network will use this data to test how the training process is progressing. The measure of error used during the training of the models is the mean absolute error (MSE).

It is important to consider that, in order to have a robust estimation of the performance of the network in each epoch, the validation data is separated before the training start, so the network will not be ever fed with this data for training purposes.

As an extra functionality, Keras software package allows to include specific behavioral features to the developed models, known as callbacks. These callbacks allow diverse and different functionalities, from a more detailed output of the developed training process, including graphical and analytical information, to changes in the behavior of the network in order to face different events while the training process is performed. From the vast different kinds of possible callbacks only the ones used are exposed:

- CSVLogger. This callback will generate a .csv file with the evolution of the performance of the deep learning network while the training process is taking place.
- TensorBoard. TensorBoard is a specific visualization tool for machine learning processes. With the application of this callback, a log file is generated while the training process is taking place. Then, this file can be visualized with the TensorboardTool, offering a very comprehensive view of the process performed.
- ReduceLROnPlateau. This callback will decrease the learning rate of the network optimizer when the metric has stopped improving from a determined number of epochs. The learning rate gives a measure of the magnitude at which the network is going to change its weights when the input data is propagated through the network. Models usually benefits from a decrement of the learning rate when they get stagnated.

Once the followed procedure for the generation of the models is explained, the structure of the network generated is introduced.

This first approach to the confection of a deep learning network was a standard approach in the field of image processing. In this approach the objective is not to produce a massive network that requires too much data, computational power and time to be trained, but robust and extensive enough network to study its training profile as a starting point.

This first network architecture is composed by 13 layers giving a total of 60.081 parameters. From these parameters, 59,985 are trainable and the remaining 96 are not. A resume of this architecture is summarized in the Table 2.

Table 2 Structure of the first deep learning network developed in the project.

Layer type	Output shape	Number of parameters	Features
InputLayer	512x512x3	0	(512,512,3), “input”
Conv2D	498x498x16	10,816	16, (15,15), “relu”
BatchNormlization	498x498x16	64	
MaxPooling2D	166x166x16	0	(3,3)
Conv2D	158x158x16	20,752	16, (9,9), “relu”
BatchNormalization	158x158x16	64	
MaxPooling2D	79x79x16	0	
Conv2D	75x75x16	6,416	16, (5,5), “relu”
BatchNormalization	75x75x16	64	
MaxPooling2D	37x37x16	0	
DropOut	37x37x16	0	0.2
Flatten	21,904	0	
Dense	1	21905	1, “output”, “linear”

Although this first design presented an acceptable result for the simplicity of the network, it did not fulfill the requirements for the application of the network. It is important to consider that the usage of a deep learning network for the prediction of a very sensitive value for the CTF estimation, as the defocus, is a double-edged sword. On one hand, if a robustly trained model is obtained, an important and sensitive source of errors is eliminated but, on the other hand, if the model is not trained well enough, this same sensitive source of errors will spoil the results due to the wrong estimations obtained.

Also, setting a new point of view with the design of a new network establishes a double check and allows a deeper analysis. This analysis will allow to verify if the previous result is due to the simplicity of the network, unable to learn enough details from the input data in order to give a better resolution, or if there exist a lack of data that will not allow a proper training of a network.

A second architecture is designed for the construction of a new network. The aim of this new design is to increase the learning capability of the network in order to capture more features and details of the input data and increase it learning capability.

This second design is composed by 14 layers giving a total of 5.646.113 parameters. From this set of parameters, 5.646.017 are trainable and the other 96 are not. A resume of this architecture is summarized in the Table 3.

Table 3 Structure of the second deep learning network developed in the project.

Layer type	Output shape	Number of parameters	Features
InputLayer	512x512x3	0	(512,512,3), “input”
Conv2D	498x498x16	10,816	16, (15,15), “relu”
BatchNormlization	498x498x16	64	
MaxPooling2D	166x166x16	0	(3,3)
Conv2D	158x158x16	20,752	16, (9,9), “relu”
BatchNormalization	158x158x16	64	
MaxPooling2D	79x79x16	0	
Conv2D	75x75x16	6,416	16, (5,5), “relu”
BatchNormalization	75x75x16	64	
MaxPooling2D	37x37x16	0	
Flatten	21,904	0	
Dense	256	5,607,680	256, “relu”
Dropout	256	0	0.2
Dense	1	257	1, “output”, “linear”

The design of this network, although it maintains the initial treatment over the input images through the convolutional layers, increases its learning capability. For this, a new dense layer has been added to the network, being it the main source of trainable parameters of the network.

This second network did not completely fulfil the requirements for its design. In order to produce a more stable and robust network a third design is implemented.

The third and last network architecture developed is composed by 12 layers giving a total of 17,938,353 parameters, from which all of them are trainable. A resume of this architecture is summarized in Table 4.

Table 4 Structure of the third deep learning network developed in the project.

Layer type	Output shape	Number of parameters	Features
InputLayer	512x512x3	0	(512,512,3), “input”
Conv2D	498x498x16	10,816	16, (15,15), “relu”
MaxPooling2D	166x166x16	0	(3,3)
Conv2D	158x158x16	20,752	16, (9,9), “relu”
MaxPooling2D	79x79x16	0	
Conv2D	75x75x32	12,832	32, (5,5), “relu”
MaxPooling2D	37x37x32	0	
Conv2D	33x33x64	51,264	64, (5,5), “relu”
Flatten	69,696	0	
Dense	256	17,842,432	256, “linear”
DropOut	256	0	0.2
Dense	1	257	1, “output”, “linear”

As it can be seen, this network presents a new increase in the number of trainable parameters. This increment comes, not as in the previous case adding a new dense layer but feeding this layer with a more intensive information. For this, a new convolutional filter layer has been added and the rest of them have been modified in order to extract more features and details of the input images.

This third deep learning network increased significantly its performance presenting better results in the defocus estimation compared with the previous ones. Due to the results obtained from this model no new designs have been implemented.

Both models have been trained with the same database due to two main reasons. First, in order to take advantage of all processed micrographs and, second, because it is the only procedure that allows a fair comparison between the models.

The whole training dataset contains 8,144 elements, from which 7329 will be used for training and 815 for validation, considering each element as a triad of images.

At this point the training process of the network would be concluded and now it is possible to move on to the testing process. A schematic of the resumed workflow followed in the training process is shown in Figure 26.

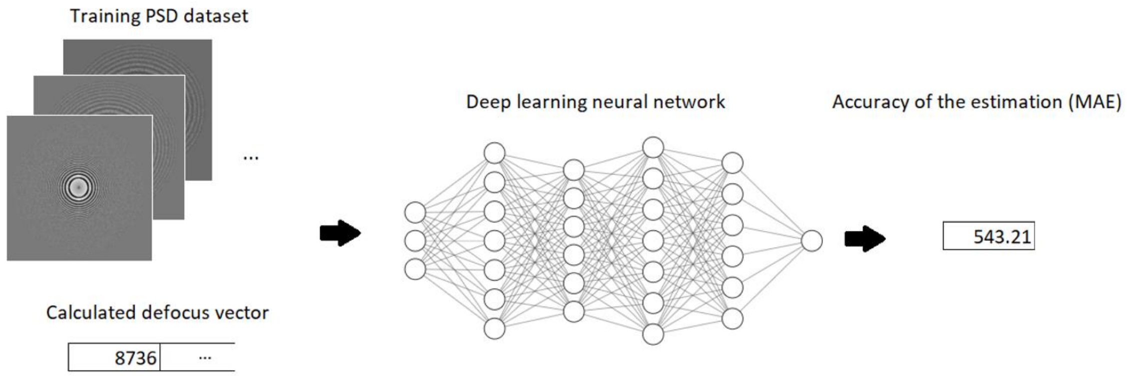

Training PSD dataset
...
Calculated defocus vector
8736
...
Deep learning neural network
Accuracy of the estimation (MAE)
543.21

Figure 26: Schematic of the whole workflow of the deep learning network training process for the micrograph defocus estimation

Once the trained models are available it is possible to advance to the testing process. This process is much simpler than the previous since only it is necessary a sample data and the ground truth information. Then, this data will be processed by the network and a defocus value will be estimated for each input triad of PSD.

Finally, the predicted defocus values from the testing dataset are compared with the ground truth information and the MAE can be calculated to characterize the quality of the prediction.

A schematic of the resumed workflow followed in the testing process is shown in Figure 27.

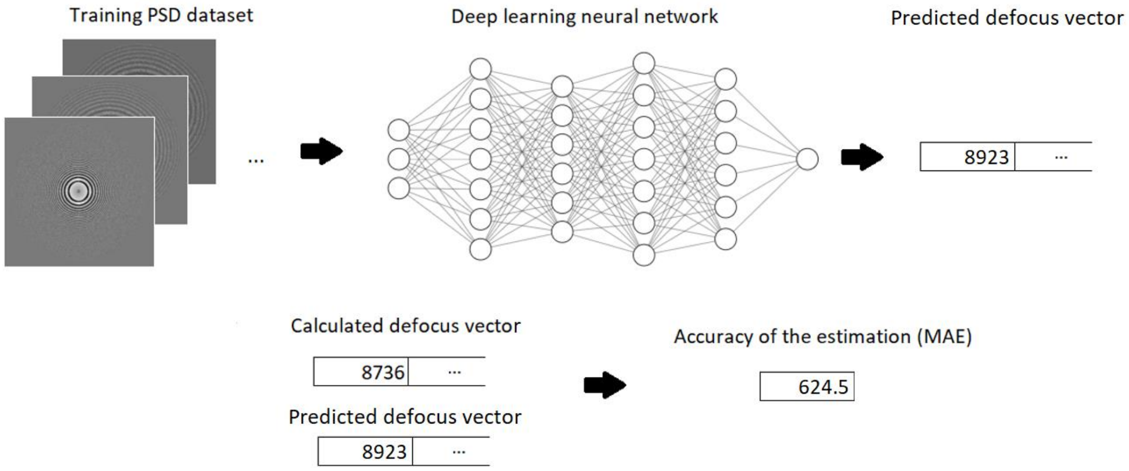


Figure 27: Schematic of the whole workflow of the deep learning network testing process for the micrograph defocus estimation

In order simplify the process and considering that the validation data gives a robust enough estimation of the performance of the model, no testing step is performed in the development of this project.

4 RESULTS

After the exposition of the materials and methods employed in the development of this project, the results obtained from these are presented. For this purpose, this chapter is divided in the same two different sections than the previous one. First, presenting the results obtained from the corrections performed in the CTF estimation protocol, and second, presenting the results obtained from the deep neural network for defocus estimation.

4.1 CTF estimation corrections

In this first section the results obtained from the applied CTF corrections are presented. Due to the relation of the whole set of corrections performed and its joint behavior the results presented compare the performance of the algorithm after and before the whole set of corrections were applied.

An individual analysis of the improvement obtained from each individual correction is nonsense. The reason for this is due to the compose action of several corrections, synergies are generated which raises a higher improvement than the separate effect of each one of them.

Now, diagnosing the improvement of the performance of the algorithm after and before the corrections is not obvious. The reason for this is that two different behaviors must be considered in order to give a robust measure of the performance. First correct estimation of the CTF, and second the correct discarding of those CTF that do not achieve a quality compromise.

In order to do so, the performance measure will be calculated over a matrix that will compare the real quality of the estimation against the estimation. To judge the quality of the estimation it is necessary to analyze each, one by one, which is a time-consuming task if large datasets are going to be analyzed. In order to be robust in the analysis of the improvement performed of the corrections, a large dataset with 1792 elements has been used.

From the confusion matrix four different groups are defined:

- True positive. Those micrographs the are correctly estimated and fulfill the requirements criteria.
- True negative. Those micrographs that were correctly discarded due to the do not fulfill the requirements criteria.
- False negative. Those micrographs that are correctly estimated but did not have to be discarded.
- False positive. Those micrographs that are not correctly estimated but passes through the quality criteria.

The set of elements formed by the first two groups represents the correct behavior of the protocol, while the set of elements from the second two groups constitutes the sources of errors. Two matrices are constructed one before and after the corrections were applied, comparing its results. In Table 5 it is possible to see the classification of these 4 groups in the confusion matrix

Table 5 Confusion matrix for the classification of the results obtained and the evaluation of the performance of the CTF estimation protocol.

Total population	**Condition position**	**Condition negative**
Predicted condition positive	True positive	False positive
Prediction condition negative	False negative	True negative

Now some criteria have been selected in order to give a measure of the performance of the CTF estimation protocol and a quality of its estimation, it is possible to compare the impact of the corrections comparing the confusion matrix obtained before and after the corrections were applied.

In Table 6 it can be seen the results matrix for the algorithm before the corrections were applied.

Table 6 Confusion matrix obtained from the CTF estimation protocol before the corrections where applied.

Total population	Condition position	Condition negative
Predicted condition positive	1457	236
Prediction condition negative	1	98

Once the corrections were applied a significant improvement was obtained both in the absolute values of each group and its relative distribution inside de matrix, meaning that not all the false positives were moved to true negative, but also some estimations were also corrected, leading these element to the true positives group.

In Table 7 it can be seen the results matrix for the algorithm after the corrections were applied.

Table 7 Confusion matrix obtained from the CTF estimation protocol after the corrections where applied.

Total population	Condition position	Condition negative
Predicted condition positive	1622	10
Prediction condition negative	0	160

In order to give a more compact measure of the obtained results that makes easier the comparison of both matrices, before and after the corrections were applied, the F1 score and the Matthew correlation coefficient are used. These scores are going to give a measure of the wellness of the algorithm performance considering the 4 different groups of the matrix.

In the literature, there is some criticize to the usage of the F1 score. The reason for this is that their detractors argue that this coefficient is not considering the relevance of each group, but it is giving the same importance to all kind of errors and matches and refusing any ponderation between them.

Being aware of this issue, the measure is used. The reason for this is that, first the raw data it is presented and a deep comparison between the different groups will be performed while the benefits of simplicity and understanding compensates its backwards.

The F1 coefficient is defined as:

$$\mathrm{F1} = 2 \cdot \frac{\text{precision} \ \cdot \ \text{recall}}{\text{precision} \ + \ \text{recall}}$$

Defining precision and recall as:

$$\text{precision} = \frac{\text{true possitives}}{\text{true possitives} + \text{false positives}}$$

$$\text{recall} = \frac{\text{true possitives}}{\text{true possitives} + \text{false negatives}}$$

Now this first score the result gives in the confusion matrix before the corrections were applied is:

$$\text{precision} = \frac{1457}{1457 + 236} = 0.8606 \qquad \text{recall} = \frac{1457}{1457 + 1} = 0.9993$$

$$\mathrm{F1} = 2 \cdot \frac{\text{precision} \ \cdot \ \text{recall}}{\text{precision} \ + \ \text{recall}} = 2 \cdot \frac{0.8606 \cdot \ 0.9993}{0.8606 + \ 0.9993} = 0.9248$$

And its homologous after the corrections were applied:

$$\text{precision} = \frac{1622}{1622 + 10} = 0.9939 \qquad \text{recall} = \frac{1622}{1622 + 0} = 1$$

$$\mathrm{F1} = 2 \cdot \frac{\text{precision} \ \cdot \ \text{recall}}{\text{precision} \ + \ \text{recall}} = 2 \cdot \frac{0.9939 \cdot \ 1}{0.9939 + \ 1} = 0.9969$$

To reinforce the information given by the previous score, the Matthews correlation coefficient is also applied to the obtained results. This score is defined as:

$$\mathrm{MCC} = \frac{(\mathrm{TP} \cdot \mathrm{TN} \text{ - } \mathrm{FP} \cdot \mathrm{FN})}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

denoting TP as true positives, TN as true negatives, FP as false positives and FN as false negatives.

Now this coefficient is defined, its result calculated for the confusion matrix before the corrections were applies is:

$$\mathrm{MCC} = \frac{(1457 \cdot 98 \text{ - } 236 \cdot 1)}{\sqrt{(1457 + 236)(1457 + 1)(98 + 236)(98 + 1)}} = 0.4990$$

And its homologous after the corrections were applied:

$$\mathrm{MCC} = \frac{(1622 \cdot 160 \text{ - } 10 \cdot 0)}{\sqrt{(1622 + 10)(1622 + 0)(160\ + 10)(160\ + 0)}} = 0.9672$$

All the results presented in this section will be compared and analyzed in the next discussion section.

4.2 Deep learning network for defocus estimation

In this second section of this chapter, the results obtained from the designed deep learning networks for the defocus prediction are presented. But first it is necessary to analyze the training process of the model.

The main reason for this is that, depending on how the training progression was performed, it is possible to rely on the obtained results or not. In principle, if the training data is not performed properly, the testing process will arise poor results.

Remember that the error measurement of the obtained estimation employed while training and testing is the mean absolute error (MAE).

In case the generated model is overfitted, one of the main concerns in the design and train of deep learning architectures, a very specific behavior of the MAE obtained by the training and the validation data can be observed. This overfitting behavior is characterized by a sharp decrement of the MAE function in the training data, since the network is “memorizing” and not learning this data, while the MAE function obtained by the validation data increases sharply, since the network is not learning and the validation element do not match with the training (and memorized) ones.

To check the correct training of the model it is necessary to analyze and compare the evolution of the error functions presented by the training and the validation datasets during the process. The details of this comparison process will be explained in the discussion chapter.

In first place the results obtained from the first simplest model are shown. For an easier interpretation of the performance of this model, the error profile of both datasets (training and validation) are shown at the same time. In the Table 8 it can be seen the MAE obtained at the end of each epoch for the training and the validation data.

Table 8 MAE for both validation and train data in the training step of the first developed model.

Epoch	Train MAE	Validation MAE	Epoch	Train MAE	Validation MAE	Epoch	Train MAE	Validation MAE
1	14923.9461	16483.8667	**35**	642.6408	2135.522	**69**	645.7195	2413.0824
2	4938.4459	3967.1706	**36**	644.529	1955.1537	**70**	650.004	2005.5305
3	2781.569	7406.6253	**37**	648.6108	1804.4199	**71**	641.3758	1854.4794
4	2,191	4434.3644	**38**	636.4016	1835.3795	**72**	647.1924	2237.7793
5	2124.8578	3108.6466	**39**	646.5268	1693.9652	**73**	646.7732	2243.9877
6	1878.028	3840.9335	**40**	639.1028	1942.3927	**74**	649.932	2286.8069
7	1738.0563	1857.3791	**41**	650.2256	1887.6984	**75**	648.4023	1701.4646
8	1578.073	6552.6205	**42**	653.7123	2240.7864	**76**	647.9737	2254.155
9	1502.7047	2687.4888	**43**	638.8355	1924.7676	**77**	645.9368	2012.5067
10	1448.0198	5456.0541	**44**	642.8428	1856.8356	**78**	645.5324	2287.0633
11	1291.2272	2363.7729	**45**	648.8692	1930.4765	**79**	637.2915	1797.9453
12	1231.4586	1597.927	**46**	648.5715	2062.3185	**80**	645.8953	2219.5222
13	1158.1423	3090.2952	**47**	650.2258	2099.0128	**81**	655.5077	2336.5213
14	1114.6565	4252.9029	**48**	638.0292	1823.7257	**82**	643.7384	1862.8117
15	1050.021	1927.2213	**49**	639.6658	1714.2971	**83**	643.3276	1679.5038
16	1001.4539	1090.0683	**50**	651.2495	2169.9248	**84**	633.6328	1806.0378
17	963.3323	1753.5307	**51**	629.282	2080.6183	**85**	626.2174	2322.9727
18	898.294	1098.5901	**52**	641.355	2305.5273	**86**	646.545	1747.5427
19	886.3934	2069.3034	**53**	631.0106	2103.6066	**87**	645.7141	2135.3724
20	838.3466	1850.2737	**54**	654.7787	1739.4262	**88**	651.9804	2314.1894
21	810.9587	1206.2354	**55**	640.1239	2143.4493	**89**	636.5743	1788.2038
22	694.177	2333.9149	**56**	642.5668	2181.2686	**90**	639.281	2260.5833
23	670.944	2037.2854	**57**	642.8349	1893.2544	**91**	644.138	2427.1342
24	677.1188	2279.2257	**58**	641.1354	1818.8469	**92**	647.1844	1994.7515
25	677.4127	2351.1055	**59**	643.9639	2230.8986	**93**	634.6277	1896.8726
26	665.8807	2419.7443	**60**	637.4104	1833.9714	**94**	634.3699	1929.8352
27	652.4883	1784.0337	**61**	643.7096	2150.7264	**95**	645.3282	1885.3223
28	643.7601	2205.7128	**62**	644.5611	1800.7671	**96**	641.7834	1841.7785
29	638.8731	2379.7879	**63**	642.9749	2278.9927	**97**	647.6663	1758.5578
30	646.1048	1798.7329	**64**	640.0991	1755.3121	**98**	637.0184	2122.094
31	656.3399	1900.9764	**65**	652.636	1862.2526	**99**	637.2182	2059.7121
32	643.1721	1914.6919	**66**	638.1169	1984.0148	**100**	646.1381	2082.7864
33	652.9998	1942.2035	**67**	652.5643	1763.0338			
34	638.5752	1833.4693	**68**	646.1592	1677.4069			

In order to facilitate the understanding of the previously presented data, it has been plotted in Figure 28.

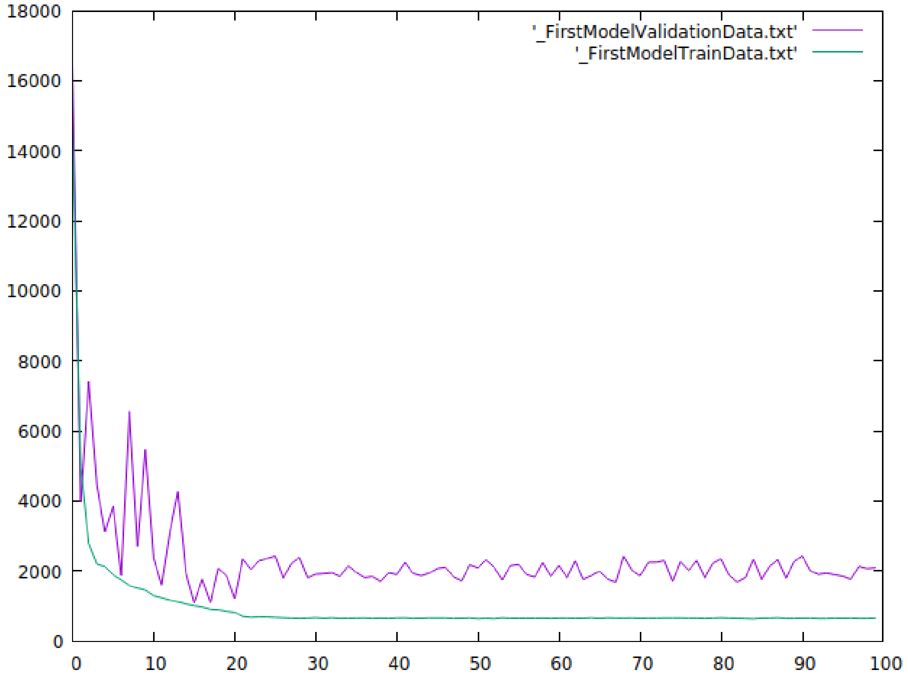


Figure 28: Values obtained at the end of each epoch for the training data (red) and the validation data (blue) measuring their mean absolute error for the first designed model.

Once the MAE values from all epochs for both train and validation were obtained it is possible to characterize the behavior of the generated model. Attending to the error measurement of the estimation performed over the training dataset, from iteration 25 it presents a stable profile around a MAE of 645.03.

Regarding the performance of the model it is necessary to attend to the validation data. This dataset is the one used in order to accomplish the testing procedure of the model while the training process is taking place. The lowest MAE value that the trained model presented once it has abandoned the transitory regime is 1677.4069, at iteration 68.

Once the exposition of the results concerning to the first model is finished the results of the second model are presented. As for the previous model the results regarding to the training and validation processes are shown together. In the Table 9 it can be seen the MAE obtained at the end of each epoch for the training and the validation data.

Table 9 MAE for both validation and train data in the training step of the second developed model.

Epoch	Train MAE	Validation MAE	Epoch	Train MAE	Validation MAE	Epoch	Train MAE	Validation MAE
1	5265.3221	3921.5278	**35**	5265.3221	625.7383	**69**	772.1474	592.9078
2	1685.9492	3817.9805	**36**	1685.9492	556.8276	**70**	772.0543	612.2017
3	1321.3871	1507.3217	**37**	1321.3871	570.373	**71**	766.1735	637.195
4	1144.6739	2637.551	**38**	1144.6739	613.6317	**72**	772.2699	648.1382
5	1046.2012	1057.1036	**39**	1046.2012	667.5173	**73**	782.8852	630.3086
6	1002.4411	563.306	**40**	1002.4411	660.1959	**74**	783.8548	558.2686
7	969.3886	971.7463	**41**	969.3886	704.3576	**75**	773.0874	597.0657
8	950.4835	1069.81	**42**	950.4835	605.2893	**76**	760.7962	655.5763
9	895.6867	765.5455	**43**	895.6867	703.9587	**77**	764.5396	635.196
10	902.6185	1129.4586	**44**	902.6185	640.8378	**78**	758.441	586.288
11	893.6601	336.2581	**45**	893.6601	695.2381	**79**	776.2029	660.8635
12	873.2935	520.0651	**46**	873.2935	597.2834	**80**	771.9667	629.5203
13	875.9147	808.1545	**47**	875.9147	600.4609	**81**	771.369	673.9074
14	866.2153	1449.1964	**48**	866.2153	655.7635	**82**	770.0508	616.4582
15	895.7293	524.8058	**49**	895.7293	619.6062	**83**	759.1199	628.1327
16	841.8867	882.3111	**50**	841.8867	625.7872	**84**	770.3139	680.6082
17	797.4445	677.6053	**51**	797.4445	603.2508	**85**	767.5731	589.4382
18	774.9127	1183.6225	**52**	774.9127	654.5428	**86**	769.6572	633.812
19	775.3095	942.7841	**53**	775.3095	737.1643	**87**	771.6747	637.1917
20	788.5488	980.315	**54**	788.5488	660.325	**88**	781.1145	657.2766
21	785.1263	591.8588	**55**	785.1263	604.6705	**89**	765.1878	599.2198
22	764.1961	680.1444	**56**	764.1961	631.7743	**90**	765.3799	672.5181
23	766.6758	550.037	**57**	766.6758	704.7565	**91**	752.1925	692.8028
24	767.5778	625.1243	**58**	767.5778	600.7221	**92**	766.8335	660.2816
25	769.1709	631.6594	**59**	769.1709	584.0653	**93**	761.1048	594.6647
26	783.713	691.5395	**60**	783.713	587.8008	**94**	783.061	617.3551
27	775.5811	632.6427	**61**	775.5811	663.2713	**95**	781.6897	614.0408
28	781.7389	608.8003	**62**	781.7389	610.172	**96**	759.6743	594.3741
29	774.14	630.1872	**63**	774.14	670.7733	**97**	770.0165	671.0063
30	771.477	596.3289	**64**	771.477	639.7269	**98**	774.0807	591.4648
31	772.7271	635.9029	**65**	772.7271	661.9095	**99**	761.8572	726.7094
32	769.4824	598.6687	**66**	769.4824	632.9932	**100**	778.9581	625.5055
33	773.3475	599.8772	**67**	773.3475	671.4829			
34	779.9682	639.036	**68**	779.9682	625.7383			

In order to facilitate the understanding of the previously presented data, it has been plotted in Figure 29.

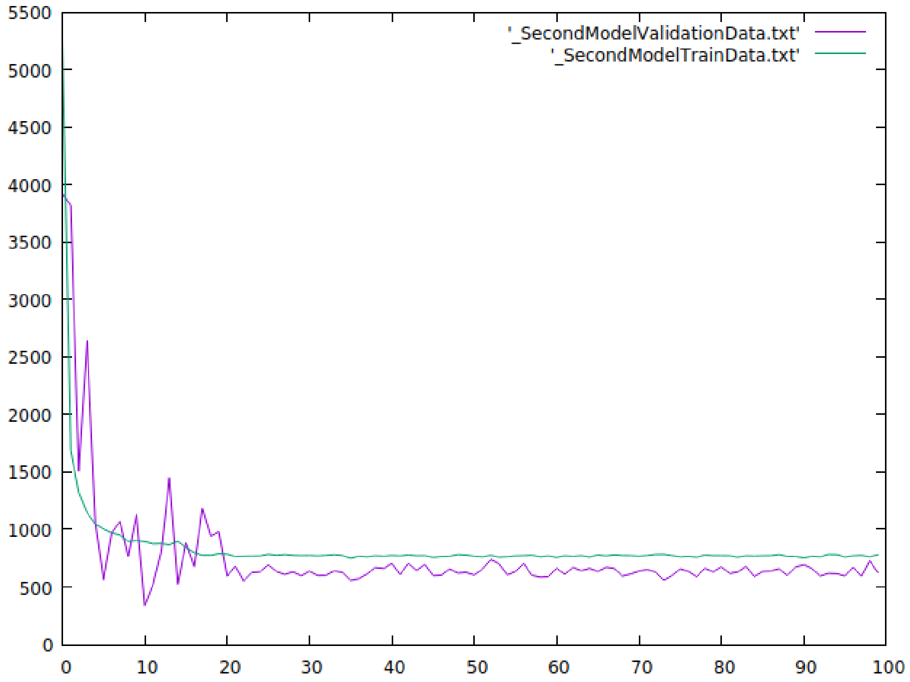


Figure 29: Values obtained at the end of each epoch for the training data (red) and the validation data (blue) measuring their mean absolute error for the second designed model.

Once all the MAE values for both train and validation were obtained it is possible to characterize the behavior of the generated model. Attending to the error measurement of the estimation performed over the training dataset, from iteration 25 it presents a stable profile around a MAE of 769.61.

Regarding the validation data, the lowest MAE value that the trained model presented once it has abandoned the transitory regime is 556.8276, at iteration 36.

Finally, the same previous procedure is done for the last third model constructed. As for the previous model the results regarding to the training and validation process are shown together. In the Table 10 it can be seen the MAE obtained at the end of each epoch for the training and the validation data.

Table 10 MAE for both validation and train data in the training step of the third developed model.

Epoch	Train MAE	Validation MAE	Epoch	Train MAE	Validation MAE	Epoch	Train MAE	Validation MAE
1	6906.0705	5578.7515	**35**	954.101	504.4455	**69**	942.8634	504.1992
2	4246.2555	4188.6799	**36**	954.9026	511.2336	**70**	940.9698	504.1992
3	3393.3498	3578.6936	**37**	936.6101	505.8664	**71**	944.3294	504.1992
4	2909.4139	3048.8931	**38**	943.3834	504.6351	**72**	938.4253	504.1992
5	2304.9447	3520.9587	**39**	947.2607	504.329	**73**	942.9315	504.1992
6	1900.0349	3138.7949	**40**	942.7136	504.0495	**74**	931.8519	504.1992
7	1712.2279	4051.9137	**41**	936.3492	504.7355	**75**	945.707	504.1992
8	1750.4901	1813.2277	**42**	942.1123	504.5796	**76**	944.2089	504.1992
9	1369.9822	1982.2114	**43**	931.7128	504.5702	**77**	940.3295	504.1992
10	1318.1988	874.4662	**44**	945.1441	504.4749	**78**	947.1132	504.1992
11	1387.1404	1460.172	**45**	920.3414	504.2922	**79**	937.959	504.1992
12	1189.2656	1592.9411	**46**	949.5572	504.2831	**80**	944.2289	504.1992
13	1276.3478	1518.306	**47**	947.3945	504.2045	**81**	935.9772	504.1992
14	1115.8838	1338.4267	**48**	959.8998	504.2013	**82**	935.398	504.1992
15	1159.7688	928.6857	**49**	950.2943	504.1994	**83**	936.4687	504.1992
16	1009.5135	658.3785	**50**	942.4473	504.1991	**84**	942.7208	504.1992
17	985.6337	710.5122	**51**	944.0733	504.2005	**85**	932.2695	504.1992
18	993.071	1002.9568	**52**	945.1506	504.1986	**86**	946.453	504.1992
19	982.6094	552.6769	**53**	948.9198	504.2	**87**	937.8504	504.1992
20	983.668	551.1602	**54**	945.0734	504.1985	**88**	931.7735	504.1992
21	985.6328	623.3457	**55**	946.7009	504.1997	**89**	936.1044	504.1992
22	973.1463	573.5585	**56**	940.1063	504.1975	**90**	932.9595	504.1992
23	985.9028	679.3578	**57**	945.8268	504.1978	**91**	947.0551	504.1992
24	966.0499	619.6917	**58**	943.9019	504.1975	**92**	949.5227	504.1992
25	959.3828	551.9126	**59**	937.2469	504.1971	**93**	946.9618	504.1992
26	961.9479	530.5338	**60**	936.642	504.1988	**94**	927.4164	504.1992
27	933.8925	488.1128	**61**	945.7086	504.1987	**95**	933.7468	504.1992
28	952.2826	499.0185	**62**	926.2942	504.1992	**96**	932.889	504.1992
29	938.3903	495.0139	**63**	936.484	504.1992	**97**	952.4123	504.1992
30	949.8653	518.8249	**64**	951.0844	504.1992	**98**	945.8782	504.1992
31	948.6656	499.7923	**65**	926.476	504.1992	**99**	941.557	504.1992
32	935.0021	537.9519	**66**	933.2736	504.1992	**100**	925.2371	504.1992
33	950.8735	500.2287	**67**	941.5843	504.1992			
34	954.6252	507.0442	**68**	948.0085	504.1992			

In order to facilitate the understanding of the previously presented data see, it has been plotted in Figure 30.

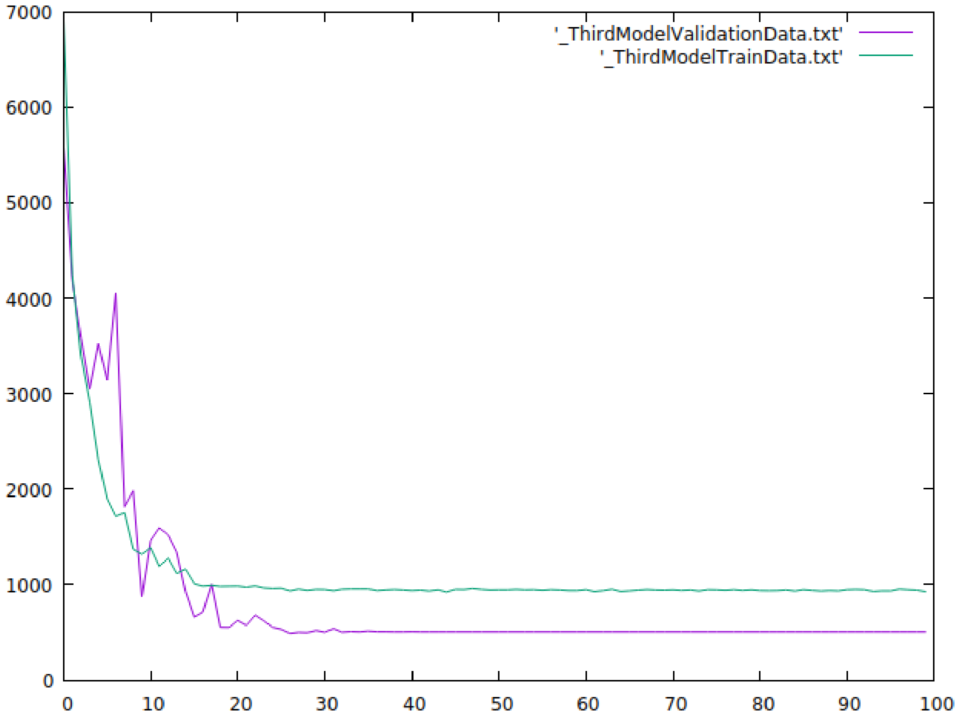


Figure 30: Values obtained at the end of each epoch for the training data (red) and the validation data (blue) measuring their mean absolute error for the third designed model.

Once all the MAE values for both train and validation were obtained, it is possible to characterize the behavior of the generated model. Attending to the error measurement of the estimation performed over the training dataset, from iteration 25 it presents a stable profile around a MAE of 942.34.

Regarding the validation data, the trained model falls from iteration 62 into a stable MAE of 504.1992, after leaving the transitory regime. In this case, instead looking for the lowest MAE to find the epoch at which the model presents the best performance, the performance resulting model was better characterized by the stable region presented at the end of its training process.

Results will be compared and analyzed in the next discussion section.

5 DISCUSSION

Once the results are presented it is necessary to achieve some discussion in order to put them in context and understand its relevance. For this purpose, this chapter, as the two previous one, is divided into sections. First, for the analysis in the corrections applied in the CTF estimation protocol and second, for the deep learning network for defocus estimation designed.

5.1 CTF estimation corrections

In this first discussion section, the results obtained from the corrections performed in the CTF estimation algorithm are analyzed. In order to do so, it is necessary to analyze each one of the different groups generated by the confusion matrix in order to understand where and how these corrections had an impact in the protocol.

But before going deep in the study of the performance of the protocol, the macroscopic measures of the results obtained will be analyzed first. Both scores or coefficients present different behaviors that will be compared and contextualized with the results obtained. But before going deep in the analysis and comparison of both statistics it is important to understand the different implications of the groups that are disposed in the confusion matrix, being some confusions or error more harmful than others for the whole processing workflow.

Although the interpretation of the data is lost while these scores are used, they easily resume the obtained results in just one parameter which is a comfortable procedure in order to easily characterize the behavior of a classifier. Due to the simplicity of these coefficients it is important to understand how they are constructed, and which kind of behaviors from the estimation protocol will have more impact in its result and which of them will be underrated.

Starting with the F1 estimation, it shows a 7.8% increment from 0.9248 to 0.9969. Considering that this score can takes values between 0 and 1, the quality of the estimation according to this coefficient is highly accurate even before the corrections were applied. The reason for this, and one of the main criticisms granted to this score, is that it does not apply any ponderation between the different groups in the confusion matrix. This behavior leads to a difficult characterization of the estimators or classifiers in which most of the elements are correctly estimated in case it is needed to analyze the sources of errors, as it is the case of the CTF estimation protocol.

The second scored used is Matthew correlation coefficient. Compared to the F1 score, the improvement of the protocol using this coefficient is far exalted presenting a 93.83% increment from 0.4990 to 0.9969. If it is considered that this score is bounded from -1 to 1, this increment is even more significant.

The reason for this difference between both coefficients is due to how each of them is constructed. The Matthew correlation coefficient is more informative rather than the F1 score in the binary classification problem because it considers the balances between the four different groups of the confusion matrix.

The difference in the behavior between the two scores is even more outstanding when most of the elements are properly estimated and it is necessary to deal with false positives and false negatives analysis, as in the case of the CTF estimation algorithm. For these cases, Matthew correlation estimation gives a more robust measure between the ratios of the different groups that form the confusion matrix. Due to this, the correction of falsely classified elements, basically false positives, has a larger implication in this estimator.

Since this protocol is both performing the estimation and analyzing the wellness of it, the disposition of this matrix can be significantly affected if deep changes are performed in the behavior of the protocol. In a perfect case scenario, and counting with a high-quality dataset, most of the elements should be placed in the True positive group and, as the quality of the dataset decreases, more elements will be displaced from this to the true negative group.

In case of some element do not belong to one of this groups, supposing a classification error, the consequences of one or the other would be different. Without a doubt, false positive mistakes are always worse than false negatives. If losing value information is always something undesired (false negatives), the inclusion of mistaken data into the workflow must be the first correction priority (false positives). The reason for this is that these false negative elements could spoil the results from a workflow that, without the inclusion of these elements, will arise good results. This is the main reason why, although analyzing the macroscopic values is a good practice, some deeper study must be performed.

As it can be seen, although most of the elements were correctly classified, there is a non-neglectable number of false positives before the corrections were applied. These elements are the first ones whose behavior must be corrected. Also, the number of false negatives is insignificant and neglectable. In order to correct this behavior two different approaches can be performed, both improving the estimation or the selections criteria although. At the end, a combination of both is performed.

If both confusion matrices are compared, the main correction is performed over the false positive group where after the correction the number of elements present fell from 236 to 10, supposing a decrement of the 95.76%. This is an extraordinary improvement in the performance of the protocol.

Attending to the rest of the groups, important improvements can be seen in the properly classified elements, increasing a 11.32% for the true positive group and a 63,27% for the true negative. Although an increment in the true negative group could not be considered as an improvement in comparison to the true positive group, some considerations must be taken. Firstly, not all the CTF need to be properly estimated since, if the quality of the PSD does not fulfill the criteria, the correct decision is to discard it although a good estimation is performed. Also, since the estimation process of the CTF cannot be perfect, the best decision that can be taken for the bad estimated profiles is to discard them. The false negative group is ignored due to its lack of elements.

If a deeper study is performed over the false positive group, it is possible to characterize some common features or behaviors of the corrected elements. Most of them presented an error in the detection of the first zero. This behavior and its correction can be seen in Figure 31:

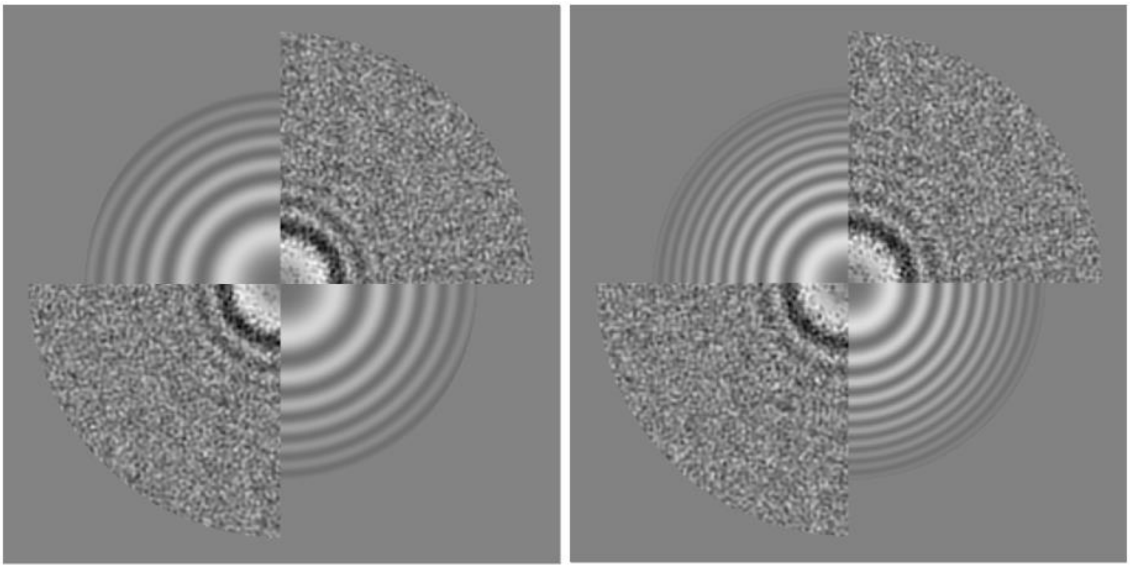


Figure 31: False positive estimation (left) and its comparison after the application of the corrections (right)

The other two main source of false positives that has been corrected were PSD presenting an excessive noise or astigmatism that were not discarded. An example of these to behaviors is presented in Figure 32:

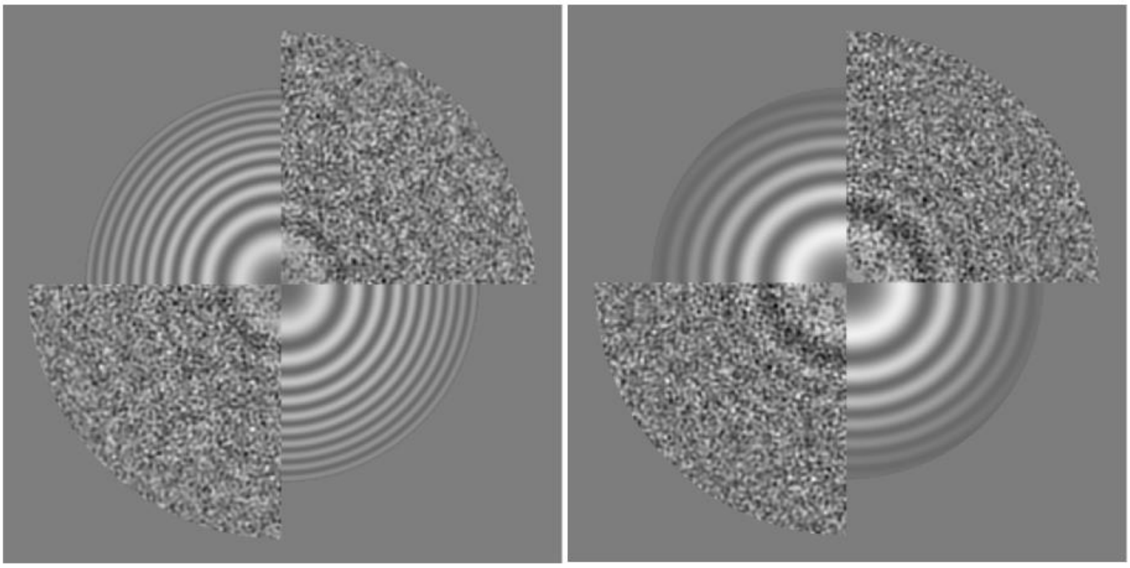


Figure 32: False positive estimation due to an undetected excess of noise (left) and false positive estimation due to an undetected astigmatism (right).

Finally, but a lower rate, some of the estimation where not even show (some not even included in the output set) due to the mistaken metadata parsing process and other displaying issues. This problem has been also corrected and an example of it can be shown in Figure 33.

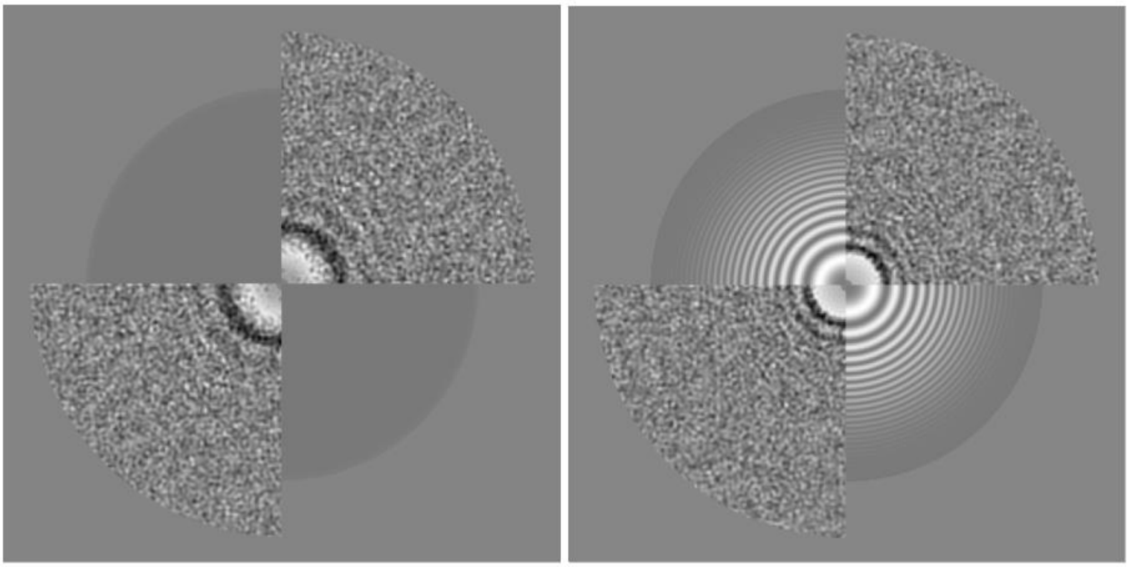


Figure 33: False positive estimation due to a mistaken parsed metadata (left) and its comparison atter the corrections estimated at a different sampling rate (right).

After going in detail into the comparison of the false positive group before and after the corrections were applied, an overview of how the corrections affected to this group in comparison with the rest is mandatory due to the significative improvement. The corrections applied had two main consequences to the elements belonging to this group, both almost in the same proportion.

First, the CTF estimations that were correctly performed moved these elements to the true positive groups while second, the CTF estimations that were not corrected or which criteria did not fulfill the requirements were moved to the true negative group. Almost all the 10 remaining elements of the false positive group after corrections presented a no detection of an astigmatic behavior in the PSD.

Attending at the final disposition of the confusion matrix after the correction were applied, although almost all the elements are properly classified, it is a clear decompensation between the false positive and the false negative group. This behavior, attending to the previous exposition, is undesired since it is always better to miss some good data rather that feeding the workflow with a corrupted one.

Although this behavior could be corrected applying a more restrictive estimation criteria, moving the false positives to the true negatives group and losing from other side some true positives into the false negatives, this was not applied. The reason for this is mainly because the cost-benefit ratio did not worth it, losing some true positive estimations to correct a few false positives.

5.2 CTF estimation corrections

In this second section of the discussion chapter the results obtained from the developed deep learning network for defocus estimation are analyzed. As it can been in the results sections, three different networks have been developed adding some changes from one to the next in order to achieve the better performance of the output model at the end of the training process.

Machine learning solutions are useful tools for problem solving, more when some behavior or data want to be characterized in order to produce a wise output. The application proposed in this project fits very well to these programming approaches since a lot of coherent information (an image) is going to be used to predict a single numerical value (the defocus of the micrograph).

But before analyzing the results, some comments need to be done about the problematic that this application has. First, constructing deep neural network, changing its architecture and parameters it is something that, in principle requires a deep effort of design and validation. Due to the usage of TensorFlow as a calculation engine and the software library Keras working as a user-friendly interface in order to design and develop these networks, this task is highly simplified. With the usage of these tool changing the architecture of the network or its parameters is an affordable task, allowing the accessible and clear application of different nuances.

Once the design problem is covered there is another, even harder, task that must be solved. This problem is to gather the enough computational power in order to train the network. Deep neural network training requires a huge amount of calculation power, even more if it is making some prediction from input images. To do so, this task cannot be performed in a normal user computer, and a more powerful device is required. To solve this, these networks have been trained in a computational server form the CNB, the Carver machine.

There are two main causes why it is necessary the usage of this machine. First, a personal computer does not usually have enough memory to allocate all the calculations required in the train process of a network and, second, the usage of a powerful GPU to perform the training step is deeply encouraged. Although a network can be trained employing only a CPU, the usage of a GPU makes this process around 50 times faster, taking from one hour and a half to a couple of minutes for each epoch trained. This is not a worldwide measure of the performance of a GPU for this task, just a measure of the computing times experimented in the consecution of this project.

Once all the tools needed to develop the deep learning network are available it is time to prepare the data and design the network. The most time-consuming task from the computational point of view is the preparation of the training database, although the positive factor is that it is only needed to be done once. For training the network, the usage of powerful GPU significantly lightens the computational process.

The reason why it is given this importance to all the computational complexity and the resources necessary to accomplish the calculation is due to the significant payload that they suppose in order to develop a deep learning solution. The amount of data needed for the training and the complexity of the models requires powerful tools.

Due to a lack of experience in the application of machine learning solutions the first model developed is a conventional approach to a network able to process and learn from images that has been tested for other applications. The results of this network were worse than expected, and the MAE of 1677.4069 was significantly high considering the range of values in which the defocus usually is bounded.

Analyzing the progression of the MAE of the train and the validation data through the different epochs it is possible to see how, although the training data has a smoother profile, the validation data present an unstable one. Also, the validation data present a significant higher MAE compared with the training data, although both of then presents an unacceptable error rate. The interpretation done about this result is that, due to the simple architecture presented by this attempt of deep learning network, it may not be able to learn from the captured features of the input images. Due to this it is not possible to produce a correct prediction of the input image.

In order to solve this behavior a second network is designed, with the only change that a new dense layer is added. The intention sought when adding this new layer to the network is to significantly increase the number of trainable parameters. All the new parameters added should be able to learn the captured information by the convolutional layers.

Attending to the obtained result from this second design, the pursued objective was partially achieved. As it can be seen in the result section, the MAE presented by this new model significantly decrease, reaching the value of 556.8276. As it can be seen in the learning process both the train and the validation error functions did not present any qualitative change.

Another difference that can be observed is that, once the both train and validation profiles are stable and the transitory region has finished, the validation profile is situated below the train one. Although this is not a typical behavior, the small difference between them discards any concern.

Apart from the decrement in the in the error rate, the progression of the profile after the changes in the new network were applied did not present any improvement in the stability of the error function, still presenting an irregular behavior.

The new model arises an error that is closer to an acceptable prediction for a defocus estimation. Although the improvement introduced in the prediction after this change in the design of the network, the error rate and the still unstable profile presented by the validation data aims to look for some modification in the architecture of the model that keeps improving its performance.

In order to look for corrections of the actual model, it is necessary to interpret its results and understand the consequences in the behavior of the protocol. The interpretation is that the new dense layer introduced, and consequently, the significative increment in trainable parameters, allows the network to capture and model more complex and detailed behaviors. This new captured behavior improves the prediction, reducing the error in the estimated defocus value. Due to this, instead of adding new more dense layers and trainable parameters to the network a different approach is tried.

The last included modification to the network consists in a different design of the convolutional layers. The aim was to feed the network with more information increasing the features extracted from the images, now that there is a significative learning capability of the network. In order to do so, a new convolutional layer has been added and some modifications have been done in all of them. In order to improve the feature detection, the number of filters (channels) will be progressively increased from 16 to 64 by a factor of 2 while the size of the filters is reduced from 15 to 9 and finally to 5.

Also, the batch normalization layers are removed. The reason for this is that there is not enough variation in the input the data in order to produce some covariant shift effect that decreases a network performance.

Finally, with the previous modifications, the third and final model is constructed and trained. The result obtained from this model fulfilled the expectations from the corrections. The decrement of the MAE reaching 504.1992, although it is a smaller decrement than in the previous corrections, suppose a new significative reduction of the error obtained from the model.

Also, and more significantly, the reduction of the instability in the error function is huge, even more if it is considered the difference between the first erratic behavior and the posterior flat one after the second set of corrections was applied.

With the first correction the learning capability of the network is addressed and with the second the quality and quantity of the information it is fed with improves.

Some final comments must be done about some general behaviors observed in every model. The initial unstable behavior of the validation data, common for all the three models is due to a slightly elevated learning rate. Thus, the network tries to learn from each epoch more image features than it should. Due to this elevated learning rate, the network presents this erratic results in the prediction of the validation data at the beginning of the training process.

Also, one of the reason due to all the train data error functions tends to smooth behavior is, in part, due to the application of a reduction in the learning rate when a plateau region is reached. The main implication of this application is that, when the error function did not present significant changes for five epochs, the learning rate is reduced.

Also, no overfitting behavior is observed in any of the presented models, so no correction in the design or the training data were needed to correct this situation. The reason why it can be assured that no overfitting took place is because the behavior presented by any of the considered models is this project did not present any of the typical consequences of an overwitted training process.

6 CONCLUSIONS

Once the project has been concluded and the results obtained have been analyzed some conclusions are extracted from the developed work.

The first remarkable point in the course of the project is the entry barrier that suppose working on an already developed project, more being aware the size of the protocol and the environment in which it is immersed (the Xmipp software package). This kind of protocols that performs complex calculation requires a deep initial study of their workflow before applying any correction which suppose an increment in the complexity of the developed work. This entry barrier is even tougher if the added difficulty of the need of theoretical knowledge of the performed process is needed too.

On the other hand, these previous drawbacks carry the profit that, in case the project presents an improvement or a new solution, it will have real applications in the field, and it will not be reduced to a theoretical range of action. Since the corrections are performed on a professional software tool, the increment in the performance will have implications in real applications.

The CTF estimation is a complex process where the corrections applied not always improved its performance. Deep analyzing the implications of each correction is necessary to understand its impact and viability in the final configuration of the protocol. The identification of the sources of erroneous behavior in the protocol is a slow dedicated process, even more considering that not all the sources of error are due to image processing but due to data manipulation and other code bugs.

Also, the individual effect of each correction may not cause a significant change in the behavior of the algorithm but the combination of all of them suppose a correction in the behavior of the protocol as a whole. Each correction applied implies an intensive tedious process of trial and error in order to assure that the performed changes in the behavior of the protocol are not casualty or that they do not have any adverse implication under another paradigm in its execution.

An objective measure of the improvement performed in the protocol is hard to calculate. Measuring the implications of each correction is not representative due to the synergies coming from the corrections combination. In order to give an overall characterization of the effect of the corrections performed, a more detailed result measure must be given. This measure must consider independently correctly admitted and discarded estimations, true positives and true negatives, and misclassified estimations, false positives and false negatives.

From these results it is observed a significant improvement in the behavior of the protocol. Depending on which macroscopic score it is used the impact of the corrections varies, thus deep analysis has been performed over the confusion matrix. Attending to this matrix, the main effect of the corrections is affecting the false positive classified elements, reducing its number more than a 95%. This is a very positive result since the false positive group is the most harmful for the correct performance of the workflow.

From the corrections performed one of the main conclusions obtained is the importance of a correct calculation of the defocus for a proper estimation of the CTF. This is the main reason why the continuation of the project was the search of a more robust procedure for the calculation of the defocus form a PSD.

The solution chosen for a better defocus calculation procedure is the implementation a deep neural network. These kinds of solutions adapt nicely to the proposed problematic since, from a very powerful and intensive training data, the PSD dataset, it is only necessary to predict a number, the defocus.

Also, the different approaches in the design of a deep learning network have been studied, comparing the different possibilities and approaches, changing some characterization parameters and comparing the different results obtained. Once the proper design of the network was achieved, a robust method for defocus estimation is obtained.

The design procedure of these deep learning networks was based in the improvement in the performance of an original design until a robust and accurate architecture was obtained for the network. The last design met the compromises of stability and error prediction, presenting a MAE of 504.1992.

6.1 Future work

Although the developed project has achieved the objectives established from its beginning, some posterior designs could be implemented in order to give a broader application to the developed work and to increase its action range.

The first and more straight forward continuation for the developed work would be the development a new deep learning model that allows the defocus calculation form micrographs coming from a 200 kV electron microscope. This process would be easier to perform since only an enough large dataset of 200 kV micrographs is needed. Then the procedure is identical as the one performed for the 300 kV model.

Also, some improvements can be performed in the network since, as it has been mentioned in the discussion section, its behavior is still improvable. Different network architectures and data usage are under development in order to obtain a more robust model to perform this task.

Once the models are generated, the natural next step is their integration in the software package Xmipp through the creation of a new protocol. This new protocol should take some input micrographs, generate the dataset with the PSD at the same resolutions with which the model has been trained and then calculate their defocus. Once the defocus is calculated, the protocol could conclude its execution. Since the CTF estimation protocol from Xmipp has an option to import the defocus from another estimation, the combination of both protocols is straightforward.

Outside the previous exposed future lines of work, from the knowledge acquired during the development of this project and in view of the promising obtained results, other applications employing deep neural network can be designed. Once of the possible applications could be stablished over the goodness of the estimation performed, allowing its proper prediction with the employment of a deep neural network. For this, a deep learning network could be trained per each criterion or, if possible, only one, eliminating the need of the estimation criteria in the code.

Finally, some less disruptive corrections could be performed. Due to the variety of tools that have been used, one of them is the inclusion of safe code practices on the developed work. A good example of this is the downsampling tool from Xmipp, in order to avoid undesired or unexpected behaviors.

As a final comment, working with an already designed tool or protocol and applying corrections to it is a hard but always feasible project. The truly complicated objective is to reach a robust and efficient design able to always perform correct estimations with a stable behavior. Due to this, always new sources of error can be found, and corrections can be applied in order to improve the behavior of an algorithm and its performance, even more if it is such an extent protocol immersed in a very complex environment as in the case of this project.

7 REFERENCES

[1] De la Rosa JM, Oton J, Marbini R, Zaldivar A, Vargas J, Carazo JM, Sorzano COS. Xmipp 3.0: An improved software suite for image processing in electron microscopy. *Journal of Structural Biology*, vol 184, issue 1, 2013, pp 321-328

[2] Sorzano COS, Marabini R, Velazquez-Muriel J, Bilbao-Castro JR, Scheres SHW, Pascual-Montano A, Carazo JM. XMIPP: a new generation of an open-source image processing package for electron microscopy. *Journal of Structural Biology*, vol 148, issue 2, 2004, pp 194-2014

[3] Sorzano COS, Jonic S, Núñez-Ramírez R, Boisset N, Carazo JM. Fast, Robust, and accurate determination of transmission electron microscopy contrast transfer function. *Journal of Structural Biology*, vol 160, issue 2, November 2007, pp 249-262.

[4] Bogner A, Jouneau PH, Thollet G, Basset D, Gauthier C. A history of scanning electron microscopy developments: Toward “wet-STEM” imaging. *Micron*, vol 1, issue 4, June 2001, pp 390-401.

[5] Vilas JL. Local Quality Assessment of CryoEM Reconstructions and Its Applications. Vol 1. *CSIC Thesis*, March 2019.

[6] Haguenau F, Hawkes PW, Hutchison JL, Satiat-Jeunemaitre B, Simon GT, Williams DB. Key events in the history of electron microscopy. *Microscopy and Microanalysis*, vol 9, 2003, pp 96-138.

[7] Masters BR. History of the Electron Microscope in Cell Biology. *eLS*, May 2001.

[8] Ercius P, Alaidi O, Rames MJ, Ren G. Electron Tomography: A Three-Dimensional Analytic Tool for Hard and Soft Material Research. *Advanced Materials*, vol 27, issue 38, June 2015.

[9] Sorzano COS, Otero A, Olmos EM, Carazo JM. Error analysis in the determination of the electron microscopical contrast transfer function parameters from experimental power Spectra. *BMC Structural Biology*, vol 9, issue 18, March 2009.

[10] Zegers P. Speech Recognition Using Neural Networks. *University of Arizona Master of Science Thesis*, 1998, pp 37-46.

8 APPENDIX

8.1 PrepareDataset.py

```
import os
import sys
import xmippLib as xmipp
import shutil


class DeepDefocus:
  def importCTF(self, fnDir, dataFlag):
    fileList = []
    for file in os.listdir(fnDir):
      if file.endswith("_enhanced_psd.xmp"):
        fnRoot = os.path.join(fnDir, file)
        md = xmipp.MetaData(fnRoot.replace("_xmipp_ctf_enhanced_psd.xmp", "_xmipp_ctf.xmd"))
        objId = md.firstObject()
        dU = md.getValue(xmipp.MDL_CTF_DEFOCUSU, objId)
        dV = md.getValue(xmipp.MDL_CTF_DEFOCUSV, objId)
        kV = md.getValue(xmipp.MDL_CTF_VOLTAGE, objId)
        enabled = md.getValue(xmipp.MDL_ENABLED, objId)
        if dataFlag == 1:
          if enabled == 1:
            fileList.append((fnRoot, 0.5*(dU+dV), kV))
        else:
          fileList.append(fnRoot)
    print("Files read from origin")
    return fileList


  def downsampleCTF(self, fileList, stackDir, subset, dataFlag):
    if dataFlag == 1:
      for file in fileList:
        fnRoot, defocus, kV = file
        fnBase = os.path.split(fnRoot)[1]
        destRoot = stackDir + fnBase.replace("_xmipp_ctf_enhanced_psd.xmp", "_psdAt_%d.xmp"
% subset)
        if os.path.isfile(os.path.join(stackDir, "metadata.txt")):
          metadataPath = open(os.path.join(stackDir, "metadata.txt"), "r+")
          metadataLines = metadataPath.read().splitlines()
          lastLine = metadataLines[-1]
          i = int(lastLine[0:9]) + 1
        else:
          metadataPath = open(os.path.join(stackDir, "metadata.txt"), "w+")
          metadataPath.write("  ID        DEFOCUS      kV   SUBSET  FILE\n")
          i = 0
        shutil.copy(fnRoot, destRoot)
        metadataPath.write("%9.7d%11d%8d%9d  %s\n" % (i, defocus, kV, subset, destRoot))
        i += 1
      print("Files copied to destiny and metadata generated")


    else:
      for fnRoot in fileList:
        metadataPath = open(os.path.join(stackDir, "metadata.txt"), "r+")
        fnBase = os.path.split(fnRoot)[1]
        destRoot = stackDir + fnBase.replace("_xmipp_ctf_enhanced_psd.xmp", "_psdAt_%d.xmp"
% subset)
        metadataLines = metadataPath.read().splitlines()
        metadataLines.pop(0)
        for line in metadataLines:
          storedFile = line[40:]
          storedFileBase = os.path.split(storedFile)[1]
```

```
                    if storedFileBase == fnBase.replace("_xmipp_ctf_enhanced_psd.xmp", "_psdAt_1.xmp") \
                            or storedFileBase == fnBase.replace("_xmipp_ctf_enhanced_psd.xmp",
"_psdAt_2.xmp") \
                            or storedFileBase == fnBase.replace("_xmipp_ctf_enhanced_psd.xmp",
"_psdAt_3.xmp"):
                        shutil.copy(fnRoot, destRoot)
            print("Files copied to destiny")

    def prune(self, stackDir):
        metadataPath = open(os.path.join(stackDir, "metadata.txt"), "w+")
        lines = metadataPath.read().splitlines()
        lines.pop(0)
        nameFiles = []
        for line in lines:
            fileName = line [40:-1]
            nameFiles.append(fileName)

        for file in stackDir:
            for nameFile in nameFiles:
                nameFile.replace()
                if nameFile == file:
                    break
                file.replace(".xmp", "_ERRASE.xmp")

if __name__ == "__main__":
    if len(sys.argv) == 1:
        print("Usage: python prepareDataset.py <dirIn> <dirOut> <subsetNumber>
<importDataFlag(0/1)>")
    fnDir = sys.argv[1]
    stackDir = sys.argv[2]
    subset = int(sys.argv[3])
    dataFlag = int(sys.argv[4])
    stackDir = stackDir + "/"
    deepDefocus = DeepDefocus()
    allPSDs = deepDefocus.importCTF(fnDir, dataFlag)
    deepDefocus.downsampleCTF(allPSDs, stackDir, subset, dataFlag)
```

8.2 PrepareStack.py

```
import numpy as np
import os
import sys
import xmippLib as xmipp
from time import time


def prepareData(stackDir):
    metadataFile = open(os.path.join(stackDir, "metadata.txt"))
    metadataLines = metadataFile.read().splitlines()
    metadataLines.pop(0)
    Ndim = len(metadataLines)
    imagMatrix = np.zeros((Ndim, 512, 512, 3), dtype=np.float64)
    defocusVector = []
    i = 0
    for line in metadataLines:
        storedFile = line[39:]
        subset = int(line[30:38])
        defocus = int(line[10:21])
        img1Path = storedFile.replace("_psdAt_%d.xmp" % subset, "_psdAt_1.xmp")
        img2Path = storedFile.replace("_psdAt_%d.xmp" % subset, "_psdAt_2.xmp")
        img3Path = storedFile.replace("_psdAt_%d.xmp" % subset, "_psdAt_3.xmp")
        img1 = xmipp.Image(img1Path).getData()
        img2 = xmipp.Image(img2Path).getData()
        img3 = xmipp.Image(img3Path).getData()
        imagMatrix[i, :, :, 0] = img1
        imagMatrix[i, :, :, 1] = img2
        imagMatrix[i, :, :, 2] = img3
        defocusVector.append(defocus)
        i += 1
    imageStackDir = os.path.join(stackDir, "preparedImageStack.npy")
    defocusStackDir = os.path.join(stackDir, "preparedDefocusStack.npy")
    np.save(imageStackDir, imagMatrix)
    np.save(defocusStackDir, defocusVector)

if __name__=="__main__":
    stackDir = sys.argv[1]
    print("Preparing stack...")
    start_time = time()
    prepareData(stackDir)
    elapsed_time = time() - start_time
    print("Time spent preparing the data: %0.10f seconds." % elapsed_time)
```

8.3 Batch_DeepDefocus.py

```
#!/usr/bin/env python2
import cv2
import math
import numpy as np
import os
import string
import sys
import time
from time import time

batch_size = 128 # Number of boxes per batch

if __name__=="__main__":
    os.environ["CUDA_VISIBLE_DEVICES"] = "1"

    from keras.callbacks import TensorBoard, ModelCheckpoint
    import keras.callbacks as callbacks
    from keras.models import Model
    from keras.layers import Input, Conv2D, MaxPooling2D, BatchNormalization, Dropout, Flatten, Dense
    from keras.optimizers import Adam
    import tensorflow as tf
    from keras.models import load_model

    def constructModelOne():
        inputLayer = Input(shape=(512, 512, 3), name="input")
        L = Conv2D(16, (15, 15), activation="relu")(inputLayer)
        L = BatchNormalization()(L)
        L = MaxPooling2D((3, 3))(L)
        L = Conv2D(16, (9, 9), activation="relu")(L)
        L = BatchNormalization()(L)
        L = MaxPooling2D()(L)
        L = Conv2D(16, (5, 5), activation="relu")(L)
        L = BatchNormalization()(L)
        L = MaxPooling2D()(L)
        L = Dropout(0.2)(L)
        L = Flatten()(L)
        L = Dense(1, name="output", activation="linear")(L)
        return Model(inputLayer, L)
```

```
  def constructModelTwo():
    inputLayer = Input(shape=(512, 512, 3), name="input")
    L = Conv2D(16, (15, 15), activation="relu")(inputLayer)
    L = BatchNormalization()(L)
    L = MaxPooling2D((3, 3))(L)
    L = Conv2D(16, (9, 9), activation="relu")(L)
    L = BatchNormalization()(L)
    L = MaxPooling2D()(L)
    L = Conv2D(16, (5, 5), activation="relu")(L)
    L = BatchNormalization()(L)
    L = MaxPooling2D()(L)
    L = Flatten()(L)
    L = Dense(256, activation="relu")(L)
    L = Dropout(0.2)(L)
    L = Dense(1, name="output", activation="linear")(L)
    return Model(inputLayer, L)

  def constructModelThree():
    inputLayer = Input(shape=(512, 512, 3), name="input")
    L = Conv2D(16, (15, 15), activation="relu")(inputLayer)
    L = MaxPooling2D((3, 3))(L)
    L = Conv2D(16, (9, 9), activation="relu")(L)
    L = MaxPooling2D()(L)
    L = Conv2D(32, (5, 5), activation="relu")(L)
    L = MaxPooling2D()(L)
    L = Conv2D(64, (5, 5), activation="relu")(L)
    L = Flatten()(L)
    L = Dense(256, activation="relu")(L)
    L = Dropout(0.2)(L)
    L = Dense(1, name="output", activation="linear")(L)
    return Model(inputLayer, L)

  model = constructModel()
  model.summary()

  if len(sys.argv)<3:
    print("Usage: scipion python batch_deepDefocus.py <stackDir> <modelDir>")
    sys.exit()
  stackDir = sys.argv[1]
  modelDir = sys.argv[2]

  print("Loading data...")
  imageStackDir = os.path.join(stackDir, "preparedImageStack.npy")
  defocusStackDir = os.path.join(stackDir, "preparedDefocusStack.npy")
  imagMatrix = np.load(imageStackDir)
  defocusVector = np.load(defocusStackDir)

  print("Train mode")
  start_time = time()
  model = constructModel()
  model.summary()
  optimizer = Adam(lr=0.0001)
  model.compile(loss='mean_absolute_error', optimizer='Adam')
  elapsed_time = time() - start_time
  print("Time spent preparing the data: %0.10f seconds." % elapsed_time)

  callbacks_list = [callbacks.CSVLogger("./outCSV_06_28_1", separator=',', append=False),
             callbacks.TensorBoard(log_dir='./outTB_06_28_1', histogram_freq=0, batch_size=128,
                          write_graph=True, write_grads=False, write_images=False,
embeddings_freq=0,
                          embeddings_layer_names=None, embeddings_metadata=None,
                          embeddings_data=None),
             callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=5, verbose=1,
mode='auto',
                                             min_delta=0.0001, cooldown=0, min_lr=0)]
  history = model.fit(imagMatrix, defocusVector, batch_size=128, epochs=100, verbose=1,
validation_split=0.1, callbacks=callbacks_list)
  myValLoss=np.zeros((1))
  myValLoss[0] = history.history['val_loss'][-1]
  np.savetxt(os.path.join(modelDir,'model.txt'), myValLoss)
  model.save(os.path.join(modelDir,'model.h5'))
  elapsed_time = time() - start_time
  print("Time in training model: %0.10f seconds." % elapsed_time)
```

Madrid..........de ....de...........

D./Dña.: .... alumno de la Universidad San Pablo CEU de Madrid, da permiso para que su Trabajo Fin de Grado pase a estar depositado en la Biblioteca de la Universidad, dónde podrá ser consultado por cualquier persona que esté interesado en ello.

Y para que surta efecto donde proceda, firma la presente

Fdo:...

Licencia de autorización para incorporación de materiales en CEU Repositorio Institucional

D./Dña DNI/PASAPORTE----------------------------------

DOMICILIO EN---

MANIFIESTA:

Primero.- Su interés en que la obra/s que a continuación se detalla/n quede/n depositada/s en el REPOSITORIO de la FUNDACIÓN UNIVERSITARIA SAN PABLO-CEU, la FUNDACIÓN PRIVADA UNIVERSITAT ABAT OLIBA CEU y sus Centros académicos dependientes, en adelante CEU Repositorio Institucional.

El CEU Repositorio Institucional comprende en la actualidad los proyectos:

- OCW (Open Course Ware) de difusión de materiales docentes organizados por asignaturas.
- ReI: (Repositorio Institucional) de difusión de la producción científica, académica e institucional del CEU
- Así como cualquier otro proyecto de naturaleza similar que pudiera desarrollarse en el futuro con estos mismos fines o análogos.

Todos los materiales depositados estarán bajo la protección de licencia Creative Commons en los términos expuestos en el anexo II

Segundo.-

□ Ser titular de los derechos de propiedad intelectual de la obra/s depositada/s, como autor de la misma, y que esta es una obra original.

□ Ser cotitular de los derechos de propiedad intelectual de la obra/s depositada/s, y que cuenta con la autorización de los restantes cotitulares cuyos nombres aparecen en la obra depositada para hacer la cesión.

□ Que no existe una previa cesión a terceros de los derechos que en este documento se contemplan y que, en caso de existir una cesión previa, tiene la autorización de dichos titulares de derecho a los fines que en este acuerdo se establecen.

(Márquese con una x lo que proceda)

Tercero.-Que cede por el presente documento al CEU REPOSITORIO, y con el fin de dar la máxima difusión de la/s obra/s citada/s de forma gratuita y no exclusiva, y con fines de docencia e investigación, los derechos de archivo, digitalización, reproducción, difusión en línea y puesta a disposición electrónica, y transformación de la obra/s que a continuación se detalla y/o extractos de la misma con el objeto de hacer posible su utilización por todos los usuarios del Repositorio según las directrices que se establecen en el presente documento.

Cuarto.- que garantiza que este acuerdo no infringe ningún derecho de propiedad industrial, intelectual, derecho a la cita, al honor, intimidad o imagen o cualquier otro derecho de terceros.

OBJETO Y FIN DEL ACUERDO

Es la difusión, promoción cultural y puesta en marcha del Repositorio institucional que, sin ánimo de lucro y con fines de docencia e investigación, está llevando a cabo la FUNDACIÓN UNIVERSITARIA SAN PABLO-CEU, LA FUNDACIÓN PRIVADA UNIVERSITAT ABAT OLIBA CEU y sus Centros académicos dependientes, a través de sus Bibliotecas, haciendo posible su utilización por todos los usuarios del Repositorio y sus aplicaciones (OCW, ReI, entre otras).

CONDICIONES DE LA CESIÓN

- Sin perjuicio de la titularidad de la obra, que sigue perteneciendo al autor, éste entregará al CEU REPOSITORIO, en soporte electrónico o papel la documentación que le sea requerida para hacer efectiva la cesión de los derechos establecidos.

- Quedan excluidos de este acuerdo y reservados al autor todos los derechos que le correspondan, no previstos en el mismo.

- La FUNDACIÓN UNIVERSITARIA SAN PABLO-CEU, la FUNDACIÓN PRIVADA UNIVERSITAT ABAT OLIBA CEU y sus Centros académicos dependientes, en virtud del presente acuerdo, adquieren el derecho de poder difundir las obras objeto de la presente cesión de forma total o parcial, siempre haciendo constar los datos necesarios para la identificación inequívoca de las mismas (nombre y apellidos del autor, titulo de la misma, nº ISBN / ISSN y Depósito legal, entre otros)

- CEU Repositorio Institucional podrá:

    1. Realizar las transformaciones necesarias en la/s obra/s con el fin de su adaptación a cualquier tecnología necesaria para su incorporación a Internet, utilización de la obra en formatos electrónicos, o adaptación a cualquier sistema de seguridad y protección.

    2. Reproducir la obra en soporte digital e incorporarla a una Base de Datos.

    3. Almacenar la obra en servidores a los efectos de garantizar su seguridad y conservación

    4. Comunicarla y ponerla a disposición de los usuarios/público a través de un archivo abierto institucional, accesible de forma libre y gratuita a través de Internet.

- Los Centros académicos dependientes de la FUNDACIÓN UNIVERSITARIA SAN PABLO-CEU y la FUNDACIÓN PRIVADA UNIVERSITAT ABAT OLIBA CEU informarán a los usuarios del CEU REPOSITORIO sobre la Licencia que protege la obra, no asumiendo responsabilidad alguna sobre la forma y manera en que los usuarios hagan uso posterior de los documentos cedidos.

- El autor, cuando deposite en el CEU REPOSITORIO una obra no publicada, lo realizará bajo los términos de una licencia “Creative Commons”.

- El CEU REPOSITORIO aconsejará al autor sobre el uso de las licencias “Creative Commons” y explicará el funcionamiento de las mismas.

- El autor se responsabiliza de la veracidad de los datos, la originalidad de la/s obra/s, el contenido de las mismas y el goce en exclusiva de los derechos cedidos.

- El autor exonera al CEU REPOSITORIO de toda responsabilidad que pudiera surgir como consecuencia de reclamaciones realizadas por parte de terceros en relación con los mismos, incluyendo las indemnizaciones por daños y perjuicios que pudieran ejercitarse contra la FUNDACIÓN UNIVERSITARIA SAN PABLO-CEU y la FUNDACIÓN PRIVADA UNIVERSITAT ABAT OLIBA CEU así como asume la responsabilidad en el caso de que la Institución fuera condenada por infracción de derechos derivada de las obras objeto de la cesión.

- El autor renuncia a cualquier reclamación frente a la FUNDACIÓN UNIVERSITARIA SAN PABLO-CEU y la FUNDACIÓN PRIVADA UNIVERSITAT ABAT OLIBA CEU por el modo de utilización de las obras que hagan los usuarios cuando no se ajuste a la legislación vigente.

- Ambas partes se comprometen a comunicar a la otra, cuando tenga conocimiento de ello, la existencia de cualquier reclamación de un tercero relacionada con los documentos objeto de la cesión, así como de cualquier infracción de los derechos de propiedad intelectual.

DURACIÓN

Esta cesión tendrá una duración de ---5------- año/s a contar desde la fecha de su firma. A falta de revocación expresa, comunicada de forma fehaciente, con un mes de antelación a la fecha de su vencimiento, el plazo de esta cesión se entenderá prorrogado por periodos sucesivos de --5--------- (años)

POLITICA DE RETIRADA Y EXCLUSIVIDAD

El Autor podrá solicitar la retirada de la obra del CEU REPOSITORIO por causa justificada, sin perjuicio de que la decisión última sobre dicha solicitud corresponda al CEU Repositorio. Asimismo, el CEU REPOSITORIO podrá retirar la obra, previa notificación al autor, en supuestos justificados o en caso de reclamaciones de terceros.

Los documentos que se hayan incluido en el CEU REPOSITORIO por personas vinculadas en su momento a la FUNDACIÓN UNIVERSITARIA SAN PABLO-CEU y la FUNDACIÓN PRIVADA UNIVERSITAT ABAT OLIBA CEU se regirán por lo establecido en el párrafo anterior.

El autor tiene, derecho a introducir otras copias en repositorios de otras instituciones bajo acuerdo de no exclusividad.

JURISDICCIÓN Y LEY APLICABLE

El presente documento se regirá de conformidad con la legislación española en todas aquellas situaciones y consecuencias no previstas en forma expresa en el presente documento y, en concreto, de acuerdo con las prescripciones de la legislación española sobre propiedad intelectual vigente y demás legislación aplicable.

Para la resolución de cualquier controversia que pudiera surgir en la aplicación e interpretación del presente documento se realiza un sometimiento a la jurisdicción de los Juzgados y Tribunales de Madrid Capital, con expresa renuncia a cualquier otro fuero que pudiera corresponder

En Madrid, a de de

Fdo.: D.

ANEXO II
Creative Commons
(Licencia de Uso)

El autor como titular de los derechos de propiedad intelectual puede establecer determinadas condiciones al uso que los usuarios hagan de su trabajo, por medio de las licencias, protegiendo su obra de usos indebidos si está en acceso abierto.

Licencias Creative Commons

http://es.creativecommons.org/proyecto/

Creative Commons es una corporación americana sin ánimo de lucro. Si quieres conocerla mejor puedes visitar su web.

http://es.creativecommons.org/licencia/

La licencia de los materiales incorporados al CEU Repositorio Institucional es :

cc BY NC SA **Reconocimiento – No Comercial – Compartir Igual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

ANEXO I
(Relación de materiales que se incorporan a CEU Repositorio Institucional)

Artículos

Capítulos

Comunicaciones

Materiales docentes

En , a de del 2…..

Fdo.:D../Dña. --

ACTA DE EVALUACIÓN DEL TFG

Tribunal-Miembros	**1:** **2:** **3:**
Título TFG	
Estudiante	
Director TFG	

Los Resultados de Aprendizaje (RA) que componen la rúbrica aseguran la adquisición de las competencias comprometidas en el TFG del GIBM que se detallan en el Anexo.

Se valorará de 0 a 100 el nivel de adquisición del Estudiante de los Resultados de Aprendizaje tras la revisión del TFG y su defensa oral:

RÚBRICA DE EVALUACIÓN	**Miembro 1 Tribunal**	**Miembro 2 Tribunal**	**Miembro 3 Tribunal**	**Valoración Media**
RA1: Study and analyze the scientific, technical and socio-economic viability of a Biomedical Engineering project and understand the relationships between science, technology and business.				
RA2: Understand the principles of scientific methodology and apply it to solving problems in the field of Biomedical Engineering.				
RA3: Knowing how to find and interpret information in major scientific databases of engineering and medicine.	.			
RA4: Write, present and defend a research paper.				
CALIFICACIÓN GLOBAL TFG (0 -100) **CALIFICACIÓN GLOBAL NORMALIZADA (0-10)**				

Firma:	**Firma:**	**Firma:**
Miembro 1 Tribunal:	Miembro 2 Tribunal:	Miembro 3 Tribunal:

En Madrid a ____de _____de 20___

ANEXO: Competencias del Trabajo Fin de Grado del GIBM

Los Resultados de Aprendizaje que componen la rúbrica de evaluación del TFG en el GIBM, aseguran la adquisición de las competencias comprometidas en su memoria de verificación:

	COMPETENCIA ESPECÍFICA DEL TFG EN GIBM
CE-34	Gestionar y coordinar un proyecto de Ingeniería Biomédica empleando herramientas comunes de gestión de proyectos y saber realizar, tramitar y gestionar documentos técnicos.
CE-35	Estudiar y analizar la viabilidad científico-técnica y socioeconómica de un proyecto de Ingeniería Biomédica y conocer las relaciones entre ciencia, tecnología y empresa.
CE-36	Conocer los principios de la metodología científica y aplicarlos a la resolución de problemas en el campo de Ingeniería Biomédica.
CE-37	Saber buscar e interpretar información en las principales bases de datos científicas, tanto de la rama de la ingeniería como de la medicina.
CE-48	Aplicar los conocimientos adquiridos durante los estudios a la resolución individual de un problema en el ámbito de la Ingeniería Biomédica.

	COMPETENCIAS BÁSICAS DEL TFG EN GIBM
BAS-2	Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
BAS-3	Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
BAS-4	Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
BAS-5	Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.