LUND UNIVERSITY

FACULTY OF SCIENCE

X-Ray Fluorescence of Metal Halide Perovskites

Samuel Narciso Silva

Thesis submitted for the degree of Bachelor of Science
Project duration: 2 months (15 hp)

Supervised by Prof. Jesper Wallentin and Dr. Lucas Marçal

Department of Physics
Division of Synchrotron Radiation Research
January 2022

Abstract

This thesis has the aim to provide a framework for analyzing X-ray fluorescence (XRF) data obtained from German Electron-Synchrotron Group (DESY). The theoretical analysis shows that XRF competes with Auger emission, and works mainly for elements with high atomic number. Here, incident X-rays of 13.7 keV, generated by the synchrotron radiation, impinges the material, and fluorescent X-ray spectra are recorded. For the analysis, custom Python scripts have been written to manage large datasets and plot results. They are the major contribution to this work, as they reveal how data is organized and how it should be interpolated, for the XRF maps to be plotted. For the fitting of the individual element data, PyMCA, a Python-based XRF analysis software, is used. The software is able to accurately fit the counts as they are measured, and to assign them to specific elements. From this, an elemental map is provided. The maps shown include K, Br, I, and Pb, while in spectra, Cs is considered as well. The material investigated is a perovskite, $CsPbBr_3$, which can be used to enhance efficiency of solar cells. This material is combined with a KI solutions in different ways, where K is used to improve the perovskite light absorption, and is detected through XRF. The obtained data indeed show presence of K, where counts depend on how K is deposited. Furthermore, individual pixel fit reveals that Br and I tend to segregate when material is exposed to light, as I counts drop. Lastly, PyMCA is able to obtain element counts more accurately that region-of-interest plot. All these effects and similar, can be examined using PyMCA, and custom Python scripts that organize, and interpolate data. Each pixel can be analyzed separately, showing that big data analysis can be a useful tool in science.

Acknowledgment

I would like to thank my main supervisor, Prof. Jesper Wallentin, and my assistant supervisor, Dr. Lucas Marçal for helping me complete this thesis, for their knowledge of the field, and their patience with me, during the process of writing the thesis and compiling the data. I would also like to thank personnel from DESY: Michael Stuckelberger, and Christina Ossig, for providing information about their beam line, providing the data and helping me understand PyMCA and how data should be handled. Finally, I would like to thank my husband Stefan, for helping in proof-reading the text, as well as for being the support during thesis writing.
Thank you all!

Samuel.

Table of Acronyms

Br Bromide

Cs Cesium

DESY Deutsches Elektronen-Synchrotron [German Electron Synchrotron]

EDXRF Energy Dispersive X-ray Fluorescence

EM Electromagnetic

ESRF European Synchrotron Radiation Facility

HDF Heterogeneous Data Format

I Iodine

K Potassium

MCA Multi-Channel Analyzer

Pb Lead

ROI Region-Of-Interest

SDD Silicon Drift Detector

Si Silicon

XRF X-ray Fluorescence

Contents

Chapter 1

Introduction

1.1 Background

This thesis describes the use of X-rays and their interaction with matter. This interaction is mostly seen by the atoms in matter, and allows us to identify the elemental composition of a material [1, 2]. In principle, X-rays are no different than any other "light", but the energy they carry is much higher than, for instance, radio or visible light. The photon energy, ε, is related to the oscillation frequency of an electromagnetic (EM) wave, through Plank relation, where $\varepsilon = h\nu = \hbar\omega$, where ν is frequency, $\omega = 2\pi\nu$ is the angular frequency, h is Plank constant, and $\hbar = h/2\pi$ is reduced Planck constant [3]. Another important physical quantity is the wavelength, λ, which represents the distance that an EM wave travels before reaching the same electric field. The relationship between the frequency and the wavelength is given as $\nu = c/\lambda$, where c is the speed of light, a universal constant. A representation of this dependency is given in Fig. 1.1.

There are many ways in which photons can be used, produced, or annihilated. Typically, because of wave-particle duality, they can interact with matter in various ways, producing many different phenomena. When a photon transfers its energy to the material, and is annihilated, we call that absorption. When a photon is created and emitted in the material that releases energy, it is known as emission, or fluorescence, and when photon and another particle interact (with or without energy exchange), we call that scattering. All these effects are important in understanding the topic of this thesis, and they will be used to describe the findings of this work. Here, the interaction of X-rays and matter will be looked into in greater detail, as their specific energy (wavelength) allows them to probe into atomic structure itself. This is due to X-rays having a wavelength between 0.01 nm and 10 nm, a range that corresponds to the distance between electrons

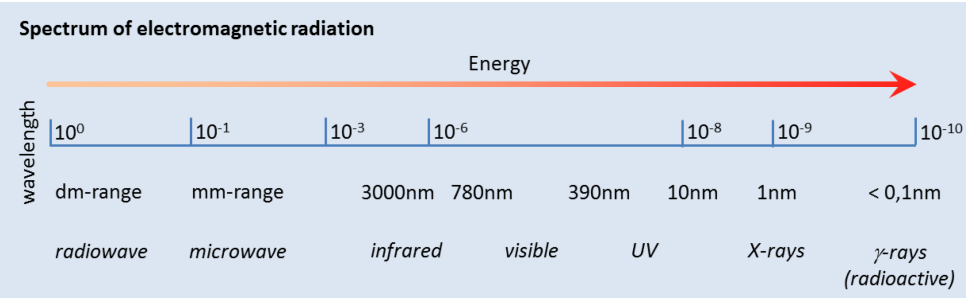

Figure 1.1: *Electromagnetic radiation spectra [4]*

in the atomic shells, up to the diameter of nanocrystals.

An important aspect for this work is the generation and manipulation of such small-wavelength high-energy X-rays. Generating these waves is not simple, and requires many steps. For this purpose, facilities known as particle accelerators are used nowadays. They have many advanced instruments that enable the energy buildup in an electron beam, which can, in turn, emit high-energy X-ray photons. Furthermore, this requires instruments that can track the exact energies of this electron beam, as well as auxiliary setups that can also focus the X-rays on a very narrow spot where a sample/material is studied.

1.2 Motivation

Obtaining renewable energy in an efficient way is very challenging in the world today. One significant contribution to renewable energy is solar energy, which is harvested using solar cells. These cells are made with semiconductor materials, where their energy bandgap can be suitable for light absorption. The most used material is silicon (Si), but it has limitations in conversion efficiency, or the amount of light that it can absorb w.r.t the total incoming light [5]. Furthermore, Si needs to be perfectly crystalline, and the process of obtaining crystalline Si is expensive and requires specialized facilities. Therefore, alternative materials are investigated in order to improve the efficiency of the solar cells. The aim is to have a material that can be relatively easily applied on a surface, and that has a high conversion of light into electricity, without many losses. Such material should also be used together with Si, in order to further boost the conversion efficiency, but they should not raise production cost significantly.

One of the materials that may be interesting are metal-halide perovskites [6]. Metal-halide perovskites are specific materials, which have specific crystal structure that allows them to efficiently absorb light and convert them into electricity, as both the electron-rich (anions) and electron-poor (cation) elements are present within the material [7]. These materials show great potential in enabling higher energy output at reduced material usage, and can potentially be the disruptive technology that will enable more efficient clean energy sources in the future.

There are many aspects of perovskites that can be mentioned, such as that by changing their composition, they are able to absorb more or less of the specific portion of the electromagnetic spectra. This may be important when incorporating perovskites with Si, since Si has limitations at absorbing all of the electromagnetic spectra that the Sun is producing. However, when perovskites are deposited on Si surface, many issues appear which degrade the overall solar cell performance. An promoter is needed, to make the surface as smooth as possible, and for this, potassium (K) is found to be a good candidate [8]. K-enhanced perovskites may even be a safer option, since perovskites typically use lead (Pb), which is a toxic material. Investigating how K incorporates into the perovskite is technologically significant, and is investigated in part of this work. Aside from this, selecting an anion material is important as well. In this case, bromide (Br) and iodine (I) are typically used. However, introducing both Br and I in a perovskite can also have advantages in enhancing the efficiency. Such perovskites will also be affected by the surface they are deposited on, with Br segregating from I [9]. It is important to understand the process of segregation, as it can be incident light that segregates the material.

For technological relevance, thin perovskite layers are investigated, and an area-wide imaging method is needed to asses how different elements incorporate in the overall perovskite structure. One way of investigating is through observation of fluorescent X-rays.

The *X-ray fluorescence* is a method that helps us to investigate quantitatively, and qualitatively, the composition of the material in the sample that is being investigated, independent of its physical state, which means that the method is suitable for liquids, solids, powders, etc. In simple terms, the X ray fluorescence diffraction allows one to determinate: "What elements is the sample composed of?" and "What is the concentration of the elements that the sample is composed of?".

1.3 The work

The fluorescent X-rays, emitted from the material, will have characteristic energies, corresponding to the energy difference of transition between electronic states in the particular elements. These emitted X-rays are also known as characteristic X-rays. A technique that analyzes the characteristic X rays through the identification and measurement of their energies, is named Energy-Dispersive X-Ray Fluorescence (EDXRF). An energy-dispersive spectrum shows the amount of detected characteristic X-ray photons versus their energy. By counting the number of photons for each of the discrete energies, corresponding to certain energy transitions, we determine the materials present and their concentration. With those data, a diagram can be created using a programming software, such as Python, or MatLab, to plot the spectra where different peaks correspond to different elements. In this way, the areas under the peaks give us the total amount of photons that were detected during the measurement time. The total number of photons is proportional to the amount (concentration) of the particular element for which the emission of photons of that particular energy has occurred, which can be expressed per microgram of the chemical element per gram of sample (µg/g).

In this thesis, metal-halide perovskites, which are scanned using EDXRF, are analyzed. The obtained fluorescence data is fed into Python-based software, PyMCA, developed by researchers at European Synchrotron Radiation Facility (ESRF), which allows for quick identification of the materials that are present on the sample, as well as understand their concentration throughout the scanned sample area. This is achieved by performing a semi-automated fit to the measured cumulative data, and a batch fit which allows for material identification at each scanned pixel. The core of this thesis is to further enhance this software with custom Python-based scripts, that improve data visualization. The objective is to aid further research on perovskites and their potential application in solar cell industry.

Chapter 2

Theory

Considering electrons in an atomic shell, as mentioned before, placed in their respective shells, the X-ray interaction phenomenon will be the one that can directly excite electron at all energy levels within the atomic structure. Fig. 2.1 shows a simplified schematic of an atomic shell, including continuum, where electrons would become free, if they have sufficient energy. Primarily, the electron is the lowest shell (the deepest energy level) absorbs the X-ray photon, and is excited to to the continuum, so that it escapes the atom, leaving a vacancy behind. The atom is not in the energy minimum any longer, and another electron from higher energy orbital will occupy the vacancy. Following this transition, there are two possibilities, that compete with each other. The first possibility is the excess energy gets transferred to a third electron, which sits in a higher atomic orbital, that can now escape as well. This is called Auger effect [10]. In the second possibility, as electron moves to the vacancy on the lower energy level, it emits a characteristic X-ray photon. This effect would then be X-ray fluorescence. Depending on the origin of the electron that occupies the lower atomic orbital, transitions are labeled, as shown in Fig. 2.1: K_α for L-to-K transition, or K_β for M-to-K transition.

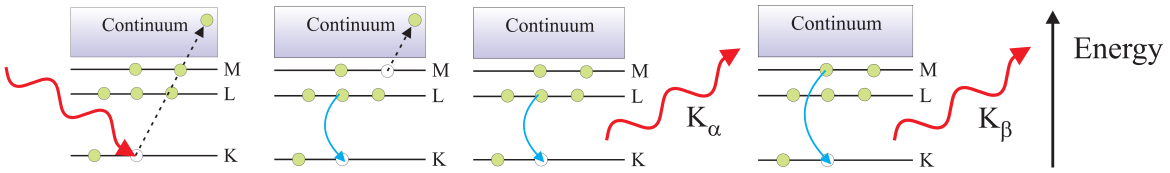

Figure 2.1: *Photon absorption and subsequent effects (left to right): Auger electron emission, and X-ray Fluorescence with varying photon energies [11]*

X-ray fluorescence (XRF) and Auger spectroscopy are competing processes, and are defined by several factors, such as atomic number of the material, or the incident photon energy. But they both star with absorbing X-ray energy before emitting it again. Therefore, we will approach this problem with a brief focus on the photon absorption, followed by photon emission, in the XRF process.

2.1 X-ray Fluorescence

Following the absorption of a high-energy X-ray photons, the material emits characteristic X-rays, which was dependent on the element that was irradiated. What is considered here is that an incident beam of X-rays is only at a single energy (monochromatic), which simplifies the analysis [12]. The beam hits a thin sample (thickness h), with intensity I_0, and the energy E_0. As the rays penetrate the surface of the material, they get absorbed, and the intensity drops, according to Lambert-Beer's law [10]:

$$I = I_0 \mathrm{e}^{-\mu_s \rho x} \tag{2.1}$$

where μ_s is the mass absorption coefficient of the entire sample at the specific energy E_0, ρ is material density, and x is the path length, or penetration depth of the X-ray beam into the material. Each material may have a different absorption coefficient, depending on their absorption cross-section, or their affinity to absorb X-rays. This parameters depend heavily on incoming X-ray energy. Therefore, the incoming X-ray intensity, that can be absorbed by the unit volume, defined as ρdx, and produce the fluorescent X-ray, is depending on the concentration of the element, C, and its absorption coefficient, μ, given as:

$$dI = I_0 \left(C \mu \rho dx\right) \mathrm{e}^{-\mu_s \rho x}. \tag{2.2}$$

In order for fluorescence of the particular X-ray photon to occur, the probability of ejecting the lowest (K shell) energy electron needs to be higher than the probability of ejecting electrons from higher atomic shells (L or M shells). Furthermore, the probability of emitting XRF photon needs to be higher than emitting Auger electron as well. First, the probability of emitting K shell electron is defined by the K shell absorption jump ratio, or:

$$J_K = \frac{r_K - 1}{r_K} \tag{2.3}$$

where r_K is the absorption jump at the absorption edge of the K shell electron. This means that there is a steep increase in the absorption ratio when the incoming X-ray energy is equal or slightly higher than the binding energy of the K shell electron. The ratio between the two levels is defined as $r_K = \mu_{max}/\mu_{min}$, where μ_{max} and μ_{min} are maximum and minimum absorption coefficients, respectively. The absorption jump is illustrated in Fig. 2.2(a).

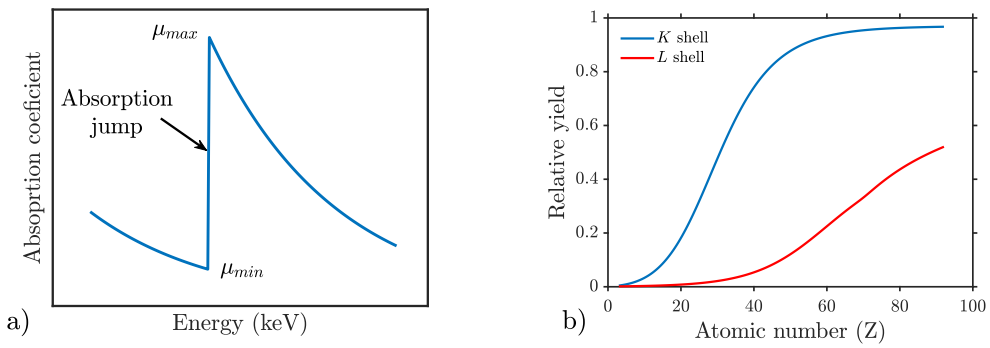

Figure 2.2: *a) Illustration of absorption coefficient VS X-ray energy, showing absorption jump, and minimum and maximum absorption coefficient values. b) Relative yield of X-ray fluorescence for K and L shell electrons [12].*

The second parameter defining the emission of X-ray photons is the probability of XRF in competition to Auger electron emission, usually given as ω_K parameter. Since Auger emission relies on the fact that electrons in higher atomic shells will be expelled as a product of de-excitation of other electrons to the K shell, the process is strong for elements with low atomic number Z. As the atomic number gets higher, the fluorescence becomes the dominating emission effect. Therefore, the fluorescence yield increases significantly at high Z, as shown in Fig. 2.2(b). Even fluorescence yield of L shells increases for large elements. The comparison between Auger and XRF yields is illustrated in Fig. 2.3, showing strong Auger emission at low Z, and high XRF at high Z. Since XRF does not rely on electron emission, it does not depend on electron conductivity. Therefore, it can be used to image insulators, as well as conductors. The charge buildup in dielectric interferes with the Auger electrons, and makes it difficult to analyse the Auger response, while such limitation does not exist for XRF [13].

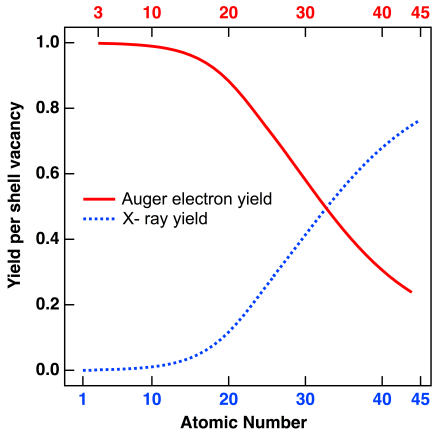

Figure 2.3: *X-ray Fluorescence yield, as compared to Auger emission [14].*

The important point for XRF is the notation, which helps identify the characteristic lines. In case of the first higher orbital, it is denoted as α, while the second higher orbital would have the label β. A detailed illustration of the possible transitions re shown in Fig. 2.4, while the naming convention is known as Siegbahn notation [15]. Possible transitions are, of course, governed by selection rules, allowing for specific change in orbital and spin-orbital angular momentum [ref]. The full table with transitions (emission lines) for different elements is provided in Appendix A.

Finally, the probability that a specific XRF transition will occur represents a portion of the total XRF probability. If, for instance, if we consider only $K\alpha$ transition, as compared to other K shell transitions, the probability is $g_{K\alpha} = I(K\alpha)/(I(K\alpha)+I(K\beta))$. Taking all three effects into account, the probability for XRF emission versus incident photon, over Auger electron emission and other X-ray fluorescence transitions is known as excitation factor, and is given as:

$$Q = J_K g_{K\alpha} \omega_K. \tag{2.4}$$

Following the emission of the fluorescent X-ray photon, the specific atom that emits the photon acts like a point source, and it will emit photon in all directions with equal probability. However, we can only detect those X-ray photons that land onto the detector, which does not cover the entire space, but only a small area. First, the X-ray photon intensity is attenuated similarly to incident photon, according to $e^{-\mu'_s \rho x}$, where μ'_s is the attenuation coefficient for the fluorescent X-ray. Furthermore, the fraction of the photons

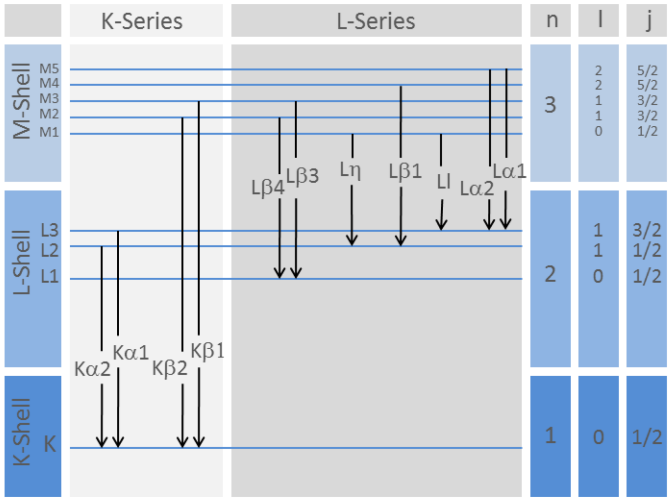

Figure 2.4: *Possible XRF transitions for the first three atomic shells [4]*

reaching the detector is $\Omega/4\pi$, where Ω is the detector solid angle. If the detector is placed at the distance d from the XRF source, and it is a circular shape, with radius a, then the solid angle is $\Omega \approx \pi d_2/a^2$. Thus, the total fluorescence intensity from the specific element, and for the specific transition, will be [12]:

$$I_{XRF} = I_0 C J_K g_{K\alpha} \omega_K \left(\frac{\Omega}{4\pi}\right) \frac{\mu}{\mu_s + \mu_s'}. \tag{2.5}$$

For thin sample, the eq. 2.5 is slightly different, and it can be seen as a fraction of the intensity of the thick sample. Similarly, the small percentage of the intensity is spent of secondary and tertiary emission. Secondary emission occurs when fluorescent X-ray photons are absorbed by neighboring atoms, which then produce their own fluorescent X-rays. When secondary XRF repeats the same process, a tertiary XRF emission occurs, but the probability for this emission is very low.

2.2 X-ray Sources - Synchrotron Accelerators

Synchrotron radiation is an electromagnetic radiation emitted by charge moving at relativistic speeds (i.e., close to the speed of light) along a curved path. In particular, this condition applies to particles circulating in electron or positron accelerators. The name of this radiation derives from a specific type of accelerator, the electron synchrotron [11, 16].The system typically consists of a charged particle source, and a linear accelerator segment, which then feeds the storage ring. In the storage ring, these charged particles travel with constant energy. The radius of curvature of the storage rings is of the order of several centimeters to tens of meters. The electrons accelerated inside the synchrotron have high kinetic energy and, consequently, produce light, which has intensity in the range of million times than that of the Sun. Electrons are then filtered out to specific energetic values that each workstation or so called "beam" will used to perform further experiments using only electrons with a specific energy.

During the process of deflection, the electrons in the storage ring pass thorough straight sections, where they are exposed to an oscillating magnetic field, which electrons follow. Devices placed in these straight sections are called undulators, or wigglers. As a result

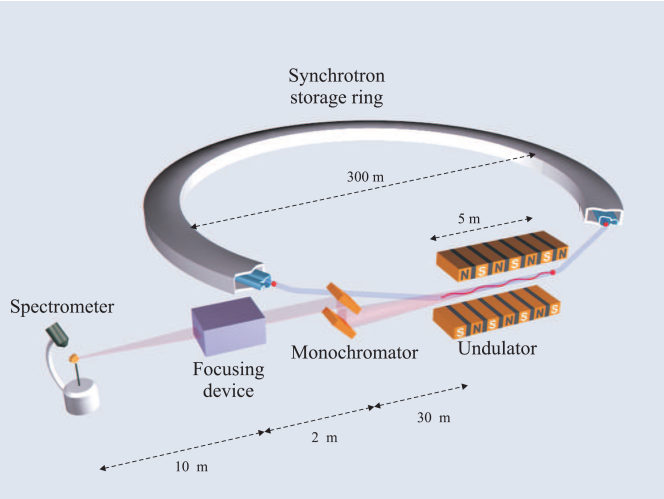

Figure 2.5: *Illustration of synchrotron radiation using undulators [11]*

of the oscillating magnetic field, electrons start emitting high-energy electromagnetic radiation, which are known as X-ray photons [17]. The magnets are located tangential to the orbit. This electromagnetic radiation (X-ray photons) produced by the synchrotron is emitted in a narrow cone, where it hits the sample that is being investigated. A schematic example of such a system is shown in Fig. 2.5. The X-ray delivery system may also include the monochromator, and a focusing device. The monochromator can filter out unwanted incoming wavelengths, as it is usually built from a material that allows diffraction of X-rays with specific wavelengths. Using the principle of Bragg diffraction ($d \sin \theta = n\lambda$), the crystal with known lattice constant is placed at specific angle, thus allowing only those wavelengths that are fulfilling Bragg condition to pass through. Finally, the focusing device is used to shape and concentrate the beam, in order to reduce the spot size.

When X-rays read the measurement station (or beam), they can be filtered, in order to reduce the photon intensity. The filtering enables the detector not no get saturated while capturing data. After the X-rays hit the sample, the XRF will be produced, and it will travel to the detector. Collimators, or beam limiting devices, are placed between the sample and the t-detector, in order to focus the beam, by passing, or directing. those XRF photons that have a travel path in the general direction towards detector. The measurement system is shown in Fig. 2.6.

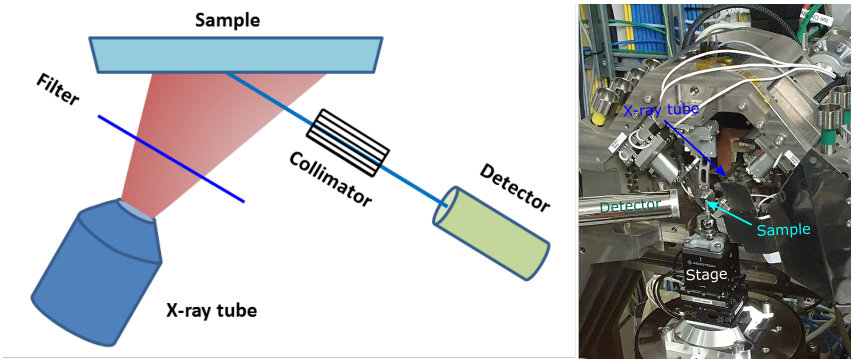

Figure 2.6: *Left: a simplified illustration of Energy Dispersive XRF, including a filter and a collimator between the X-ray source, sample, and a detector [4]. Right: setup of the P06 beam line.*

2.3 X-ray Detectors

The method, shown in Fig. 2.6, is used to capture XRF photons. For that purpose, a semiconductor detector, alongside the associated electronics, is used to sort and count each photon that comes in, according to their energy. The semiconductor detectors are also known as Silicon Drift Detectors (SDD), shown in Fig. 2.7 [18]. The SDD material is high quality Si, which has minimal defects. The anode and cathode are designed in such a way that, when voltage is applied, the entire Si is depleted of charges. When the photon hits the detector surface, it will create a number of electron-hole pairs, which then drift to their respective electrodes. Electrons travel along the path where depletion areas meet, so they are guided to the anode. The time delay between the hit and the detection of the incoming electrons gives information about the exact position where XRF photons landed, and it allows to count photons that have arrived simultaneously. The generated charge is proportional to the absorbed X-ray energy. The field-effect transistor, placed on the back of the anode, is sensitive to the input charges and converts them into voltage (integrates them), where the voltage level determines the photon energy. As there can be random variation in the number of charges that each X-ray hit produces, the energy peak spectrum is broad.

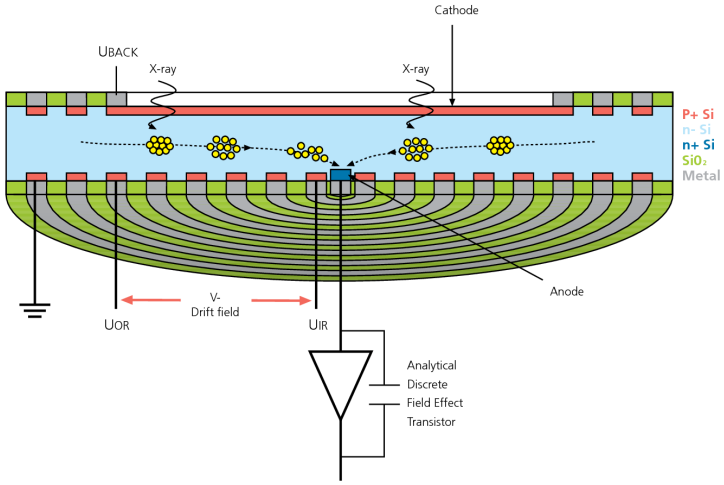

Figure 2.7: *Schematic illustration of Silicon Drift Detector (SDD) [19].*

Following the integrator stage, the output voltage level can only increase, which means that is needs to be reset. Otherwise, it will reach saturation, and X-ray photons would not be detected. The level change in the integrator stage is then taken by the amplifier, which generates a voltage pulse. The time constant of this pulse determines the energy resolution of the XRF photons. The system, following the detector, is known as Multi-Channel Analyzer (MCA), and it converts analog pulses into digital output [20]. It is able to detect and process up to 100 000 pulses per second, allowing for fast data capture and processing. If the number of photons reaching the detector is too high, the MCA will not be able to accurately capture and count all hits, which is why a filter, or an increased distance between the sample and the detector is needed, to prevent overloading of the detector. Once data stream is taken out from the system, it can be processed further, as is described in the following chapter.

Chapter 3

Methods and Tools

This section will describe the methods used to obtain and visualize the XRF data. The main principle behind the synchrotron accelerator usage in collecting data from material surfaces, as well as data subsequent processing, will be described. The main part of this work is PyMCA, a software tool that enables fast and accurate peak identification, which results in material composition of the specific element on the sample surface. Measured data is obtained from *Deutsches Elektronen-Synchrotron* (DESY) beamline *P06*, in Hamburg, Germany.

3.1 Energy-Dispersive X-ray Fluorescence Spectra

Following the detection of the X-rays, the incoming data stream is converted into counts, which will form the XRF spectrum, as illustrated in Fig. 3.1. The spectrum consists of peaks which correspond to specific fluorescent X-ray transitions that all the elements in the sample have produced. The advantage of using energy-dispersive X-ray fluorescence (EDXRF) imaging is that the detector is detecting different photon energies, which correspond to different transitions. This way, many elements can be sampled at the same time, as indicated by many peaks in Fig. 3.1. The channel number corresponds to XRF energy, where each channel is approximately 10 eV. The incident photon energy for all runs presented in this thesis is 13.7 keV, and therefore, 1400 channels are shown.

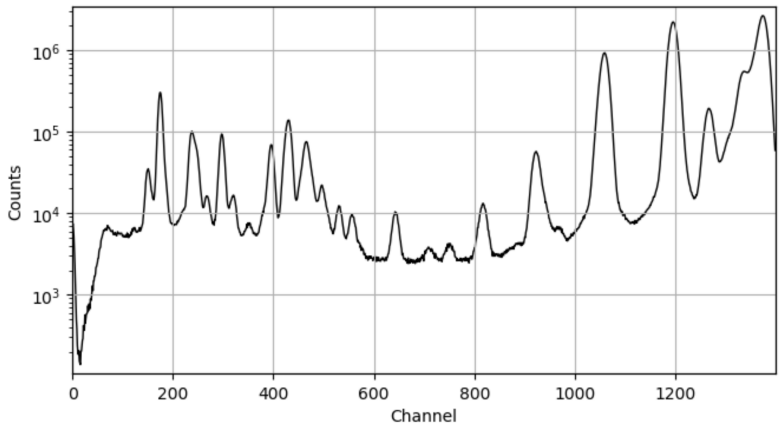

Figure 3.1: *A typical cumulative energy-dispersive XRF spectra, showing number of hits (counts) per energy.*

An important point is that the data is captured as a stream, where the step motors move the sample along the scan path. As the scan is moved and the data is recorded at each point, at the places where scan direction is changed and the motor slows down and accelerates again is being recorded a few times instead of only once. This makes it difficult to reconstruct the fluorescence map, and thus the data is interpolated with a redefined position map, to correct for this error. Following the interpolation, software can then read the fluorescence map in the correct way. Note that each pixel on the map contains a similar XRF spectra, as shown earlier.

The spectra also has wide (>350 eV) spectral lines for specific transitions. The disadvantage of EDXRF is that the spectral lines are wide, or rather, that the energy resolution is not as high, due to random charge generation in the detector, as stated in the Chapter 2. This needs to be taken into account when fitting the elemental map to each spectral line, as there may be many lines from more than one element which combine to form a single wide peak. For the highest energies, the wide peaks are not originating from any element, but they exist because the incident X-ray photons have scattered from the sample surface and travelled to the detector, where they are counted. They need to be modeled separately. For a more accurate peak detection, a wavelength-dispersive X-ray fluorescence can be used, but it is limited in observing only single element at a time, which is not useful for complex materials, such as metal-halide perovskites.

3.2 PyMCA

A very important MCA tool, used also in this thesis, is PyMCA, a software developed by the European Synchrotron Radiation Facility (ESRF) [21]. An example of a PyMCA window is shown in Fig. 3.2. It allows us to analyse data from XRF scans and determine quantitatively and qualitatively the chemical composition of the sample. It is a Python-based software that enables data visualisation and the dispersion of energy peaks of the XRF spectra. So far, it is considered the best software for the analysis of XRF data.

In order for PyMCA to perform the analysis of the data from the beamline, the data needs to be unpacked from many databases which are generated in the beamlike,

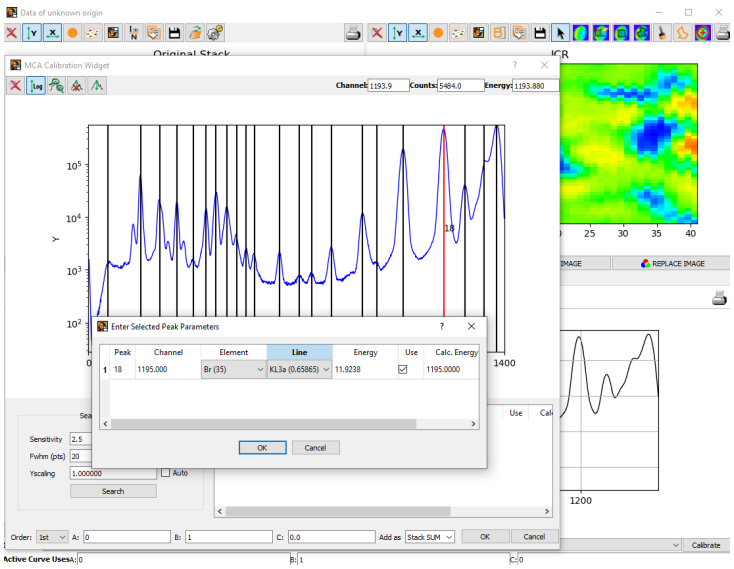

Figure 3.2: *XRF spectra calibration routine for peak fitting in PyMCA.*

and stored as one file, with HDF5 file type. This data is packaged using Python script shown in Appendix B1, consisting a major portion of the thesis work, as interpreting the incoming data is essential in generating the HDF5 files. PyMCA then takes this data set, and displays the cumulative spectra, with all the pixel counts summed up. This cumulative spectra than needs to be calibrated since the spectra peak counts are shown as a function of MCA channels, and not energy. Based on the elements present in the sample, one can attribute a characteristic peak to the expected XRF line of the element, as shown in Fig. 3.2. This should be repeated for as many peaks as can be identified, to help PyMCA accurately place peaks in energy. The software will than calibrate the spectra peaks in function of counts per energy.

3.2.1 Cumulative and batch fitting using PyMCA

After the calibration, PyMCA is configured to perform a fit that would overlap with the measured cumulative data. The configuration of the fit may include the incoming and outgoing angle of the X-rays, the distance between the detector and the source from the sample, the position of the background, with respect to the peaks, or the incident X-ray photon energy. The most importat is the peak selection, since one needs to estimate all the peaks and their counts in the cumulative fit, and all peaks should be taken into account. Several iterations are needed, with fine adjustment of fitting conditions, in order to minimize the error between the measured and fit data. Fig. 3.3 shows completed fit procedure on the cumulative XRF spectra (single-pixel spectra is shown later), where nearly all peaks are considered, accounting for a total of 18 elements.

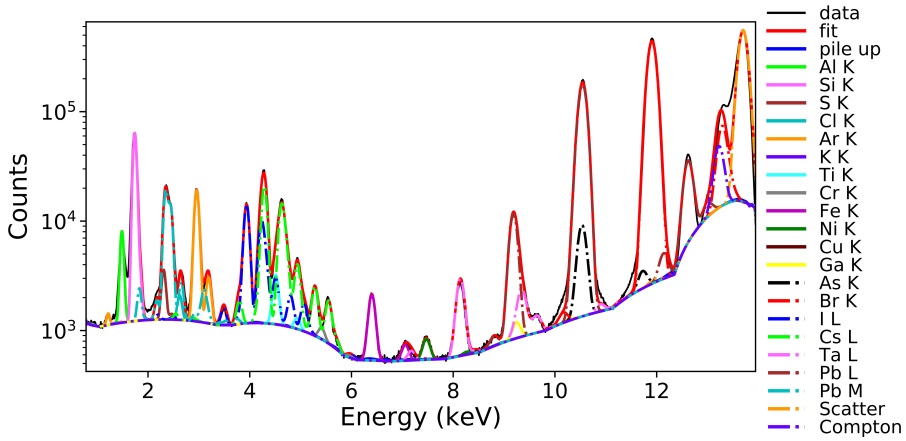

Figure 3.3: *PyMCA fit and elemental identification of the summed-up XRF spectra.*

The bottom line, shown in Fig. 3.3, shows the background count, which are added counts for all pixels that do not contribute to individual fluorescence peaks. These counts may originate from stray photons in the measurement chamber, or false detection due to sensitivity of the detector amplifier, or during processing and conversion in the MCA tools. They are typically not considered when the complete fluorescence map is generated, and should be removed. The removal of background peaks is not considered in this thesis, since the procedure is complicated , as peaks should be normalized with respect to the number of pixels, scan dwell time and the incident photon flux. They all may vary in case of fast scans (fly-scans), such as the ones that are shown in Results chapter. The error they introduce is expected to be small.

So far, only the fit of the cumulative spectra is performed. As a result, the configuration file is generated, with settings corresponding to the fit shown in Fig. 3.3. As each pixel contributes to the cumulative spectra, one can expect that the same fit would work for each pixel individually. In order to obtain counts of each element, in individual pixels, batch fitting is performed. The same HDF5 data set, alongside the configuration file, is imported into PyMCA, where the program will automatically fit each pixel and sum all counts for each element and emission line specified. As a result, the total counts per energy, for each element and its emission line, are saved in a separate HDF5 file, that can be used to obtain fluorescence maps. Additionally, fit files are generated, which contain more information for each pixel.

3.2.2 Grid interpolation

PyMCA data set provides many details for each pixel and for the map as a whole. Each pixel data and fit can be obtained separately and observed in the plotting tool. Fig. 3.4 shows the interpolated fluorescence map of a typical scan. The total counts show regions where many hits were detected, marked in red, while regions marked in blue are those that have very little fluorescence photons detected. The interpolated map quality depends on the initial number of pixels. In the case shown in Fig. 3.4, 2092 pixels are interpolated over a grid of 150×150 pixels, using nearest neighbor interpolation. This method allows for extending of the data in between measured points, but only to a certain extent. In order for the map quality to improve, a denser map is needed. Python script that performs interpolation and plotting of XRF maps is given in Appendix B2.

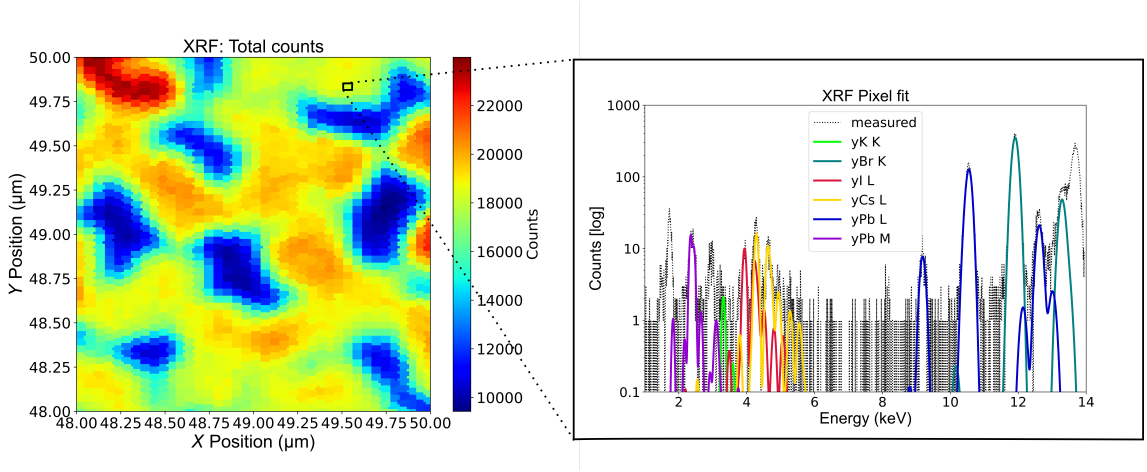

Figure 3.4: *Python-interpolated XRF map, with measured and element fit data shown for a single pixel.*

Fig. 3.4 also shows pixel measured data (dotted line), alongside some of the major elements obtained by fitting (solid lines). The data is retrieved from fit files generated by PyMCA batch fitting functionality. Each element emission line can be treated separately, which allows for a detailed plot with respect to energies and positions. The fit match the measured data quite well, with the exception that the background seems to be excluded from the total counts, making the fits not reach the peak counts as in measured data. Furthermore, the total counts is much less that the counts observed for cumulative plot shown in Fig. 3.3. This way, all pixels can be plotted and treated separately, especially is only a portion of energies of pixels is to be examined in detail.

3.3 PyMCA fitting vs. ROI plot

An important aspect in obtaining XRF maps and pixel pots relates to the use of PyMCA, to generate the correct count of elements. Another way to isolate element counts is to observe a region-of-interest (ROI). The method is simpler, as we can select a specific energy range in the XRF spectra, and simply plot cumulative counts in that area. And while this may work very well at high energies or for isolated peaks, it fails to describe low-energy regions, or regions where individual peaks of different elements are overlapping, but need to be fit and to be summed up, to generate the map.

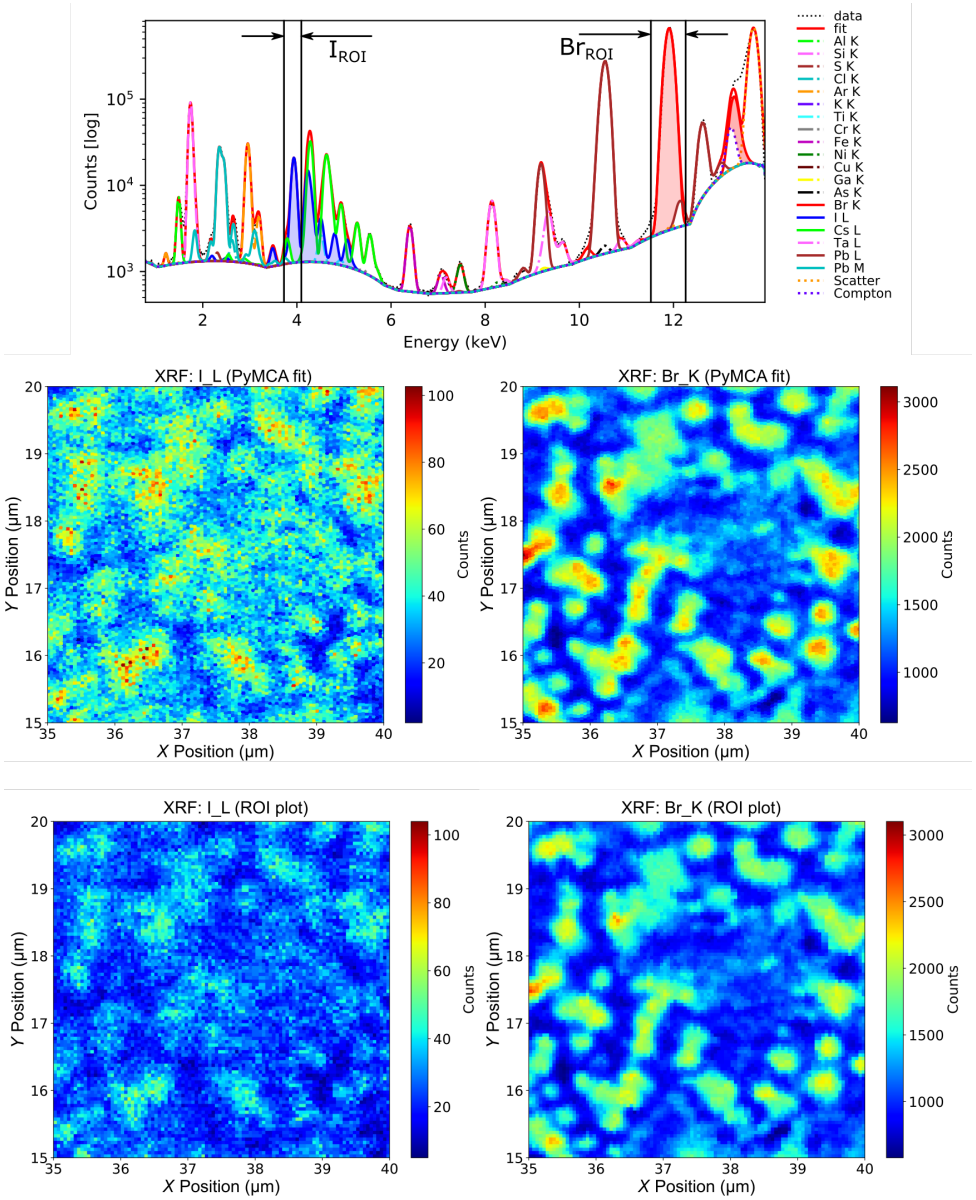

Figure 3.5: *Difference between PyMCA fit at all energies, versus ROI cumulative plot, which captures all counts in the specified energy range.*

Such example is shown in Fig. 3.5. The spectra shows the shaded areas that correspond to PyMCA fit, while ROI regions may not include all of the peaks that contribute to the specific line. The upper panels show Br and I XRF maps with correct counts, as calculated by PyMCA fit. Here, correct regions are selected, and overlapping counts are removed

automatically. The lower panel shows XRF map where totals counts are taken for Br and I as a cumulative sum in the ROI. In both cases, ROI plot captures a single peak only, and if multiple peaks are present, they will not be counted in the total sum. While Br peaks captures counts quite well, as it is positioned high in energy, I counts are significantly lower, since only one peak can be isolated. Other peaks are dominated by Cs counts, but some I counts are associated with them as well. PyMCA has the ability to recognize and assign the correct counts to both I and Cs, while ROI plot will not be able to make a difference between I and Cs, and will always show them as a either of the elements. There lies the strength of PyMCA, as a tool to accurately present XRF maps, and to separate each element contribution to peaks.

Chapter 4

Results and Discussion

This chapter will present the main results of the thesis: the analysis of XRF data, which are organized using PyMCA and custom made Python scripts, as described in the previous chapter. In this chapter, we will investigate different elements of perovskite thin films, both the radiation-stable $CsPbBr_3$ and the technologically-relevant triple cation $Cs_{0.06}MA_{0.15}FA_{0.79}Pb(I_{0.85}Br_{0.15})_3$ [22]. Furthermore, the perovskites will be deposited on the Si substrate and treated in several ways. Alongside the perovskite, K and I will be deposited as well, either as a mixture before or after the perovskite, or together with the perovskite. The investigation of these elements is organized into two different projects, where the first looks into K detection, and the second looks into Br and I mixing.

4.1 Investigation of K in $CsPbBr_3$

The first attempt to interpret XRF maps is done on comparing data for K, as an element with low Z number, with Pb, an element with a high Z number. As mentioned in the *Introduction*, K can be used to enhance the interface between the perovskite and Si [8]. Fig. 4.1 shows the total counts captured by X-rays emitted near the sample edge. This region is chosen to minimize re-absorption of the X-rays, which is very significant towards the middle part of the sample and drastically reduces the XRF signal collected at lower energies. In the first case (upper panels), K was deposited on the surface as KI solution, before depositing the perovskite (shown as Perovskite / K). The signal obtained from K line of potassium in this case is very weak. In another case (bottom panels), both the

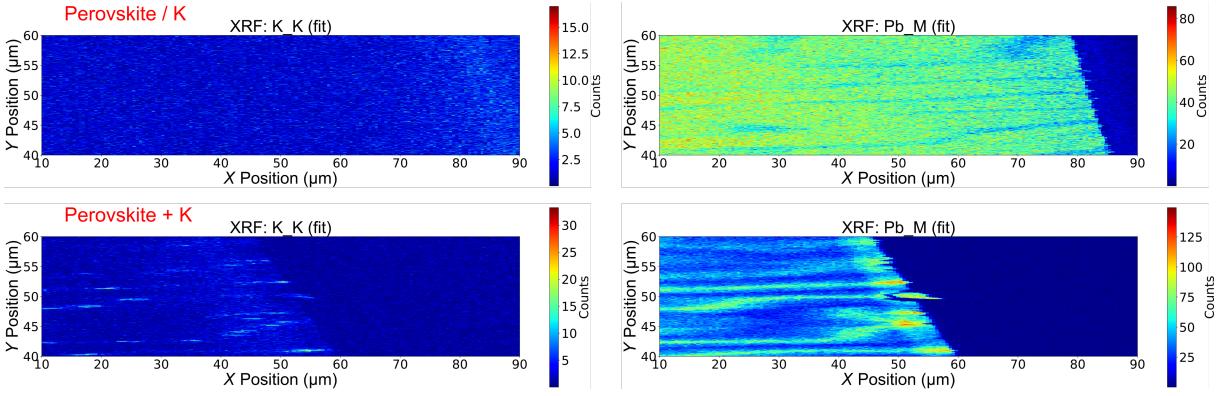

Figure 4.1: *Investigation of K in $CsPbBr_3$. The maps are compared against Pb, for three different cases: perovskite-on-K, and K+perovskite.*

K solution and perovskite are deposited as a mixture (shown as Perovskite + K), and a stronger signal is visible on XRF map. Comparing the peak count for Pb and K, the increase in Pb counts is 40%, while the number of counts for K is doubled. This suggests that mixing K and perovskite will improve K detection by XRF. The technique shows that it is possible to detect even low Z elements, but with a lower yield. Additionally, the mixture looses its uniformity over the sample surface, as shown by count inhomogeneities obtained in the Pb map.

In order to ensure that we are not only observing the background emission (errors in detection), we have to compare total counts per pixel for each case, with the case where no K is present (reference sample). An XRF map where no K is present is shown in Fig. 4.2. The Pb map, in this case, evidences fluctuations in the film morphology, while a near to zero signal is present in the K map, as expected. Note that this scan was not performed near a sample edge but in the middle, which can artificially reduce the photon count at lower energies. The residual counts represent background, or error data, which is not removed, in this case.

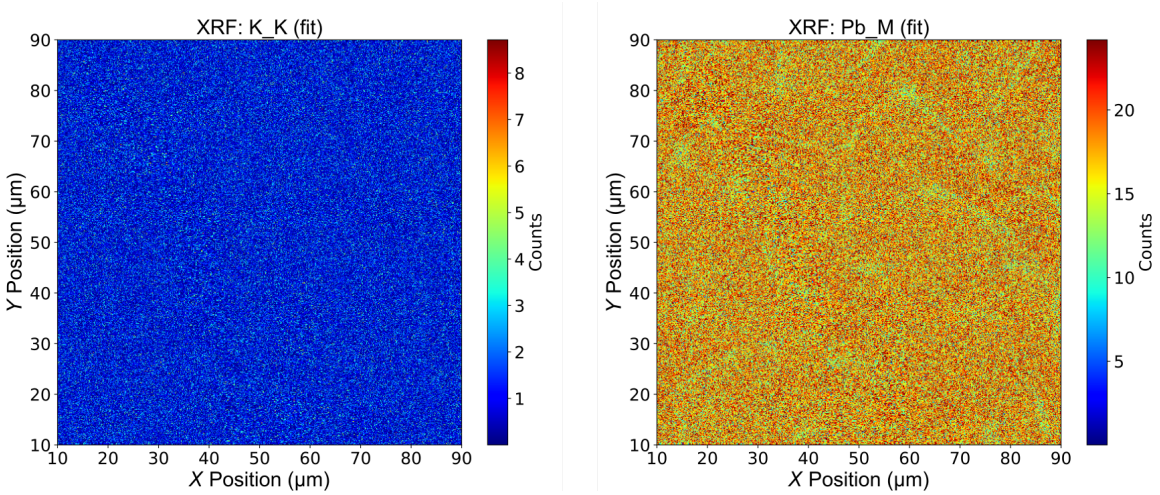

Figure 4.2: *Investigation of K in $CsPbBr_3$, in case where only perovskite is present.*

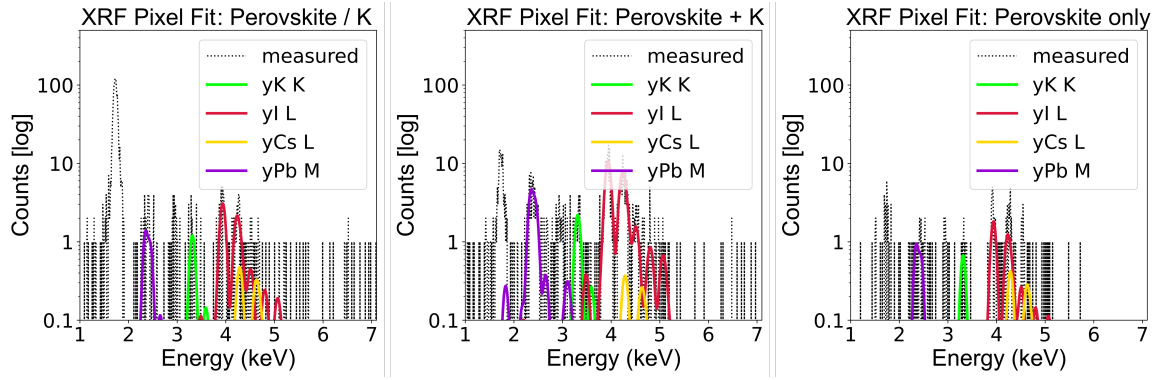

Figure 4.3: *Investigation into individual pixel fit for $CsPbBr_3$, in cases shown in Fig. 4.1 and 4.2*

Fig. 4.3 shows the fit for a single pixels at low energies, where both the K-line of K, and M-line of Pb are located. The pixel chosen here contains maximum counts for K. We also plot L-line of I and Cs as well, as they are a part of the KI solution and

the perovskite. It is important to be able to observe I and Cs L-lines as well, as they overlap. However, L-line of I starts at a lower energy, which is a clear indicator that the the element is present in the mixture. The number of counts for I is stronger than that of K, which can be attributed to several reasons, including their difference in Z. When comparing the counts for K, it is evident where the fit is actually detecting K (number of counts above the noise), from the case where we detect background only. It becomes evident from maximum number of counts for K on the scanned sample. Still, other lines scale as well, so it is difficult to say if the intensity for those particular pixels have just scaled, or if K is detected. Analyzing more pixels can reveal if we only have scaling in values, or there are actual discrepancies on different regions of the samples.

4.2 Br/I mixing in $CsPb(Br_{0.67}I_{0.33})_3$

Another project aims to determine location and distribution of Br and I in the $CsPb(Br_{0.67}I_{0.33})_3$ films. Br and I have a tendency to segregate in the presence of light or heat, so we use XRF to understand if they segregate [9]. Fig. 4.4 shows K-line of Br and L-line of I in two scans, on the same sample. The top maps show the case where the sample is not treated with light, while the bottom maps show a similar area XRF map following the light exposure. Similar structures can be seen on both maps. In general, the number of counts for I remains constant, while Br counts decrease slightly, and form XRF map it is

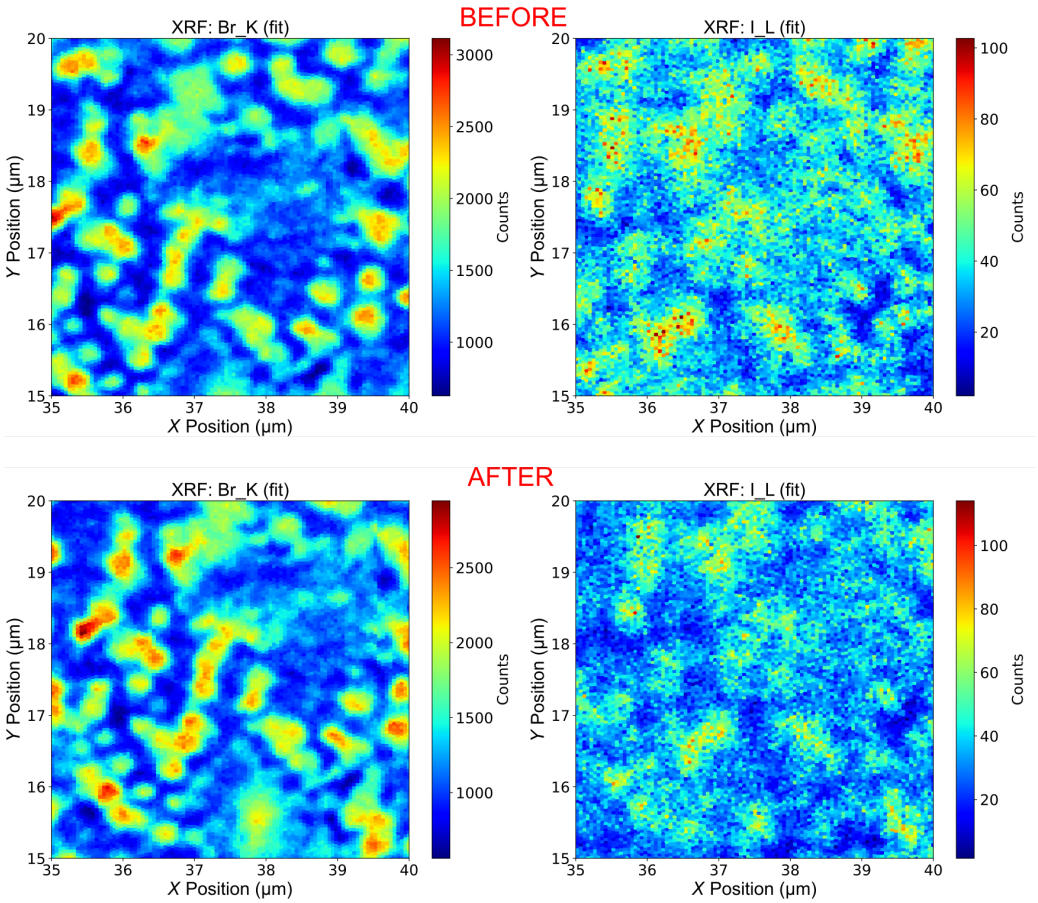

Figure 4.4: *Investigation of Br/I mixing in* $CsPb(Br_{0.67}I_{0.33})_3$. *The maps are shown before and after light treatement.*

difficult to understand if the elements are segregating under the presence of light.

Instead, we can look into the individual pixel comparison. Since majority of the area observed is the same,a grain that contains maximum Br counts is selected, as it would correspond before and after light exposure. The pixel plot and elemental fit is shown in Fig. 4.5. The energy scale is different, as we are observing K-lines of Br, and they occur between 12 and 14 keV. Additionally, L-lines of Cs and L- and M-lines of Pb are shown as well, to help in understanding the plot. While Br lines have similar cumber of counts (intensity), I lines decrease is value. It is evident that I does not occupy the same pixel as Br, and that segregation happens, since the intensity of I is reduced only slightly.

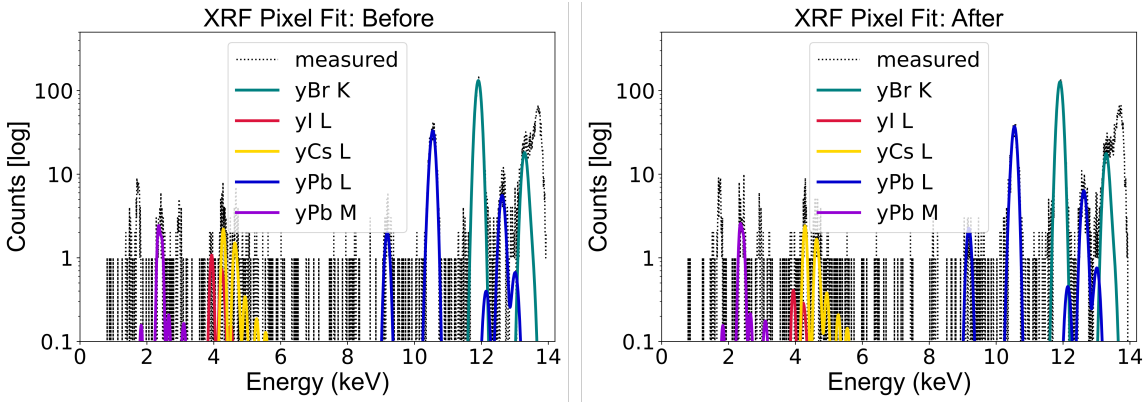

Figure 4.5: *Investigation into individual pixel fit for* $CsPb(Br_{0.67}I_{0.33})_3$*, in cases shown in Fig. 4.4.*

The complete XRF datasets are large, and more information can be taken out form each pixel, or from a cumulative plot, such as those shown in the previous chapter. They can give a better understanding of why maps look in the specific way, and how to interpret them. As far as the scope of this thesis, the necessary scripts are provided to be bale to plot and understand all the measured lines, which can lead to a better understanding. Even for data shown here, it covers only a fraction of the possibilities that PyMCA and the custom Python scripts can offer in handling and interpreting big data.

Chapter 5

Conclusion

The aim of this thesis was to develop tools and methods to understand the X-ray fluorescence data obtained at DESY. By analyzing data both in Python and in PyMCA, X-ray fluorescence allows us to image heavy elements, which are usually used to build perovskites. These materials can enhance light absorption in solar cells and resolving the properties of these materials could enable a more sustainable world in the future.

The main part of this work was to understand the data structure and content that is obtained with XRF imaging experiment from a synchrotron radiation facility. The files are streamed and stored in many individual datasets, and the custom Python script is used to stitch files into a single dataset, that can later on be used in PyMCA. In PyMCA, a fit procedure is established, where each fluorescence peak is specified, alongside the settings for the system and the beam line stricture. This enables the fit to accurately overlap measured data, as well as to automatically create the fit for each pixel separately, in a procedure known as batch fitting. Finally, and custom interpolation routine is written in Python to plot and summarize the pixel maps, while plots of individual measured and fit data can be generated for each pixel. the available dataset is large and allows for investigation of many element and material related details.

Through this work, different elemental maps are compared to study various perovskite-based samples. First, low energy detection is investigated with K mixture in perovskite. It shows that XRF can indeed be used to detect K peaks, as well as to relate them to the more pronounced Pb peaks. Secondly, a mixing between Br and I in $CsPb(Br_{0.67}I_{0.33})_3$ is investigated, where XRF map does not reveal a significant difference, but the pixel mapping shows that materials are indeed separating, as emission lines of I are reduced while Br line is kept constant. Lastly, an advantage of PyMCA, as an automated fitting and counting tool is showcased. This is done through comparison with region-of-interest (ROI) plots, which has difficulty representing low-energy elements, or elements with overlapping peaks.

Further research may include more thorough investigation of XRF data, since the framework for treating the XRF data in a correct manner is provided. As this data is part of a a large dataset, using artificial intelligence can significanly help in analyzing the data an answering some of the crucial questions. To meet this end, Python can be a powerful tool, and here, only a small part of the capabilities is used. Further scripting can make the analysis more complete. It can also be an example how new approach to handling data can benefit science in general.

Bibliography

[1]W. Pauli, "Über den zusammenhang des abschlusses der elektronengruppen im atom mit der komplexstruktur der spektren", Zeitschrift für Physik **31**, 765–783 (1925).

[2]N. Bohr, "On the constitution of atoms and molecules", The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **26**, 1–2 (1213).

[3]D. Griffiths, *Introduction to quantum mechanics* (Prentice Hall, Upper Saddle River NJ, 1995).

[4]R. Schramm, *X-ray fluorescence analysis: practical and easy 2nd edition* (FLUXANA, 2015).

[5]W. Shockley and H. J. Queisser, "Detailed balance limit of efficiency of p-n junction solar cells", Journal of Applied Physics **32**, 510–519 (1961).

[6]J. S. Manser, J. A. Christians, and P. V. Kamat, "Intriguing optoelectronic properties of metal halide perovskites", Chemical Reviews **116**, 12956–13008 (2016).

[7]M. V. Kovalenko, L. Protesescu, and M. I. Bodnarchuk, "Properties and potential optoelectronic applications of lead halide perovskite nanocrystals", Science **358**, 745–750 (2017).

[8]M. Abdi-Jalebi, Z. Andaji-Garmaroudi, S. Cacovich, C. Stavrakas, B. Philippe, J. M. Richter, M. Alsari, E. P. Booker, E. M. Hutter, A. J. Pearson, S. Lilliu, T. J. Savenije, H. Rensmo, G. Divitini, C. Ducati, R. H. Friend, and S. D. Stranks, "Maximizing and stabilizing luminescence from halide perovskites with potassium passivation", Nature **555**, 497–501 (2018).

[9]Y. Zhou, F. Wang, H.-H. Fang, M. A. Loi, F.-Y. Xie, N. Zhao, and C.-P. Wong, "Distribution of bromine in mixed iodide–bromide organolead perovskites and its impact on photovoltaic performance", J. Mater. Chem. A **4**, 16191–16197 (2016).

[10]IUPAC, *Compendium of chemical terminology, 2nd ed. (the "gold book")*, Compiled by A. D. McNaught and A. Wilkinson (Blackwell Scientific Publications, Oxford, 1997).

[11]J. Als-Nielsen and D. McMorrow, *Elements of modern X-ray physics* (John Wiley & Sons, Ltd., 2011).

[12]V. Thomsen, "Basic fundamental parameters in X-ray fluorescence", Spectroscopy **22** (2007).

[13]D. R. Baera, A. S. Leaa, J. D. Gellerb, J. S. Hammondc, L. Koverd, C. J. Powelle, M. P. Seahf, M. Suzukig, J. F. Wattsh, and J. Wolstenholmei, "Approaches to analyzing insulators with Auger electron spectroscopy: Update and overview", Journal of Electron Spectroscopy and Related Phenomena **176**, 80–94 (2010).

[14]Toshiyouri, *Own work, CC BY-SA 4.0*, `https://commons.wikimedia.org/w/index.php?curid=39333723`.

[15]M. Siegbahn, "Relations between the K and L series of the high-frequency spectra", Nature **96**, 676 (1916).

[16]A. W. Chao, K. H. Mess, M. Tigner, and F. Zimmermann, *Handbook of accelerator physics and engineering* (World Scientific, 2013).

[17]K. Wille, "Introduction to insertion devices, part of synchrotron radiation and free electron lasers", in Proceedings, CERN Accelerator School, CAS (Apr. 1996), pp. 61–75.

[18]E. Gatti and P. Rehak, "Semiconductor drift chamber — An application of a novel charge transport scheme", Nuclear Instruments and Methods in Physics Research **225**, 608–614 (1984).

[19]"Silicon drift detectors explained", in Semantic Scholar (2015).

[20]A. C. Melissinos and J. Napolitano, *Experiments in modern physics* (San Diego, CA: Academic Press, 2003).

[21]V. Solé, E. Papillon, M. Cotte, P. Walter, and J. Susini, "A multiplatform code for the analysis of energy-dispersive X-ray fluorescence spectra", Spectrochimica Acta Part B: Atomic Spectroscopy **62**, 63–68 (2007).

[22]N. A. Mica, R. Bian, P. Manousiadis, L. K. Jagadamma, I. Tavakkolnia, H. Haas, G. A. Turnbull, and I. D. W. Samuel, "Triple-cation perovskite solar cells for visible light communications", Photon. Res. **8**, A16–A24 (2020).

Appendix A

Emission lines for various elements

The emission lines shown here are taken from X-Ray Data Booklet [ref]:

X-Ray Data Booklet Table 1-2. *Photon energies, in electron volts, of principal K-, L-, and M-shell emission lines.*

Element	$K\alpha_1$	$K\alpha_2$	$K\beta_1$	$L\alpha_1$	$L\alpha_2$	$L\beta_1$	$L\beta_2$	$L\gamma_1$	$M\alpha_1$
3 Li	54.3								
4 Be	108.5								
5 B	183.3								
6 C	277								
7 N	392.4								
8 O	524.9								
9 F	676.8								
10 Ne	848.6	848.6							
11 Na	1,040.98	1,040.98	1,071.1						
12 Mg	1,253.60	1,253.60	1,302.2						
13 Al	1,486.70	1,486.27	1,557.45						
14 Si	1,739.98	1,739.38	1,835.94						
15 P	2,013.7	2,012.7	2,139.1						
16 S	2,307.84	2,306.64	2,464.04						
17 Cl	2,622.39	2,620.78	2,815.6						
18 Ar	2,957.70	2,955.63	3,190.5						
19 K	3,313.8	3,311.1	3,589.6						
20 Ca	3,691.68	3,688.09	4,012.7	341.3	341.3	344.9			
21 Sc	4,090.6	4,086.1	4,460.5	395.4	395.4	399.6			
22 Ti	4,510.84	4,504.86	4,931.81	452.2	452.2	458.4			
23 V	4,952.20	4,944.64	5,427.29	511.3	511.3	519.2			
24 Cr	5,414.72	5,405.509	5,946.71	572.8	572.8	582.8			
25 Mn	5,898.75	5,887.65	6,490.45	637.4	637.4	648.8			
26 Fe	6,403.84	6,390.84	7,057.98	705.0	705.0	718.5			
27 Co	6,930.32	6,915.30	7,649.43	776.2	776.2	791.4			
28 Ni	7,478.15	7,460.89	8,264.66	851.5	851.5	868.8			
29 Cu	8,047.78	8,027.83	8,905.29	929.7	929.7	949.8			
30 Zn	8,638.86	8,615.78	9,572.0	1,011.7	1,011.7	1,034.7			
31 Ga	9,251.74	9,224.82	10,264.2	1,097.92	1,097.92	1,124.8			
32 Ge	9,886.42	9,855.32	10,982.1	1,188.00	1,188.00	1,218.5			
33 As	10,543.72	10,507.99	11,726.2	1,282.0	1,282.0	1,317.0			
34 Se	11,222.4	11,181.4	12,495.9	1,379.10	1,379.10	1,419.23			
35 Br	11,924.2	11,877.6	13,291.4	1,480.43	1,480.43	1,525.90			
36 Kr	12,649	12,598	14,112	1,586.0	1,586.0	1,636.6			
37 Rb	13,395.3	13,335.8	14,961.3	1,694.13	1,692.56	1,752.17			
38 Sr	14,165	14,097.9	15,835.7	1,806.56	1,804.74	1,871.72			
39 Y	14,958.4	14,882.9	16,737.8	1,922.56	1,920.47	1,995.84			
40 Zr	15,775.1	15,690.9	17,667.8	2,042.36	2,039.9	2,124.4	2,219.4	2,302.7	

63	Eu	41,542.2	40,901.9	47,037.9	5,845.7	5,816.6	6,456.4	6,843.2	7,480.3	1,131
64	Gd	42,996.2	42,308.9	48,697	6,057.2	6,025.0	6,713.2	7,102.8	7,785.8	1,185
65	Tb	44,481.6	43,744.1	50,382	6,272.8	6,238.0	6,978	7,366.7	8,102	1,240
66	Dy	45,998.4	45,207.8	52,119	6,495.2	6,457.7	7,247.7	7,635.7	8,418.8	1,293
67	Ho	47,546.7	46,699.7	53,877	6,719.8	6,679.5	7,525.3	7,911	8,747	1,348
68	Er	49,127.7	48,221.1	55,681	6,948.7	6,905.0	7,810.9	8,189.0	9,089	1,406
69	Tm	50,741.6	49,772.6	57,517	7,179.9	7,133.1	8,101	8,468	9,426	1,462
70	Yb	52,388.9	51,354.0	59,370	7,415.6	7,367.3	8,401.8	8,758.8	9,780.1	1,521.4
71	Lu	54,069.8	52,965.0	61,283	7,655.5	7,604.9	8,709.0	9,048.9	10,143.4	1,581.3
72	Hf	55,790.2	54,611.4	63,234	7,899.0	7,844.6	9,022.7	9,347.3	10,515.8	1,644.6
73	Ta	57,532	56,277	65,223	8,146.1	8,087.9	9,343.1	9,651.8	10,895.2	1,710
74	W	59,318.24	57,981.7	67,244.3	8,397.6	8,335.2	9,672.35	9,961.5	11,285.9	1,775.4
75	Re	61,140.3	59,717.9	69,310	8,652.5	8,586.2	10,010.0	10,275.2	11,685.4	1,842.5
76	Os	63,000.5	61,486.7	71,413	8,911.7	8,841.0	10,355.3	10,598.5	12,095.3	1,910.2
77	Ir	64,895.6	63,286.7	73,560.8	9,175.1	9,099.5	10,708.3	10,920.3	12,512.6	1,979.9
78	Pt	66,832	65,112	75,748	9,442.3	9,361.8	11,070.7	11,250.5	12,942.0	2,050.5
79	Au	68,803.7	66,989.5	77,984	9,713.3	9,628.0	11,442.3	11,584.7	13,381.7	2,122.9
80	Hg	70,819	68,895	80,253	9,988.8	9,897.6	11,822.6	11,924.1	13,830.1	2,195.3
81	Tl	72,871.5	70,831.9	82,576	10,268.5	10,172.8	12,213.3	12,271.5	14,291.5	2,270.6
82	Pb	74,969.4	72,804.2	84,936	10,551.5	10,449.5	12,613.7	12,622.6	14,764.4	2,345.5
83	Bi	77,107.9	74,814.8	87,343	10,838.8	10,730.91	13,023.5	12,979.9	15,247.7	2,422.6
84	Po	79,290	76,862	89,800	11,130.8	11,015.8	13,447	13,340.4	15,744	—
85	At	81,520	78,950	92,300	11,426.8	11,304.8	13,876	—	16,251	—
86	Rn	83,780	81,070	94,870	11,727.0	11,597.9	14,316	—	16,770	—
87	Fr	86,100	83,230	97,470	12,031.3	11,895.0	14,770	14,450	17,303	—
88	Ra	88,470	85,430	100,130	12,339.7	12,196.2	15,235.8	14,841.4	17,849	—
89	Ac	90,884	87,670	102,850	12,652.0	12,500.8	15,713	—	18,408	—
90	Th	93,350	89,953	105,609	12,968.7	12,809.6	16,202.2	15,623.7	18,982.5	2,996.1
91	Pa	95,868	92,287	108,427	13,290.7	13,122.2	16,702	16,024	19,568	3,082.3
92	U	98,439	94,665	111,300	13,614.7	13,438.8	17,220.0	16,428.3	20,167.1	3,170.8
93	Np	—	—	—	13,944.1	13,759.7	17,750.2	16,840.0	20,784.8	—
94	Pu	—	—	—	14,278.6	14,084.2	18,293.7	17,255.3	21,417.3	—
95	Am	—	—	—	14,617.2	14,411.9	18,852.0	17,676.5	22,065.2	—
41	Nb	16,615.1	16,521.0	18,622.5	2,165.89	2,163.0	2,257.4	2,367.0	2,461.8	
42	Mo	17,479.34	17,374.3	19,608.3	2,293.16	2,289.85	2,394.81	2,518.3	2,623.5	
43	Tc	18,367.1	18,250.8	20,619	2,424	2,420	2,538	2,674	2,792	
44	Ru	19,279.2	19,150.4	21,656.8	2,558.55	2,554.31	2,683.23	2,836.0	2,964.5	
45	Rh	20,216.1	20,073.7	22,723.6	2,696.74	2,692.05	2,834.41	3,001.3	3,143.8	
46	Pd	21,177.1	21,020.1	23,818.7	2,838.61	2,833.29	2,990.22	3,171.79	3,328.7	
47	Ag	22,162.92	21,990.3	24,942.4	2,984.31	2,978.21	3,150.94	3,347.81	3,519.59	
48	Cd	23,173.6	22,984.1	26,095.5	3,133.73	3,126.91	3,316.57	3,528.12	3,716.86	
49	In	24,209.7	24,002.0	27,275.9	3,286.94	3,279.29	3,487.21	3,713.81	3,920.81	
50	Sn	25,271.3	25,044.0	28,486.0	3,443.98	3,435.42	3,662.80	3,904.86	4,131.12	
51	Sb	26,359.1	26,110.8	29,725.6	3,604.72	3,595.32	3,843.57	4,100.78	4,347.79	
52	Te	27,472.3	27,201.7	30,995.7	3,769.33	3,758.8	4,029.58	4,301.7	4,570.9	
53	I	28,612.0	28,317.2	32,294.7	3,937.65	3,926.04	4,220.72	4,507.5	4,800.9	
54	Xe	29,779	29,458	33,624	4,109.9	—	—	—	—	
55	Cs	30,972.8	30,625.1	34,986.9	4,286.5	4,272.2	4,619.8	4,935.9	5,280.4	
56	Ba	32,193.6	31,817.1	36,378.2	4,466.26	4,450.90	4,827.53	5,156.5	5,531.1	
57	La	33,441.8	33,034.1	37,801.0	4,650.97	4,634.23	5,042.1	5,383.5	5,788.5	833
58	Ce	34,719.7	34,278.9	39,257.3	4,840.2	4,823.0	5,262.2	5,613.4	6,052	883
59	Pr	36,026.3	35,550.2	40,748.2	5,033.7	5,013.5	5,488.9	5,850	6,322.1	929
60	Nd	37,361.0	36,847.4	42,271.3	5,230.4	5,207.7	5,721.6	6,089.4	6,602.1	978
61	Pm	38,724.7	38,171.2	43,826	5,432.5	5,407.8	5,961	6,339	6,892	—
62	Sm	40,118.1	39,522.4	45,413	5,636.1	5,609.0	6,205.1	6,586	7,178	1,081

Appendix B

Python scripts

B.1 Convert raw data into 1D array

```
"""
This script imports the main scan file and associated set of scan files,
reads the energy limit in the main file, as well as all pixels in the data
files, packages them in 1D array and saves in the _raw.h5 file

Files should be placed in the folder SCAN_~sample_number~
This foder should contain:
    scan_~sample_number~.nxs - the main file where information is placed
    xspress3_scan_~sample_number~ - folder containing data in many .nxs files

If some of these do not exist, the script will fail!

The output is scan_~sample_number~.h5 - 1D dataset used in cumulative fit

@author: Samuel Narciso Silva
"""

import os
import h5py
import numpy as np

path = 'C:/Users\Public\Documents\Python_Scripts/' # main location
sample_number = '00000'     # sample run, given as a 5-digit text

path_folder = os.path.join(path, 'SCAN_' + sample_number) # main folder

# Check if data directory exists
if not os.path.exists(path_folder):
    print('\n\nThe path does not exist: ' + path_folder + '\n\n')
    quit()

# find main scan file, .nxs extention, and create a path for output _raw.h5 file
for filename in os.listdir(os.path.join(path_folder)):
    if 'scan_' + sample_number in filename[0:10] and filename.endswith('.nxs'):
        scanfile = os.path.join(path_folder, filename)
        outputfile = os.path.join(path_folder, filename[0:10]+'.h5')

# check for data folder
folder_name = 'xspress3_scan_'
file = [0]*len(os.listdir(os.path.join(path_folder,folder_name+sample_number)))
for filename in os.listdir(os.path.join(path_folder,folder_name+sample_number)):
    if 'scan_' + sample_number  in filename[0:10] and filename.endswith('.nxs'):
        file[int(filename[11:16])] = os.path.join(path_folder, folder_name+sample_number,
     filename)
file = [x for x in file if x != 0] #remove fields that do not have assigned strings

with h5py.File(scanfile, 'r') as main_file:
    # get energy value, to define channel limit
    energy = main_file['scan/data/energy'][0]

```

```
# channel limit for observing XRF - should not be above the source energy!
limit = int(np.ceil(energy/1000)*100)

data = []
# read files in the xspress3 folder, create a single dataset by stacking rows!
for filename in file:
    with h5py.File(filename, 'r') as scan:
        data_temp = scan['entry/instrument/xspress3/channel00/histogram'][:,0:limit]
    if len(data) != 0:
        data = np.vstack((data, data_temp))
    else:
        data = data_temp

# wire data into output file
with h5py.File(outputfile, 'w') as output:
    output['energy'] = energy
    output.create_dataset('xrf', data = data, compression='gzip')
```

B.2 Interpolate and plot XRF maps

```
"""
This script reads the .h5 file in the batch folder, alongside position data.
Elemental maps are interpolated, in order to obtain counts for each pixel.
The data is then saved as figure file

Files should be all placed in the folder SCAN_~sample_number~
This foder should contain:
    scan_~sample_number~_positions.h5 - the position file
    scan_~sample_number~_batch - folder where batch fits are located
    scan_~sample_number~_batch/scan_~sample_number~.h5 - fit file
    scan_~sample_number~_out - folder where output plots will be stored

If some of these do not exist, the script will fail!

The outputs are figures as PNG and SVG files

@author: Samuel Narciso Silva
"""

import os
import h5py
import numpy as np
from scipy.interpolate import griddata
import matplotlib
import matplotlib.pyplot as plt

path = 'C:/Users\Public\Documents\Python_Scripts/' # main location
sample_number = '00000'     # sample run, given as a 5-digit text
# what elementsa re expected to be found
elements = ['K', 'Br', 'Cs', 'I', 'Pb'] # elements that are to be investigated
K_line   = [  1,    1,    0,   0,    0] # K lines plot selector
L_line   = [  0,    1,    1,   1,    1] # L lines plot selector
M_line   = [  0,    0,    0,   0,    1] # M lines plot selector
# interpolation grid size
dX = 150
dY = 150

path_folder = os.path.join(path, 'SCAN_' + sample_number) # main folder
# Check if sample directory exists
if not os.path.exists(path_folder):
    print('\n\nThe path does not exist: ' + path_folder + '\n\n')
    quit()

batch_folder = os.path.join(path_folder,'scan_' + sample_number + '_batch')
# Check if batch directory exists
if not os.path.exists(batch_folder):
    print('\n\nThe path does not exist: ' + batch_folder + '\n\n')
    quit()

out_folder = os.path.join(path_folder, 'scan_' + sample_number + '_out')
# check if output folder exists
```

```
if not os.path.exists(out_folder):
    print('\n\nThe path does not exist: ' + path_folder + '\n\n')
    quit()

# find position file in sample name directory
for filename in os.listdir(os.path.join(path_folder)):
    if 'scan_' + sample_number in filename and filename.endswith('_positions.h5'):
        positionfile = os.path.join(path_folder, filename)

# find batch fit file in sample name directory
for filename in os.listdir(os.path.join(batch_folder)):
    if 'scan_' + sample_number in filename[0:10] and filename.endswith('.h5'):
        input_file = os.path.join(batch_folder, filename)

# read position file for interferometer positions, as well as set positions
with h5py.File(positionfile, 'r') as pos_file:
    x_set = pos_file['data/dialed_fast/data'][:]
    y_set = pos_file['data/dialed_slow/data'][:]
    x_pos = pos_file['data/int_y_12/data'][:]
    y_pos = pos_file['data/int_z_12/data'][:]

# define files that are to be read and plotted, based on the predefined matrix
codename = []
for i in range(0,len(elements)):
    if K_line[i] == 1: codename.append(elements[i]+'_K')
    if L_line[i] == 1: codename.append(elements[i]+'_L')
    if M_line[i] == 1: codename.append(elements[i]+'_M')

data = []; data_label = [];
data_path = 'scan_'+sample_number+'_raw/xrf_fit/results/parameters/'
with h5py.File(input_file, 'r') as data_file:
    label = list(data_file[data_path].keys())
    for el in range(0,len(codename)):
        if codename[el] in label:
            data_label.append(codename[el])
            data.append(np.array(data_file[data_path+codename[el]][:].T))

# bug in batch fitting: check index for elements which are not a number,
# and remove them from data and positions
NANi = np.zeros(len(x_pos), dtype = 'int')
for i in range(0,len(x_pos)):
    if np.isnan(data[0][i]):
        NANi[i] = int(i)
NANi = NANi[NANi!=0]
x_pos = np.delete(x_pos,NANi)
y_pos = np.delete(y_pos,NANi)
for el in range(0,len(data)):
    data[el] = np.delete(data[el],NANi)

# create interpolation grid
xmin = np.min(x_pos)+0.05; xmax = np.max(x_pos)-0.05
ymin = np.min(y_pos)+0.05; ymax = np.max(y_pos)-0.05
pos = (x_pos,y_pos)
XX, YY = np.mgrid[xmin:xmax:dX*1j, ymin:ymax:dY*1j]
# create interpolated data
data_interp = []
for el in range(0,len(data)):
    data_interp.append(np.zeros((dX,dY)))
# interpolation routine
for el in range(0,len(data_interp)):
    FF = griddata(pos,data[el][:],(XX,YY),method='nearest')
    data_interp[el][:,:] = FF.T

# create XY grid from set X and Y
x_set_min = np.min(x_set); x_set_max = np.max(x_set)
y_set_min = np.min(y_set); y_set_max = np.max(y_set)
limits = [x_set_min,x_set_max,y_set_min,y_set_max]

# plots
matplotlib.rcParams.update({'font.size': 18}) # set global tick label

# plot fits for elements
for el in range(0,len(data_interp)):
```

```
125     plt.figure(figsize=(10,8))
126     font = {'fontname':'Arial', 'size':'22'}
127     plt.imshow(data_interp[el],'jet',extent=limits)
128     cbar=plt.colorbar(); cbar.ax.set_ylabel('Counts')
129     plt.xlabel('$X$ Position ($\mu$m)',**font)
130     plt.ylabel('$Y$ Position ($\mu$m)',**font)
131     plt.title('XRF: '+data_label[el]+' (fit)',**font)
132     plt.savefig(os.path.join(out_folder, data_label[el]+'_fit_counts.png'), dpi=300,
        format='png')
133     plt.savefig(os.path.join(out_folder, data_label[el]+'_fit_counts.svg'), dpi=300,
        format='svg')
```