Classification of Two-Dimensional Gas Chromatography Data

Problem presented by

Alexandra Harvey & Emily Matthews

DSTL

ESGI130 was jointly hosted by
University of Warwick
Smith Institute for Industrial Mathematics and System Engineering

Report author

Matteo Croci (University of Oxford)
Piotr Morawiecki (Centre for Industrial Applications of Mathematics and System Engineering)
Valentin Sulzer (University of Oxford)
Florian Theil (University of Warwick)

Executive Summary

Gas chromatography (GC) is a popular tool for chemical analysis. Some samples are so complex that a single column does not have enough power to separate all of the analytes. In this instance a higher resolution GC method, known as comprehensive two-dimensional gas chromatography (GCxGC), is used. DSTL want to be able to use data from GCxGC to attribute samples to a particular region or cultivar. However, the nature of the data means that several difficulties must be overcome before being able to do this: noise from sample, peak mis-alignment, and low quantity of samples. In this report, we investigate several methods to overcome such difficulties, and then classify the data. We are very successful in telling apart blanks from seeds, but obtain limited success when trying to classify between seeds. The method that shows the most promise is k-Nearest Neighbours classification by Wasserstein distance. However, this is still quite sensitive to the noise created by the solvent in the sample. Thus, we suggest that more blank runs be obtained, so that the 'ground truth' behaviour of the solvent is better understood, allowing us to remove the effect of the solvent from seed data. We also hope that the methods explored here will be more successful on the full raw data than they were on the limited 'peaks' data available to us for the purpose of this study.

Version 1.0
September 15, 2017
iv+15 pages

Contributors

Matteo Croci (University of Oxford)
Piotr Morawiecki (Centre for Industrial Applications of Mathematics and System Engineering)
John Prater (University of Warwick)
Valentin Sulzer (University of Oxford)
Florian Theil (University of Warwick)

Contents

1 Introduction

Gas chromatography (GC) is a popular tool for chemical analysis. In GC, a sample is passed through a column and the analytes within the sample are retained by the column depending on their affinity to the column phase. Therefore each analyte will move through the column at different rates and reach the detector where a retention time is recorded. Samples such as air, soil and other environmental samples are so complex that a single column does not have enough power to separate all of the analytes. In this instance a higher resolution GC method, known as comprehensive two-dimensional gas chromatography (GCxGC), is used. GCxGC combines two columns, hence two retention times and the volume of each analyte is used for identification. GCxGC allows for less than pico-gram detection of the chemicals in a sample.

DSTL want to be able to use data from GCxGC to identify features of the background of a sample, with the aim of attributing samples to a particular region or cultivar. In particular, they have GCxGC data for multiple seeds, which are of the same type but come from different countries and cultivars. The aim of this project is to provide methods that can determine what country or cultivar a new sample might be from.

Several difficulties exist with the data available for gas chromatography, and this project in particular. Firstly, and perhaps most importantly, there is a lot of noise in the data coming from the presence of solvent that is used to dilute the seeds. Secondly, there is some error in the position of the peaks, such that between samples peaks corresponding to the same chemical compound do not appear in exactly the same place. Much of the effort in the literature is focused on 'peak alignment' to resolve this issue, e.g. [1, 2]. Finally, only a limited quantity of amount of data is available.

In this report we outline the steps taken at ESGI 130 in Warwick to overcome these challenges and attempt to classify the data. Our main approach is to treat the data as a raw image and apply standard image classification techniques. We use this method, rather than more direct peak identification, because we do not have access to spectral data that could be used to confirm the significance of peaks.

In Section 2, we present the data that was available to us, and the two pre-processing methods that we use in order to apply image classification methods (aggregation and Gaussian regression). In Section 4, we discuss one method that we attempted, using correlation of cumulative distributions. This method finds good correlation between experiments that were conducted on the same day, suggesting that environmental conditions might play an important role. In Section 5, we introduce a completely different method that we used to classify the images. Classification using the Euclidean distance successfully differentiates between whether a sample is blank or has a seed, but is no better than random guessing to classify country and cultivar. Using the Wasserstein distance yields slightly more success when classifying country and cultivar (50% success rate). Finally, we conclude and give

recommendations for future work (both theoretical and experimental) in Section 6.

Sample/Blank	Number	Country	Cultivar	Day
B	1	-	-	1
B	2	-	-	1
S	1	A	1	1
S	2	B	2	1
S	3	A	2	1
S	4	A	1	1
S	5	B	2	1
B	3	-	-	2
S	6	A	1	2
S	7	A	3	2
S	8	A	4	2
S	9	C	5	2
S	10	A	3	2
B	4	-	-	3
S	11	A	4	3
S	12	C	5	3
S	13	D	5	3
S	14	B	4	3
S	15	E	6	3
B	5	-	-	4
S	16	D	5	4
S	17	B	4	4
S	18	E	6	4
S	19	F	5	4
B	6	-	-	5
S	20	G	6	5
S	21	F	5	5
S	22	G	6	5

Table 1: Summary of available data

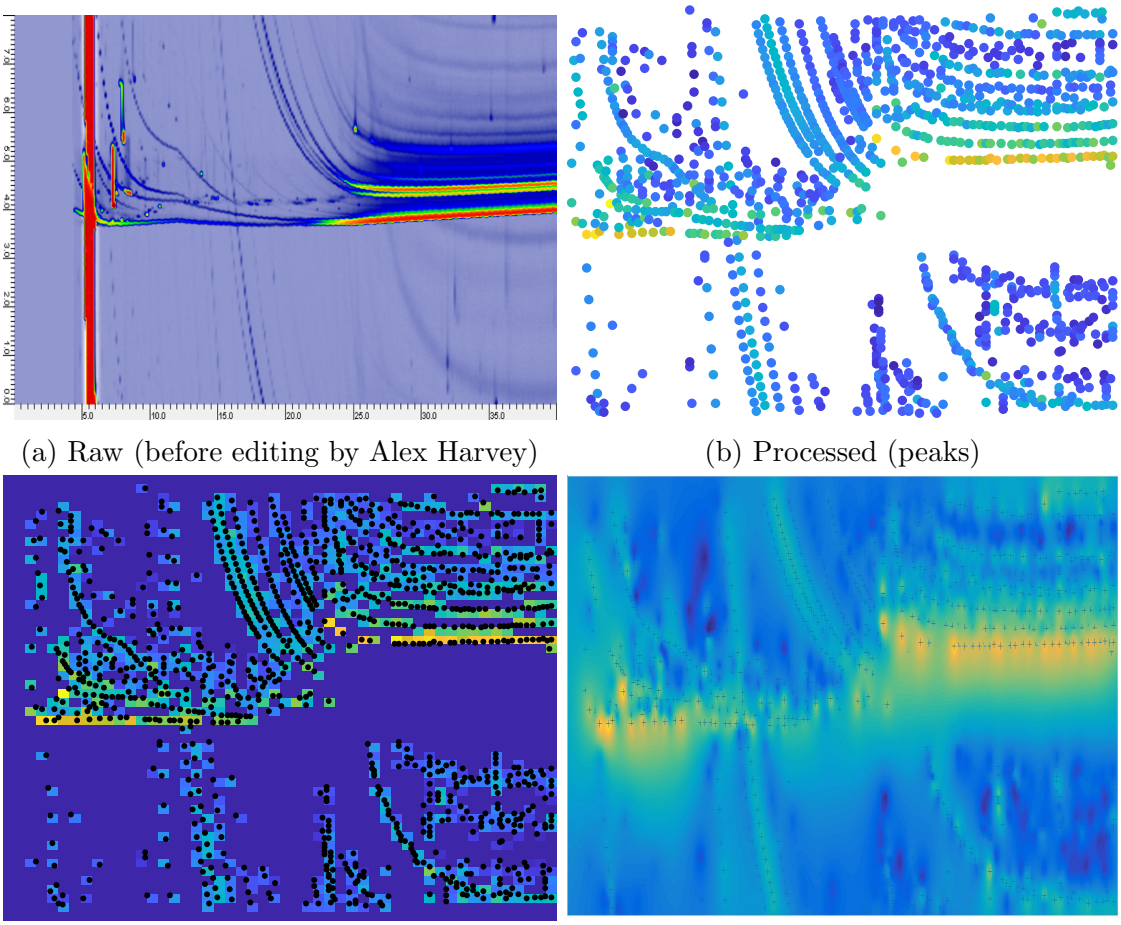
2 Available data and pre-processing

DSTL have run tests on 22 samples, each from a known country and cultivar, as well as 6 blank tests (Table 1).

The gas chromatography machine (TOFMS) provides data in two forms. The first is the volume, $v(x, y)$, received at each boiling point, x, and polarity, y (Figure 1). This data can be represented as a sum of n Gaussian functions, such that

$$v(\mathbf{x}) = \sum_{i=1}^{n} \frac{v_i}{2\sigma_i^2} e^{-(\mathbf{x}-\mathbf{x}_i)^2/2\sigma_i^2}, \tag{1}$$

where $\mathbf{x} = (x, y)$ and the Gaussian functions are assumed to be isotropic. The second output of the gas chromatography software is a list $(\mathbf{x}_i, v_i)$, $i = 1, \ldots, n$, of



(a) Raw (before editing by Alex Harvey)

(b) Processed (peaks)

(c) Reconstructed (aggregation, 50×50 grid)

(d) Reconstructed (Gaussian regression)

Figure 1: Stages of deconstruction and reconstruction for Sample 12 (Country C, cultivar 5).

the positions, $\mathbf{x}_i$, and volumes, v_i, of the peaks of the Gaussian (1) (Figure 1b). However, the standard deviations σ_i are *not* provided by the software.

2.1 Image reconstruction

For the ESGI130 study group, we are only able to access the second output, *i.e.* the list of the peaks, for all but Samples 1, 10 and 12. Hence, we begin by exploring methods to recreate the raw data for the remaining samples. The first of these methods, aggregation, consists of separating the image into a regular grid of size $m \times n$ and summing the volumes of all the peaks that fall within each grid point (x_p, y_q). The idea behind this method is that, if the aggregation grid is coarse enough, peaks that are misaligned between different images will be caught in the same grid point. This method is sometimes referred to in the literature as 'tiling'. The aggregation of the Sample 12 peaks data on a 50×50 grid is shown in Figure 1c.

The second method is to use Gaussian process regression to fit a surface to the data. Because we only have information on the locations of peaks and the volume beneath each one, and not the heights of the peaks, we must fit the covariance function to the data. We attempt to fit Gaussian processes with the Squared Exponential, Rational Quadratic and Ornstein Uhlenbeck covariance functions. However, we are unable to obtain good fit to the data with this method – the marginal likelihood is of order 10^3. The resulting image for Sample 12 is shown in Figure 1d.

For this project, we find that the results obtained from aggregating the data into buckets and those using Gaussian regression, are similar. Hence, we will use the aggregation method for the rest of the report. We note that, although these methods might be useful for projects external to DSTL where the full data cannot be revealed, in reality the best approach is to use the full data, thus avoiding the errors introduced by the aggregation and Gaussian regression processes.

3 Outlier detection

One way to classify the samples is to find the coordinates of the peaks corresponding to the chemical compounds in the sample. Those can be found by finding the peaks that appear in sample but do not occur in blank chromatogram (Figure 2). Since the images are not perfectly aligned, both in peak size and total volume, it is difficult to find such peaks. The problem is even worse for data in the form of the set of peaks.

The outlier detection method is supposed to return a list of the most promising peaks. Those should appear in as many samples as possible (because a given compound often appears in many similar organic samples), but not appear in blank data. Each peak can be 'scored' using the following procedure:

1. Pick a peak. Let $\mathbf{x}$ be the coordinates of the peak.

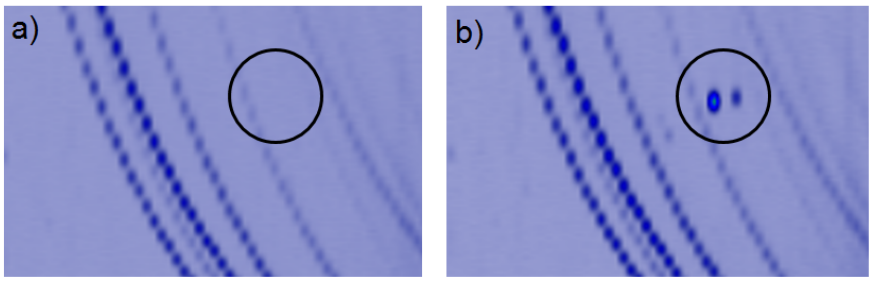


Figure 2: Example of outlier.

2. For each sample calculate the total volume, S_i, of peaks within a radius τ of $\mathbf{x}$.
3. For each blank calculate the total volume, B_i, of peaks within a radius τ of $\mathbf{x}$.
4. Calculate the peak's score by subtracting the means of values from points 2 and 3 (score $= \langle S_i \rangle - \langle B_i \rangle$).

The constant τ is a free-parameter that should be big enough to include possible shifts of a given peak. During tests τ was set as 3% of the total chromatogram size. From data about all the peaks one can then choose a selected percentage of peaks as the most promising peaks. An example of potential peaks distribution is presented in Figure 3. Points with warmer colours correspond to higher peak scores and grey points correspond to the blank set.

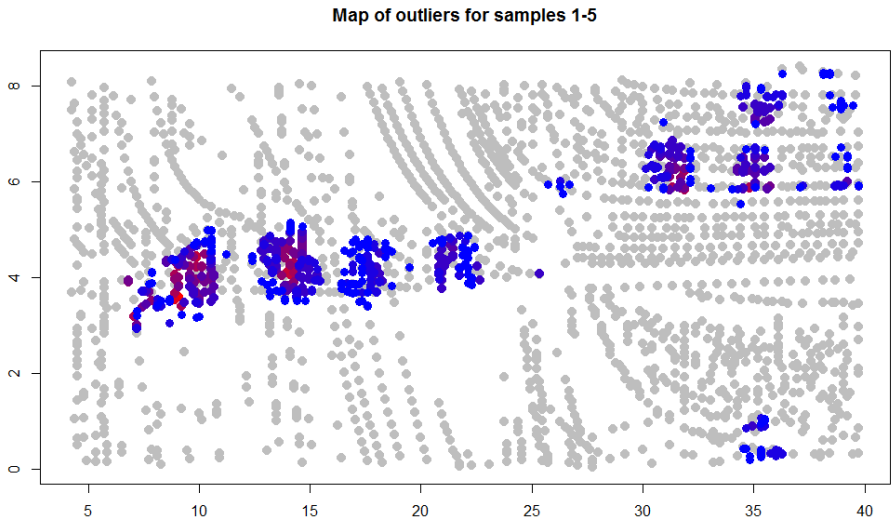


Figure 3: Top 10% peaks chosen from samples 1-5 using outlier detection method.

The most promising peaks can be used as an input for further processing methods, decreasing computation time and hopefully increasing accuracy. One can take either only the selected peaks or add all the peaks lying in their surroundings. By doing this, the chance of finding the corresponding peaks in samples that haven't received

a high score increases. However, peaks from background noise may appear, thus reducing quality of data.

The procedure described above can work on pixel data as well, potentially leading to even better results. To adapt the procedure, one just needs to average pixels lying within a certain radius in points 2 and 3. The rest of the procedure remains unchanged.

4 CDF Correlation

The chromatogram data has the default form of probability distributions. Peaks in the chromatogram are often misaligned and don't overlap each other. This makes it hard to compare values that should be at the same position. Instead of transforming the chromatogram, one can use the cumulative distribution function (CDF) instead. Peaks close to each other are not going to influence the CDF much, which will have a similar value over the whole data range (Figure 4). Only the difference in distribution of peaks may cause the CDF function to differ in a wider range of values. The idea is to compare the CDF of two samples to check whether they have similar peaks (but not necessary aligned) or not (Figure 5).

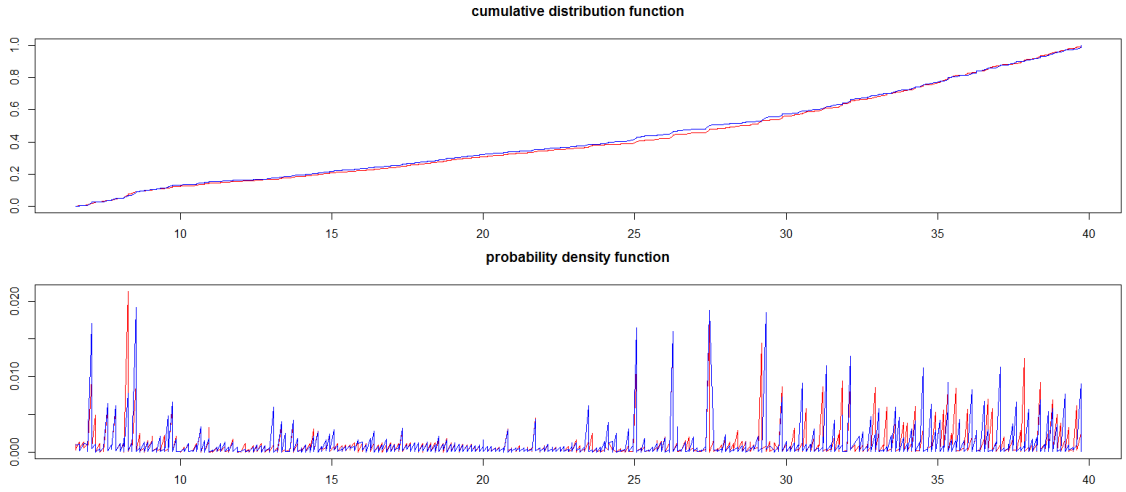


Figure 4: The comparison of CDF and PDF for samples 2 (red) and 5 (blue).

The CDF can be calculated in two dimensions (by cumulating volume along x- or y-axis). For comparison the chromatograms were normalized, so the total volume is always 1. To compare the CDFs two methods were used:

- calculating the mean absolute difference (results on Figure 6),
- calculating their correlation (results on Figure 7).

In the first case, no significant relationship was spotted between the correlation of CDFs and the countries and cultivates of samples. The second test didn't lead

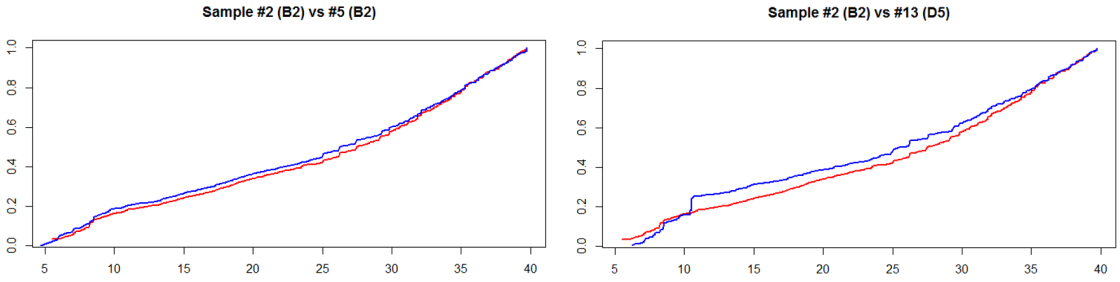


Figure 5: Expected behaviour on example of two similar and two different samples.

to clear relationships between these variables, but another interesting result was observed. Samples that were examined on the same day were highly correlated. Therefore this test may allow to check with high accuracy whether two samples were examined on the same day. The reason is artefacts, which turn out to be a bit different each day. As they are much higher than the searched peaks from chemical compounds, artefacts dominated the results of the experiment.

DIFFERENCE BETWEEN CDF (x axis only)

	A1	B2	A2	A1	B2	A2	A3	A4	C5	A3	A4	C5	D5	B4	E6	D5	B4	E6	F5	G6	F5	G6
A1	0,0%	3,0%	2,6%	2,4%	2,3%	2,3%	6,6%	5,8%	3,1%	4,1%	2,3%	2,5%	2,3%	3,9%	2,9%	2,2%	2,6%	4,1%	3,9%	3,2%	4,2%	3,6%
B2	3,1%	0,0%	2,7%	1,1%	1,9%	4,5%	9,2%	8,4%	2,8%	6,0%	4,8%	1,9%	4,1%	6,4%	3,0%	3,2%	4,1%	2,4%	1,5%	4,6%	1,7%	2,2%
A2	2,7%	2,7%	0,0%	1,8%	3,4%	4,2%	8,8%	8,0%	4,2%	5,5%	4,1%	3,1%	3,8%	6,0%	4,0%	2,5%	3,8%	4,4%	3,2%	4,8%	4,0%	4,0%
A1	2,4%	1,1%	1,8%	0,0%	1,9%	4,4%	9,2%	8,3%	2,8%	5,9%	4,3%	1,5%	3,4%	6,3%	2,7%	2,5%	3,7%	2,7%	2,0%	4,0%	2,6%	2,4%
B2	2,3%	1,8%	3,2%	1,8%	0,0%	2,7%	7,2%	6,4%	1,0%	4,0%	3,1%	0,9%	2,6%	4,6%	1,4%	2,0%	2,4%	2,4%	2,3%	3,1%	2,2%	1,8%
A2	2,3%	4,4%	4,0%	4,2%	2,7%	0,0%	4,6%	3,8%	2,5%	2,3%	0,9%	3,0%	1,8%	2,2%	2,4%	2,2%	1,5%	4,7%	4,7%	2,7%	4,4%	3,2%
A3	6,9%	9,3%	8,8%	9,1%	7,5%	4,8%	0,0%	1,6%	6,6%	4,2%	4,9%	7,9%	5,9%	3,0%	6,6%	6,8%	5,5%	9,6%	9,6%	5,4%	9,2%	7,5%
A4	5,9%	8,2%	7,7%	8,0%	6,4%	3,8%	1,4%	0,0%	5,5%	4,0%	3,9%	6,8%	4,8%	2,0%	5,6%	5,8%	4,5%	8,5%	8,5%	4,4%	8,1%	6,4%
C5	3,1%	2,8%	4,1%	2,8%	1,2%	2,6%	6,2%	5,6%	0,0%	3,4%	2,9%	1,6%	2,5%	3,9%	0,8%	2,0%	1,9%	3,1%	3,2%	3,0%	2,6%	1,6%
A3	4,3%	6,0%	5,5%	5,9%	4,2%	2,4%	4,1%	4,2%	3,5%	0,0%	2,8%	4,6%	3,3%	3,2%	3,4%	3,5%	2,2%	6,4%	6,4%	4,2%	5,9%	4,2%
A4	2,5%	4,8%	4,1%	4,4%	3,2%	0,9%	4,8%	4,0%	2,9%	2,8%	0,0%	3,1%	1,5%	2,1%	2,6%	2,1%	1,3%	4,9%	4,9%	2,3%	4,8%	3,2%
C5	2,6%	1,9%	3,1%	1,5%	0,9%	3,1%	7,9%	7,1%	1,5%	4,6%	3,1%	0,0%	2,4%	5,1%	1,4%	1,6%	2,5%	1,9%	1,9%	2,9%	2,2%	1,4%
D5	2,4%	4,1%	3,8%	3,5%	2,7%	1,9%	5,8%	5,0%	2,5%	3,3%	1,5%	2,4%	0,0%	3,0%	2,1%	1,6%	1,4%	4,2%	4,3%	1,4%	4,4%	2,6%
B4	4,1%	6,5%	6,0%	6,3%	4,8%	2,2%	2,9%	2,1%	4,0%	3,2%	2,1%	5,1%	3,0%	0,0%	3,8%	4,0%	2,7%	6,8%	6,9%	2,7%	6,3%	4,7%
E6	3,0%	3,0%	4,0%	2,7%	1,4%	2,5%	6,6%	5,7%	0,6%	3,4%	2,6%	1,4%	2,1%	3,8%	0,0%	1,8%	1,4%	3,1%	3,2%	2,6%	2,8%	1,3%
D5	2,2%	3,2%	2,5%	2,5%	2,1%	2,3%	6,8%	6,0%	2,0%	3,5%	2,1%	1,6%	1,6%	4,0%	1,7%	0,0%	1,5%	3,3%	3,4%	2,6%	3,5%	2,2%
B4	2,7%	4,1%	3,8%	3,8%	2,5%	1,5%	5,4%	4,6%	1,8%	2,2%	1,3%	2,5%	1,4%	2,6%	1,5%	1,5%	0,0%	4,3%	4,3%	2,3%	3,9%	2,2%
E6	4,2%	2,4%	4,4%	2,7%	2,4%	4,9%	9,6%	8,8%	3,1%	6,3%	4,9%	1,9%	4,2%	6,9%	3,1%	3,3%	4,3%	0,0%	2,4%	4,7%	2,3%	2,5%
F5	4,0%	1,4%	3,2%	2,0%	2,4%	4,9%	9,7%	8,8%	3,2%	6,4%	4,9%	2,0%	4,3%	6,9%	3,1%	3,4%	4,3%	2,4%	0,0%	4,7%	0,9%	2,3%
G6	3,3%	4,6%	4,8%	4,0%	3,3%	2,8%	5,3%	4,5%	3,1%	4,2%	2,3%	2,9%	1,4%	2,7%	2,6%	2,6%	2,3%	4,7%	4,7%	0,0%	4,9%	3,0%
F5	4,3%	1,7%	4,0%	2,6%	2,3%	4,5%	9,2%	8,3%	2,6%	5,9%	4,8%	2,2%	4,3%	6,3%	2,8%	3,5%	3,9%	2,3%	0,9%	4,9%	0,0%	1,9%
G6	3,7%	2,2%	4,0%	2,4%	1,9%	3,3%	7,5%	6,6%	1,5%	4,2%	3,1%	1,4%	2,5%	4,6%	1,3%	2,2%	2,1%	2,5%	2,3%	3,0%	1,9%	0,0%

Figure 6: Values of absolute mean difference between CDF of all 22 samples.

We can try to remove the background simply by subtracting its CDF from the sample (Figure 8). This may improve the results of the described tests. However, as the method initially did not bring promising results the focus was pulled on other methods and this recommendation has not been implemented.

5 Image classification approach

Each data sample (either raw or reconstructed data) comes in the form of an image and thus another option is to apply image classification techniques to try to tell samples of different characteristics apart. In particular, for a new image, we can ask the questions:

CORRELATION OF CDF (mean of x and y axes)

	A1	B2	A2	A1	B2	A2	A3	A4	C5	A3	A4	C5	D5	B4	E6	D5	B4	E6	F5	G6	F5	G6
A1	1,00	0,79	0,80	0,84	0,81	0,84	0,80	0,77	0,68	0,59	0,78	0,72	0,61	0,68	0,60	0,66	0,64	0,55	0,49	0,40	0,40	0,45
B2	0,77	1,00	0,85	0,96	0,96	0,83	0,83	0,71	0,96	0,63	0,79	0,91	0,67	0,78	0,88	0,82	0,83	0,57	0,88	0,60	0,84	0,82
A2	0,75	0,86	1,00	0,93	0,91	0,94	0,95	0,80	0,88	0,91	0,85	0,84	0,70	0,82	0,81	0,86	0,86	0,40	0,73	0,50	0,66	0,68
A1	0,82	0,96	0,91	1,00	0,98	0,93	0,93	0,85	0,96	0,74	0,91	0,95	0,80	0,89	0,91	0,90	0,91	0,53	0,86	0,70	0,81	0,83
B2	0,79	0,96	0,90	0,98	1,00	0,92	0,91	0,50	0,95	0,74	0,89	0,95	0,80	0,89	0,91	0,88	0,90	0,53	0,87	0,68	0,82	0,84
A2	0,79	0,82	0,94	0,93	0,92	1,00	0,98	0,93	0,84	0,85	0,95	0,89	0,86	0,92	0,83	0,91	0,90	0,39	0,70	0,69	0,62	0,69
A3	0,78	0,83	0,94	0,94	0,91	0,98	1,00	0,92	0,88	0,86	0,96	0,90	0,85	0,94	0,87	0,93	0,93	0,43	0,75	0,70	0,67	0,75
A4	0,75	0,70	0,79	0,85	0,84	0,94	0,92	1,00	0,74	0,70	0,97	0,87	0,90	0,95	0,79	0,86	0,87	0,36	0,63	0,77	0,55	0,67
C5	0,68	0,96	0,87	0,96	0,95	0,85	0,88	0,74	1,00	0,73	0,84	0,95	0,73	0,85	0,95	0,88	0,91	0,51	0,94	0,69	0,91	0,90
A3	0,47	0,62	0,91	0,74	0,72	0,85	0,86	0,68	0,71	1,00	0,75	0,71	0,66	0,74	0,70	0,83	0,83	0,06	0,61	0,47	0,56	0,59
A4	0,76	0,79	0,85	0,91	0,89	0,96	0,96	0,97	0,83	0,75	1,00	0,92	0,91	0,97	0,87	0,93	0,93	0,39	0,74	0,81	0,66	0,77
C5	0,69	0,92	0,83	0,96	0,96	0,89	0,90	0,86	0,95	0,68	0,92	1,00	0,86	0,94	0,96	0,93	0,95	0,45	0,92	0,86	0,88	0,93
D5	0,60	0,65	0,70	0,78	0,80	0,86	0,84	0,88	0,70	0,64	0,89	0,84	1,00	0,90	0,75	0,88	0,83	0,33	0,66	0,87	0,59	0,71
B4	0,67	0,78	0,81	0,89	0,89	0,93	0,93	0,95	0,85	0,74	0,97	0,94	0,92	1,00	0,91	0,93	0,96	0,39	0,79	0,85	0,73	0,84
E6	0,60	0,90	0,81	0,92	0,93	0,84	0,87	0,79	0,96	0,71	0,86	0,96	0,76	0,90	1,00	0,88	0,95	0,43	0,95	0,74	0,92	0,94
D5	0,67	0,81	0,88	0,91	0,88	0,93	0,94	0,87	0,87	0,83	0,93	0,93	0,90	0,93	0,88	1,00	0,94	0,36	0,84	0,86	0,77	0,86
B4	0,62	0,83	0,86	0,91	0,90	0,91	0,93	0,87	0,91	0,82	0,93	0,96	0,86	0,96	0,95	0,95	1,00	0,38	0,89	0,80	0,84	0,90
E6	0,57	0,65	0,45	0,62	0,60	0,48	0,51	0,47	0,61	0,19	0,53	0,60	0,47	0,51	0,56	0,50	0,51	1,00	0,53	0,50	0,48	0,54
F5	0,47	0,89	0,72	0,87	0,87	0,72	0,75	0,64	0,94	0,60	0,75	0,92	0,70	0,81	0,95	0,84	0,90	0,39	1,00	0,74	0,98	0,98
G6	0,52	0,66	0,57	0,75	0,72	0,73	0,73	0,79	0,71	0,46	0,84	0,86	0,88	0,86	0,77	0,86	0,82	0,37	0,76	0,99	0,68	0,83
F5	0,39	0,85	0,67	0,81	0,84	0,65	0,68	0,54	0,91	0,55	0,66	0,88	0,61	0,74	0,91	0,78	0,84	0,32	0,98	0,67	1,00	0,96
G6	0,47	0,84	0,69	0,85	0,85	0,73	0,77	0,69	0,91	0,60	0,79	0,93	0,74	0,85	0,95	0,85	0,91	0,39	0,98	0,79	0,96	1,00

Figure 7: Values of correlation between CDF of all 22 samples.

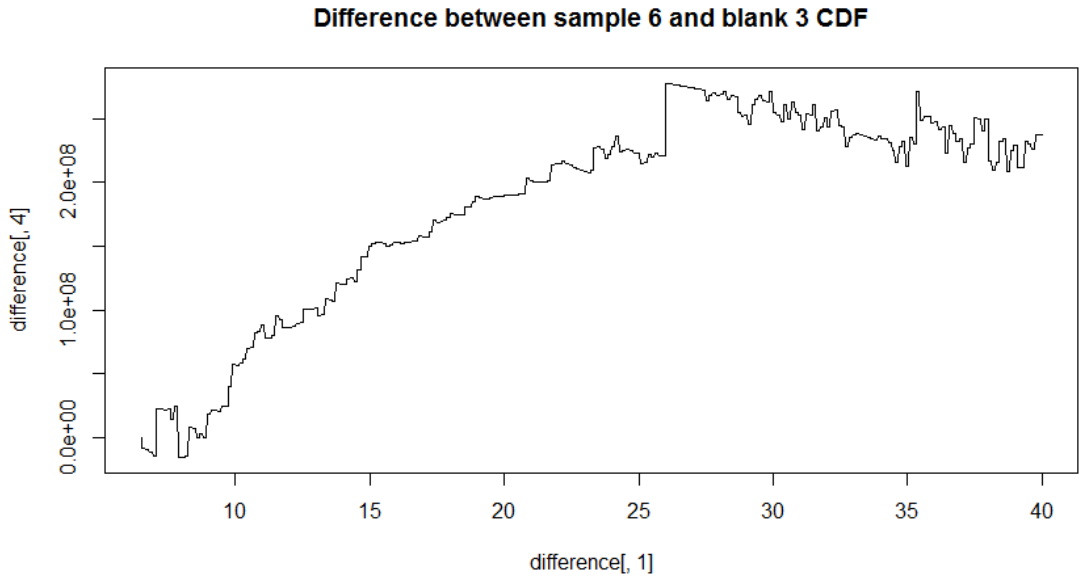


Figure 8: Difference between the sample 6 and the blank 3, which was recorded just before sample 6 examination.

1. Does this test contain a sample, or is it a blank?
2. Which country is the sample from?
3. Which cultivar is the sample from?

As we will see, question 1 is more easily answered, while the others present some complications.

5.1 Classification by Intensity

To classify the samples, we use a k-Nearest Neighbours (kNN) classifier, cf eg [4, 3]. This algorithm works as follows.

Each image can be represented as a $m \times n$ matrix, which can be viewed as a position vector, *i.e.* a point in $\mathbb{R}^{mn}$. The details of the rearrangement are irrelevant the classification is based only on differences between the coefficients. We then separate all the points (one for each image) into a training and a testing set and we provide the algorithm with the correct labelling for each image in the training set (*e.g.* this point represents a sample from seed 1 grown in country A; OR this point represents a blank test). For each point in the test set, kNN finds the k closest points in the training set and labels the test point with the most frequent label of the neighbours.

The closest points are chosen according to a user-specified distance function. We ran the kNN classification with the Euclidean and Wasserstein distances.

The Wasserstein distance, or Earth Mover's distance, between two images can be informally thought of as the effort required to turn one pile of dust into another – *i.e.* the cost of moving the dust times the distance that it has to be moved by.

Formally, the Wasserstein distance between two probability measures $\mu, \nu \in M_1(\Omega)$ on some probability space Ω is defined by

$$d_{\mathrm{W}}(\mu, \nu) = \inf \left\{ \int_{\Omega \times \Omega} c(x, y) \, \mathrm{d}\gamma(x, y) \ : \ \gamma \in J(\mu, \nu) \right\},$$

where

$$J(\mu, \nu) = \left\{ \gamma \in M_1(\Omega \times \Omega) \ : \ \int_{\Omega} \gamma(x, y) \, \mathrm{d}y = \mu(x) \text{ and } \int_{\Omega} \gamma(x, y) \, \mathrm{d}x = \nu(y) \right\}$$

denotes the set of joint distributions which have μ and ν as marginal measures. The integrand c is a cost function that measures distance between the x and y. The choice of the cost function should be informed by the problem. If $\Omega \subset \mathbb{R}^d$ for some $d \in \{1, 2, \ldots\}$ the choice $c(x, y) = |x - y|_2$ can be sensible. To illustrate the idea behind the definition of d_{W} assume that the supports of μ and ν are disjoint. In such a situation the total variation distance between μ and ν is 2 irrespective of the distance between the supports. On the other hand, if $\mu = \delta(\cdot - a)$ and $\nu = \delta(\cdot - b)$, then

$$d_{\mathrm{W}}(\mu, \nu) = c(a, b),$$

i.e. the Wasserstein distance is non-trivial even in situations where the supports are disjoint.

It is not hard to see why the Wasserstein distance performs well when used to measure the similarity between chromatograms: Assume that two chromatograms consist slightly misaligned peaks. A local metric such as total variations will depend

only very weakly on the misalignment. On the other hand, the Wasserstein distance depends strongly on the misalignment and will be small if the misalignment is small.

Computing the Wasserstein distance takes the form of a linear programming problem and can be computed in almost linear time. A Matlab Implementation with source code can be found at `https://www.mathworks.com/matlabcentral/fileexchange/22962-the-earth-mover-s-distance`. A more efficient implementation in C is available in the form of MEX (Matlab executable) files at: `http://www.mathworks.com/matlabcentral/fileexchange/12936-emd--earth-movers-distance--mex-interface`).

5.2 Principal Component Analysis (PCA)

Let $k_1,\ k_2 \in \mathbb{R}$. A slightly more sophisticated approach is to split the data into a $mn \times k_1$ training set X and a $mn \times k_2$ testing set Y, consisting of k_1 and k_2 column vectors respectively, and perform a PCA on X. To do this, we take the truncated eigendecomposition of the covariance matrix of X, XX^T, obtaining the eigenvector matrix $V = [v_1, v_2, \ldots, v_c]$, where $v_i,\ i = 1, \ldots, c$ are the c largest eigenvectors of XX^T with λ_i being their corresponding eigenvalues. PCA is a dimensionality reduction technique that can be used to capture the hidden main features of the data, provided that these are approximately lower-dimensional. The dimensionality of X and Y is then reduced through a projection into the space spanned by V, thus obtaining smaller feature vectors on which we run a kNN classification.

5.3 Results

We usually obtained the best results when choosing only one nearest neighbour. We believe this to be due to the small number of available samples: for some countries and cultivars, there may only be one example of that country/cultivar in the training set, so we should look for at most one nearest neighbour.

kNN with the Euclidean distance is successfully able to distinguish the blank samples from those that contained seeds, with a 100% success rate for a 50×50 peak aggregation grid size and a train-to-test ratio of 60%.

These two factors (grid size and train-to-test ratio) significantly affect the success rate and the computational burden, we show the results of some heuristic testing in the following table.

What grid aggregation does is to reduce the noise in the samples which causes peak misalignment. The grid size must be large enough to compensate for the noise (the same peaks must fall within the same tile even if the peaks are misaligned). Too large grids result in aggressive aggregation, making it impossible to distinguish between samples. Too fine grids make the computational cost increase and fail in reducing the noise.

grid size	train-to-test ratio	failed classifications	CPU time
50-60 × 50-60	50%	1	21s
50-60 × 50-60	60-70%	0	30s
70-80 × 70-80	50-60%	0	37-50s
70-80 × 70-80	70%	1	37-50s
90 × 90	50-70%	1	61s
100 × 100	50%	3	76s
100 × 100	60-70%	2	76s

Table 2: Comparison of the effect of different grid sizes and train-to-test ratios on the success rate and on the CPU time in the case of seed vs blank sample classification.

sample	real	1. guess	2. guess	3. guess	4. guess	5. guess	6. guess	7. guess	first guess	second guess
1	A	D	B	E	F	G	A	C	0	0
2	B	D	B	E	F	G	A	C	0	1
3	A	D	B	E	F	G	A	C	0	0
4	A	D	B	E	F	G	A	C	0	0
5	B	D	B	E	F	G	A	C	0	1
6	A	D	B	E	F	G	A	C	0	0
7	A	D	B	E	F	G	A	C	0	0
8	A	D	B	E	F	G	A	C	0	0
9	C	D	B	E	F	G	A	C	0	0
10	A	D	B	E	F	G	A	C	0	0
11	A	D	B	E	F	A	C	G	0	0
12	C	D	B	E	F	A	C	G	0	0
13	D	D	B	E	F	A	C	G	1	0
14	B	D	B	E	F	A	C	G	0	1
15	E	D	B	E	F	A	C	G	0	0
16	D	A	B	C	G	F	D	E	0	0
17	B	A	B	C	G	F	D	E	0	1
18	E	A	B	C	G	F	D	E	0	0
19	F	A	B	C	G	F	D	E	0	0
20	G	D	B	E	F	A	C	G	0	0
21	F	D	B	E	F	A	C	G	0	0
22	G	D	B	E	F	A	C	G	0	0
									5%	18%
									23%	

sample	real	1. guess	2. guess	3. guess	4. guess	5. guess	6. guess	first guess	second guess
1	1	5	4	6	2	1	3	0	0
2	2	5	4	6	1	2	3	0	0
3	2	5	4	6	1	2	3	0	0
4	1	5	4	6	1	2	3	0	0
5	2	5	4	6	1	2	3	0	0
6	2	5	4	6	1	2	3	0	0
7	3	5	4	6	1	2	3	0	0
8	4	5	4	6	1	2	3	0	1
9	5	5	4	6	1	2	3	1	0
10	3	5	4	6	1	2	3	0	0
11	4	5	4	6	1	2	3	0	1
12	5	5	4	6	1	2	3	1	0
13	5	5	4	6	1	2	3	1	0
14	4	5	4	6	1	2	3	0	1
15	6	5	4	6	1	2	3	0	0
16	5	1	2	3	4	5	6	0	0
17	4	1	2	3	4	5	6	0	0
18	6	1	2	3	4	5	6	0	0
19	5	1	2	3	4	5	6	0	0
20	6	5	4	6	1	2	3	0	0
21	5	5	4	6	1	2	3	1	0
22	6	5	4	6	1	2	3	0	0
								18%	14%
								32%	

Figure 9: Classification using kNN with Euclidean distance: low success rate

Similarly, we have to be careful in splitting the samples between training and test sets. If the train-to-test ratio is too high, we risk to over-fit the data and we reduce the number of testing samples that we can use to validate the algorithm. Over-fitting happens when a classification algorithm gets too tailored to the available data and therefore becomes unable to classify new samples. If the train-to-test ratio is too low, our algorithm will not be able to learn the distinguishing features of our data set and it will fail to classify the samples.

kNN with Euclidean distance fails to distinguish samples containing the same cultivars or seeds coming from the same countries (even aggregating the countries by climate did not bring any improvement), with a very low success rate (below 25%, see Figure 9).

The Wesserstein distance achieves much better results (arount 50% success rate, see Figure 10). However, given the large computational cost of calculating such a distance, we must restrict our analysis exclusively to the region given by $5.5 \leq x \leq 15$, $0 \leq y \leq 3.5$ so that we can consider less points. This region is chosen because it seems to contain the largest difference in peaks between blanks and seed samples. As the Wasserstein distance increases if volume needs to be moved around

sample	real	1. guess	2. guess	3. guess	4. guess	5. guess	6. guess	7. guess	first guess	second guess
1	A	B	A	E	C	G	F	D	0	1
2	B	A	C	E	B	G	D	F	0	0
3	A	A	E	C	G	B	D	F	1	0
4	A	A	B	C	E	G	D	F	1	0
5	B	C	B	E	A	G	D	F	0	1
6	A	A	B	C	G	E	D	F	1	0
7	A	A	B	C	G	E	D	F	1	0
8	A	A	C	B	G	E	D	F	1	0
9	C	C	G	A	B	E	D	F	1	0
10	A	A	B	C	G	E	D	F	1	0
11	A	B	A	C	G	D	E	F	0	1
12	C	B	C	A	E	G	D	F	0	1
13	D	F	G	D	B	C	A	E	0	0
14	B	B	A	C	E	G	D	F	1	0
15	E	B	C	G	A	E	D	F	0	0
16	D	F	G	D	B	E	C	A	0	0
17	B	B	C	A	G	E	D	F	1	0
18	E	A	B	C	D	G	E	F	0	0
19	F	D	G	F	B	A	C	E	0	0
20	G	C	B	E	D	A	F	G	0	0
21	F	D	F	G	A	B	E	C	0	1
22	G	B	A	C	E	D	F	G	0	0
								total	41%	23%
									64%	

sample	real	1. guess	2. guess	3. guess	4. guess	5. guess	6. guess	first guess	second guess
1	1	2	1	6	3	4	5	0	1
2	2	1	3	5	6	4	2	0	0
3	2	1	6	3	5	4	2	0	0
4	1	2	1	3	5	6	4	0	1
5	2	5	4	6	3	2	1	0	0
6	2	3	4	5	2	6	1	0	0
7	3	3	4	5	6	1	2	1	0
8	4	3	5	4	6	1	2	0	0
9	5	5	6	3	4	2	1	1	0
10	3	3	2	4	5	1	6	1	0
11	4	4	3	2	5	1	6	1	0
12	5	2	5	4	6	3	1	0	1
13	5	5	6	4	3	1	2	1	0
14	4	4	2	3	5	1	6	1	0
15	6	2	5	6	4	3	1	0	0
16	5	5	6	4	3	1	2	1	0
17	4	4	2	5	3	1	6	1	0
18	6	1	2	5	3	6	4	0	0
19	5	5	6	4	1	2	3	1	0
20	6	5	2	6	3	4	1	0	0
21	5	5	6	1	4	2	3	1	0
22	6	2	3	5	4	6	1	0	0
							total	45%	14%
								59%	

Figure 10: Classification using kNN with Wasserstein distance: medium success rate

(remember the pile of dust analogy in the previous section), we believe this measure is significantly inflated by the presence of large quantities of solvent (lots of solvent volume to be moved around = large increase in the Wasserstein distance which is not due to any seed-specific feature). If the effect of the solvent could be removed in some sort of pre-processing, then this distance should yield better results, but figuring out how to do this is left for future work.

After trying different heuristic approaches, most of which ended up in poor results, we found one that seems to yield good results, with a high success rate (in around 85% of the cases, a sample with the correct label was between the first 2 nearest neighbours that presented different labels, see Figure 11). To obtain these results, we computed the Wasserstein distance between the volume vectors, completely neglecting the peak position (although all the peaks where contained in the same small region, see previous paragraph). Although these results are quite good, we are currently unable to provide a theoretical justification for this approach and we are worried that such an outcome might derive from chance.

Unfortunately, Principal Component Analysis does not work well for this problem, so we do not show any results for this as they are not significant. This is probably due to the fact that the data set is too small for a dimensionality reduction technique to work: the data we have available might effectively be high-dimensional.

5.4 Recommendations and Future Work

We believe that the Wasserstein distance is the best metric for classifying the images. This is because only the Wasserstein distance corrects for the misalignment of peaks between two different images. In particular, if a peak corresponding to the same chemical compound appears in one grid box for one image, and in an adjacent grid box in another image, both the Euclidean distance and the Principal Component

sample	real	1. guess	2. guess	3. guess	4. guess	5. guess	6. guess	7. guess	first guess	second guess
1	A	A	B	D	C	E	F	G	1	0
2	B	A	B	C	D	E	F	G	0	1
3	A	A	B	D	C	E	F	G	1	0
4	A	B	A	D	C	E	F	G	0	1
5	B	B	A	C	D	E	F	G	1	0
6	A	A	B	D	C	E	F	G	1	0
7	A	A	C	D	B	E	F	G	1	0
8	A	A	D	B	C	E	F	G	1	0
9	C	C	A	E	B	F	D	G	1	0
10	A	A	C	B	E	F	D	G	1	0
11	A	B	A	D	C	E	F	G	0	1
12	C	E	C	A	D	B	F	G	0	1
13	D	D	A	B	C	E	F	G	1	0
14	B	A	B	D	C	E	F	G	0	1
15	E	F	C	G	B	A	E	D	0	0
16	D	D	E	C	B	A	F	G	1	0
17	B	B	A	F	E	C	D	G	1	0
18	E	C	D	A	E	B	F	G	0	0
19	F	E	F	G	B	A	C	D	0	1
20	G	F	E	B	G	C	A	D	0	0
21	F	G	F	E	B	A	C	D	0	1
22	G	G	F	E	B	C	A	D	1	0
								total	55%	32%
									86%	

sample	real	1. guess	2. guess	3. guess	4. guess	5. guess	6. guess	first guess	second guess
1	1	1	2	5	4	3	6	1	0
2	2	1	2	3	5	4	6	0	1
3	2	1	2	3	4	5	6	0	1
4	1	2	1	3	4	5	6	0	1
5	2	2	1	3	5	4	6	1	0
6	2	2	1	3	4	5	6	1	0
7	3	4	5	3	2	6	1	0	0
8	4	3	4	2	5	6	1	0	1
9	5	5	3	6	4	2	1	1	0
10	3	4	5	3	6	2	1	0	0
11	4	4	3	5	6	2	1	1	0
12	5	6	5	3	4	2	1	0	1
13	5	5	4	3	2	6	1	1	0
14	4	4	3	5	6	2	1	1	0
15	6	5	6	4	3	2	1	0	1
16	5	5	6	4	3	2	1	1	0
17	4	4	3	5	6	2	1	1	0
18	6	5	3	6	4	2	1	0	0
19	5	6	5	4	3	2	1	0	1
20	6	5	6	4	3	2	1	0	1
21	5	6	5	4	3	2	1	0	1
22	6	6	5	4	3	2	1	1	0
							total	45%	41%
								86%	

Figure 11: Classification using kNN with alternative Wasserstein distance: high success rate

Analysis will fail to see that these two boxes are adjacent, whereas the Wasserstein distance will pick this up.

The main limitation of this result is that we may have overfitted by, at each step, comparing the image that we want to investigate with every other image in the set.

Our main recommendation for future work is to attempt these methods on the raw image data, rather than our estimation of this data.

Another important direction for future work is to conduct sensitivity analysis on the image classification. In particular, it would be important to determine which part of the images provides the most information for the image classification. To do this, we recommend either taking away sections of the image one at a time, or adding sections of the image one at a time, and at each point investigating how good the image classification is. Metrics for assessing the image classification might be determining what percentage of blanks are correctly classified, or how well clustered the blanks and samples are.

6 Conclusions and recommendations

In conclusion, we achieve very limited success in classifying the samples. We are able to classify with 100% success (for the right grid size) whether a test contains a seed or is a blank. We can also identify on which day the experiments were conducted, which suggests that the environment has a significant effect on the results. Beyond these conclusions, we believe that there is too little data to draw any confident conclusions on the success of the various methods that we tried.

To extend the work that we have done, our main recommendation is to test the methods that we have described above on the full, raw data, rather than on our reconstructed data. Depending on which methods work best on the full data, sen-

sitivity analysis should be conducted to find which region of the image is the most important for classification. Such analysis is sometimes called feature construction, or greedy forward/backward selection.

If future experiments are possible, we suggest changing the experimental set-up to try to minimise the effect of which day the experiment is conducted on. It would also be very important to conduct more "blank" experiments, so that we can confidently establish a ground truth on which parts of the chromatogram are solvent, and which parts are seed. We believe that being able to dismiss some parts of the chromatogram as solvent would significantly improve the success rate of the Wasserstein method.

Another method that could be tried are alignment methods, such as point set registration, which we didn't have time to implement and test. Finally, if information detailing where peaks corresponding to chemical compounds are expected to be found is available, the classification methods should be tested on those regions exclusively.

References

[1] Trung Nghia Vu & Kris Laukens. Getting Your Peaks in Line: A Review of Alignment Methods for NMR Spectral Data. *Metabolites*. 2013 Jun; 3(2): 259276.

[2] Seongho Kim, Imhoi Koo, Aiqin Fang & Xiang Zhang. Smith-Waterman peak alignment for comprehensive two-dimensional gas chromatography-mass spectrometry. BMC Bioinformatics 2011, 12:235.

[3] K. Weinberger & L. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research* **10** (2009) 207-244.

[4] D. Paulus & J. Hornegger. *Applied Pattern Recognition, Fourth Edition: Algorithms and Implementation in C++*, Vieweg (2003).