International Environmental Modelling and Software Society (iEMSs)
2012 International Congress on Environmental Modelling and Software
Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany
R. Seppelt, A.A. Voinov, S. Lange, D. Bankamp (Eds.)
http://www.iemss.org/society/index.php/iemss-2012-proceedings

An application of data mining to fruit and vegetable sample identification using Gas Chromatography-Mass Spectrometry

G. Holmes [a], **D. Fletcher**[a] **and P. Reutemann**[a]

[a] *University of Waikato, Private Bag 3105, Hamilton 3240, New Zealand*
(geoff@waikato.ac.nz, dale@waikato.ac.nz, fracpete@waikato.ac.nz)

Abstract: One of the uses of Gas Chromatography-Mass Spectrometry (GC-MS) is in the detection of pesticide residues in fruit and vegetables. In a high throughput laboratory there is the potential for sample swaps or mislabelling, as once a sample has been pre-processed to be injected into the GC-MS analyser, it is no longer distinguishable by eye. Possible consequences of such mistakes can be the destruction of large amounts of actually safe produce or pesticide-contaminated produce reaching the consumer. For the purposes of food safety and traceability, it can also be extremely valuable to know the source (country of origin) of a food product. This can help uncover fraudulent attempts of trying to sell food originating from countries deemed unsafe. In this study, we use the workflow environment ADAMS to examine whether we can determine the fruit/vegetable, and the country of origin of a sample from a GC-MS chromatogram. A workflow is used to generate data sets using different data pre-processing methods, and data representations from a database of over 8000 GC-MS chromatograms, consisting of more than 100 types of fruit and vegetables from more than 120 countries. A variety of classification algorithms are evaluated using the WEKA data mining workbench. We demonstrate excellent results, both for the determination of fruit/vegetable type and for the country of origin, using a histogram of ion counts, and Classification by Regression using Random Regression Forest with PLS-transformed data.

Keywords: GC-MS; sample identification; data mining; workflows

1 INTRODUCTION

Analysis of pesticide residues in fruit and vegetables is an important part of food safety monitoring. Guidelines and regulations are issued by government agencies, like the EU parliament or the Food and Drug Administration (FDA) in the USA, to protect consumers. Pesticide residues are often detected and quantified using Gas Chromatography-Mass Spectroscopy analysis (GC-MS), which we describe in more detail later. GC-MS typically has limits of detection from the sub parts-per-million down to the low parts-per-billion, depending on the type and setup of the instrument. Modern analysis involves searching for an ever increasing number of pesticide compounds, typically numbering in the hundreds.

Different products can have very different permitted pesticide residue levels. For this reason, ensuring that the sample being analysed is– as labelled– is very important. Mislabelling can happen through human error during sample preparation in the laboratory, or by the client before being sent to the laboratory. A laboratory will often receive samples that have already been prepared, and the original material can not be visually identified. Mislabelling can have serious consequences, either in food safety, with produce being sent to market with pesticide levels above permitted thresholds, or economically, with produce being destroyed unnecessarily.

The question we examine in this paper, is whether we can use the chromatographic data alone to distinguish between types of produce samples. Thus enabling us to identify whether a sample was labelled correctly and the appropriate pesticide level(s) applied. The machine learning approach presented here uses a qualitative approach to GC-MS (i.e., determining sample type), in contrast to the usual quantitative approach in laboratories (i.e., determining compound concentrations).

One way of answering this question is to determine a list of compounds that are expected to differentiate the various fruits and vegetables, and explicitly measure those in addition to the pesticides. Modelling would then be done on the determined concentration of these compounds. However, assuming that a suitable list of compounds could be readily found, this approach places a significant extra burden on the analysis (in terms of both time and cost). Also, the historical database of sample data would have to be re-analysed to measure the concentrations of these extra compounds. Depending on the compounds chosen for extra analysis, the GC-MS application may have to be altered (e.g., additional internal standards, or increased run-time), which would mean the existing database would not be compatible with newly analysed samples. Instead, the approach we use is to see what we can determine automatically by a combination of data pre-processing (primarily noise reduction techniques) and machine learning starting from the raw chromatogram as produced by the instrument, with purely automated techniques. With this approach, the entire database of chromatograms is available for analysis, and a system which alerts to the possibility of an incorrectly labelled sample would require no changes to laboratory practice.

The remainder of the paper is structured as follows: first, we give a short introduction into the GC-MS domain in Section 2, before explaining some of the pre-processing techniques in Section 3. Then, since we are using a workflow system for most of the tasks presented here, Section 4 explains the workflow engine in detail. After that, Section 5 shows the conducted experiments and their results, before Section 6 concludes the paper.

2 GC-MS

According to WikiPedia [2012], GC-MS is “a method that combines the features of gas-liquid chromatography and mass spectrometry to identify different substances within a test sample. Applications of GC-MS include drug detection, fire investigation, environmental analysis, explosives investigation, and identification of unknown samples”. In Holmes et al. [2010], we showed the successful application of data mining in environmental analysis, predicting concentrations of polycyclic aromatic hydrocarbons in soil and water samples.

2.1 Sample analysis

The sample under investigation is injected into the column head by the sample injector (see Figure 1). The carrier gas then propels the sample through the capillary column. Depending on the chemical and physical properties of the molecules (e.g., size), these will elute (exit) the column at different times (retention time). When they elute, the mass spectrometer breaks up the molecules into charged fragments, e.g., using an ion source, and the fragments are subjected to a magnetic field. Depending on their mass-to-charge ratio (m/z), the flight path of the fragments will differ and end up in different ion traps used for counting (abundance count).

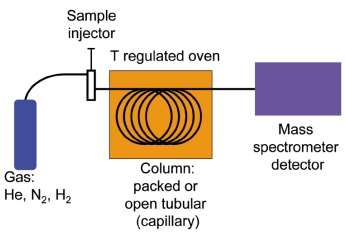

Figure 1: Instrument schematic.

2.2 Chromatograms

A GC-MS instrument produces data for a sample termed a **chromatogram**. This is essentially a succession of ion abundance counts produced by the mass spectrometer. Figure 2 depicts a potato sample. The top panel shows the total ion count, which is the sum of all ion abundances at each time stamp. The bottom panel shows the mass-spectrometer (mass-spec) data from a single time stamp. Here, the x-axis is m/z of the ion, and the y-axis is the abundance.

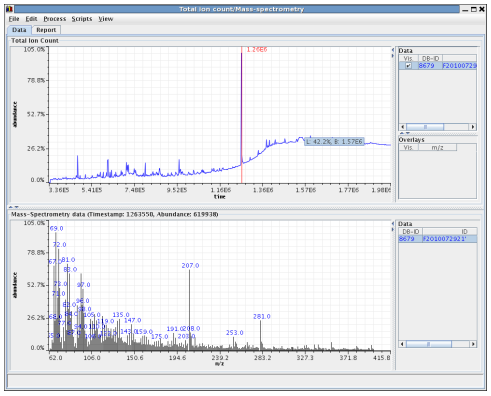

Figure 2: Raw chromatogram.

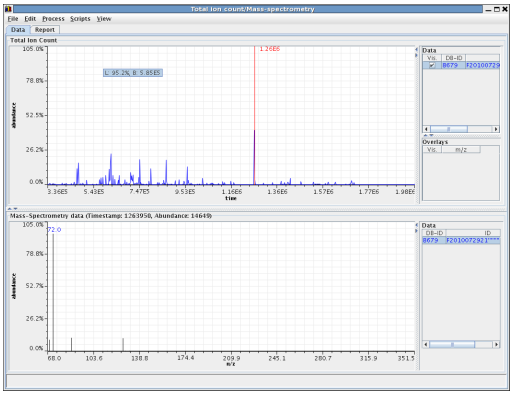

Figure 3: Cleaned up chromatogram.

One way of viewing this data, is that the peaks in the top panel (total ion count panel) represent compounds eluting from the gas chromatograph, and the bottom panel provides further detail about the compound provided by the mass-spectrometer. Compounds have an expected mass spectrometry profile - the ion abundance ratios provide a fingerprint for a particular compound. The combination of the expected retention time of the peak in the top panel, and the mass-spec fingerprint in the bottom panel can usually identify the compound producing the peak. Difficulties arise when compounds elute at the same time, or, as the top panel in the above example shows, with the rising baseline, when contamination leads to ion abundances being present across large sections of the chromatogram, and thus confusing the mass-spec fingerprint. Later, we describe some techniques to reduce noise in the ion abundances.

2.3 Pesticide data

The pesticide data used in this study was generated by two Thermo Scientific instruments (Trace GC 2000 Series Voyager GC/MS) in full-scan mode, collected over nearly two years. The data set consists of approximately 8000 labelled chromatograms, from over 100 different types of fruit and vegetables, and 120 countries. Each chromatogram has over 4000 scans for each of approximately 350 m/z ratios (ions). This results in around 1.4 million data points for each sample.

3 Pre-processing

The raw data, as generated by the instrument management software is not generally amenable to processing by machine learning systems. Firstly, chromatograms are multi-dimensional, and need flattening in some way. This is compounded by chromatograms having a differing number of data points in each dimension (both in the number of scans, and in the ion abundances per scan). Machine learning software, such as WEKA (Hall et al. [2009]), use the attribute-value format for data storage, where an attribute is expected to have the same meaning across examples. Therefore some form of normalisation is necessary across samples. The issue of retention time drift also has an impact on

normalisation. Either due to contaminants building up in the column of the gas chromatograph, or the constituents of the sample, or previous samples run through the instrument, or even routine maintenance, the time taken for compounds to emerge from the instrument may vary from run to run. As a result, the x-axis (time) shifts non-linearly over time. During the normal course of operation, a column can also be trimmed to remove contaminants, or replaced entirely, resulting in a very different chromatographic profile. Several alignment algorithms, for example, Correlation Optimised time Warping (Tomasi et al. [2004]), have been proposed to account for these shifts.

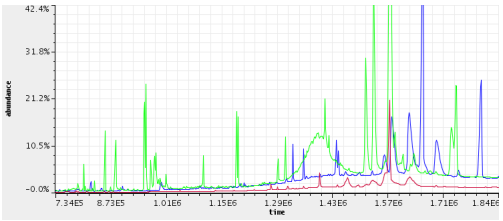

Figure 4: Three “beans with pods”.

It is clear from examining chromatograms over time however, that the non-linear retention time shift exhibited is difficult to overcome. With complex, noisy chromatograms containing many peaks that differ between quite different samples (Figure 4), alignment algorithms can produce very different alignments depending on parameter settings and the freedom permitted. With this in mind, we have investigated a flattening scheme that does not use the time axis, i.e., no use of alignment methods. Instead we have experimented with various strategies using the m/z ratio, and counting the number of occurrences, or presence/absence of m/z ratios, throughout the chromatogram. An issue with this technique, is the presence of noise, particularly that caused by contamination of the column. This leads to ion abundances being measured across large sections of the chromatogram. To compensate for this, we have experimented with a range of noise removal algorithms (and their combinations), combined with flattening techniques. Examples of such techniques include:

- *Windowed ion noise removal (WIN)* - a sliding window of abundances are examined, and a multiple of the lower-quartile of the abundances is subtracted from each abundance.
- *Top x ion abundances (TOP)* - remove all but the x ions with the topmost abundances.
- *Global ion noise removal (IN)* - evaluate each ion for abundances being present over significant parts of the chromatogram. If considered noise, a noise factor is subtracted.
- *Minimum abundance (MIN)* - remove ion abundances if less than a given threshold.
- *Minimum consecutive ion abundance (CON)* - ion abundances are zeroed if they do not appear in a given number of consecutive scans. This is especially useful for cleaning up left-over ion abundances after noise removal filters have been run.

The number of combinations of pre-processing, noise removal, and classification algorithms is very large, and processing is slow due to the size of the data. In order to automate the analysis as much as possible, we use a workflow system to generate data sets. This is described in the following section. Figure 3 is the same potato sample shown earlier, but after noise removal algorithms have been applied.

4 WORKFLOW

4.1 Introduction

The workflow engine of the ADAMS[1] framework (Advanced Data mining and Machine learning System), which is used as the basis for this application, uses a different approach than most current workflow applications. Systems, like Kepler (Ludäscher et al.

[1]The ADAMS base platform, excluding the GC-MS modules, is expected to be released as open-source at the end of 2012.

[2006]) and RapidMiner (Mierswa et al. [2006]), use a “canvas” approach in designing the workflows. In this approach, the user places the various operators (or “actors” in Kepler and ADAMS terminology) on a large canvas and then connects the various inputs and outputs manually. Though this is a very intuitive approach to design, it is also a very time consuming one. When inserting an additional pre-processing step, potentially in multiple places in numerous processing branches, the user ends up moving and rearranging a lot of actors in order to keep the design tidy. Despite methods, like meta-actors that encapsulate multiple actors, zooming in/out or bird’s-eye-view, large workflows quickly become hard to maintain.

Most of our workflows tend to have a tree-like structure, i.e., 1-to-n connections: data comes from a single source, undergoes some general transformation and forks into multiple branches with different transformations, before ending up in, for example, files. A tree-structure could therefore determine how the data flows.

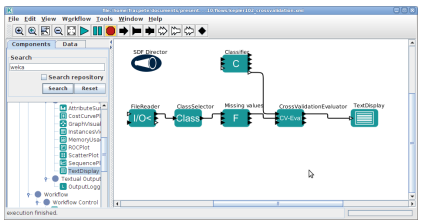

Figure 5: Kepler example workflow.

In a conventional tree, processing steps that follow each other would be in a parent-child relationship. ADAMS uses a slightly different approach as it distinguishes between primitive actors and ones that handle other actors (fixed or variable number). In the latter case, the managed actors become the children of the managing actor, which also knows how the data flow is to be handled. For instance, the *Flow* actor uses the output of one sub-actor as the input to the next sub-actor. The *Branch* actor, on the other hand, forwards the same data to all of its sub-actors, i.e., its branches. The *Tee* actor, like the Unix *tee* command, forks the data into a separate branch. This approach allows for a compacted tree structure. Visual cues are used to hint at how the data flows in the tree: for example, the *Flow* actor uses vertical lines between the sub-actors and the *Branch* and *Tee* actors use horizontal lines for their sub-actors.

Using a tree structure limits the flow connections to 1-to-1 and 1-to-n connections. n-to-1 or n-to-m connections are not possible. In addition, the same output cannot be used several times within the flow. ADAMS adds several mechanisms to alleviate this shortcoming: *global actors, containers, variables, storage*. First, *global actors* are used to allow n-to-1 connections. Special actors reference globally defined actors, therefore synchronising the data flow to these global actors. Second, *containers*, are used to encapsulate multiple outputs in a single data structure as key-value pairs. The *ContainerValuePicker* actor can be used to extract the various values from containers. Third, *variables* can be used to dynamically update options in the flow at run-time, simulating multiple inputs. Fourth, each flow offers internal *storage* for key-value pairs for storing arbitrary data structures. Data can be stored and retrieved multiple times in the flow. This allows for connections across sub-trees in the flow.

4.2 GC-MS Application

The application consists of three phases, each represented by a workflow:

1. **data collection phase:** the training data for the machine learning model is generated.
2. **training phase:** a machine learning model is built using the previously generated data and stored on disk.
3. **prediction phase:** incoming data is pre-processed in the same way, the model applied to obtain predictions and a PDF report is sent via email.

Phase two, the *training phase*, consists of a very simple workflow (load data, train model, save model) and is hence omitted. The other phases are explained in more detail below.

Data collection. Data collection is a time consuming process, due to the size and number of available chromatograms, and can take up to twelve hours. For testing various pre-processing techniques, the data is only retrieved once and then distributed into multiple branches, generating multiple data sets. Figure 6 shows how the data is loaded from the database, using the *ChromatogramIdSupplier* and *ChromatogramDbReader* actors. After that, common pre-processing takes place: this includes noise removal (Windowed ion noise, Global ion noise) and the trimming of the m/z data (removal of m/z points with low abundance, retaining only the topmost common m/z points). Using *Tee* actors, the chromatogram can be branched off and further pre-processed, if required, before being appended to a data set (see Figure 6). For turning a chromatogram into data suitable for WEKA, we only considered m/z ratios from 35 to 395. For each chromatogram a histogram of m/z ion counts is generated, which reduces the number of data points from around 1.4 million to less than 400. The data sets can then be evaluated on multiple classification algorithms using the WEKA Experimenter, to determine the best pre-processing/classification algorithm combination. See section 5 for details and results.

Prediction phase. In order to make the sample identification as easy as possible, the workflow performing the predictions is run as a background process, with the analyst only having to place the raw chromatogram files in a directory to be processed. The workflow (see Figure 7) monitors this directory and starts the processing whenever a file appears. First, it loads the raw chromatogram and subjects it to the same pre-processing that generated the best results in the training phase. Images of the total ion count (TIC) and base ion count (BIC; only most abundant ion at each time stamp) are generated and output. The trained model is applied to the chromatogram and the classification and class distribution obtained to be included in the report. Based on the predicted classification label, the appropriate picture is chosen and, together with a table comprised of the five most likely classifications and the TIC and BIC images, added to a PDF document. The generated PDF report is then sent to the analyst's email address (see Figure 8).

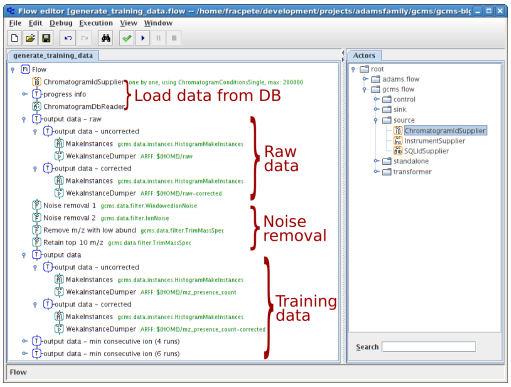

Figure 6: Training data generation.

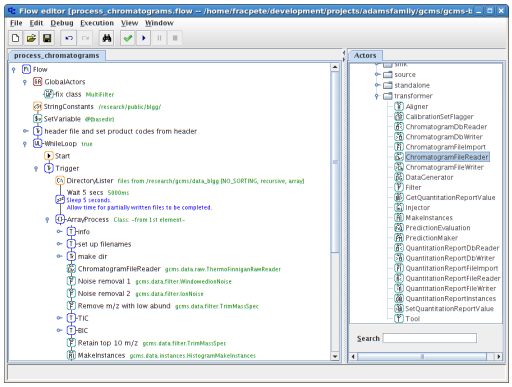

Figure 7: Report generation.

5 Experiments and Results

In this paper, we aim to determine two things; whether we can distinguish the type of fruit/vegetable, and the geographic origin of a sample from its GC-MS chromatogram alone. We use the WEKA suite for comparison of modeling techniques using the pre-processed data generated using the workflow system previously described. The data sets were analysed using the comprehensive 10 by 10 cross-validation with significance testing to 5% significance via the corrected paired t-test (Nadeau and Bengio [2003])

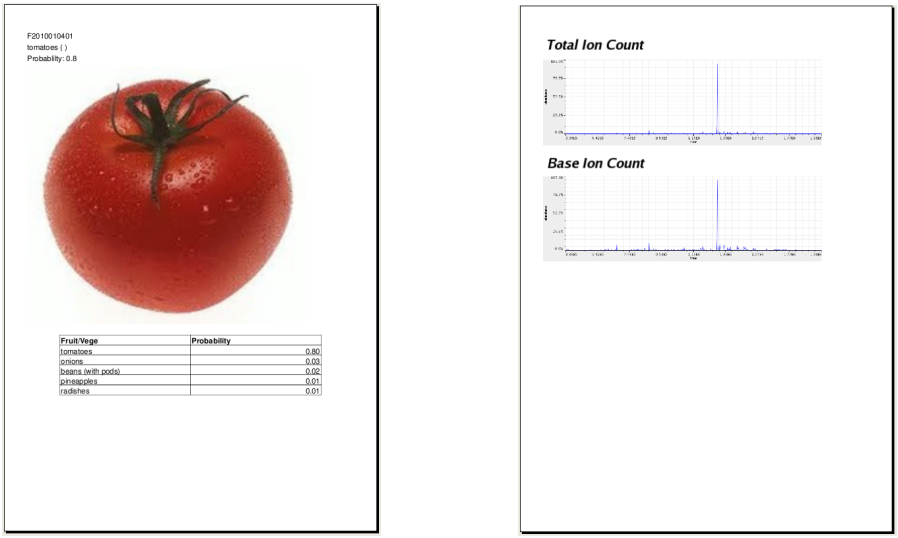

Figure 8: Sample PDF report.

as used in the WEKA Experimenter. The results shown are from five algorithms, Naive Bayes, J48 (WEKA's C4.5 implementation), Random Forest, and two examples of Classification Via Regression, one using Partial Least Squares (PLS), and the other using a Random Regression Forest of PLS-transformed data.

5.1 Fruit/vegetable type

Table 1 shows results for the five machine learning algorithms, using the raw and pre-processed (noise removal) chromatographic data. The numbers shown are the percentage correctly predicted on average over 100 randomised cross-validation splits. The standard deviations are given after the $\pm$. Significant improvement in the prediction accuracy is apparent in the pre-processed data, with the best result approximately 95% correct for the Classification Via Regression model, using Random Regression Forest with PLS-transformed data. The wide range of results shows the impact of pre-processing on an algorithm's performance, with the best combination of these two being the ultimate goal. A *CVR-RRF* model built on the *Clean* dataset is used in production.

Dataset	NB	J48	RF	CVR-PLS	CVR-RRF
Raw	11.92 ± 1.32	$51.31 \pm 1.83\circ$	$67.79 \pm 0.60\circ$	$45.40 \pm 1.20\circ$	$66.69 \pm 1.18\circ$
WIN	10.81 ± 0.85	$40.99 \pm 1.16\circ$	$62.58 \pm 1.53\circ$	$70.81 \pm 1.34\circ$	$88.17 \pm 0.69\circ$
Clean	76.23 ± 0.90	77.20 ± 1.44	$92.36 \pm 0.58\circ$	$89.44 \pm 0.99\circ$	$94.77 \pm 0.78\circ$

Table 1: $\circ$,$\bullet$ stat. significant improvement/degradation; Clean=WIN+IN+MIN+TOP+CON.

Incorrectly classified chromatograms mainly stem from vegetables that are hard to distinguish, like onion/shallot and beans/peas with and without pods, and vegetables with very few occurrences in the database, like olives and figs (less than five samples).

5.2 Country of origin

For the analysis of the geographic origin of the sample, there is less data due to the origin not necessarily being recorded in the database. We have chosen to split the data set into produce type, to see if the origin of a particular fruit or vegetable can be determined. For example, to determine if we can distinguish an apple grown in x from an apple grown in y. This reduces the amount of data for testing, as many types of produce have primarily one source in the data we have at hand.

Dataset	NB	J48	RF	CVR-PLS	CVR-RRF
Beans	75.83 ± 6.80	77.79 ± 11.12	$89.86 \pm 5.28\circ$	$94.67 \pm 6.37\circ$	$94.67 \pm 6.37\circ$
Grapes	88.12 ± 7.69	84.87 ± 7.49	94.68 ± 3.34	$96.32 \pm 2.29\circ$	$96.72 \pm 3.23\circ$

Table 2: ○,● statistically significant improvement or degradation.

Table 2 shows results distinguishing "Beans with pods" between three geographic origins, Kenya, Morocco and Thailand, and "Table Grapes" between India and South Africa (insufficient data for other origins). The results show we can clearly distinguish the geographic origin for those fruit/vegetable types. Further investigation would be needed to understand whether it is truly geography - such as trace elements from local growing conditions, or perhaps that the different geographic locations grow different subtypes of a particular fruit/vegetable. In the database available to us, very few subtypes are labelled. Labelling will increase over time as new samples are taken, making further study possible once sufficient data is collected.

6 Conclusion and Future work

We have shown in this paper that data mining can be used to support analysts in laboratories in terms of quality control, minimizing the impact of accidental sample swaps. Clearly, proper pre-processing of the data is paramount for achieving good results, obtaining as few misclassifications as possible. The promising results of the "origin" test could lead to future work, involving how this approach can be applied, for example, for testing produce shipments for potential fraud. Furthermore, it still needs be verified that our approach works across more than just two instruments as used here.

7 Acknowledgement

We would like to thank BLGG AgroXpertus, Wageningen, NL, for providing us with the GC-MS data and generously letting us publish our findings.

References

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11, 2009.

Holmes, G., D. Fletcher, and P. Reutemann. Predicting polycyclic aromatic hydrocarbon concentrations in soil and water samples. In *Proc International Congress on Environmental Modelling and Software*, 2010.

Ludäscher, B., I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience*, 18:1039–1065, 2006.

Mierswa, I., M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)*, 2006.

Nadeau, C. and Y. Bengio. Inference for the generalization error. *Machine Learning*, 52 (3):239–281, 2003.

Tomasi, G., F. van den Berg, and C. Andersson. Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *Journal of Chemometrics*, 18(5):231–241, 2004.

WikiPedia. Gas chromatography-mass spectrometry — Wikipedia, the free encyclopedia, 2012. [Online; accessed 23-Feb-2012].