LU-TP 20-22
June 2020

Correcting biological infrared spectroscopy data for atmospheric gases and Mie scattering

Stéphan Pissot

Department of Astronomy and Theoretical Physics, Lund University

Master thesis supervised by Carl Troein.

LUND
UNIVERSITY

Abstract

Infrared absorption microscopy is a powerful chemometric tool with a wide variety of applications. It is, however, subject to considerable disturbances from atmospheric gases and scattering effects. As experimental fixes are not always applicable or advisable, data therefore needs to be computationally corrected before any attempt at an interpretation. This work designed, improved and validated several methods to correct such imaging data with varying success. It also provided a reliable way to assess the efficacy of such correction, as well as a set of methods to produce synthetic test data.

Popular description

Biochemical screening techniques (such as cancer diagnosis, water testing, etc.) used today tend to be unreliable, slow, and most importantly expensive. Infrared spectroscopic imaging has the potential to make this process easier, faster, and cheaper. It is, however, subject to disturbances that make it very difficult to interpret. Techniques to get rid of these disturbances exists, but have the drawback of making the whole process either slower, more expensive, or both. Solving the problem with an algorithm, however, would allow to get rid of the drawbacks while keeping the advantages of the technique.

Molecules have long been known to vibrate at specific frequencies when excited by electromagnetic radiation, in a similar way as a guitar string emits a note when picked by the guitarist's finger. By recording how a sample reacts to this excitation, scientists can determine its composition in much the same way as the musician can hear the notes that make up a chord.

In real life, however, samples are not as simple as the sound emitted by a carefully played 6-string guitar in a quiet room, but would be better compared to a piano sonata being played in the middle of Times Square at rush hour. Numerous and hard to predict vibrations appear and overlap with the fine melody. So much so that it is difficult to identify what corresponds to the actual note being played and what is background noise.

In Times Square, the sound of cars and people passing by is intense. In a biological sample, this "noise" can come from a lot of things, but originates mainly from the water vapor and carbon dioxide which surround it and which also vibrate in those frequencies of interest.

It is of course possible to isolate the sample from much of these perturbations by keeping it in a very controlled environment, but this requires additional time and equipment. Nobody wants to (or can) rent a recording studio every time they want to listen to music. This is why the most promising way to get rid of the unwanted vibrations is instead to remove them with the use of a computer algorithm.

Indeed, even though the background noise is unpredictable, it still has some key differences from the music which make it possible—to some extent—to separate them. We therefore could, in theory, "listen in" to the vibrations emitted by, for example, cancer cells markers in the middle of the chaos of a biopsy. Then, the equipment needed to produce this kind of diagnosis would fit in a backpack and cost thousands of dollars, instead of complete labs worth millions today. It would be almost instant, and very easily deployed in almost any circumstances, and not just in a hospital.

Additionally, this technique would not be restricted to cancer screening, and not even to medical applications. Being able to identify the precise composition of a sample is useful in numerous fields and could help to, for instance, check the drinkability of a water source, or understand how certain fungi can break down plastics.

Contents

List of acronyms

AsLS	Asymmetric Least-Squares
FTIR	Fourier Transform Infrared
MCR-ALS	Multivariate Curve Resolution–Alternating Least Squares
(E)MSC	(Extended) Muliplicative Signal Correction
OCTAVVS	Open Chemometrics Toolbox for Analysis and Visualization of Vibrational Spectroscopy data
(C)RMieSC	(Clustered) Resonant Mie Scattering Correction

Acknowledgements

I would like to thank Dr. Carl Troein for his continued advice and assistance throughout the whole duration of this thesis. I would like to thank Michiel Op De Beeck for providing data without which none of this work would have been possible, as well as help regarding its interpretation.

1 Introduction

1.1 Infrared spectroscopy

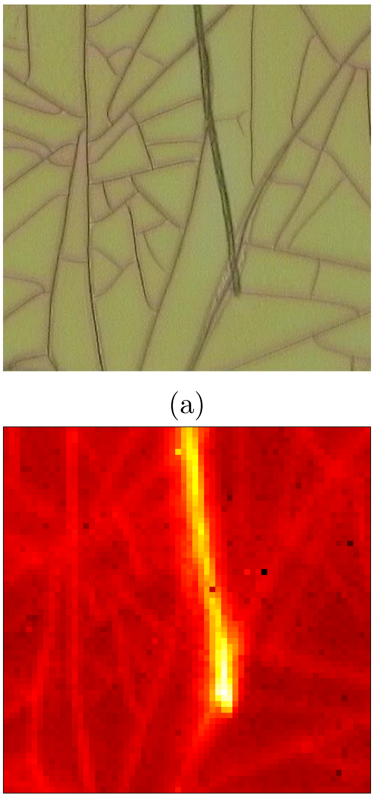
Fourier Transform Infrared (FTIR) Spectroscopy is a chemometric technique useful in identifying organic compounds. As a microscopy technique, it allows researchers and technicians to not only identify the nature of chemical species but also their spatial distribution in a sample. As such, it is has proven useful in the analysis of biological structures and the way they interact with their environment.

When molecules are subjected to electromagnetic waves of certain frequencies, their bonds can start to vibrate. The nature of the bonds, as well as their environment, affect the frequencies at which they resonate, and those frequencies are often in the infrared part of the spectrum (additionally, bonds can vibrate in different modes, leading to several absorption peaks). Therefore, when infrared light of the appropriate wavelength hits such bonds, part of it is absorbed. This leads to peaks in the absorption spectrum of the sample, of different width, position and amplitude depending on the bonds responsible for them. For instance, solids and liquids will have wider absorption peaks than gases, where the excitation modes are better defined. Infrared absorption microscopy consists in obtaining those absorption spectra for all points of an image. See Figure 1 for an example.

To obtain these images, an infrared light beam is fed through a Michelson interferometer[1], then through a microscopy stage, and the resulting beam is received by a *Focal Plane Array (FPA)* [1](in our case, this array is of 64×64 pixels). The resulting interferogram is then converted back to a spectrum by calculating its Fourier transform.

There are several sampling modes available to obtain absorption spectra: transmission (light passing through the sample and out on the other side), transflection (light passing through the sample, reflected on the base of the slide—usually coated in gold—and through the sample again), or attenuated total reflection (light passes through a crystal on which the sample is placed, reflects on the interior surface and through the other side; the evanescent wave which appear at the interface interacts with the sample). We studied exclusively transflection samples.

[1] An optical setup splitting a beam, reflecting it on a mirror and interfering with itself. The position of the mirror is then the Fourier transform of the wavelength.



(a)

(b)

Figure 1: Example of captured data. (a) shows a visible light image. (b) shows total intensity in the IR spectrum.

Numerous applications of this technique, such as medical diagnosis, are being developed. However, such applications require a precise and reliable way to analyse the image provided by infrared absorption spectroscopy apparatus. To that end, the Computational Biology and Biological Physics Group[2], in collaboration with the Department of Biology, at Lund University has designed OCTAVVS (Open Chemometrics Toolbox for Analysis and Visualization of Vibrational Spectroscopy data) as a toolkit for the processing and analysis of such images [2, 3].

1.2 Spectral image processing

The first step in processing infrared absorption imaging data consists of clearing the data of any perturbation which the experimental process might have introduced. A reference spectrum (usually of the apparatus without any sample in the chamber) is always subtracted from the spectra to account for the instrument influence (the light emitted by the source, its transmission across the optics, and its measurement at the detector each have wavenumber-dependent variations). It still fails to account, however, for the influence of other factors. The two main such perturbations are of atmospheric (caused by water vapour and carbon dioxide in the light's path) and scattering (due to the light scattering against structure within the sample) nature. As the infrared signature of biological species in naturally occurring concentrations can be weak, those perturbations can quickly become a limiting factor in the analysis of the data. Hence the need for algorithmic correction of those atmospheric and scattering perturbations.

This work was concentrated on improving those corrections within the OCTAVVS framework. As no reliable way to evaluate the efficacy of the correction was found in literature, a sizeable part of this work also focused on devising a way to validate these correction methods.

Since the 1980s and the development of the *Multiplicative Signal Correction*, there has been considerable effort to provide better and faster ways of correcting infrared spectra. Solheim et al. [4] provided the most current way of generating extinction matrices for scattering correction. This work did not focus on improving Solheim's method, but instead on perfecting the way they are used within OCTAVVS' RMieSC implementation.

[2] part of the Department of Astronomy and Theoretical Physics

1.2.1 Current procedures for atmospheric correction

One problem arising when using infrared absorption spectroscopy in normal atmospheric conditions (that is, without purging the apparatus' chamber with pure nitrogen or argon), is the atmospheric perturbations. Water vapour and carbon dioxide both have absorption peaks in the IR spectrum. Therefore, the presence of those species in the path of the light leads to some unwanted contributions to the output data. This is particularly problematic for the water vapour, which has absorption peaks in *fingerprint region* (a region of interest in the analysis of organic compounds, located between 500 and 1500 cm^{-1}) and less for carbon dioxide, which just represents one isolated peak in a less crucial region of the spectrum. These absorption spectra are shown in Figure 2.

The most effective way to eliminate those unwanted contributions consists of purging the offending gases. This is most commonly done with an inert gas such as argon or—in our case–nitrogen, as it does not affect the infrared light passing through it. However, such a method is time-consuming and requires additional equipment, making it more expensive and less adaptable to different situations. To say nothing of the fact that, even using those gases, it is difficult to achieve total elimination of water vapour. As its low cost of operation is one of the strengths of infrared microscopy, it is advisable to attempt to keep it that way. Therefore, an algorithmic correction of atmospheric perturbations is required.

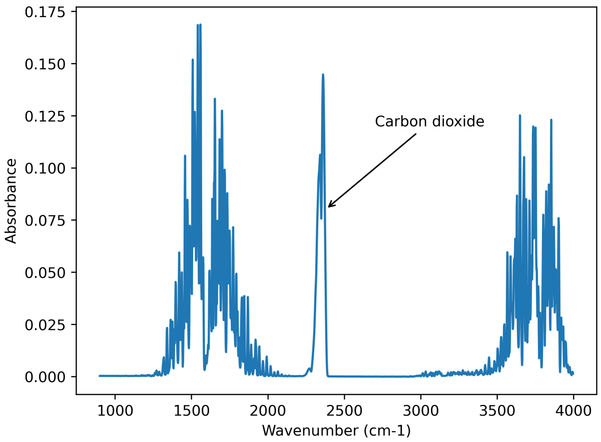


Figure 2: Spectrum of a mixture of atmospheric gases (water vapour and carbon dioxide). The carbon dioxide peak is clearly visible at around 2400 cm^{-1}.

Previous work Over the years, several methods of denoising the signal have been designed, with varying efficacy [8, 9].

Firstly, a straightforward approach is to filter the spectra to smooth them [10]. This is very efficient and fast at eliminating atmospheric-related peaks, but does not discriminate between atmospheric and biological data, therefore also eliminating potentially significant peaks.

Secondly, it is possible to use a purely atmospheric spectrum measured separately as a reference and remove it from the sample spectra. This has the advantage over the previous method to only affect the contribution one wants to have removed. Moreover, it tends to not fully remove the unwanted contributions, due to slight variations in the amplitude and location of peaks.

OCTAVVS method The method currently used in the OCTAVVS preprocessing software eliminates water vapour and carbon dioxide perturbations in the following way:

The carbon dioxide is simply cut away using a spline [2]. This is justified by the fact that this area of the spectrum does not contain much significant data. As such, OCTAVVS' elimination of this peak is mainly justified by the risk that it would affect further correction and analysis steps.

The rest of the atmospheric correction relies on the fitting of an EMSC model to the raw data using a previously acquired atmospheric reference. For reasons which will be made clear in Section 2.2.2, fitting is done on the derivative of both the data and the reference spectrum.

The next step consists in a *local peak correction* affecting smaller groups of peaks through a moving window.

Some filtering is then applied, to smooth out any sharp peaks left, as the water vapour spectrum contains those high-frequency elements.

1.2.2 Current procedures for scattering correction

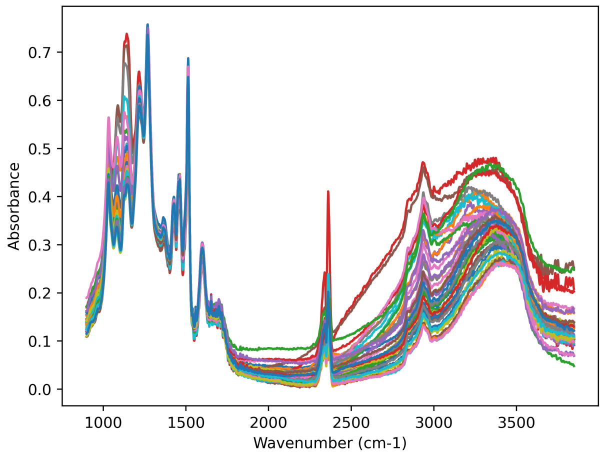


Figure 3: Example of how scattering affects spectra. The sample pictured is of *Paxillus involutus* hyphae on a lignin substrate. The use of the transflection sampling mode—where light passes through the full thickness of the sample twice—can explain why this image is particularly affected.

On top of atmospheric gases, another deleterious effect commonly affects infrared spectra: scattering. When the samples being studied have some spatial structure (in our case cells and cracks in the substrate), the beam of infrared light can be scattered by them. As different frequencies are differently scattered, this leads to a perturbation of the apparent spectra.

In certain applications, such as the measure of particles size [11], measuring scattering effects can be useful, but in our case, they only contribute to the deformation of the signal. Peaks are distorted, and their positions potentially shifted, which in turns makes any further analysis difficult or even impossible. This is shown in Figure 3. Here, contrary to the atmospheric perturbations, no other solution than computational exists to solve this problem. Therefore, extensive research has been undertaken to solve this problem. Most models used for this correction are based on the Mie solutions to Maxwell's equations, which explain the behaviour of electromagnetic waves scattered by spheres, knowing their refractive index and radius.

Resonant Mie scattering Knowing the absorption spectrum of the sample (or, in our case, approximating it using a reference and refining the approximation through an iterative process), the *Kramers-Kronig relations* can be used to calculate the refractive index of the sample:

$$n_{KK}(\nu) = \frac{2}{\pi}\mathbf{P}\int_0^\infty \frac{s \times k(\nu)}{s^2 - \nu^2}\mathrm{d}s \tag{1}$$

Where n_{KK} is the real refractive index, $\mathbf{P}$ denotes the Cauchy principal value of the integral, and k is the extinction coefficient, proportional to the absorption Z [12].

In turn, the *van de Hulst approximation*[13] shows how scattering by a sphere can affect such a spectrum [12]. Using a range for the expected size of the spheres (i.e. the scale of the structure) and their refraction index, one can use the Mie solutions to generate extinction spectra representative of what must happen in the sample. These spectra can then be used, through an EMSC model, to correct the scattering out of the data:

$$Z_{app}(\nu) = a + b\bar{x}(\nu) + d_1\nu + \sum_i g_i p_i(\nu) + e(\nu) \tag{2}$$

Where the p_i are the scattering extinction spectra. This model is an extension of Equation 4 and is called *Resonant Mie scattering EMSC*. Parameters g_i—and, as in standard EMSC, a, b and d_1—are usually found using least-square fitting to the apparent spectrum Z_{app}. Correction then follows in the same manner as standard EMSC:

$$Z_{corr}(\nu) = \frac{Z_{app}(\nu) - a - d_1\nu - \sum_i g_i p_i(\nu)}{b} \tag{3}$$

This is the method used in the OCTAVVS toolkit, in a clustered approach.

Correction algorithms The generation of extinction spectra from diameters and refraction indices of sphers is non-trivial and computationnally intensive. Bassan et al. [12] were, in 2010, the first to provide an implementation for this problem. Then, in 2018, Konevskikh at al. [14] designed a faster implementation relying on a meta-model for faster computation of the extinction spectra.

Both of those implementations are available in OCTAVVS and used in this work (for correction or for generation of synthetic scattering data). For simplicity, they are referred to as *the Bassan algorithm* or *the Konevskikh algorithm.*

Structure of this document We will first present the methods used for each of the three parts of this work: atmospheric correction, scattering correction, and classification of spectra to validate those corrections. Then, we will show what results were obtained on each of these parts. Finally, we will discuss general results about corrections and their impact.

2 Methods

2.1 General spectroscopy

All corrections and validations presented here were carried out on raw infrared absorption microscopic data. This data showed hyphae[3] of the fungus *Paxillus involutus* on a lignin substrate [5]. The output of the experiments (and hence the input for our corrections) is the *apparent* spectrum at each point of the image. These apparent spectra can be decomposed in several additive components, representing the substrate, the atmospheric contributions, and any other biological species potentially significant. Therefore, when knowing the aspect of the perturbation(s) we would like to remove, the usual approach consists in working back from this sum.

$$Z_{app}(\nu) = a + b\bar{x}(\nu) + d_1\nu + d_2\nu^2 + \cdots + d_n\nu^n + e(\nu) \tag{4}$$

Where ν is the wavenumber; Z_{app} is the raw, or apparent, spectrum; a a constant baseline, $\bar{x}$ the average spectrum; and $d_1\nu + d_2\nu^2 + \cdots + d_n\nu^n$ a polynomial baseline.

Then, a corrected spectrum could be approximated by:

$$Z_{corr}(\nu) = \frac{Z_{app}(\nu) - a - d_1\nu - d_2\nu^2 - \cdots - d_n\nu^n}{b} \tag{5}$$

Such a correction method is named an Extended Multiplicative Signal Correction, or EMSC. This method, an improvement on the simpler *Multiplicative Signal Correction* was first introduced in 2003 by Martens et al. [6].

In the original application of EMSC, the authors applied it to Raman spectra of salmon oils, and used the model up to its quadratic term. Extending it beyond the sixth or seventh order might cause over-fitting problems [7].

The higher order terms are useful in correcting unexplained ways in which the spectra are distorted, but since our model accounts for most of the distortions we expect to observe, those terms are not necessary. In this work, the EMSC is used with a linear baseline only. However, it is extended with scattering terms in the RMieSC approach described in Section 1.2.2.

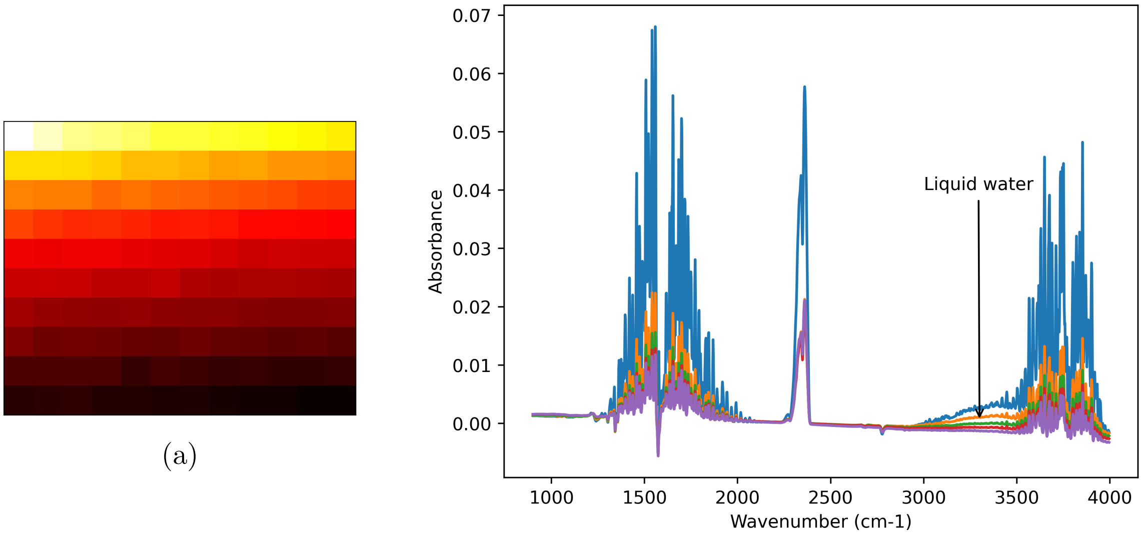
2.2 Atmospheric correction

The first part of this work focused on the improvement of OCTAVVS' atmospheric correction (mentioned in Section 1.2.1). To that end, we collected some data onto which to train and apply our algorithms:

[3] Hyphae are the filaments that make up the mycelium of a fungus.

2.2.1 Input data

Depending on the type of correction used, different data was used: adequate reference spectra to obtain an accurate representation of the contributions we had to eliminate; and input data against which to assess the efficacy of our algorithms.



(a)

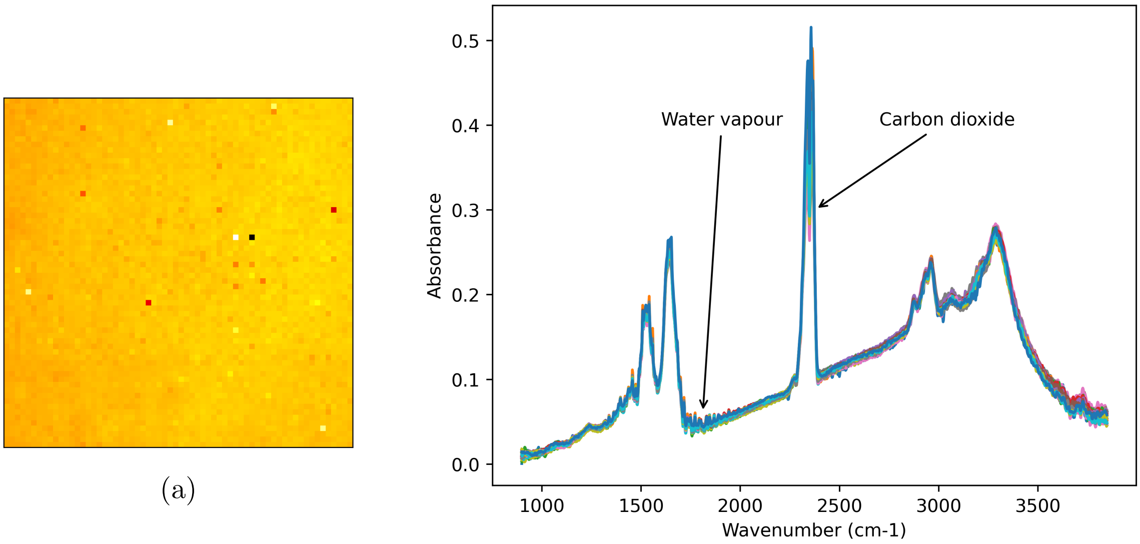
(b)

Figure 4: Time series of the enclosure being purged of water vapour after introduction of a Petri dish of water at 60°C. (a) shows a visual representation of the time series. The 120 spectra captured can be seen decreasing in overall intensity as water vapour is purged out of the chamber of the microscope. (b) shows the corresponding spectra. The overall intensity can be seen decreasing, as expected. In addition to the water vapour and carbon dioxide peaks, absorption bands from liquid water can be seen in the region between 3000 and 3500 cm^{-1}.

Atmospheric reference The reference data supplied to the algorithm originated from the results of two experiments. Each started with the introduction of a Petri dish filled with water at 60°C. One of the experiments consisted of progressively purging away water vapour from the spectrometer enclosure using dry nitrogen gas. The other left the dish to cool down on its own. Both were conducted over a time of 120 minutes, with one measurement every minute.

The PCA methods mentioned in Sections 5 and 2.2.2 were applied to a “full-set” file containing both of those experiments. This guaranteed that as much variability as possible was captured.

It is important to note that the time-series used as an atmospheric reference and the experimental data to be processed do not originate from the same device.



(a)

(b)

Figure 5: Example of a test image particularly affected by atmospheric perturbation and depicting a uniform slide of casein. The sharp absorption peaks of water vapour are particularly noticeable in the $1200 - 2000$ cm^{-1} region. The carbon dioxide peak is also present at 2400 cm^{-1}. (a) shows the total intensity; (b) shows a set of representative spectra. Note that differences between spectra are mainly due to thickness and atmospheric gas content variation.

Raw data The raw infrared absorption images on which the atmospheric correction methods were tested consisted of a set of particularly "watery" data. An example is depicted in Figure 5. These very compromised images were chosen to make the effect of the correction as clear as possible. However, this limited the potential performance of a correction: if some data was effectively lost to atmospheric perturbations, no amount of correction could recover it. Therefore, for the sake of comprehensiveness, the algorithms were also applied to less compromised data. This allowed to make sure that they did not try to remove nonexistent atmospheric perturbations. Indeed, a badly designed correction could theoretically add defects where they previously were none, which would be counter-productive.

These experiments also had different substrates, as the shape of substrate spectra could influence the fitting of the model to the data. An adequate correction needs to perform well whatever the substrate might be. The badly compromised images were acquired from a casein sample, and the less compromised images from a lignin one.

These experiments were of known composition, being only of a substrate (casein or lignin), allowing us to know, to some extent, what output to expect after proper correction.

2.2.2 Algorithm

Why is filtering unacceptable? As the water vapour peaks—which make up the bulk of what we are trying to correct—are features of high frequency in the wavenumber domain, it is tempting to simply filter them out of the otherwise smooth spectra. However, water vapour is not the only compound to have such sharp peaks, and using this method would expose us to the elimination of some potentially important data. Additionally, filtering can skew the absorption levels around the peaks, which is not acceptable in our applications, where the concentration of chemical species are crucial outputs of the analysis. Therefore, we need a method which only removes the atmospheric perturbation without taking the risk of corrupting the rest of the data.

Moreover, the spectra of water vapour and carbon dioxide are known. This is a very useful piece of information which a simple filter cannot use.

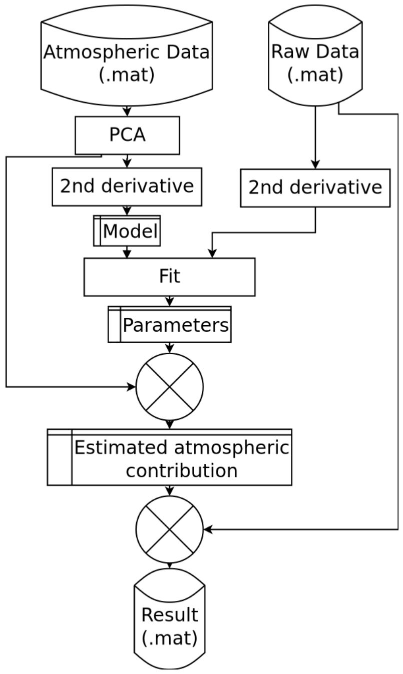


Figure 6: Flowchart of PCA-based atmospheric correction method.

Principal Component Analysis To improve on the current correction methods, we had to account for some variability in the atmospheric contribution (variability for which we did not have a model). To that end, we needed to extract several uncorrelated components capable of representing all or a large portion of the atmospheric contribution, which could then be used to build a linear model to fit to the data. Extracting those types of components is the objective of Principal Component Analysis (PCA), and it is, therefore, this tool that we used.

PCA-based approach We extracted the components of our model from the data mentioned in Section 2.2.1. Specifically, we applied PCA to the full set of 120 atmospheric spectra (depicted in Figure 4). This allowed us to extract several independent components which reflected the variability in atmospheric spectra. This method (summarized in Figure 6) was assumed to present a potential improvement over the OCTAVVS one by accounting for more variability in the atmospheric perturbations. We found some of those components to be due to different physical processes at play in the atmospheric spectra: for instance, component 1 and 2 in Figure 7 show the spectrum of liquid water, probably deposited on some of the equipment's optics and soon evaporated. In realistic use-cases, liquid water is unlikely to be as important a factor as water vapour.

Selection of the number of PCA components to retain was done by evaluating the performance of the correction for 1 to 8 of those components. Performance was quantified using the scoring system devised in Section 2.2.3.

This procedure presented a disadvantage: the spectra appeared to *spread* after the correction.

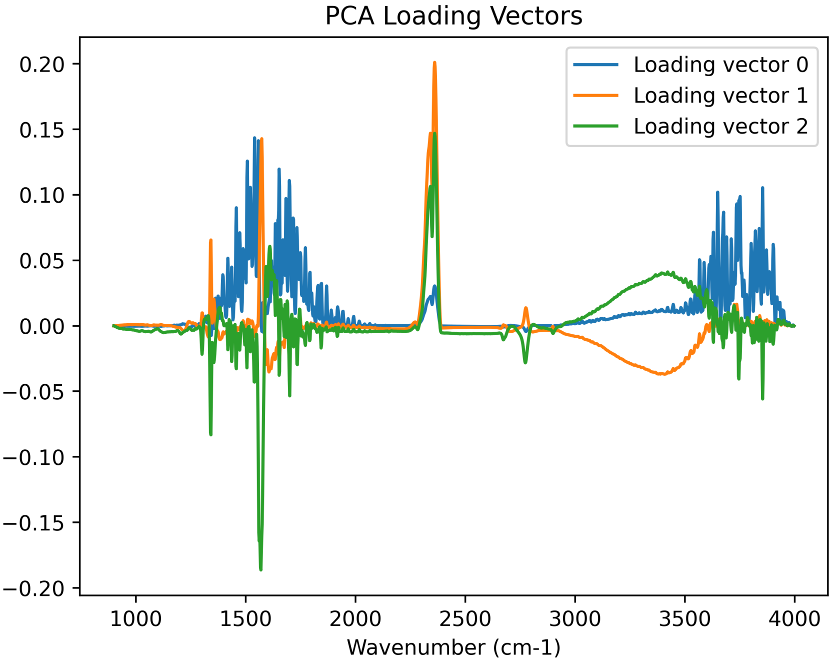


Figure 7: First three loading vectors from the PCA of the atmospheric reference data.

Spreading here refers to spectra which showed similar amplitudes in the raw data getting pulled away from each other, disturbing their overall shape. The more PCA components were considered in the model, the clearer this effect was. This is shown in Section 3.1.1.

This spreading was due to the components of the model (from the PCA of the atmospheric data) having a baseline. Since the model is fitted to the raw data by its second derivative, constant or slowly varying parts of the components become poorly defined. Therefore, such a baseline can be introduced into the correction and get subtracted from the data. To remedy this defect, we corrected the baseline of the atmospheric spectra prior to PCA, by drawing a straight line between the absorbance at the first and last point of the region, or by using a stiff AsLS [4] or a Savitzky-Golay filter. The results for each of these baseline-correction methods are studied in Section 3.1.1.

Gaussian peaks approach The previous approach (PCA-based model from the calibration experiment) showed that very few independent components could be extracted

[4] *Asymmetric Least-Squares* usually abreviated as *AsLS*

from the water spectra, and that it was not enough freedom for the model to significantly outperform OCTAVVS.

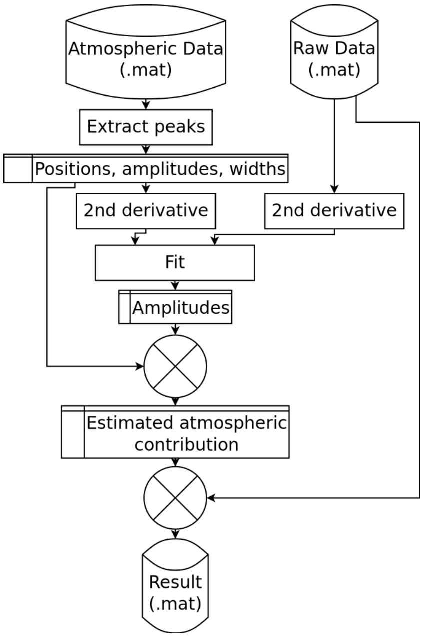
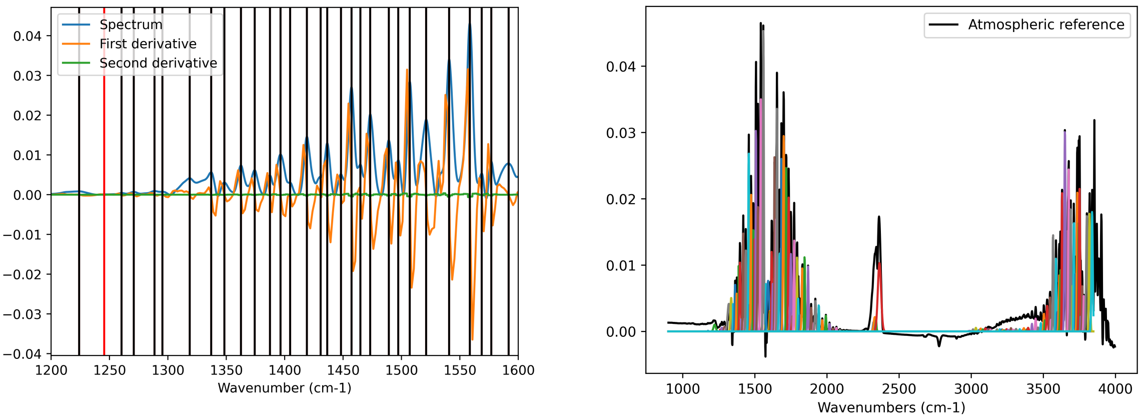


Figure 8: Flowchart of the Gaussian peaks atmospheric correction method.

To fix this, we completely shifted the way we built our atmospheric model: instead of considering the atmospheric spectrum as the sum of the PCA components, what if we considered it as the sum of its peaks? To that end, 157 Gaussian peaks were fitted to the data. First, their number and positions were determined using the derivatives of the spectrum: the first derivative was interpolated to find at which points it went from positive to negative, indicating a peak. Then, those points were selected if the second derivative (also through interpolation) was above a threshold, allowing to select only sharp enough peaks (see Figure 9a). This simple procedure required a single pass and was therefore much faster than an iterative approach.

In the literature, the absorption peaks of water vapour have been modelled by Lorentz [10] as well as by Gaussian distributions [15]. Figure 12 shows the shape of these distributions. A comparison of the fitting of Gaussian and Lorentzian distributions to the atmospheric reference showed the Gaussian peaks to be the better choice: We measured the sum of the squared difference between the atmospheric spectrum and the sum of fitted peaks. We found a value of $1.4 \cdot 10^{-3}$ with Lorentzians and $5.4 \cdot 10^{-4}$ with Gaussians. This difference of accuracy is noticeable in Figure 10. The positions of detected peaks, and intensities at those positions were used, along with an arbitrary width, as a first guess of the Gaussian peaks parameters. Then, these parameters (positions, widths, and amplitudes) were optimized to fit the atmospheric spectrum, as seen in Figure 9b.

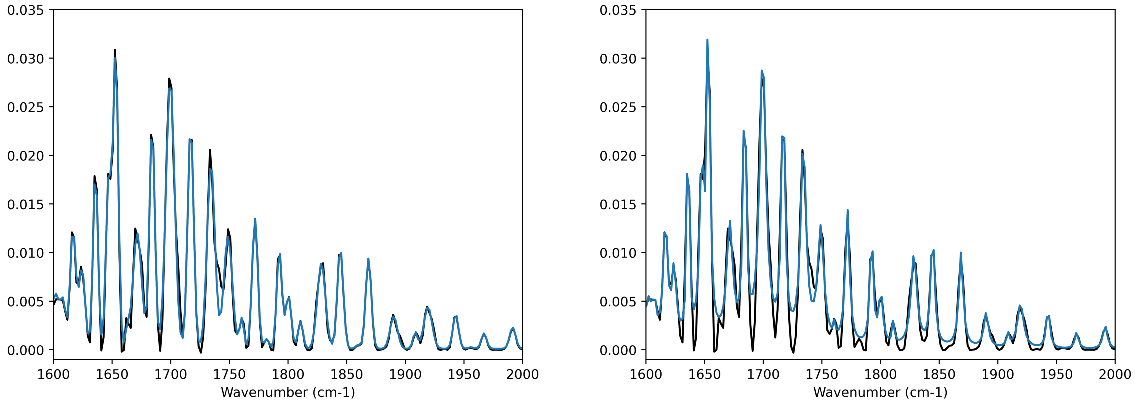
This process provided a model of the atmospheric spectrum as a sum of Gaussian peaks. These peaks, having been fitted to raw experimental data (through their second derivative, see Section 2.2.2), could be removed from damaged data, as a correction. This process is summarized in Figure 8. However, with such a large number of peaks each having an amplitude, position and width, there was a risk of overfitting as well as potentially long computation times. Thus, such parameters had to be constrained.



(a) Peak detection. In black: the final set of positions; in red: peaks which were dismissed as too close to 0 or not sharp enough. Region from 1200 to 1600 cm^{-1} enlarged for clarity.

(b) Least-square fitting of Gaussian peaks widths, amplitudes and positions to an atmospheric reference spectrum.

Figure 9: Detection and fitting of peaks to an atmospheric reference spectrum.



(a)

(b)

Figure 10: Comparison of the fitting of (a) Gaussian or (b) Lorentzian distributions to the atmospheric absorption spectrum. Region from 1600 to 2000 cm^{-1} enlarged for clarity. The better fit of the Gaussian distributions is clearly visible in the valleys between peaks.

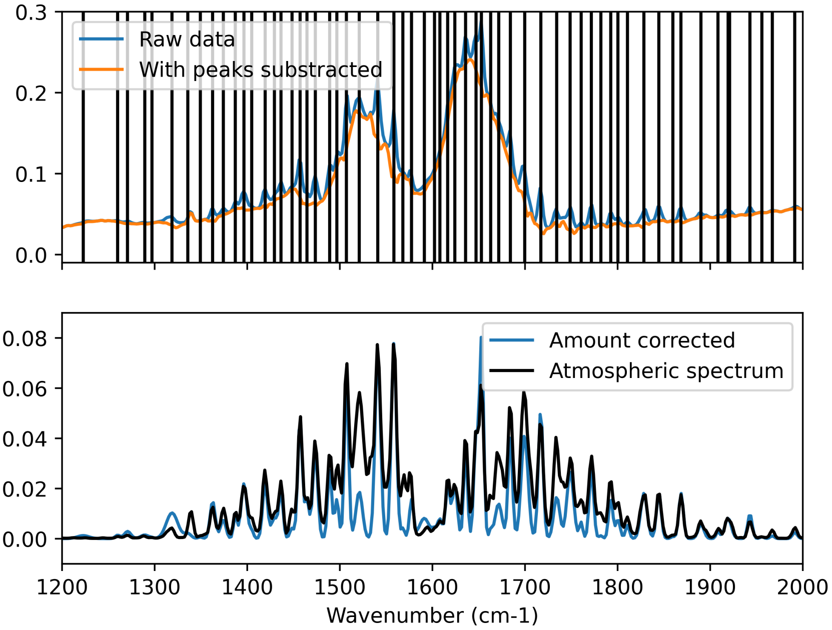


Figure 11: Result of the Gaussian peaks approach to atmospheric correction on a particularly affected spectrum. Top: Raw (blue) and corrected (orange) spectra. Bottom: Difference between raw and corrected spectrum; atmospheric spectrum for reference. Example spectrum depicted is the best corrected among the spectra in the image when measured using our scoring system. Positions of peaks used for the correction are indicated by black vertical lines.

- Position: Due to the infrared absorption spectroscopy process, we can safely assume that the positions of the peaks would not vary. The positions used were, therefore, the same as the ones extracted from the calibration experiment.
- Width: Similarly, the width of the peaks did not vary much between measurements.
- Amplitude: This was the only parameter likely to vary in the experimental data.

Since only the amplitude of the peaks could vary, they could easily and efficiently fitted to the data as a linear model:

$$Z(\nu) = \sum_i Z_i \exp\left(-\frac{(\nu - \nu_i)^2}{w_i}\right) \tag{6}$$

With Z_i (ν_i, w_i, respectively) the amplitude (position, amplitude, respectively) of the peaks. The Z_i, ν_i and w_i are found by fitting to the atmospheric data, but only Z_i vary when fitting to the apparent spectra. Here, 157 of those Gaussian peaks were found.

This model allows us to correct spectra as depicted in Figure 11. However, as seen in the aforementioned figure, this method still either over- or under-corrects some peaks. It also runs the risk of correcting any peak which would happen to coincide with a water vapour peak.

Work with the second derivative In most of the correction methods mentioned above, we fitted the second derivative(s) of our model to the second derivative of the data. Bruun et al. [8] used this method when developing their correction, as well as separating the spectra in regions of interest. This is due to the atmospheric spectrum being added to spectrum of the substrate, which has much wider features. Therefore, we need to only fit the atmospheric peaks of the experimental data to our model, while disregarding the larger features (which are part of the substrate spectrum): we need to focus on high frequency features (that is, high frequency in the wavenumber domain).

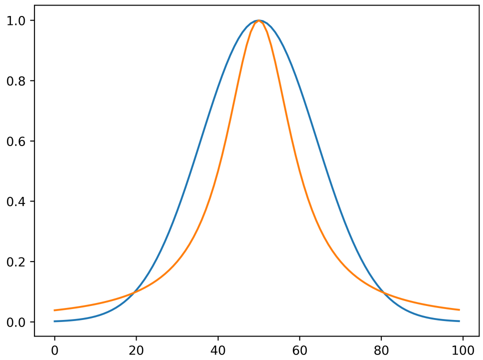


Figure 12: Gaussian (blue) and Lorentzian (orange) or Cauchy distribution.

To that end, we tried several “high-pass” methods:

- First, second or third derivative.
- Butterworth filter, with varying parameters. Because water vapour peaks are high-frequency features, they could be selected using a high-pass filter. The filter should not affect the features going through it, so the Butterworth filter was selected for its very flat passband.
- Savitzky-Golay filter, with varying parameters. This filter had shown good results in denoising IR spectra in the literature [10].

We found the second derivative to lead to the best overall correction results when measured with our score. (See Section 2.2.3 for an explanation of the scoring system.) This is most likely due to it being the most accurate in selecting the appropriates features. The acceptable results obtained with the Butterworth filter might point towards the possibility of achieving similar results as the second derivative through better tuning of the filter. This would require testing of numerous Butterworth parameters over a large sample of spectra with varying water vapour contribution (as the wider carbon dioxide is not taken into account here).

Fitting method As mentioned above, we needed to fit the second derivatives of the atmospheric model to the raw data in order to evaluate how much of it was to be removed. Since the amount of water vapour varies unpredictably between neighbouring pixels, this fitting had to be done for each of the 4096 spectra contained in each image. For an experiment of 96 images, this leads to over 393 000 applications of this process. This is one of the reasons why the correction could only be carried out in realistic time through a linear, unconstrained optimization. Luckily, this EMSC model is linear and was easily passed through Numpy's linear least-square solver *lstsq* [16], allowing the whole algorithm to complete in a few seconds.

Focus on H_2O over CO_2 In most of this work, we focus our attention on correcting water vapour perturbation in the bands while disregarding the carbon dioxide peaks at 2345 cm^{-1}. This is because the carbon dioxide peak affects a spectral band devoid of biologically important resonances, whereas water vapour impacts the "fingerprint region" (between 1800 and 1000 cm^{-1}) crucial in identifying certain compounds.

Analysis by band We divided our correction in three parts, one for each of the three regions of interest:

- Water vapour 1: 1200 – 2000 cm^{-1}
- Carbon dioxide: 2250 – 2420 cm^{-1}
- Water vapour 2: 3350 – 4000 cm^{-1}

This allows for a more in-depth analysis of how different parts of the atmospheric spectra relate to each other. As the first and third band are both due to the same compound (water vapour), we expect the amplitudes of the corrections there to evolve together. By plotting how much of the atmospheric correction needed to be applied in each region, we indeed notice the existence of a relationship between them. This is shown in Figure 20.

This test was carried out using the PCA-based method[5] as it is—by nature—a band-by-band correction (contrary to the Gaussian peaks method, which is a peak-by-peak correction).

2.2.3 Validation

In this section, our improved technique is compared with the OCTAVVS method[6] and to the commercially available OPUS software method [17].

The *OCTAVVS method* uses filtering and replacement of the CO2 spike by a spline. Since our correction score is biased towards such techniques (as it favours a smooth spectrum) even though they result in some information loss, we deactivated those features. This allowed for a more adequate comparison to the results of the *PCA-based* and *Gaussian peaks methods.*

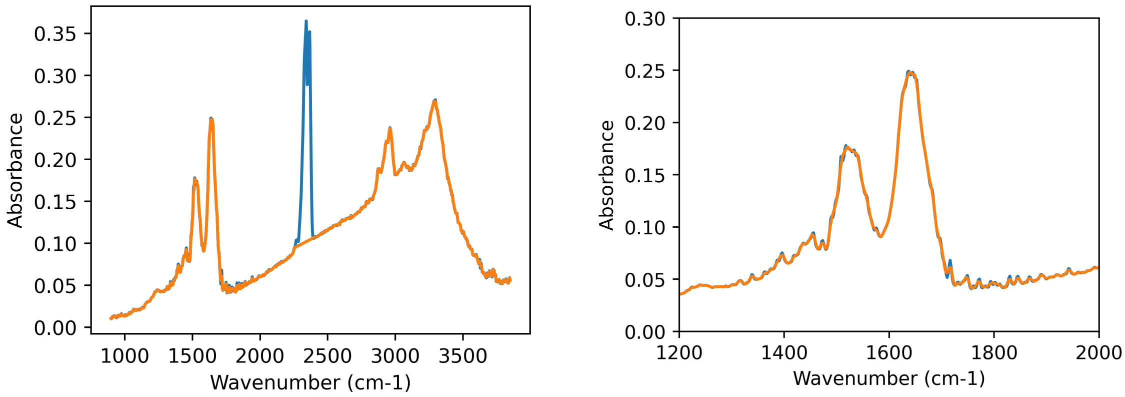
[5] See Section 2.2.2.

[6] See Section 1.2.1.

The following comparisons were performed on three files judged to contain a large amount of water vapour perturbation. Each of the files contained the scan of a casein film on 4096 pixels.

Visual comparison Since the sharps, narrow peaks caused by the water vapour, and the wide, isolated carbon dioxide peak, are evident in our data set, it is relatively easy to visually assess if the correction eliminates those peaks or not. Therefore, regular use was made of the visualization functions of the OCTAVVS preprocessing tool, and most of the scores mentioned below were devised to quantify and measure effects noticed through visual observation.

However, although it provides valuable insights, this method of evaluating results is neither reliable of measurable. Hence the need to numerically score our attempts.



(a) (b)

Figure 13: Comparison between a raw and *estimated ideal* spectrum, used for the correction score. In blue: raw spectrum. In orange: the same spectrum after filtering and cutting of the carbon dioxide peak, resulting in the *estimated ideal spectrum.* (a) shows the full spectrum. (b) shows the detail of the first water vapour region.

Correction score The correction score determines how much of the atmospheric perturbation was effectively removed by the correction. Since the correct, “clean” spectrum is not known, however, we have to rely on another way to separate perturbation from data. The method chosen here is to separate them based on the aspect of their features: the substrate is assumed to be made of wider peaks, while the atmospheric spectrum is composed of narrow ones. We applied a Savitzky-Golay filter (of window size nine at the third order) on the raw data to approximate the spectra we would like to obtain, while replacing the carbon dioxide peak with a straight slope. See Figure 13 for an example. Then, we considered the squared difference between these *estimated ideal spectra* and the corrected spectra as the amount of atmospheric perturbation left after correction. This was done for all 4096 pixels in the image. We focus on the ratio of this quantity over the amount of atmospheric perturbation present in the raw data. This reflects the improvement of the spectra: the lower the score, the better the correction.

$$\text{score}_{\text{correction}} = \frac{1}{n_{\text{samples}}} \sum_{i \in \text{Samples}} \frac{\sum_\nu (\text{corrSpectrum}_i - \text{savGo}(\text{corrSpectrum}_i))^2}{\sum_\nu (\text{rawSpectrum}_i - \text{savGo}(\text{rawSpectrum}_i))^2} \tag{7}$$

However, the way this score was devised makes it biased towards smoother solutions, whether or not they are accurate. As none of the techniques developed and studied here relies on the filtering of the data (with the exception of the OPUS software, used further for comparison with commercially-available methods), such an effect should only be due to a good correction.

Spreading score In the PCA-based methods, while the correction gets better with more components, it also introduces a new type of error, referred here as *spreading*, which the correction score cannot take into account. By spreading we designate the effect by which spectra originally similar were affected by an unpredicted additive constant.

The spreading was found to be due, at least to some extent, to PCA components used for the linear model having baselines. Then, when their second derivative was fitted to the second derivative of the data, the baselines were carried into the correction. This is addressed by removing the baseline out of the PCA components before fitting the model to data. However, the prevalence of this effect for different parameters still has to be quantified. Hence the spreading score: we evaluate by how much the corrected spectra shift away from their mean, and compare it to how far the original spectra were from their mean.

Therefore, a spreading score above 1 shows that the correction "spreads" out the spectra, while a score below 1 shows that they tend to gather together. As eliminating the atmospheric perturbation would lower the spectra variability, we expect a good correction to have a spreading score slightly below 1.

$$\text{score}_{\text{spread}} = \frac{1}{n_{\text{samples}}} \sum_{i \in \text{Samples}} \frac{\sum_\nu (\text{corrSpectrum}_i - \text{mean}(\text{corrSpectra}))^2}{\sum_\nu (\text{rawSpectrum}_i - \text{mean}(\text{rawSpectra}))^2} \tag{8}$$

2.3 Scattering correction

In this Section, we study the behaviour of the clustered scattering correction mentioned in Section 1.2.2, and devise a way to validate its results without relying on visual observation.

In order to improve the scattering correction from OCTAVVS, we focused on improving the clustered part of the method, as the correction of individual spectra already gave good results (as measured with our score describded in Section 2.3.2) and relied on algorithms known to work reliably [4, 2]. The clustering, however, was deemed responsible for the introduction of defects in the output spectra: mainly, since spectra clustered were corrected using the same reference, they would become more similar, potentially eliminating key differences in chemical composition between them. This is studied in the next section.

After explaining our approach to improving this part of the software, we will detail the method used to validate its results against synthetic data.

2.3.1 Algorithm

Resonant Mie scattering correction The approach currently used within OCTAVVS relies on a Resonant Mie Scattering (RMieSC) model which can be applied in a clustered way (CRMieSC).

The main part of the algorithm consists in calculating the extinction spectra for a large number of potential particles using the *van de Hulst approximation* described in Section 1.2.2. Then, fitting the sum of those contributions (plus a reference) to the apparent spectra, we reach an Extended Multiplicative Signal Correction (EMSC) model. Subtracting this from the apparent spectra effectively corrects the scattering effects. The resulting (corrected) spectra can then be used as a better reference for the next iteration. This method eventually converges[12], providing us with a scattering-corrected spectrum.

A high number of iterations (here, upwards of 30 iterations) is required to approach convergence. However, each iteration adds a small risk of introducing artefacts into the corrected data. To limit this effect, the residuals of the spectra's correction are used to determine the amount of scattering removed after each iteration. Then, spectra which are not improved are not considered in the following iterations.

In the clustered approach, pixels are grouped in clusters before each iteration; the set of extinction spectra are then calculated based on the cluster centre and used to correct each of the pixels in that cluster[7].

Reference improvement The clustered approach used in OCTAVVS considers the cluster centres as references for the correction of all pixels in the cluster; uses this reference to derive a RMieSC model; and then subtracts the fitted model, including the reference spectrum, from each spectrum in the cluster (see Section 1.2.2).

This RMieSC model at iteration $n > 1$ can be summarized as follows:

$$Z_i^{app}(\nu) = a_n + b_n Z_{clus_j}^{corr,n-1}(\nu) + d_{1,n}\nu + \sum_k g_{k,n} p_{k,n}(\nu) + e_n(\nu) \tag{9}$$

Where Z_i^{app} is the apparent (raw) spectrum of pixel i, and $Z_{clus_j}^{corr,n-1}$ is the previously corrected spectrum of the centre of the cluster containing pixel i, acting as a reference

[7] See Section 2.5.1 for details on clustering.

spectrum.

(Note: at the first iteration $n = 1$, since no spectrum as yet been corrected, the spectrum of the substrate is used as a reference. OCTAVVS provides spectra for lignin, casein and matrigel.) Therefore, the same reference would get subtracted from all pixels in a cluster, and it is this reference which would get updated in the resonant approach. This lead to some inaccuracy as the pixels within a cluster could stray significantly away from its centre. Also, in the iterative process, this caused individual pixels within a cluster to become more and more similar after each iteration, as they were corrected together (using the same reference and extinction spectra). As we do not want the cluster scheme to have much effect on the final aspect of the spectra, this was a considerable problem.

However, this can be considerably improved by subtracting each pixel's reference (in the iterative case, the previously corrected version of it) to it. Then, each pixel is corrected relatively to its reference, and not to one common for the whole cluster. To that end, we subtract from each pixel its individual reference, considered as the projection of the previously corrected spectrum onto the apparent spectrum.

The RMieSC model with our improved reference is then, at iteration $n > 1$:

$$Z_i^{app}(\nu) = a_n + b_n Z_i^{corr,n-1}(\nu) + d_{1,n}\nu + \sum_k g_{k,n} p_{k,n}(\nu) + e_n(\nu) \tag{10}$$

Note that the reference spectrum was changed from $Z_{clus_j}^{corr,n-1}$ in Equation 9 to $Z_i^{corr,n-1}$, the spectrum of pixel i corrected at iteration $n - 1$.

This way, the result of the subtraction only contains the scattering components which the previous step(s) failed to correct. This difference is then fed to the fitting function[8] (instead of the raw spectrum, as in the method from OCTAVVS), for the intensity of each extinction spectrum (after PCA dimensionality reduction) to be estimated. The fitted model is then used to correct the raw data itself.

Cluster mixing Additionally, to prevent the clusters from drifting away from one another, we removed the assumption that each pixel was part of a single cluster. Instead, after clustering spectra, we consider them as a mix of all clusters. Hence, the correction is computed for each pixel as part of every cluster, and those corrections are then averaged. The average correction is weighted by the inverse of its distance to the centres of the clusters, to some power:

$$Z_i^{\text{mixed}}(\nu) = \sum_{j<n_{\text{clusters}}} w_{i,j}\ Z_i^{\text{clus}_\text{j}}(\nu) \tag{11}$$

With ν the wavenumber, Z_i^{mixed} the final corrected ("mixed") absorption spectrum of pixel i, and $Z_i^{\text{clus}_\text{j}}$ the absorption spectrum of pixel i corrected as part of cluster j. $w_{i,j}$ is the

[8] The fitting function consists of a least-squares optimization on a EMSC model. See Section 2.2.2.

weight of cluster j's correction on pixel i, defined as:

$$w_{i,j} = \frac{1}{d(Z_i^{\text{pixel}}, Z_j^{\text{clus center}})^a} \tag{12}$$

Where a is the *power parameter* which affects the computation of the weights: closer clusters are favoured when it is high. The distance d is calculated as follows:

$$d(X_1, X_2) = \sum_{\nu} (X_1(\nu) - X_2(\nu))^2 \tag{13}$$

When using this *cluster mixing*, spectra are mainly corrected as their most similar counterparts, with some contribution from other clusters. This allows the method to bypass one of the main drawbacks of the clustered approach: that potentially dissimilar spectra were corrected together.

2.3.2 Synthetic data

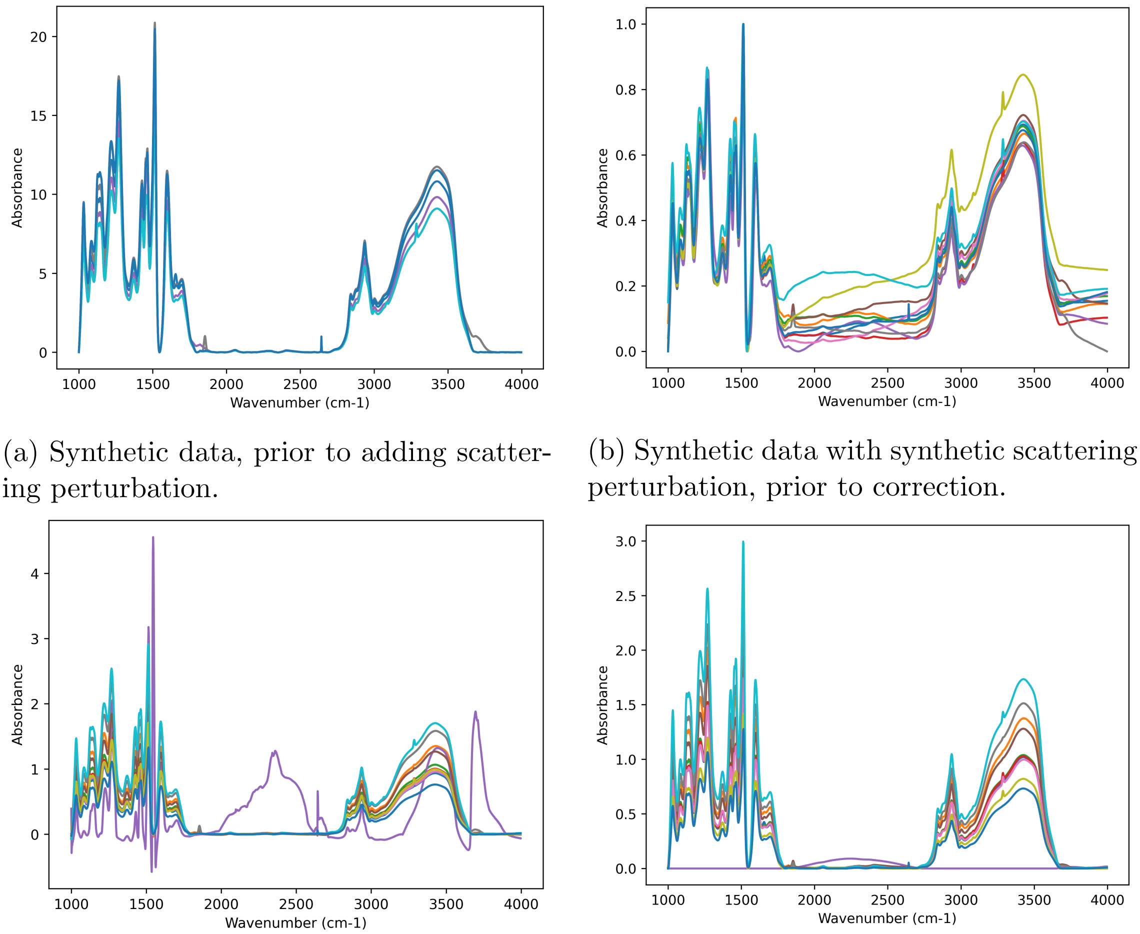
Since we do not know exactly what the output, scatter-free spectra should look like, it is difficult to assess the efficacy of the scattering correction on measured spectra. We, therefore, designed our own method of generating spectra inspired by Rasskazov et al. [18] and Bassan et al. [12]:

Firstly, we sourced a lignin spectrum (representing the substrate; captured with transmission spectroscopy and free of scattering effects), and added Gaussian peaks of random position, intensity and width (representing significant biological signal). Secondly, extinction matrices were calculated for a set of different particle sizes and refraction indexes:

- Particle sizes: 10 values from $4\pi \cdot 10^{-4}\,\text{m}$ to $8\pi \cdot 10^{-4}\,\text{m}$
- Refraction index: 10 values from 1.1 to 1.4

Among those extinction spectra, five were picked, given a random intensity, and added to the spectrum. Doing this for 4096 spectra produces two files containing biological signals: one unperturbed or "clear" image, the other one affected by scattering. (See the first two subfigures of Figure 14 for an example.) Comparing the output of the correction to the clear image, therefore, gives an estimate of how well that correction method performs.

The extinction matrices were produced from the Bassan algorithm [12], and the data was corrected using the Konevskikh algorithm [14]. This method was chosen to not take the risk of creating too easily corrected spectra.



(a) Synthetic data, prior to adding scattering perturbation.

(b) Synthetic data with synthetic scattering perturbation, prior to correction.

(c) Corrected by clustered scattering correction from OCTAVVS, before improvements.

(d) Corrected by clustered scattering correction with our improved reference.

Figure 14: Comparison of scattering correction before and after our improvements. The correction was applied on synthetic data, and 40 spectra (corresponding to the same raw ones) were plotted. All corrections were applied with 30 clusters and automatic iteration control (max. 30 iterations).

Scoring To estimate how well corrections performed on this artificial data, we designed a score to quantify it. We have access to a "true" (or "clear") image, made of synthetic biological absorption spectra over a substrate, a "scattered" image which is the same one but subjected to generated extinction spectra, and finally, we have a "corrected" image when the scattered one has been run through the correction. To assess the correction's efficacy, we measure the distance of each corrected spectrum to its original clear counterpart. This distance is measured as the sum of the squares of the difference between spectra: $\sum_{\nu}(s_1(\nu) - s_2(\nu))^2$. If the correction performed well, the clear and corrected spectra should be relatively similar. Hence, the lower the score, the better. However, this is affected by the amount of scattering introduced into the sample, so we divide this distance (between the clear and corrected spectra) by the distance between the clear and scattered spectra.

$$\text{score}_{\text{synth}} = \frac{1}{n_{\text{samples}}} \sum_{i \in \text{Samples}} \frac{\sum_{\nu} (\text{corrSpectrum}_i - \text{clearSpectrum}_i)^2}{\sum_{\nu} (\text{scattSpectrum}_i - \text{clearSpectrum}_i)^2} \tag{14}$$

A score of zero is the optimal, as it represents a total elimination of all scattering.

2.4 General validation

As mentioned in Section 2.6, we encounter a difficulty in evaluating how well corrections behave: In the atmospheric correction part of this work, scores were devised to quantify the effects of the various correction methods. [9] However, those scores are only affected by very specific aspects of the spectra (the high-frequency components for the "correction score", the variance for the "spread score"). This process is hardly applicable to scattering corrections, as the extinction spectra can—depending on the particles sizes and refraction indexes—have widely varying shapes. It is therefore difficult to separate perturbations from important data and quantify how well they are removed.

Synthetic data was also useful in estimating how well corrections perform against artificial disturbances[10], but this data does not represent any real sample. Even though it simulates reality, synthetic data can never equal it[11].

To solve this, we evaluated how well a classifier could distinguish spectra after correction and compare it to the same classification on raw spectra. This method was chosen for its similarity with potential applications of those algorithms: The aim of preprocessing treatments of infrared absorption images is to make the separation between different chemical compositions easier. Therefore, if we see a significant improvement in the classification after correction, it would prove the usefulness of our algorithms.

2.4.1 Data

Two experiments (labelled EX79 and EX80) were used. They each consist of 96 images of *Paxillus involutus* hyphae on a lignin substrate [5]. These experiments represent a fairly accurate example of what a common use-case of OCTAVVS would be: analysing the biochemical composition of cells growing on a substrate.

Our data set allowed us to distinguish between different parameters:

- Foreground/Background pixels

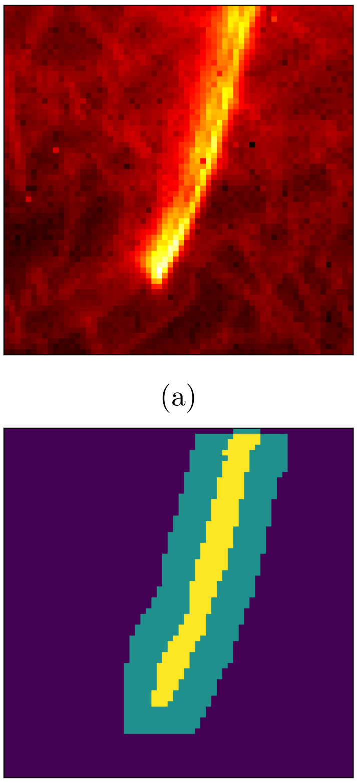
[9] See Section 2.2.3 for a detailed explanation of these scores.

[10] See Section 3.2.1 for an example of how synthetic data was used in this work.

[11] The limitations of synthetic data are discussed in Section 4.2.3.

- SL1/SL2: slides with/without ferrihydrite
- EX79/EX80: growth medium with/without ferrihydrite

2.4.2 Annotations



(a)

(b)

Figure 15: Example of annotated data. (a): Total intensity; (b): Corresponding annotation of the hypha and buffer area.

For the background vs. foreground classification, we needed a correct training set of foreground and background pixels. To that end, each pixel of each of the 96 images in both experiments was tagged as either foreground or background.

However, this failed to account for the existence of a halo of pixels which, although not fully part of the foreground, still carry some signal from it. We defined a “buffer area” five pixels wide around the foreground, in which the pixels would not be considered as either foreground or background.

This resulted in one *annotation file* per image, an example of which is depicted in Figure 15.

2.4.3 Algorithm

We trained a linear classifier on the data/annotations pairs and evaluated how well it could determine whether a pixel/image was part of a class or the other: The raw data was used to train the classifier, passed through it, and its accuracy was measured. Then, processed data was subjected to the same treatment, and if there was an improvement in accuracy, that would indicate that the correction was useful.

2.5 Complexity reduction

The images studied in OCTAVVS usually[12] contain 4096 spectra of 1530 points each. On top of this, an experiment can easily require the analysis of hundred such images. Therefore, the complexity of correction algorithms becomes a major aspect of their value.

[12] OCTAVVS is capable of treating images of various sizes and spectral resolution. The values mentioned reflect a common use-case scenario.

2.5.1 Clustering

In the scattering correction, even the most optimized models still require the generation of a large extinction matrix for each spectrum to correct. This is computationally intensive. For instance, applying the scattering correction individually on all spectra (that is, without clustering) can require several days to complete. As discussed in Section 4.3.2 it is not acceptable for the use-cases considered here to have such long computation times. Therefore, a solution is needed to reduce the number of spectra on which these calculations are applied in full. OCTAVVS solves this by clustering spectra before correction using *K-Means clustering*, a scheme which minimizes variance within the clusters.

This process relies on the assumption that spectra which look similar were likely affected by similar scattering effects, and could, therefore, be corrected by the same model. The clustering is—when compared to other steps of the algorithm—a fast operation, but trades some precision for speed. Indeed, since it approximate each pixel's reference by the reference of its cluster, the corrected spectrum is approximate too.

Part of this work focused on improving the way clustering affect the scattering correction. It is detailed in Section 2.3.1.

2.5.2 Dimensionality reduction

As mentioned above, scattering corrections require the use of numerous extinction spectra: one for each sphere diameter/refraction index pair; with OCTAVVS' default parameter ranges, this leads to 100 extinction spectra to compute at each iteration for each pixel (or cluster). However, fitting many potentially similar spectra to a single raw spectrum leads to overfitting: With many similar $p_i(\nu)$, the sum $\sum_i g_i p_i(\nu)$ in Equation 2 can start to explain individual spectral points by single extinction spectra. To limit this effect, the extinction spectra generated in OCTAVVS are subjected to a PCA dimensionality reduction before getting fitted to the data. This comes at a price: the sum of the PCA components does not explain the full extent of the extinction matrix variance. In OCTAVVS, the number of PCA components to use at each step is chosen so it explains 99.96%[13] of the variance.

2.6 Validation

As we never perfectly know what the final spectra should look like, we encounter a difficulty in evaluating how well corrections behave: We cannot see how close our algorithms bring us to the "real" spectra, because we do not know what they are. This is why we often resort to visual observation to assess whether a correction was successful. This, however, is not quantifiable.

[13] 99.96% by default, but can be changed in the user interface.

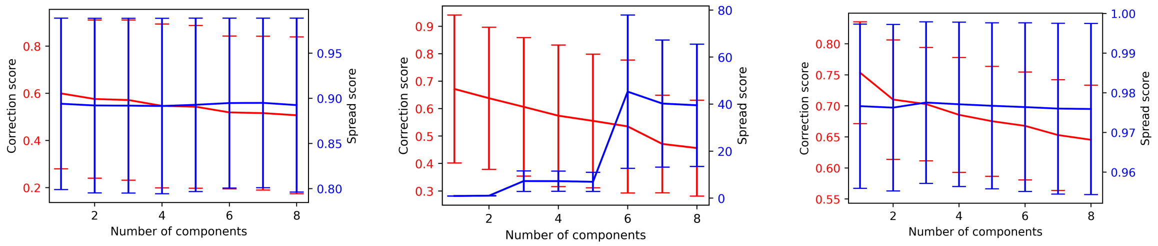
As no previous litterature provided a satisfying way of quantifying the effect of those corrections, we devised our own.

3 Results

3.1 Atmospheric correction

3.1.1 Parameter selection

We studied the "correction" and "spread" scores of our correction methods for various values of their parameters. This allowed for a refinement of the parameter space, as well as some tweaking of the methods themselves (such as the baseline correction of PCA components mentioned in Section 2.2.2). This iterative process leads us from the first PCA-based approach, to the second, and finally to the third, individual-peaks based, one. The final results are gathered in the following paragraphs.



(a) Band 1: Water vapour, 1200 to 2000 cm^{-1}.

(b) Band 2: Carbon dioxide, 2250 to 2420 cm^{-1}.

(c) Band 3: Water vapour, 3350 to 4000 cm^{-1}.

Figure 16: Variation of the correction (red) and spread (blue) scores as a function of the number of PCA components considered. One score was computed for each of the six (three noisy, three less affected) non-synthetic raw images in each band for each parameter value. Error bars show one standard deviation (one value per image).

Number of model components Figure 16 shows that, although adding PCA components to the linear model leads to a better overall correction score, it also leads to more spreading of the spectra, even with baseline correction of the PCA components. This is not acceptable in any of the potential applications of this method and therefore justifies keeping the number of PCA components low.

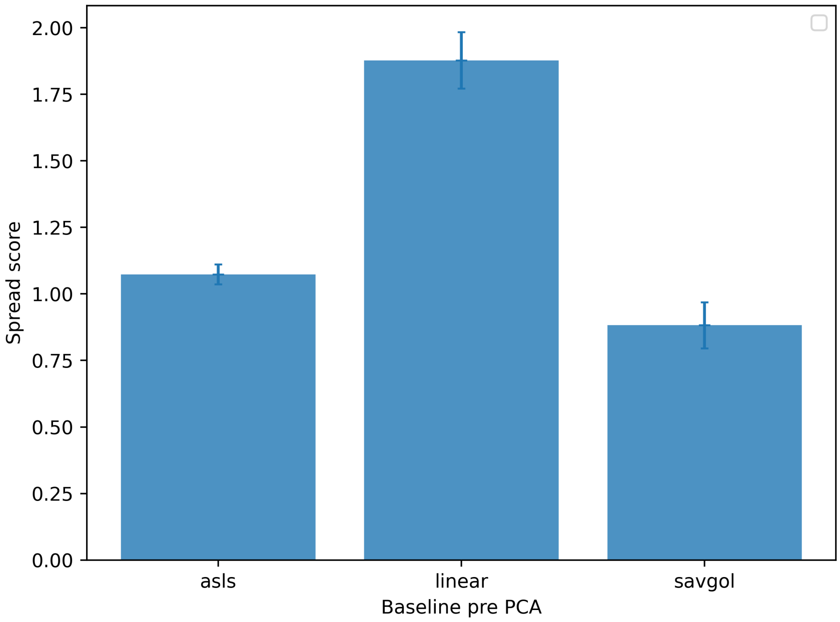
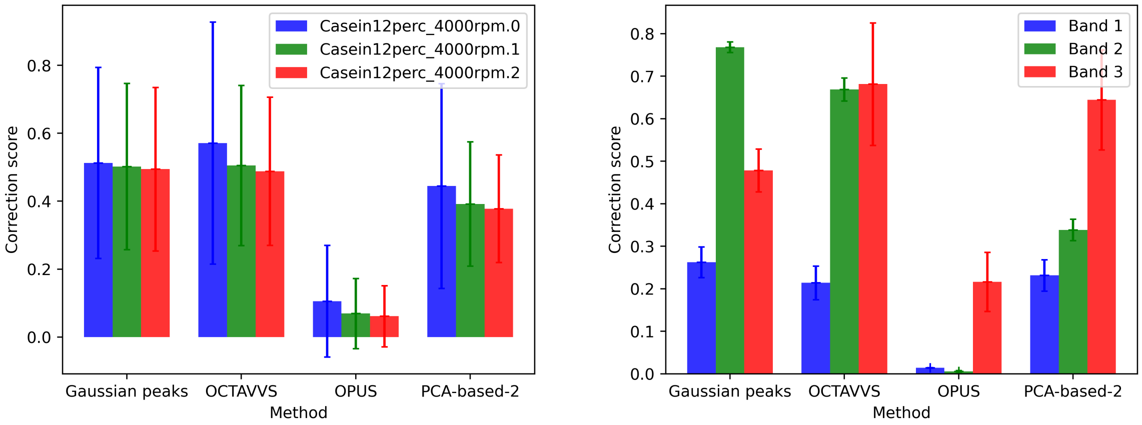


Figure 17: Comparison of the spreading score for the atmospheric correction, using different baseline correction methods prior to PCA. Error bars show one standard deviation. One score was computed on each of the two water vapour bands of each of the three “watery” files. AsLS parameters: $\lambda = 10$, $p = 0.001$. Savitzky-Golay window size: 45, order: 7.

Baseline removal As mentioned in Section 2.2.2, to limit spreading of the spectra, we had to remove a baseline from them. The three methods we used are compared in Figure 17. As the Savitzky-Golay filter (with window size 45 and order 7) gave the best result, it was selected as the baseline-correction method used in the rest of this work.

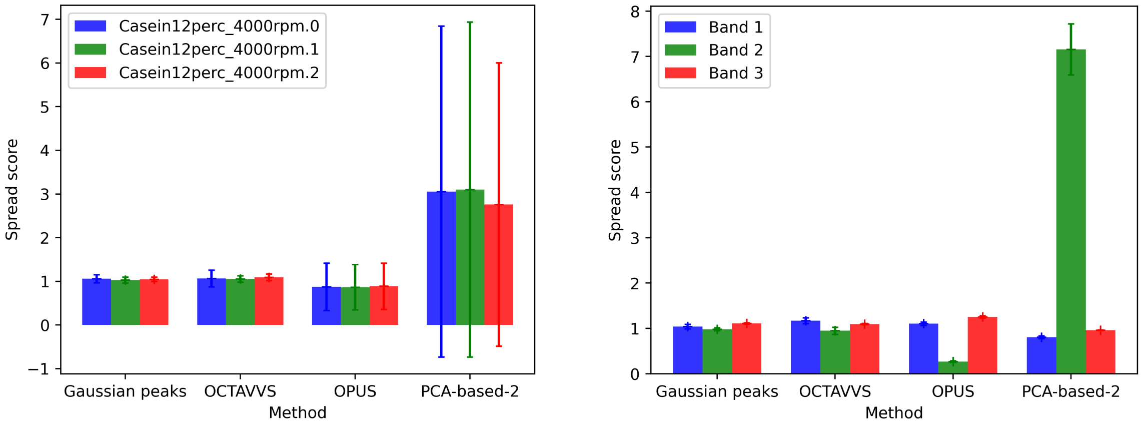
3.1.2 Methods comparison



(a)

(b)

Figure 18: Comparison of the correction scores for different methods of atmospheric correction. (a): comparison by file; (b): comparison by band (band 1 and 3: water vapour; band 2: carbon dioxide). One score was computed for each of the three “watery” images in each of the bands for each method. Error bars show one standard deviation.



(a)

(b)

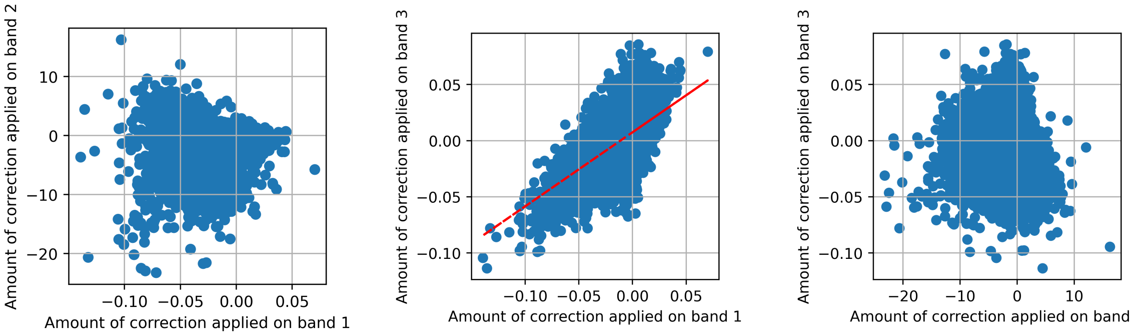
Figure 19: Comparison of the spread scores for different methods of atmospheric correction. (a): comparison by file; (b): comparison by band (band 1 and 3: water vapour; band 2: carbon dioxide). One score was computed for each of the three “watery” images in each of the bands for each method. Error bars show one standard deviation.

In Figures 18 and 19, our methods were compared to each other and to the commercially available OPUS software. Each was evaluated using the two scores (“correction” and "spread") designed in Section 2.2.3. To explain the outstanding performance of the OPUS software, one has to account for the fact that our "correction score" favours smooth spectra and is therefore heavily biased towards solutions using a noise filter. Such a low score suggests that OPUS uses such a filter to smooth the spectra, probably in conjunction with other treatments. Troein et al. came to a similar conclusion through examination of spectra corrected with OPUS, determining that "the final step of this correction consists of running the spectra through a smoothing filter, with more aggressive smoothing in the water regions than in regions without atmospheric contributions"[2].

As shown in Section 1.2.1, the atmospheric correction used in OCTAVVS can be set up to smooth the spectra after correction. As filtering is not considered in this work, and can be easily added to another method for further improvement of it, the scores labelled as OCTAVVS were obtained with the software set to not use filtering. This provides for a more level comparison between those methods.

3.1.3 Analysis by band

We compared the amount of correction in each band using the process described in Section 2.2.2 and plotted in Figure 20.



(a) Band 1 against band 2: First water region against carbon dioxide region.

(b) Band 1 against band 3: First water region against second water region.

(c) Band 2 against band 3: Carbon dioxide region against second water region.

Figure 20: Correlation of the amount of correction between bands. Each point represents the correction of one spectrum. Six images were used: three "watery", three less affected; each contained 4096 spectra.

We found a correlation between the first and third region with a slope of 0.66 ($r^2 = 0.41$). As both regions are affected by water vapour perturbation, we would expect a perfect correction to remove an amount in one band proportional to the amount in the other.

The negative values seen in Figure 20 can seem counter-intuitive as we do not expect a negative amount of water or CO2 to be possible. However, this is explained by the fact that the measuring device does a preprocessing step on the data by removing a previously measured baseline[14].

3.2 Scattering correction

3.2.1 Parameter selection

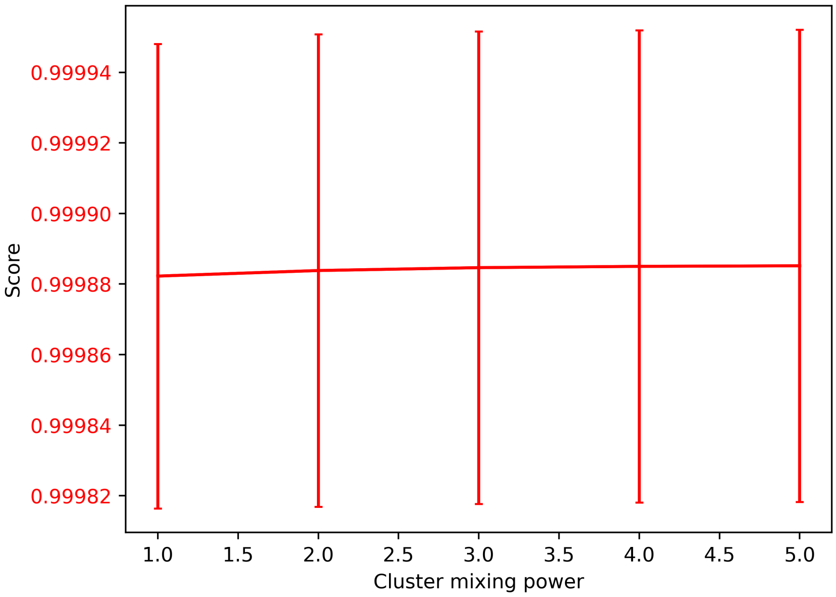


Figure 21: Variation of the score of a scattering correction with cluster mixing for different values of the power parameter a. One score was computed for each of the 50 synthetic images for each value of a. Error bars show one standard deviation.

Contrary to our atmospheric correction methods which required the exploration of a large parameter space (number of PCA components in each band, baseline methods, etc.), our scattering corrections only required the study of one parameter:

In the cluster mixing approach detailled in Section 2.3.1, we introduced the *power parameter* a affecting the computation of weights from distances.

Figure 21 shows that the power parameter a does not significantly affect the quality of the correction. Therefore, all following *cluster mixing* corrections were performed with a power parameter of 1.

[14] As described in Section 1.2.

3.2.2 Methods comparison

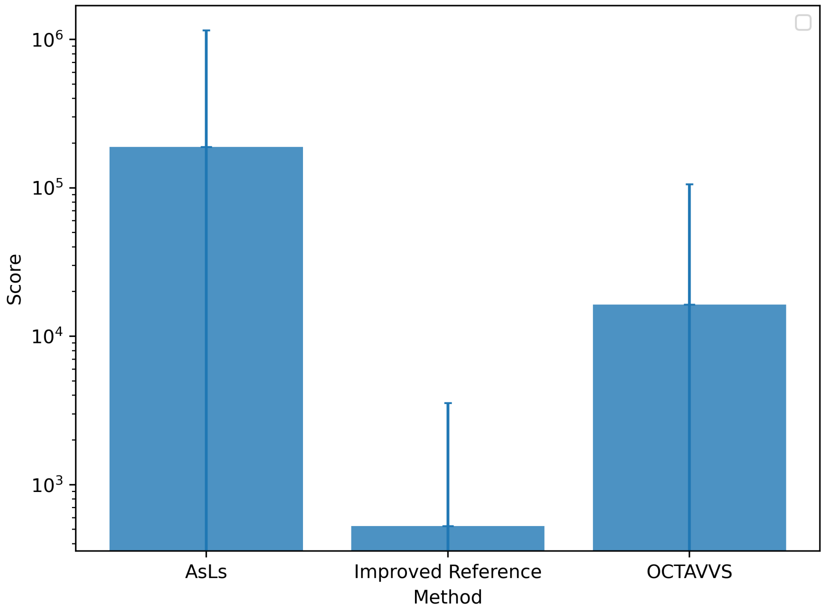


Figure 22: Comparison of the synthetic score of correcting 50 synthetic images using either the CRMieSC approach from OCTAVVS (i.e. with cluster centers as references) or our improved reference method. Error bars show the standard deviation across images (1 score per image). AsLS with $\lambda = 10^6$ and $p = 0.01$ for reference.

Figure 22 shows that the change to a more specific reference (explained in 2.3.1) lead to an improvement in the quality of the scattering correction.

Removing a stiff AsLS baseline ($\lambda = 10^6$, $p = 0.001$) gave out interesting results when used as a scattering correction as it removed the broad features of scattering perturbation without having to model them. This is a significantly less complex approach than RMie scattering correction. However, as seen in Figure 22, this baseline approach did not provide results comparable with the actual scattering correction.

3.3 General validation

Foreground detection For each image, the linear classifier was trained on all pixels, with each one having the label 1 if part of the foreground or 0 if part of the background. This is depicted in Figure 23.

Slide classification For this classification, we trained the classifier on the average foreground spectrum minus the average background spectrum, resulting in one training spec-

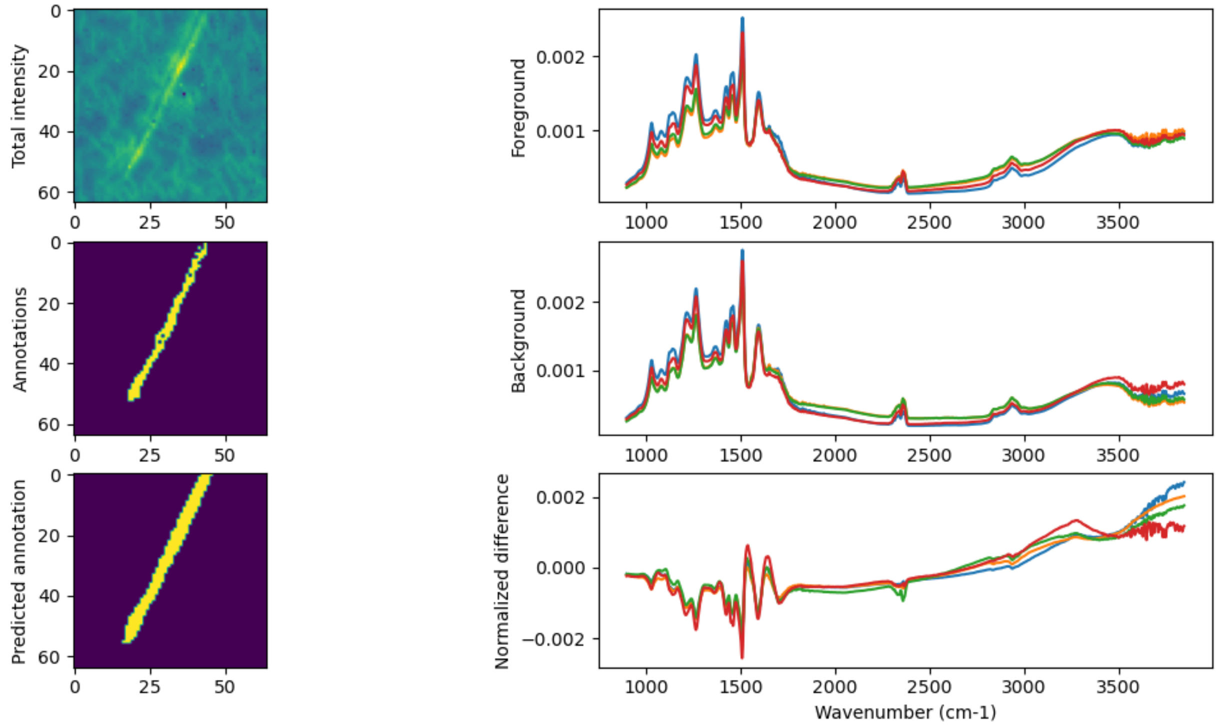


Figure 23: Example of the training process image-by-image. The spectra from a raw image (top left) are divided in foreground (top right) and background (center right) according to the corresponding annotation file (center left). This is used to train a linear classifier, which is then applied to the raw data to predict whether each pixel is from the background or foreground (bottom left). The normalized difference between foreground and background (bottom right) is shown to confirm whether an actual distinction exists.

trum per image. This allowed averaging out the variability among pixels, and therefore providing more reliable training data.

It was attempted to classify the pixels one by one. This way, scattering effects would be at their greatest, and if any improvement should exist between raw and processed data, it would be made obvious.

To that end, the classifier was—as previously—trained on the differences between average foregrounds and average backgrounds. Then, the classification was applied to the foreground pixels (the average background having been subtracted from them) and the results compared between raw and scattering-corrected data.

Experiment classification We classified images as experiment 79 or experiment 80. To that end, we used the same process as the slide classification (training on the difference between forground and background and classifying either individual pixels or full images).

3.3.1 Foreground detection

The output of the classification procedure was one accuracy value per image, corresponding to the fraction of pixels correctly classified. This process was applied to raw and scattering corrected images and the results are compared in Figure 24.

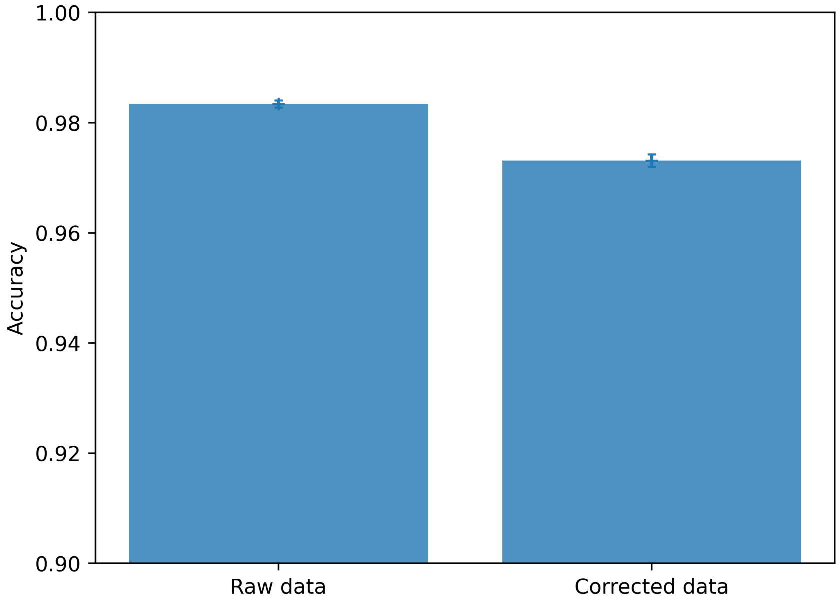


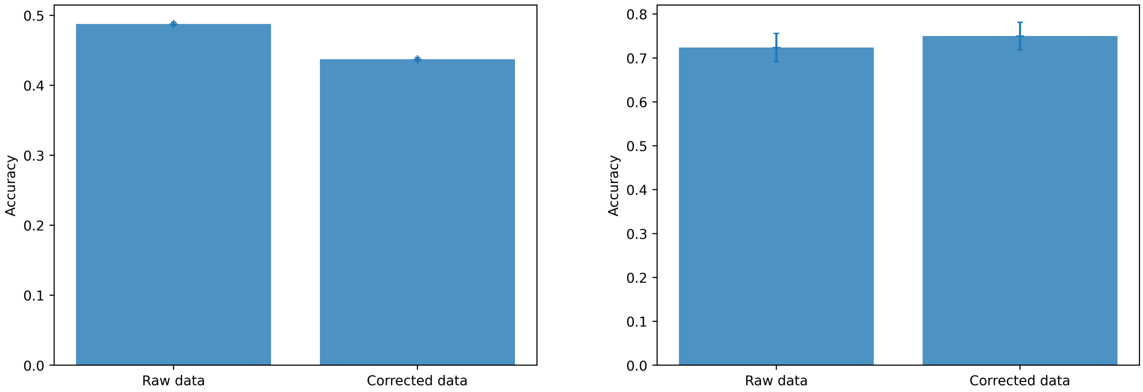
Figure 24: Average balanced accuracy (average of recall on each class) of the foreground/background classification across either raw or scattering-corrected images. Error bars represent one standard error. The high accuracy in both cases suggests that the classification process is not affected by scattering effects.

Activity of hypha and scattering effects Using foreground vs. background classification as a validation technique relies on the assumption that the hyphae are chemically active. In this case, the foreground contains chemical species which the background does not, and the aim of the classifier is to distinguish this. However, the foreground is also the part of the slide with structure, which means that it is the part experiencing most scattering effects. Therefore, it is possible that the classifier actually measures this effect instead of the chemical variation which interests us. If this was the case, we would expect the classifier to interpret cracks in the substrate (which are affected by scattering) as part of the foreground. However, visual inspection of predicted annotations did not find such false positives in any image.

3.3.2 Slide classification

Averaging foreground and background spectra may have limited the impact of scattering, as the raw images performed as well as the scattering corrected one. See Figure 25b.

Moreover, as the training set was so different from the test set, the classification did not give satisfactory results, neither on the raw nor preprocessed data.



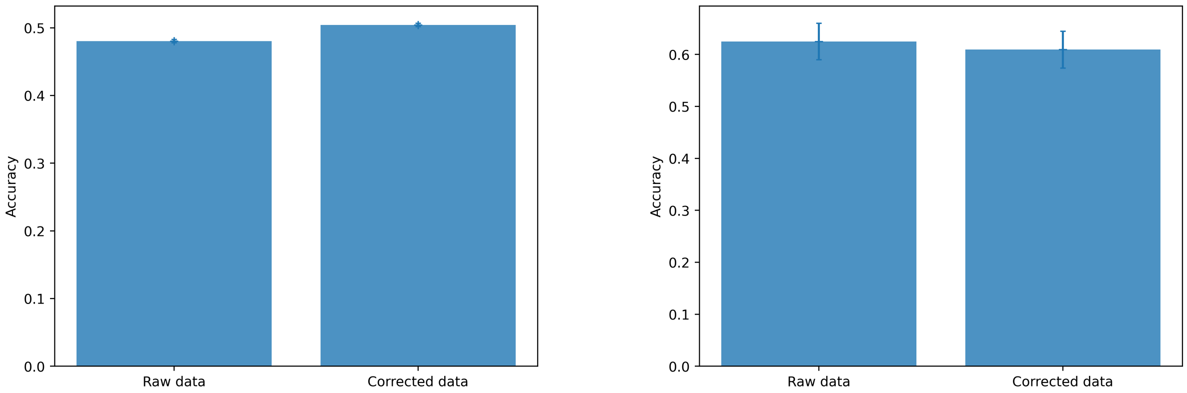
(a)

(b)

Figure 25: Average accuracy of the slide classification on average foreground-background differences across raw and scattering-corrected images. Error bars represent the standard error. (a): Classification across all background-corrected hyphal spectra; (b): Classification across images, on the difference between average foreground and average background.

3.3.3 Experiment classification

The results of the pixel-wise and image-wise classifications are depicted in Figure 26a and Figure 26b, respectively.



(a)

(b)

Figure 26: Average accuracy of the experiment classification on average foreground-background differences across raw and scattering-corrected images. Error bars represent the standard error. (a): Classification across all background-corrected hyphal spectra; (b): Classification across images, on the difference between average foreground and average background.

4 Discussion

4.1 Efficacy of the methods

In this work, the scattering correction used within OCTAVVS was improved and a reliable way to generate synthetic data was provided. To assess its efficacy, each correction method was applied to data carefully chosen for its relevance in either atmospheric or scattering perturbation. Methods were compared to each other and to the currently available one (either the OPUS commercial software, or the correction used in the OCTAVVS toolkit) in order to confirm any improvement.

4.1.1 Atmospheric correction

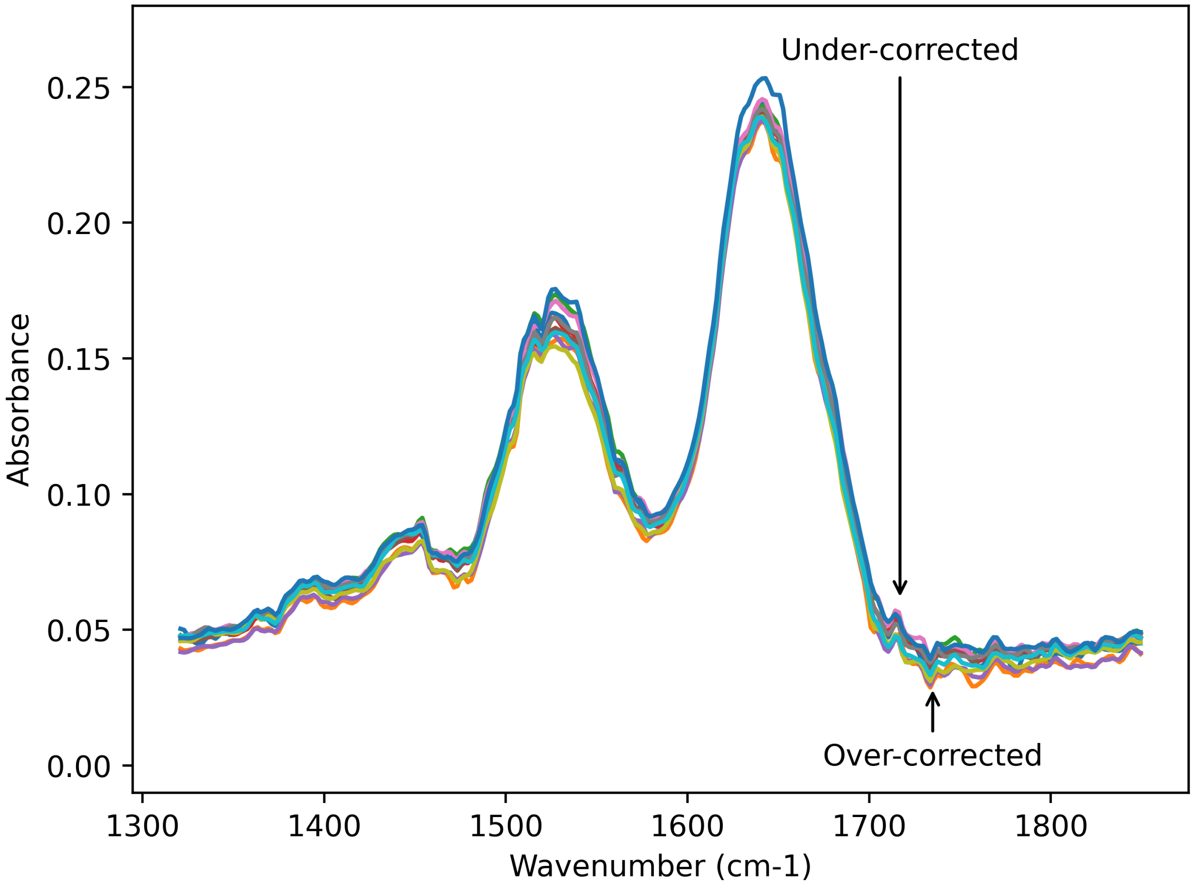


Figure 27: Example of the limitations of the atmospheric correction. These spectra are the result of a "watery" sample subjected to the correction. Some peaks, such as the one at 1715 cm^{-1}, are under-corrected. Others, such as the one at 1733 cm^{-1}, are over-corrected. Such limitations can be circumvented, to some extent, using filtering methods.

As shown in Figures 18 and 18b and through visual inspection (see Figure 27), the atmospheric corrections devised and studied here offer somewhere between similar and slightly improved performance as currently used methods. However, some peaks remain badly corrected when compared to other parts of the spectrum. This means that the correction somehow does not predict correctly the intensity of these peaks although it performs well on others. One aspect of our procedure can be to blame:

The reference set of atmospheric spectra (used to build the model) can be inaccurate or maladapted to the actual data. Since the time series mentioned in Section 2.2.1 was captured using a different device than the infrared absorption image and in potentially different atmospheric conditions (temperature, humidity, carbon dioxide contents, etc.),

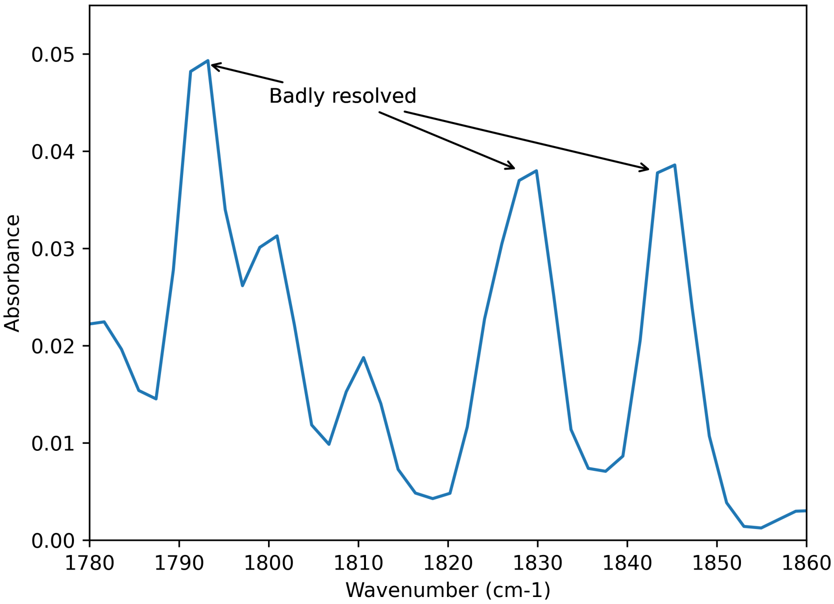


Figure 28: Example of badly resolved absorption peaks in one of our atmospheric reference spectra. Region from 1780 to 1860 cm^{-1} enlarged for clarity.

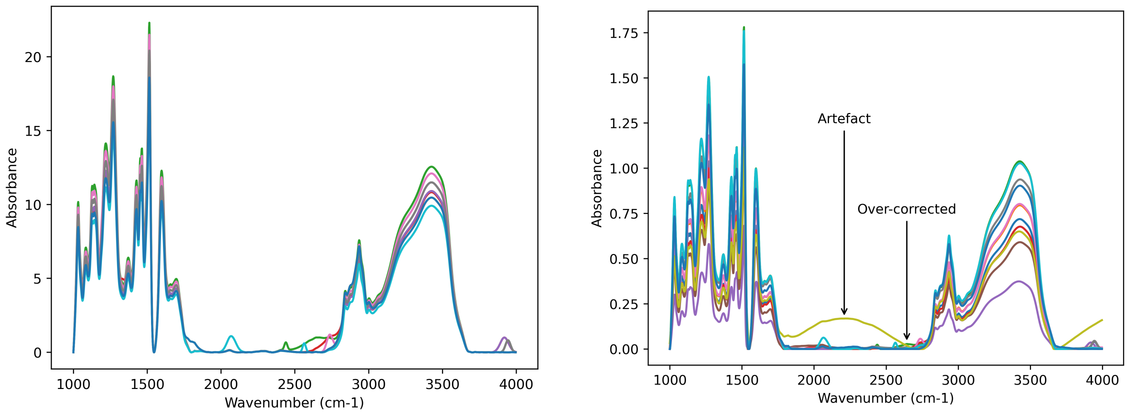
the peaks could have been distorted. For example, since the spectral resolution is higher, and the peaks extremely narrow, they could be better resolved in the reference set of atmospheric spectra than in the input data. This could be corrected by using a reference set from the same machine as the experiment itself. However, requesting a machine-specific calibration defeats the purpose of OCTAVVS, on top of being time- and ressources-consuming, and should be avoided.

Moreover, even with the higher spectral resolution used for the atmospheric reference, visual inspection of the water vapour peaks (such as depicted in Figure 28) shows them to be badly resolved.

Compared to previous work The atmospheric correction methods devised above (PCA-based and Gaussian peaks) have showed comparable or slightly improved performance (as evaluated with our correction score) over the atmospheric correction from OCTAVVS. This does not necessarily undermine the value of these methods, as it is possible that a theoretically maximum has already been reached in correction out atmospheric perturbations.

4.1.2 Scattering correction

Over-correction In the process of RMieSC correction, the extinction spectra—which have a broad shape—are fitted and removed from the data. This leads to a risk of over-



(a)

(b)

Figure 29: Example of the limitations of our scattering correction. (a): shows the "clear" synthetic data, prior to adding scattering perturbations. (b): shows the same spectra after synthetic scattering and scattering correction.

correcting broad peaks of biological origin when they are situated in a region where scattering is expected, as the algorithm would misinterpret them for scattering effects. Such an effect is shown on synthetic data in Figure 29.

Artefacts Inadequate fitting of the extinction spectra can lead to the apparition of broad features in the corrected spectra. An example is shown in Figure 29. Such defects are likely caused by the model not containing a specific extinction spectrum present in the data. Indeed, as the algorithm generates the scattering model for a finite number of sphere diameters and refractive indices, it cannot perfectly account for the infinity of values found in nature.

Validation data As scattering perturbations can have widely different shapes, quantification of the correction methods is less straightforward than in the case of the relatively predictable atmospheric perturbations. We have adressed this issue by working with synthetic scattering data and by developping a specific score to use on these.

However, validating algorithm against synthetic data comes with its own limitations, discussed in Section 4.2.3.

Compared to previous work As shown in Figure 22, our improvements to the reference in the clustering version of the RMieSC algorithm has significantly enhanced the accuracy of this tool. Clustering always trades accuracy for speed, but these improvements reduce the impact of this trade-off.

4.1.3 Relevance of classification as a validation tool

As shown in Section 3.3.1, all attempts at classifying spectra resulted in similar accuracies on raw and scattering-corrected data.

This absence of improvement does not necessarily mean that the correction methods are insufficient or inadapted. Firstly, it is possible that the classifier still manages to identify the correct features despite the scattering effects: The high accuracy results obtained from raw data (in Figure 24 for example) indeed indicate that this classification method is less sensitive to perturbations than expected.

Secondly, the averaging over several spectra—used in the classifications by slide and by experiment—may have cancelled out the scattering effects, making the raw data similar to the scattering-corrected data.

In any case, this result suggests that scattering corrections are not critical to the success of automated analysis of infrared absorption microscopy images. Such corrections might make visual analysis of spectra easier to humans, but computational tools behave differently and are not subject to the same requierments.

4.2 General limitations

4.2.1 Difficulties in quantifying corrections

In the case of both the atmospheric and the scattering corrections, one significant obstacle to their validation is the absence of "expected" output. Indeed, in experimental setups, parameters (like the thickness or exact concentration of each species at each point in the sample) are so unpredictable that we cannot assert what the "true" spectrum should look like after correction.

Using synthetic data can be used to estimate the effectiveness of the corrections to some extent (that was used as part of this work), but is limited. Synthetic data will never be able to reflect all the perturbations seen in experiments, or even all the different samples which can be studied.

4.2.2 Double perturbation effects

This work, and all previous literature, relies on the separate correction of atmospheric and scattering perturbation. For instance, in OCTAVVS, spectra are first submitted to the atmospheric correction and then to the scattering one. However, it is worth to notice that the scattering perturbation could affect the efficacy of the atmospheric correction and vice-versa: scattering of the light may make it encounter different concentration of water at different wavelengths. Whether this effect occurs and is of sufficient intensity to require correction is unknown. Preventing it could require both correction to occur at once, for instance by uniting atmospheric and scattering contributions into a single EMSC model. However, as those corrections have very different approaches, uniting them into one would not be trivial.

4.2.3 Limitations of synthetic data

Our approach to synthetic data consisted, both for atmospheric and scattering correction, in adding artificial perturbations to "clean" spectra. However, by its nature, this fails to account for several things:

- Perturbations shape: The model for the perturbations added relies on numerous assumptions which are bound to stray from reality to some extent. For instance, in the scattering synthetic data, extinctions spectra are generated either with the Konevskikh or Bassan algorithm and is therefore biased towards a solution employing similar methods.
- Parameter space: In our synthetic data, we can of course only choose from a finite number of perturbations and corresponding intensity. Nature, on the contrary, can produce an infinite number of perturbations, including some we might not have foreseen.

We can only model the effects, there will always be some variation from reality.

4.3 Specific limitations

This work was subject to limitations inherent to the use-cases studied. Such considerations are summarized in the following paragraphs.

4.3.1 Simplicity and end-user accessibility

This work, and the methods devised therein, was entirely focused on the improvement of OCTAVVS. The software is targeted at biologists and other users with little or no expected knowledge of computer programming. Therefore, all algorithms are designed with simplicity and ease of use in mind. Moreover, as users are not expected to have any specific physics background, it would not be advisable to request the setting of parameters of which they might not understand the implications. For instance, in the atmospheric correction, having to set an atmospheric reference should be avoided, and would ideally be contained in the software package, out of sight of the end-user. Likewise, in the scattering correction, the setting of the refraction index and particle sizes ranges are far from intuitive for people not familiar with this particular problem.

Therefore, one of the secondary objectives of this work was to provide methods which, while remaining effective on a wide variety of experiments, require the least possible end-user implication.

4.3.2 The issue of complexity

As mentioned above, OCTAVVS is intended to be used in bio-labs setups, and should, therefore, be intended to run on a standard desktop computer. It is not expected of the average biology lab, and hence of the average OCTAVVS user, to have access to super-computers. Thus, it is important to keep the space and time complexity to levels manageable on desktop devices. For example, a computation time of a minute per image is still acceptable, where an hour per image is not.

4.3.3 Variety of use-cases

As OCTAVVS is designed to eventually be used on widely different experiments, it is important to keep in mind that any correction therein has to perform on a variety of data. Therefore, although great care was taken to subject diverse experiments to our correction to validate them, it is important to remember that no extent of testing can ever cover the totality of use-cases. A tradeoff has to always be expected as efficacy and adaptability can rarely be both satisfied.

5 Conclusion

Overall, this thesis led its author down into more paths than he had expected:

Novel atmospheric correction methods were devised (PCA-based and Gaussian-peaks based) which account for previously neglected variability in the atmospheric spectra. These methods provided marginal improvements over the atmospheric correction from OCTAVVS.

Furthermore, the Clustered Resonant Mie Scattering Correction (CRMieSC) approach was improved by providing several methods (improved reference selection and cluster mixing) of limiting clustering-related defects. This yielded considerable improvements over the scattering correction from OCTAVVS.

Additionally, extensive work was dedicated to validating and quantifying the effect of corrections (atmospheric and scattering), both on measured and synthetic spectra. This resulted in a comprehensive scoring system which can be applied to any such correction to validate its effects, set its parameters or compare it to competing methods.

Finally, forays into classifying raw and corrected spectra showed that scattering correction does not have a substantial effect on the accuracy of classifiers and may therefore not be critical to the success of further analysis steps.

These methods in context While none of the results of this work represents a breakthrough in their field, they are one more step in the right direction. The achievement of more precise, more efficient and more resilient techniques for the correction of infrared absorption spectral imaging is a key aspect in making it the reliable, versatile chemometric tool it can be. Therefore, improvement in the precision of infrared absorption images eventually leads to the possibility of new medical diagnosis tools or water purity testing devices, among other things.

Specific use-case: fungal decomposition experiments Although great care was taken to keep the procedures as general and versatile as possible, the totality of input data used in this work came from fungal decomposition studies. In this specific use-case, our methods presented a slight improvement in making data easier to interpret for researcher. However, a reliable set of correction methods is critical to measuring the faint biochemical signals used to understand the process of fungal decomposition. Therefore, development of better pre-processing procedures for these samples is still ongoing.

Future work This work focused on the improvement of corrections, but forays into validation of these corrections proved useful. The problem of assessing whether a sample is properly corrected did not prove easily tractable. Although the attempts at scoring, classifying or using synthetic data were instructive, they could benefit from more work. Indeed, in the literature, where much is invested in perfecting the correction method, very little has been done in assessing their strength and resilience. A reliable way to carry such test is with the synthetic data mentioned above. The techniques for synthesizing data could benefit from being improved and implemented into a single framework.

Moreover, each atmospheric correction method studied here (PCA-based or Gaussian peaks) was evaluated independently, but it would be of some interest to combine them—together or with filtering—to exploit the advantages of both.

References

[1] M. J. Baker et al. "Using Fourier Transform IR Spectroscopy to Analyze Biological Materials". In: *Nature Protocols* 9.8 (Aug. 2014), pp. 1771–1791. DOI: `10.1038/nprot.2014.110`.

[2] C. Troein et al. "OCTAVVS: A Graphical Toolbox for High-Throughput Preprocessing and Analysis of Vibrational Spectroscopy Imaging Data". In: *bioRxiv* (Dec. 18, 2019), p. 2019.12.17.879387. DOI: `10.1101/2019.12.17.879387`.

[3] C. Troein. *OCTAVVS GitHub Repository*. Mar. 27, 2020. URL: `https://github.com/ctroein/octavvs` (visited on 04/21/2020).

[4] J. H. Solheim et al. "An Open-source Code for Mie Extinction Extended Multiplicative Signal Correction for Infrared Microscopy Spectra of Cells and Tissues". In: *Journal of Biophotonics* 12.8 (Aug. 2019). DOI: 10.1002/jbio.201800415.

[5] M. Op De Beeck et al. "Regulation of Fungal Decomposition at Single-Cell Level". In: *The ISME Journal* 14.4 (Apr. 2020), pp. 896–905. DOI: 10.1038/s41396-019-0583-9.

[6] H. Martens, J. P. Nielsen, and S. B. Engelsen. "Light Scattering and Light Absorbance Separated by Extended Multiplicative Signal Correction. Application to Near-Infrared Transmission Analysis of Powder Mixtures". In: *Analytical Chemistry* 75.3 (Feb. 1, 2003), pp. 394–404. DOI: 10.1021/ac020194w.

[7] N. K. Afseth and A. Kohler. "Extended Multiplicative Signal Correction in Vibrational Spectroscopy, a Tutorial". In: *Chemometrics and Intelligent Laboratory Systems* 117 (Aug. 2012), pp. 92–99. DOI: 10.1016/j.chemolab.2012.03.004.

[8] S. W. Bruun et al. "Correcting Attenuated Total Reflection—Fourier Transform Infrared Spectra for Water Vapor and Carbon Dioxide". In: *Applied Spectroscopy* 60.9 (Sept. 2006), pp. 1029–1039. DOI: 10.1366/000370206778397371.

[9] D. Perez-Guaita et al. "Atmospheric Compensation in Fourier Transform Infrared (FT-IR) Spectra of Clinical Samples". In: *Applied Spectroscopy* 67.11 (Nov. 2013), pp. 1339–1342. DOI: 10.1366/13-07159.

[10] B. Zimmermann and A. Kohler. "Optimizing Savitzky–Golay Parameters for Improving Spectral Resolution and Quantification in Infrared Spectroscopy". In: *Applied Spectroscopy* 67.8 (Aug. 2013), pp. 892–902. DOI: 10.1366/12-06723.

[11] J. L. Ilari, H. Martens, and T. Isaksson. "Determination of Particle Size in Powders by Scatter Correction in Diffuse Near-Infrared Reflectance". In: *Applied Spectroscopy* 42.5 (July 1988), pp. 722–728. DOI: 10.1366/0003702884429058.

[12] P. Bassan et al. "RMieS-EMSC Correction for Infrared Spectra of Biological Cells: Extension Using Full Mie Theory and GPU Computing". In: *Journal of Biophotonics* 3.8-9 (Apr. 22, 2010), pp. 609–620. DOI: 10.1002/jbio.201000036.

[13] H. C. Hulst. *Light Scattering by Small Particles*. Courier Corporation, Jan. 1, 1981. 504 pp. ISBN: 978-0-486-64228-4. Google Books: 6ivW_TgIdjIC.

[14] T. Konevskikh, R. Lukacs, and A. Kohler. "An Improved Algorithm for Fast Resonant Mie Scatter Correction of Infrared Spectra of Cells and Tissues". In: *Journal of Biophotonics* 11.1 (Jan. 2018), e201600307. DOI: 10.1002/jbio.201600307.

[15] J. M. Hollas. *Modern Spectroscopy*. John Wiley & Sons, Apr. 21, 2004. 483 pp. ISBN: 978-0-470-09471-6. Google Books: lVyXQZkcKKkC.

[16] *Numpy.Linalg.Lstsq — NumPy v1.20.Dev0 Manual*. URL: https://numpy.org/devdocs/reference/generated/numpy.linalg.lstsq.html (visited on 05/26/2020).

[17] *Base Package — OPUS*. URL: https://www.bruker.com/products/infrared-near-infrared-and-raman-spectroscopy/opus-spectroscopy-software/base-package.html (visited on 05/20/2020).

[18] I. L. Rasskazov et al. “Extended Multiplicative Signal Correction for Infrared Microspectroscopy of Heterogeneous Samples with Cylindrical Domains”. In: *Applied Spectroscopy* 73.8 (Aug. 2019), pp. 859–869. DOI: 10.1177/0003702819844528.

[19] G. Luo et al. “Minimum Noise Fraction versus Principal Component Analysis as a Preprocessing Step for Hyperspectral Imagery Denoising”. In: *Canadian Journal of Remote Sensing* 42.2 (Mar. 3, 2016), pp. 106–116. DOI: 10.1080/07038992.2016.1160772.

Annex I: Code listings

Listing 1: Correction score

```
def score_corr(spec_raw, spec_corr, wns=[]):
    ''' Ratio of noise in the result over noise in raw data. (i.e
       . lower is better)'''
    window, polyOrder = 9, 3
    noise_rawSpec, noise_corrSpec, score = 0., 0., 0.
    n_specs = np.shape(spec_raw)[1] - 1
    if np.min(spec_raw[:,0]) <= 2251 and np.max(spec_raw[:,0]) >=
        2419 :
        end = np.where(spec_raw[:,0] <= 2255)[0][0]
        start =  np.where(spec_raw[:,0] <= 2415)[0][0]
    for n in range(1, min(spec_raw.shape[1],spec_corr.shape[1])):
        smoothed_raw = savgol_filter(spec_raw[:,n],window,
           polyOrder)
        smoothed_corr = savgol_filter(spec_corr[:,n],window,
           polyOrder)
        if np.min(spec_raw[:,0]) <= 2250 and np.max(spec_raw
           [:,0]) >= 2420 :
            smoothed_raw[start:end] = [n*(smoothed_raw[start]-
               smoothed_raw[end])/(start-end) + smoothed_raw[
               start] for n in range(0,(end-start))]
            smoothed_corr[start:end] = [n*(smoothed_corr[start]-
               smoothed_corr[end])/(start-end) + smoothed_corr[
               start] for n in range(0,(end-start))]
        score += np.sum((spec_corr[:,n] - smoothed_corr )**2)/np.
           sum((spec_raw[:,n] - smoothed_raw)**2)
    return(score/n_specs)
```

Listing 2: Spread score

```
def score_spread(spec_raw, spec_corr):
    ''' Closer to 1 is better.'''
    mean_raw = np.mean(spec_raw[:,1:], axis=1)
    mean_corr = np.mean(spec_corr[:,1:], axis=1)
    n_specs = np.shape(spec_raw)[1] - 1
    score = 0.
    spread_raw, spread_corr = 0., 0.
    for n in range(1, min(spec_raw.shape[1],spec_corr.shape[1])):
        score += np.sum((spec_corr[:,n]-mean_corr)**2)/np.sum((
           spec_raw[:,n]-mean_raw)**2)
    return(score/n_specs)
```

Listing 3: Synthetic score

```
def score_synth(spec_raw, spec_scattr, spec_corrd):
    n_specs, n_pts = spec_raw.shape
    dist_to_corrd, dist_to_scattr, score = 0., 0., 0.
    for i in range(n_specs):
        score += (distance(spec_raw[i], spec_corrd[i])/distance(
           spec_raw[i], spec_scattr[i]))
    return(score/n_specs)
```

Listing 4: PCA-based atmospheric correction

```
    def correct(self, nbP=8, computeModel=True, savgolParams=[45,
        7], baseline_pre_PCA = 'savgol', preprocess='deriv2',
       butterParams = [5,.1], baseline_post_PCA='linear'):
        ''' Corrects and updates current spectra.'''
        atm_ref = self.atm_spectra
        wns = self.extracted_spectra[:,0]
        spectra = self.extracted_spectra[:,1:]
        atm_ref_nobaseline = []
        for n in range(1, atm_ref.shape[1]):
            if baseline_pre_PCA == 'linear' :
                atm_corr = []
                y_0, y_1, Dx = atm_ref[0,n], atm_ref[-1,n], len(
                   atm_ref[:,n])
                for w in range(Dx):
                    atm_corr.append(atm_ref[w,n] - y_0 - w*((y_1-
                       y_0)/Dx))
            elif baseline_pre_PCA == 'asls' :
                atm_corr = atm_ref[:,n] - asls(atm_ref[:,n], 10,
                   .001)
            elif baseline_pre_PCA == 'savgol' :
                atm_corr = atm_ref[:,n] - savgol_filter(atm_ref
                   [:,n], savgolParams[0], savgolParams[1])
            else:
                atm_corr = atm_ref[:,n]
            atm_ref_nobaseline.append(atm_corr)
        pca = statsmodels.multivariate.pca.PCA(atm_ref_nobaseline
           , ncomp=nbP, demean=False, standardize=False) #Run PCA
        loadings_nobaseline = []
        for n in range(pca.loadings.shape[1]):
            if np.corrcoef(pca.loadings[:,n], atm_ref[:,1])[0,1]
               <= 0:
                loading = -pca.loadings[:,n]
            else :
                loading = pca.loadings[:,n]
            load_corr = []
            if baseline_post_PCA == 'linear':
                y_0, y_1, Dx = loading[0], loading[-1], len(
                   loading)
                for w in range(Dx):
                    load_corr.append(loading[w] - y_0 - w*((y_1-
                       y_0)/Dx))
            else :
                load_corr = loading
            loadings_nobaseline.append(load_corr)
        interp_loadings = []
        for n in range(len(loadings_nobaseline)):
            f = interp1d(atm_ref[:,0],loadings_nobaseline[n],
               fill_value="extrapolate")
            interp_loadings.append(f(wns))
        if preprocess == 'deriv2':
            A = secondDeriv(np.transpose(interp_loadings))
            B = secondDeriv(spectra)
        elif preprocess == 'deriv1':
            A = Deriv(np.transpose(interp_loadings))
            B = Deriv(spectra)
        elif preprocess == 'deriv3':
            A = thirdDeriv(np.transpose(interp_loadings))
            B = thirdDeriv(spectra)
        elif preprocess == 'butter':
            fs = 1/(wns[0] - wns[1])
            A = butterworth(np.transpose(interp_loadings),
               butterParams[0], butterParams[1], fs)
```

```
            B = butterworth(spectra, butterParams[0],
               butterParams[1], fs)
        params = np.linalg.lstsq(A, B, rcond=None)[0]
        new_corr_spectra = [wns]
        for n in range(self.nbSamples):
            corr = deepcopy(spectra[:,n])
            for j in range(nbP):
                corr -= params[j][n]*interp_loadings[j]
            new_corr_spectra.append(corr)
        self.extracted_spectra = np.transpose(new_corr_spectra)
        self.full_spectra = mergeSpectra(self.full_spectra, self.
           extracted_spectra)
        self.iterations += 1
```

Listing 5: Gaussian peaks atmospheric correction

```
def fit_gausses(wn, spec, maxn):
    positions0 = detect_peaks(wn, spec)
    print('Fitting␣peaks␣to␣spectrum')
    amplitudes0 = interp1d(wn, spec)(positions0).clip(min=0)
    n_peaks = len(positions0)
    widths0 = np.array([2]*n_peaks)
    bounds_amplitudes, bounds_positions, bounds_widths = (0, np.
       inf), (wn.min(), wn.max()), (0,np.inf)
    p0 = np.concatenate([amplitudes0, positions0, widths0])
    bounds0 = np.concatenate([[bounds_amplitudes]*n_peaks, [
       bounds_positions]*n_peaks, [bounds_widths]*n_peaks])
    def model_gaussians(wns, *params):
        amplitudes, positions, widths = params[:n_peaks], params[
           n_peaks:2*n_peaks], params[2*n_peaks:]
        spec = np.zeros((len(wns)))
        for i in range(n_peaks):
            spec += peak_fn(wns, positions[i], amplitudes[i],
               widths[i])
        return spec
    def cost(params):
        return np.abs(spec - model_gaussians(wn, *params))
    if plot:
        plt.figure()
        plt.plot(wn, spec, 'k')
        plt.plot(wn, model_gaussians(wn, *p0))
        plt.ylim(-0.001, 0.035)
        plt.xlim(1600,2000)
        plt.title('Initial␣guess')
        plt.show()
    print('Least squares. This can take some time...')
    start = time.time()
    params = least_squares(cost, p0, bounds=bounds0.T).x
    end = time.time()
    print('Done', end - start)
    if plot:
        plt.figure()
        plt.plot(wn, spec, 'k')
        plt.plot(wn, model_gaussians(wn, *params))
        plt.ylim(-0.001, 0.035)
        plt.xlim(1600,2000)
        plt.xlabel('Wavenumber (cm-1)')
        fname = fig_path + 'app3_fitted_'+peak_type
        if click.confirm('Save figure as '+fname+'?', default=False): plt.savefig(fname, dpi=500, bbox_inches='tight')
        plt.show()
    err = np.sum((spec - model_gaussians(wn, *params))**2)
    print('Error on model:', err)
    peaks = {'Position':params[n_peaks:2*n_peaks], 'Amplitude':params[:n_peaks], 'Width':params[2*n_peaks:]}
    peaks_df = pd.DataFrame(data=peaks)
    return(peaks_df)
```

Annex II: Other considered methods

Other methods were considered to correct the atmospheric perturbation, which did not perform on comparable levels as the PCA-based or Gaussian approach shown above. These methods are summarized below.

Minimum Noise Fraction The Minimal Noise Fraction, or MNF, is a factor analysis developed for the decomposition of hyperspectral images. Its behaviour is similar to PCA, but distinguishing components by their Signal To Noise ratio instead of their variance in the sample. This had provided good corrections on large scale hyperspectral IR images, to the point where it performed better than PCA-based methods [19], and was therefore of interest to us. However, the experiments in infrared absorption microscopy are very much different from the large scale scans in which MNF has proved useful. This might be one of the reasons why MNF did not prove useful in our case.

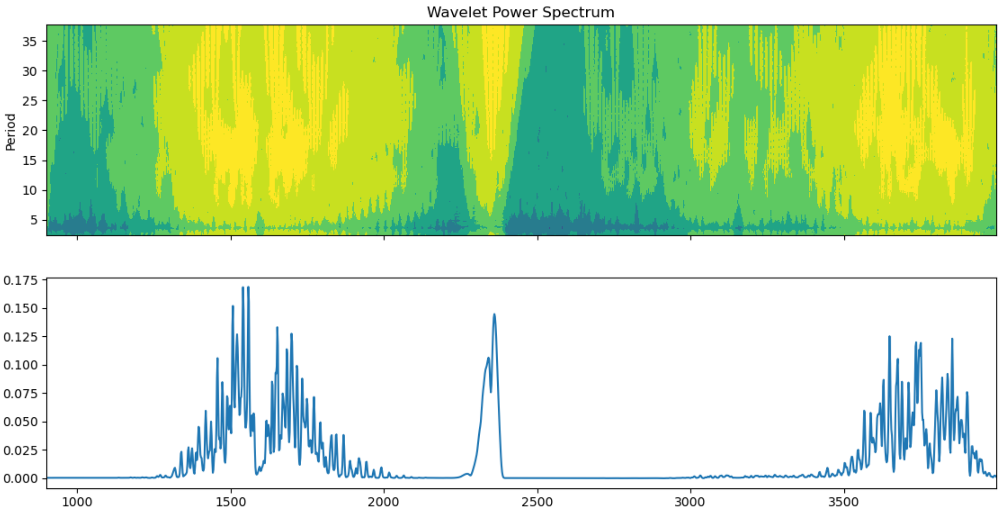


Figure 30: Wavelet analysis on the atmospheric spectrum, using a Morlet wavelet. We can see that neither the periods nor positions of the wavelets are well-defined. This makes this method unsuitable for denoising.

Wavelets Assuming that the position and amplitude of atmospheric peaks could vary lead to the Gaussian peaks approach, but also justifies using other means of modelling them. One method considered was wavelets analysis. Indeed, since atmospheric peaks could be considered as “brief oscillations” over a relatively uniform background, and of relatively predictable shape, they seem like the best a perfect candidate for wavelet denoising. Figure 30 depicts the result of wavelet analysis on an atmospheric spectrum.

This approach, however, poses a problem. Since wavelets would remove anything with similar aspect as the water vapour peaks regardless of its position, it is likely to eliminate some significant biological data. Indeed, if some chemical species of interest happens to have narrow peaks, they would get removed within this correction. A solution to this would be to constrain the positions where the wavelet correction would be applied. This did not prove effective in the tests carried out in this work, but could certainly benefit from additional consideration. For instance, a wavelet shape more adapted to the water vapour absorption peaks could be designed. A way to avoid over-correction of significant peaks would be advisable.

First PCA-based approach It was first thought to be possible to extract the atmospheric contribution to the spectrum from the experimental data itself. This would have been a significant improvement on previous correction methods as it would eliminate the need for an external atmospheric reference, using instead something already contained within the data (which would be potentially more adapted to its specific situation). Indeed, the assumption that the repartition of biological components and water vapour should be uncorrelated sounds reasonable.

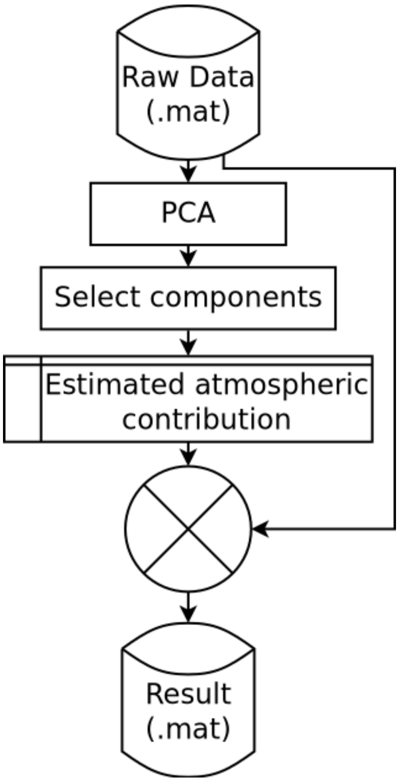


Figure 31: Flowchart of this method.

Therefore, the variance of the spectra over images containing N independently varying chemical species could be decomposed over N+2 vectors: one for water vapour, one for carbon dioxide, and one for each independently varying species. The resulting components could then be used to correct the data, following the process shown in Figure 31.

However, even if those vectors are theoretically uncorrelated, the small number of samples (usually 4096) makes it difficult for the PCA to separate variation directions from one another. Therefore, although some PCA vectors contained mainly atmospheric perturbations, they still mixed with the spectra of other species present in the sample and could not be used as a reference for a correction. This is depicted in Figure 32. Such a plot justified the use of either one or two components. However, trial and error did not show any significant improvement linked with a specific number of components. Moreover, concerns about the components mixing justified switching from this approach to another way of building a model. Selection of the adequate number of PCA components was expected to be possible with the use of a scree plot (shown in Figure 33).

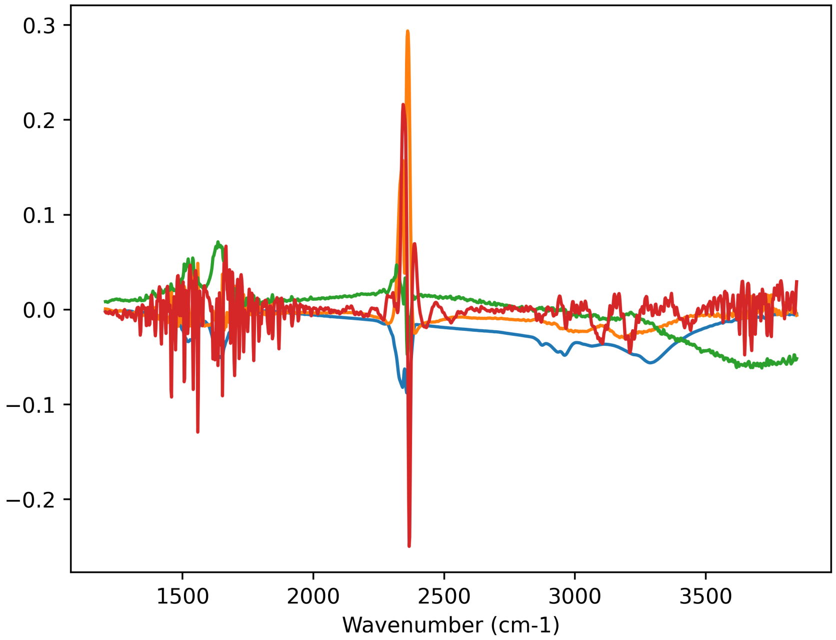


Figure 32: Loading vectors from Principal Component Analysis of a badly damaged sample of casein. The sharp peaks of the water vapour absorption spectrum are very obvious in vector 3, but are still visible in 1 and 2.

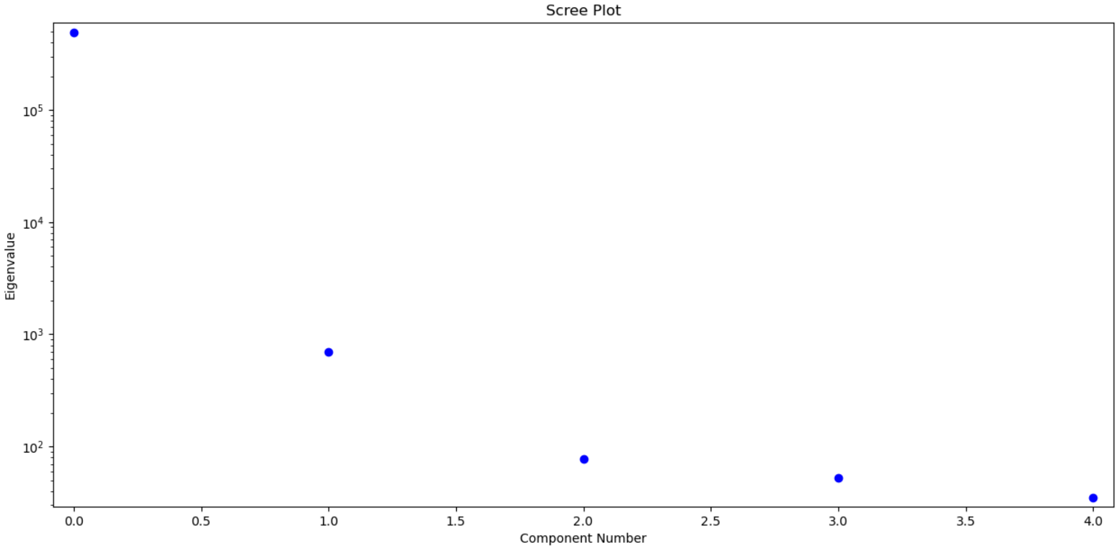


Figure 33: Scree plot of the PCA components for this first approach.