Investigating the Use of Peak Clustering for the Identification of Metabolites

James Ferguson

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

7th August 2015

Abstract

Mass spectrometry (MS) is an analytical scientific tool for identifying the constituent molecules that make up a given chemical or biological substance. Its application to to metabolomics, the study of small molecules called metabolites that are found in biological systems, has many medical applications and is therefore an area of much interest at present.

One of the main challenges associated with mass spectrometry is handling the large volumes of data produced as output. This presents a number of issues in correctly identifying which metabolites are present for a given sample.

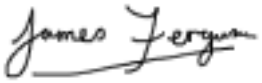
The aim of this project is to use two algorithms, namely the Gibbs sampling algorithm and variational Bayes, to combine this output into a smaller number of groups called clusters. The algorithms are used to group the output in such a way that each cluster relates to the same molecule. These clusters can then be matched to possible molecules in order to identify the sample metabolite's make-up.

There are two main stages to this project. The first involves creating a software implementation of the two clustering algorithms which can be applied to the output from a mass spectrometer for a given sample of metabololites. The second stage involves matching the clusters obtained from the implementations of the clustering algorithms to candidate molecules and analysing the results obtained from this matching process.

Of particular interest is the adduct pattern associated with the matched molecule. Adducts are produced during the ionisation stage of the mass spectrometry process by bonding each of the sample's molecules with charged ions. This ionisation process is random, and there are various different ions to which each molecule may be bonded with. It is believed that the pattern of the adducts formed by each molecule may be used to distinguish between to isomers (molecules with the same constituent molecules but having a different structure). By studying the adduct patterns identified from running the clustering algorithms developed on two samples known to contain isomers, it is believed that these adduct patterns may indeed have some predictive power. Having identified this, this dissertations sets out some areas for further work with a few to investigating this further.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: James Ferguson Signature: 

Acknowledgements

I'd like to that Dr Simon Rogers for his help in conducting this project.

Contents

Chapter 1

Introduction

1.1 Problem Context

1.1.1 Mass Spectrometry

Mass Spectrometry (MS) is an analytical scientific technique used to identify a chemical or biological sample's constituent molecules [8]. It has a wide range of applications in areas such as drug discovery, disease diagnosis as well as general chemistry and biology theory. [4]

The main application area of MS considered in this dissertation is in the field of metabolomics - that is, the study of small molecules called metabolites that are found in biological systems. Understanding the structure of the full set of metabolites in a biological system (referred to collectively as the metabolome) has many medical applications and is therefore an area of much interest at present. [3]

MS is carried out through the use of a scientific device called a mass spectrometer. A chemical or biological sample whose chemical composition is to be studied can be added to the mass spectrometer and the device then carries out the mass spectrometry process on it. Once complete, the spectrometer produces data which can then be studied in order to identify the sample's chemical structure. As described in [4], the main steps of the mass spectrometry process are as follows:

1. **Introduction** - This stage concerns the process by which the sample is added to the mass spectrometer. Before introducing a sample to the spectrometer, it must first be separated into its constituent molecules which are to be analysed (analytes). There are different methods for doing this however the method considered in this dissertation is through Liquid Chromatography - that is, this dissertation considers Liquid Chromatography Mass Spectrometry (LC-MS). Liquid chromatography relies on the fact that the different analytes will pass through a column in liquid form at different times due to their individual chemical properties (hence separating them out). The amount of time required for an analyte to pass through the chromatography stage and enter the mass spectrometer is called its **Retention Time** (RT). The RT value itself provides a large amount of useful information about the analytes and is of much use in analysing MS data.

2. **Ionisation** - Once separated out, each analyte is given a charge by ionising it (adding or removing electrons from its atoms). This ionisation process is carried out by adding an atom/molecule to each analyte with a given charge called an **adduct**. This process is random and there are variety of different adducts an analyte may be bonded with. This ionisation process means that there can be multiple peaks observed for each metabolite.

3. **Mass Detection**: This is carried out inside the spectrometer and is used to identify the **mass to charge ratio** (mass per unit charge) for each of the now charged analytes by using calculations on their movement through an electromagnetic field. This process gives a profile of mass to charge ratios for a given retention time along with their intensities (the frequency with which each ion is observed).

4. **Data Processing** The spectrometer outputs data a spectrum for each retention time which shows intensity against mass to charge ratio. This plot Each individual intensity value observed for an ion with a given mass to charge ratio is called a **peak**. (See Figure 1.1 for an example.)

5. **Quantification** - The next step in the process is to interpret the output and identify the molecules present in the sample by deriving their masses from the mass to charge ratio values for each ion. Along with being computationally intensive due to large of data produced from the mass spectrometer, accurately matching peaks to molecules presents a number of other challenges.

The focus of this dissertation is on the Quantification step and how to match the peaks produced from the mass spectrometer to a database of potential constituent molecules for a given sample. The next section outlines the main challenges faced in this stage of the MS process and defines the problem that this dissertation seeks to address.

1.2 Problem Definition

1.2.1 Motivation

Analysing the output in the Quantification step of the mass spectrometry process presents a number of challenges. The aim of this step is to take the peaks produced from the analysis and match these to molecules with a view to correctly identifying the constituent molecules of the sample being studied.

In metabolomics, one method used is to compare the spectrum of peaks produced in the MS output against a database of known masses of standard metabolites. An example of what an observed spectrum of intensity peaks may look like is shown in Figure 1.1.

The aim is then to match the peaks to these molecules. For example, this can be done by comparing the mass represented by each peak against the database of masses for known molecules. Other properties such as retention time could also be compared.

There are some challenges associated with this method however. For example, there is still limited knowledge about all of the naturally occurring metabolomes - indeed, the structure of the human

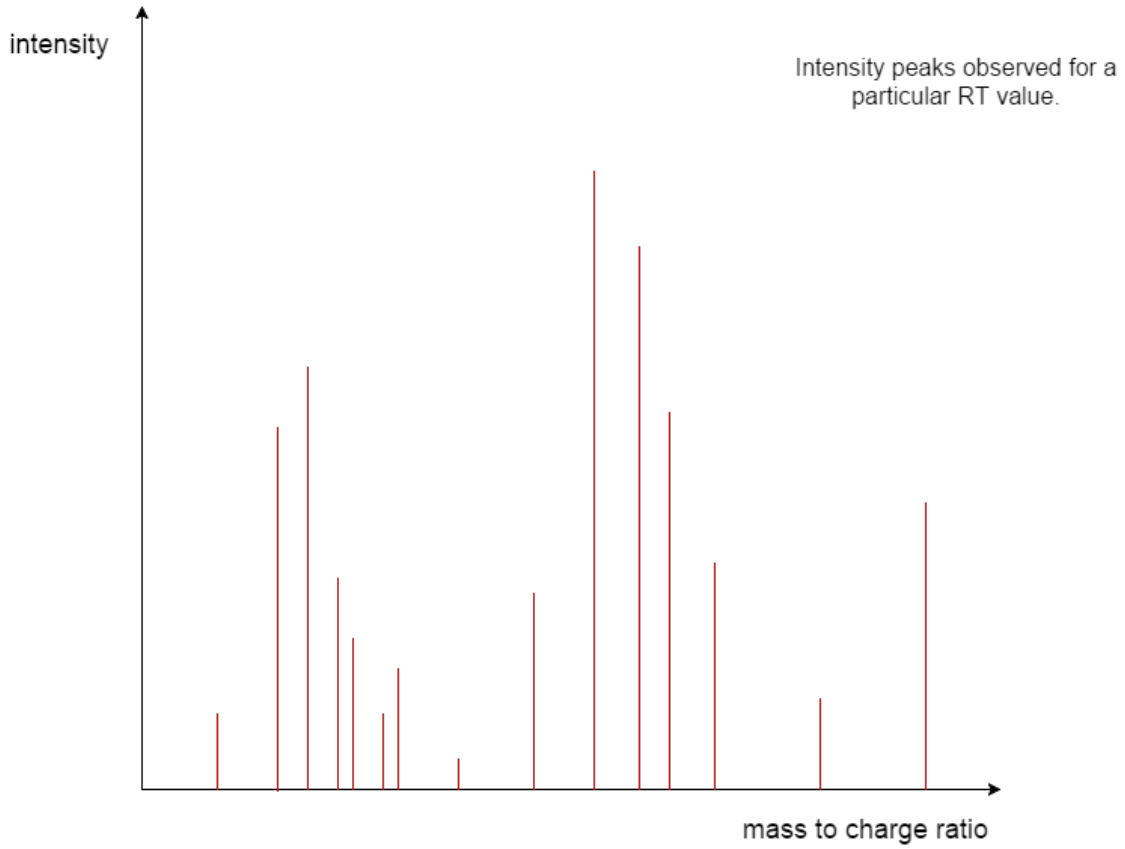


Figure 1.1: Sample plot of a spectrum for a fixed RT value. Each vertical line shown represents an intensity **peak**

metabolome is not yet fully understood. [3] This therefore places a restriction on our ability to construct a fully comprehensive database containing all potential constituent molecules for a given sample. Also, the results of a mass spectrometry experiment are greatly influenced by the experimental conditions under which it is carried out. This presents a challenge in creating standardised data for use in constructing a database. Despite these issues, a number of there are a number of openly available metabolomics databases available (see [3] for further details) and one of which is used for the analysis carried out in this dissertation.

Another challenge in metabolite identification from MS output is due to the large volume and complexity of the data produced by the mass spectrometer. In particular, as stated in [1], the number of peaks at different mass to charge ratios greatly outnumbers the collection of possible metabolites in a given sample. As such, in matching peaks to metabolites the chance of a false-positive (incorrectly matching a peak to given metabolite in the database) is high if this is not accounted for in the analysis of the peaks. One reason for the large number of peaks observed can be attributed to noise terms, impurities etc. that are not actually attributable to the molecule. Another reason is due to the ionisation process. As described in Ionisation stage of the mass spectrometry process discussed in the previous section, the same metabolite in a sample will likely form different adducts (with different mass to charge ratios) and therefore will be represented by different peaks in the MS output. Motivated by this, the problem considered in this dissertation centres on developing a computational method for matching peaks to metabolites in such a way as to reduce the number of false positives. [3]

1.2.2 Problem Definition

As discussed in the above section, the large number of peaks produced from MS can lead to peaks being incorrectly matched to molecules. The two main reasons for this are that not all peaks actually correspond to a an actual constituent molecule for the sample (they may be an impurity in the experimental process) and that peaks corresponding to actual molecules relate to their mass after the ionisation process (i.e. their mass plus the mass of an adduct) as opposed to their actual pre-ionisation **precursor mass**.

With a view to creating a method for matching peaks to molecules which reduces the number of false positives, this dissertation focuses on developing a computational tool which reduces the number of peaks to be matched by combining them into clusters. These clusters are formed using an algorithm which identifies peaks which are likely to relate to the same molecule by analysing their mass, RT and intensity values as they are obtained from a mass spectrometer.

This dissertation sets out a statistical model which can be used allocating peaks to clusters. In this model, each cluster is modelled as a bivariate normal distribution over pairs of precursor mass and RT values. The precursor mass for a peak can be calculated from its mass to charge ratio by applying a transform with parameters dependant on the adduct which has been applied to the molecule. There are a finite number of possible adducts and therefore a finite number of possible precursor masses associated with a given peak. The possible clusters that a peak can be belong to can therefore be obtained by applying the transform to its mass to charge ratio associated with each adduct and comparing this along with its RT value to the mean precursor mass and RT values of each cluster. If the values associated with a peak after applying one of the transforms on its mass are within acceptable range of those of a given cluster, then this cluster is a possible cluster to which this particular peak may belong. For each peak, this comparison can be made against each cluster for each transform in order to obtain a list of possible clusters to which the peak may belong. A mathematical clustering algorithm can then be used to allocate a particular peak to one of its possible clusters. The algorithms used in this dissertation for this purpose are the **Gibbs sampler** and the **Variational Bayes** clustering algorithms.

Having clustered the peaks together, the molecule identification problem has now been simplified - we need now only consider matching the representative masses for each cluster to the molecule of databases. If the clustering has been carried out correctly, the confidence that each cluster mass relates to an actual molecule in the database will be greater than for that of an individual peak (which may in itself we attributable to experimental error). Therefore, the use of clustering will reduce the probability of false positives in matching and improve the overall accuracy of the MS process.

1.3 Overview

This dissertation will set out the development of a software product which can be used to implements the peak clustering algorithms for a given set of MS peak data.

Chapter 2, discusses the current approaches to taken to peak matching. Here, an existing software tool for analysing MS data that also seeks to cluster peaks before matching them to molecules is discussed.

Chapter 3 sets provides an overview of the requirements of the software product developed and describes how they were gathered.

Chapter 4 provides an overview the key design decisions made in developing the software product.

Chapter 5 describes the mathematical foundation of the clustering techniques used and their implementation as a computer program. Mathematical derivations of the Gibbs sampler and variational Bayes algorithms are first provided. It is then described how these mathematical algorithms have been translated to a software implementation, with details of the software design, algorithms and data structures used in the implementation being discussed.

Chapter 6 describes presents an evaluation of the software tool developed and interprets the output it produces.

The final chapter, Chapter 7, evaluates the current status of the software tool and provides suggestions for further work.

Chapter 2

Survey

Developing effective algorithms that can be used to match peaks from mass spectrometry data to molecules is an area of much interest. As discussed in [1], the main differences in the algorithms developed to do this is in how they handle the cases where multiple peaks in the MS output can correspond to the same molecule (e.g. because of adducts formed during the Ionisation process). If the presence of these peaks in the data is not allowed for when developing an algorithm, then this will likely lead to a number of false positive matches. This is because the masses of these peaks may closely resemble those of other molecules when they are compared against a database of known molecular masses.

The use of clustering methods for this purpose is a relatively new area. In this dissertation, the Gibbs sampling and variational Bayes algorithms will be used to address this issue by clustering together peaks that likely to belong to the same molecule.

One existing software package which clusters peaks before matching them to molecules is mz-Match. It does this using a different method to the approach taken in this dissertation however. As described in [2], it does this using a greedy clustering algorithm. This algorithm seeks to identify peaks relating to the same molecule using the fact that such peaks should have similar retention times and intensity profiles (intensity values plotted at each retention time). The main steps in the algorithm, as set out in [2], are as follows:

1. while not all peaks have been clustered
    1.1. Identify the first peak with the greatest intensity value
    1.2. Using this peak form a new cluster
    1.3. For each non-clustered peak, compare its intensity profile to that of the cluster-forming peak (by calculating the Pearson correlation, see [2])
2. Terminate

The main issue with this algorithm, as identified in [2], is that once a cluster of peaks has been formed then all of the peaks contained in it are no longer considered for the rest of the algorithm. For example, it may be the case that a particular peak is not allocated to a cluster because of how it compares with the peak used to form the cluster. However, had it been compared with one of

the other peaks in the cluster then it would have been clustered with this peak. This effect results in some peaks not being allocated with any others and are allocated to their own individual cluster. Peaks allocated to such clusters can introduce false-positive classifications when matching against a database of molecules.

In comparison with this method, the Gibbs sampler and variational Bayes clustering methods considered in this dissertation follow a Bayesian approach. In each iteration of these algorithms, the current cluster allocations can be updated in light of new information. This will help address the issue identified with the algorithm used in the mzMatch software. On the whole, using the Gibbs sampler and variational Bayes algorithms with return fewer matches than the greedy algorithm used in mzMatch however fewer matches will be false positives.

Chapter 3

Requirements

3.1 Requirements Gathering

The requirements for the peak clustering software product that has been developed were gathered through consultation with the project client - Dr Simon Rogers from the School of Computing at the University of Glasgow.

Several meetings were held during which the client described problems relating to peak clustering of mass spectrometry data. Following each meeting, the problems described by the client were considered and the key requirements which the software product needed to fulfil were elicited. A software solution would then be prepared to meet the identified requirements and demonstrated to the client. Following each demonstration, the client could suggest changes where the software didn't quite meet their needs and propose further areas for consideration that would then lead on the further requirements being established. Repeating this process, the requirements were gathered and refined iteratively throughout the development of the project.

3.2 Product Requirements

Initially, the main requirements for the software related to implementing the Gibbs sampling and variational Bayes algorithms and using these implementations to cluster peaks of data and match these clusters to molecules. In addition to the requirement that the software must implement these algorithms, other requirements such as the format of the peak data that must be read, how long the software should take to run and the format of the output it must produce were also identified.

Once these initial requirements had been met, further requirements were identified which would build on what had been developed so far. For example, an area of interest to the client was whether the adduct patterns identified for each molecule obtained from the peak clustering process had any predictive ability for identifying molecules. Further software requirements were identified in order to attempt to answer this question.

The main requirements that had to be met by the software product are as follows:

1. The product must be able to read the raw peak data produced from a mass spectrometer from a text file with a predefined format.
2. The product must contain an implementation of the Gibbs sampling algorithm and be able to use this to allocate the peaks to clusters using their precursor mass, retention time and intensity values.
3. The product must be capable of running in excess of 30 peak data files in a single run.
4. The product must be capable of processing 30 peak data files with 1 hour.
5. The product must contain an implementation of the variational Bayes algorithm and be able to use this to allocate the peaks to clusters using their precursor mass, retention time and intensity values.
6. The product must produce as output from each clustering algorithm text files showing which peak the cluster has been allocated to.
7. For a given sample, the product must be able to match a the clusters identified to its constituent molecules.
8. The product should produce plots showing the frequency that each adduct is observed for a given molecule across a number of input peak data files
9. The product should produce plots showing the mean and variance of the intensity observed for each adduct for a given molecule across a number of input peak data files

Chapter 4

Design

This section provides an overview of the key design decisions made in the development of the peak clustering software product.

4.1 Object Orientation

It was decided that an object orientated approach would be taken in the design and development of the software.

As described in the previous chapter, key requirements for the software product are that it must implement the Gibbs sampling and variational Bayes clustering algorithms. Prior to development, these algorithms were first derived mathematically. It was decided that using an object orientated approach would provide a strong framework for translating the mathematical models into software. This was done by identifying the key elements being described by the models and then translating these into classes. For example, the aim of the models is to allocate peaks from raw mass spectrometry data to clusters. In light of this, a Peak and a Cluster class were the first classes identified for implementing the clustering algorithms.

Having identified the main classes, the next step was the to identify the properties that each class should have to be able to implement the algorithms. For example, it was identified that each peak should have a mass, retention time and intensity value and therefore these should be added as properties of the Peak class.

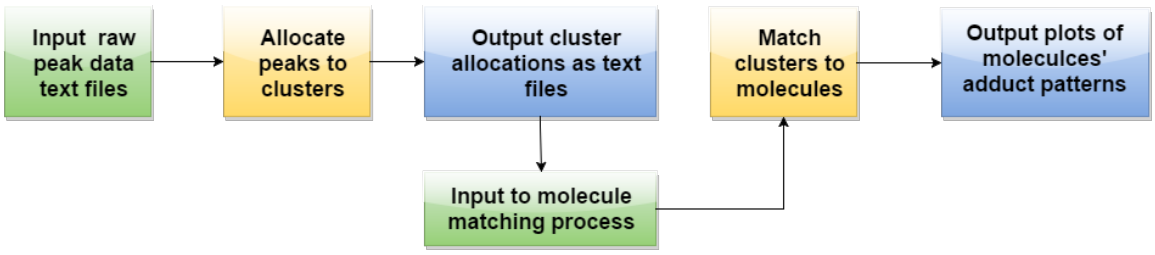
Having formulated an initial class diagram, this was used to begin implementing the algorithms as software. The class structure was then modified and updated throughout development with classes being added and updated in the overall design as required. Details of the overall class diagrams used in the development of the software product can be found in section B.1 of Appendix B.

An alternative design choice would have been to have implemented the algorithms procedurally. This approach would likely have have provided some memory efficiencies and hence faster running times than the current object orientated approach. However, it was decided that this would would be out-weighed by the overall design benefits offered by object orientation. In particular, the ability

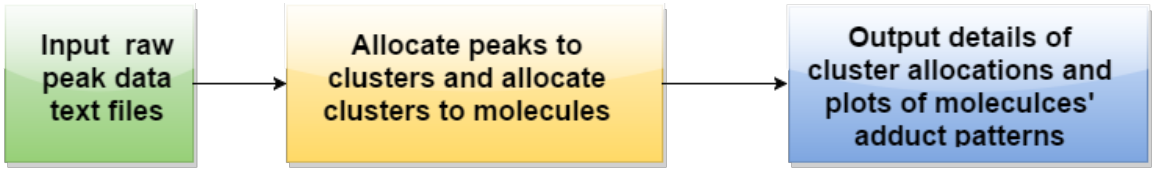
to encapsulate the key model parameters within classes is particularly helpful in gaining an overall understanding of what each element of the software product does without having to delve deeply into the code. Hence, it was decided that use of object orientation would provide an overall higher degree of clarity than offered by a procedural approach and that this would more compensate for any slight performance trade-offs.

4.2 Overall Structure

Another key design design was how the overall software product would be structured. The are essentially two main steps that the software had to implement to meet its requirements. It first has to read in the raw peak data and then cluster these peaks using the the clustering algorithms. Having done this, the next step is then to allocate clusters to molecules and generate useful output on the adduct patterns of the molecules. It was decided that this process would be implemented as a pipeline as shown below:



In the above pipeline, it can be seen that the output text files from the cluster allocation output are used as input for the molecule matching process. An alternative design would have been to instead to combine the clustering and molecule matching steps into one single process as shown below:



However, it was decided that making the two steps independent from one another would offer a number of advantages. In practice it may be desirable to run either of the two steps on its own. If they were combined into a single process then it would be necessary to wait for both processes to run each time. This approach also reduces coupling in the system since the molecule matching process is only dependent on the text files produced by the clustering algorithms. Hence, the underlying design of the clustering model could be modified and there would be no need to make any changes to the molecule matching process (as long as the format of the text files it produced remains the same).

Details of each file used in the implementation of the design are included in section B1.3 of Appendix B.

Chapter 5

Implementation

This section describes the mathematical framework behind the peak clustering algorithms which are used in the software product that has been developed. Two clustering models have been used in the software's implementation, namely, the Gibbs sampling and Variational Bayes algorithms.

First, a derivation of the overall statistical model used to implement these methods will first be provided. This statistical model will then be used as a basis for describing the two algorithms mathematically. Having laid the mathematical foundations, it will then be described how the two algorithms have been implemented in software in terms of their overall design and the key data structures which have been used in their implementation.

5.1 Statistical Model

5.1.1 Key terms and Assumptions

It is assumed that a collection of N peaks representing combinations of mass, retention time and intensity values are produced as output from the mass spectrometer. The mass values observed correspond those of each molecule's adducts however it is their precursor masses that are of interest. For a given adduct, A, a transformation T_A exists which may be used to obtain the precursor mass from the observed adduct mass. However, the adduct corresponding to each peak is not known at the outset and it will therefore be necessary to establish candidate adducts for each peak as part of the modelling process. The mass and retention time values associated with the i^{th} peak (X_m^i and X_{RT}^i respectively) are assumed to be independent random variables and will be modelled as a random pair $\mathbf{X}_i^A = (T_A(X_M^i), X_{RT}^i)$, where $i = 1, 2, ..., N$.

The aim of the clustering model is to allocate each of these N peaks to K clusters ($K \leq N$) under the assumption that the peaks belong to one of K bivariate normal distributions. That is a normal-mixture model will be fitted to the data. A mixture model is a general class of statistical model in which the population of interest can be split into sub-populations (or clusters) and a model can then be applied to each of these sub-populations individually. (See chapter 18 of [6] for further details on mixture models.) In this case, the model assumes that the population of peak data can be subdivided into K clusters in each of which a bivariate normal distribution can be fitted.

Each peak n must be allocated to exactly one of the K clusters. The cluster allocation for peak n is modelled as a vector $\boldsymbol{z}_n$ of dimension K. Assuming that this peak belongs to cluster k, $\boldsymbol{z}_n$ will contain a one in entry k, indicating that this is the peak's allocated cluster, and a zero in all other entries. That is, each entry of $\boldsymbol{z}_n$ is an indicator variable

$$z_{nk} = \begin{cases} 1, & \text{if peak n is in cluster k} \\ 0, & \text{otherwise} \end{cases}$$

and $\sum_{k=1}^{K} z_{nk} = 1$.

Prior to the cluster modelling process, a subset of possible clusters that each peak can belong to will first be identified. (See section 5.3.3 for details.) Hence, for each n, z_{nk} will be known to be zero at the outset for a number of values of k and a subset of the values $\{1, 2, ..., K\}$ need only be considered for each peak. Also, each peak can only belong to each cluster k under a single mass transformation T_k. So, if a peak belongs to cluster k, then the appropriate adduct transfrom T_k and its corresponding precursor mass $T_k(x_m)$ is known. The mass and RT values for a peak n associated with a cluster k are then:

$$\boldsymbol{X}_n^k = (T_k(X_M^n), X_{RT}^n). \tag{5.1}$$

The prior probability that a peak, n, is allocated to each cluster , is described by a vector of probabilities $\boldsymbol{\pi}_n$, where each element π_k represents the probability that a peak is allocated to cluster k ($1 \leq k \leq K$) and $\sum_{k=1}^{n} \pi_k = 1$.

The distribution of $\boldsymbol{z}_n$ is modelled in terms of these probabilities as a multinomial distribution where, for each $n \leq N$:

$$p(\boldsymbol{z}_n|\boldsymbol{\pi}) \propto \prod_{k=1}^{K} \pi_k.$$

A multinomial distribution is a generalisation of a binomial distribution where the number of possible outcomes are extended from two to $K \geq 2$. Here it assumed that each outcome can only have a single observation, that is, there is only peak per cluster. (See [6] for further details on the multinomial distribution.)

Each vector $\boldsymbol{\pi_n}$ is assumed to follow a Dirichlet distribution with parameter vector $\boldsymbol{\alpha} = [\alpha/K, \alpha/K, ..., \alpha/K]^T$ where α is a know positive constant. (The Dirichlet distribution is the conjugate prior of the multinomial distribution, see [6] for details). The probabilities for each $\boldsymbol{\pi}$ are expressed as

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}.$$

For the distributions of the K clusters to which the peaks are to be allocated, it is assumed that mass and retention time of the data belonging to each of these can be modelled using a bivariate normal distribution. Each cluster has an individual mean vector:

$$\boldsymbol{\mu}_k = [\mu_M^k, \mu_{RT}^k]^T \text{ where } (k = 1, 2, .., K),$$

where μ_M^k and μ_{RT}^k are the respective mass and RT mean parameters. The mean parameters for the clusters are unknown and it is assumed that the uncertainty in each can be expressed as a normal distribution as:

$$\mu_M^k \sim N(\mu_0^k, \sigma_{0,M}^k)$$

and

$$\mu_{RT}^{k} \sim N(\mu_0^k, \sigma_{0,RT}^k).$$

The variance is assumed to be known from the outset and is the same for each cluster. Each cluster's covariance matrix is:

$$\Sigma = \begin{pmatrix} \sigma_M^2 & 0 \\ 0 & \sigma_{RT}^2 \end{pmatrix}$$

where σ_M^2 and σ_{RT}^2 are the mass and retention time variances respectively.

5.1.2 Peak Clustering Joint Distribution

Having set out the assumptions, the peak clustering problem can now be modelled in terms of the observed peak data $\boldsymbol{X}$, the collection of cluster indicator variables for each peak $\boldsymbol{Z}$, the cluster probability vectors $\boldsymbol{\pi}$ and the cluster means $\boldsymbol{\mu}$. The precursor mass of a given peak can

The collective uncertainty in these variables can is described by their joint distribution:

$$p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}) p(\boldsymbol{Z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\boldsymbol{\alpha}) p(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \Sigma_0) \tag{5.2}$$

where

$$p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}) = \prod_{n=1}^{N} \prod_{k=1}^{K} N(\boldsymbol{X}_n^k|\boldsymbol{\mu_k}, \Sigma)^{z_{nk}}, \tag{5.3}$$

$$p(\boldsymbol{Z}|\boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}}, \tag{5.4}$$

$$p(\boldsymbol{\mu}|\boldsymbol{\mu}_0, \Sigma_0) = \prod_{k=1}^{K} N(\boldsymbol{\mu}_k|\boldsymbol{\mu}_0, \Sigma_0) \tag{5.5}$$

where $N(.)$ denotes a bivariate normal distribution.

Substituting 5.3, 5.4 and 5.5 into 5.2 gives

$$p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}) = \prod_{n=1}^{N} \prod_{k=1}^{K} (\pi_k N(\boldsymbol{X}_n^k|\boldsymbol{\mu_k}, \Sigma))^{z_{nk}} \prod_{k=1}^{K} N(\boldsymbol{\mu}_k|\boldsymbol{\mu}_0, \Sigma_0) p(\boldsymbol{\pi}|\boldsymbol{\alpha}). \tag{5.6}$$

Having derived the joint distribution for the cluster model, the aim is now to be able to fit this to a given set of peak data and hence establish the distribution parameters for each cluster and which peaks belong to each cluster. This will be done using both the Gibbs Sampling and Variational Bayes algorithms.

5.2 Gibbs Sampling Algorithm

The first algorithm that will be used to estimate the parameters of the cluster model described in 5.6 is the Gibbs Sampling algorithm. The overall aim of this algorithm is to draw multiple samples for each variable of interest from the joint distribution and then use these to obtain an estimate of each

parameter. This may be done by, say, taking the average or mode of the samples drawn.

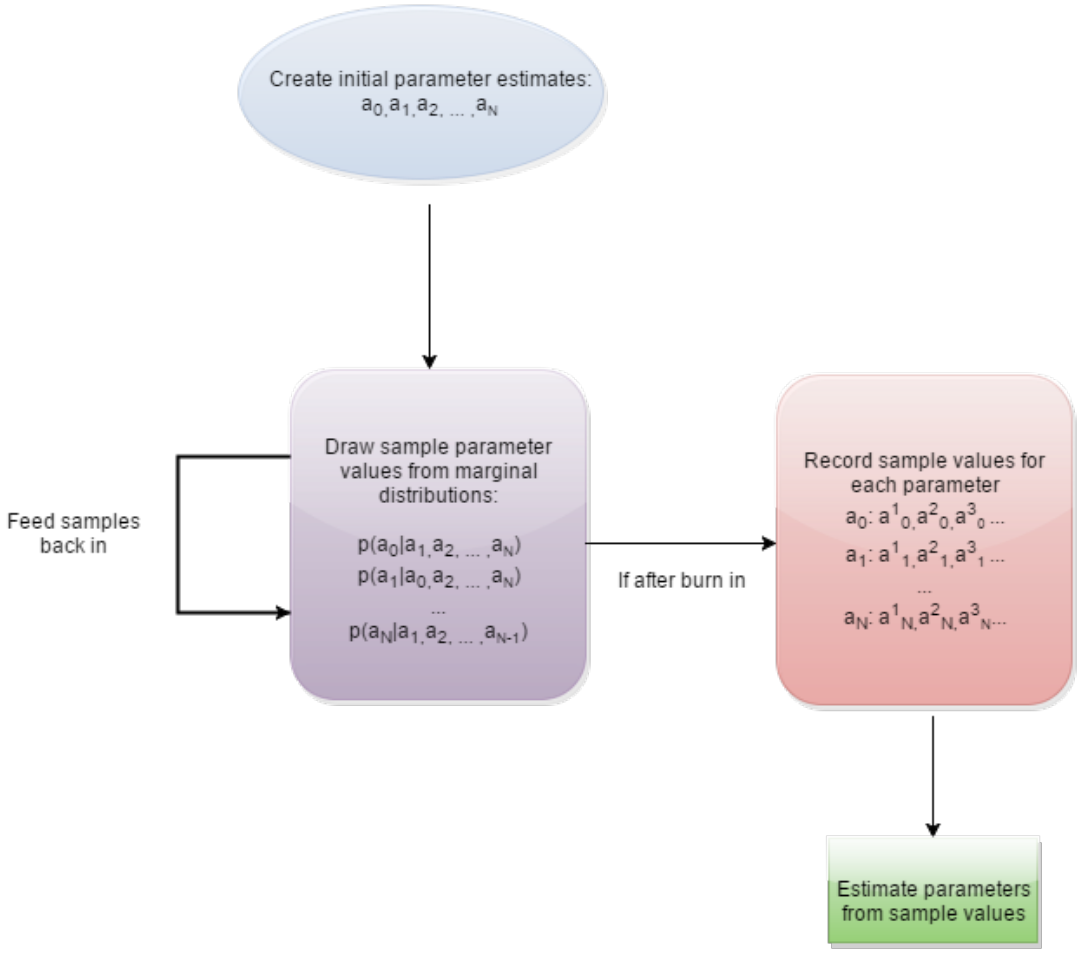
The joint distribution is approximated by first making initial estimates for each of the model parameters and then sampling from the **marginal distributions** for each model parameter conditioned on all other parameters. This is first repeated for each marginal distribution over an initial number of iterations called the **burn-in period**. The burn-in period is the number of iterations required for the distribution of the samples being drawn to converge to the joint probability distribution under consideration. After the burn-in period, the model can then be run for a further period during which each of the samples will now be drawn from the required joint distribution and the results can now be recorded.

As an illustration, consider a general model with data vector $\boldsymbol{X}$ and parameter vector $\boldsymbol{a}$. The joint distribution to be sampled from is $p(\boldsymbol{X}, \boldsymbol{a})$.

Let a_k^i denote the value of the k^{th} parameter after the i^{th} iteration. At iteration i, samples are drawn from the distributions. (See [6] for further details.)

$$p(a_k|a_1^i, a_2^i, ..., a_{i-1}^i, a_{i+1}^{i-1}, ..., a_k^{i-1}, \boldsymbol{X}) \text{ for } 0 \leq k \leq K.$$

As an overview of the process:



It should be noted that at each sampling step in the algorithm the most recently sampled parameter values are used in the marginal distributions. That is, once each parameter is sampled, this new value immediately replaces the old value held for that parameter for all subsequent iterations (including the current iteration).

5.2.1 Gibbs Sampler Derivation for the Peak Clustering Model

The marginal distributions required for the Gibbs sampler for the peak clustering model are as follows:

$$p(z_{nk} = 1|\boldsymbol{X}, \boldsymbol{\mu}, \boldsymbol{z}_{-n}, \boldsymbol{\pi}) \propto \pi_k N(\boldsymbol{x}_n^k|\boldsymbol{\mu_k}, \Sigma) \tag{5.7}$$

$$p(\boldsymbol{\mu_k}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\mu}_{-k}, \boldsymbol{\pi}) \propto \prod_{n=1}^{N} (p(\boldsymbol{x_n}^k|\boldsymbol{\mu_k}, \Sigma))^{z_{nk}} * p(\boldsymbol{\mu_k}|\boldsymbol{\mu_0}, \Sigma_0) \tag{5.8}$$

$$\sim N(\boldsymbol{x}_n^k|\tilde{\mu}_k, \tilde{\Sigma}) \tag{5.9}$$

$$p(\boldsymbol{\pi}_n|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\pi}_{-n}) \propto \prod_{k=1}^{K} \pi_k^{\sum_{n=1}^{N} z_{nk}} * \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \tag{5.10}$$

$$= \prod_{k=1}^{K} \pi_k^{\alpha_k + \sum_{n=1}^{N} z_{nk} - 1} \tag{5.11}$$

$$\sim Dirichlet(\hat{\boldsymbol{\alpha}}) \text{ (where } \hat{\alpha}_k = \textstyle\sum_{n=1}^{N} z_{nk},\ k = 1, ..., K). \tag{5.12}$$

The relationship (4.8) can be shown by first multiplying out (4.7) as follows:

$$p(\boldsymbol{\mu_k}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\mu}_{-k}, \boldsymbol{\pi}) \propto \prod_{j} (p(\boldsymbol{x}_n^k|\boldsymbol{\mu_k}, \Sigma))^{z_{jk}} * p(\boldsymbol{\mu_k}|\boldsymbol{\mu_0}, \Sigma_0)$$

$$\propto \prod_{j} \exp\left(\frac{-z_{jk}(x_{M,j}^k - \mu_M^k)^2}{2\sigma_M^2}\right) * \prod_{n=1}^{N} \exp\left(\frac{-z_{nk}(x_{RT}^j - \mu_{RT}^k)^2}{2\sigma_{RT}^2}\right)$$

Now by expanding out the above expressions and equating coefficients with that of a standard normal pdf:

$$\frac{1}{\sqrt{2\pi}\tilde{\sigma}^2} \exp\left(\frac{-(x - \tilde{\mu})^2}{2\tilde{\sigma}^2}\right),$$

the marginal distribution of $\boldsymbol{\mu_k}$ can be written as a bivariate normal distribution with mean and variance parameters $\tilde{\boldsymbol{\mu}}_k$ and $\tilde{\Sigma}_k$ respectively. The individual mass and RT parameters are:

$$\tilde{\sigma}_{k,M}^2 = \frac{\sigma_M^2 \sigma_{0,M}^2}{\sigma_M^2 + \sigma_{0,M}^2 \sum_j z_{jk} x_{M,j}^k}, \tilde{\mu}_{k,M} = \tilde{\sigma}_M^2 \left(\frac{\mu_{0,M}}{\sigma_{0,M}^2} + \frac{\sum_j z_{jk} x_{M,j}^k}{\sigma_M^2}\right) \tag{5.13}$$

and

$$\tilde{\sigma}_{k,RT}^2 = \frac{\sigma_{RT}^2 \sigma_{0,RT}^2}{\sigma_{RT}^2 + \sigma_{0,RT}^2 \sum_j z_{jk} x_{RT,j}}, \tilde{\mu}_{k,RT} = \tilde{\sigma}_{RT}^2 \left(\frac{\mu_{0,RT}}{\sigma_{0,RT}^2} + \frac{\sum_j z_{jk} x_{RT,j}}{\sigma_{RT}^2}\right). \tag{5.14}$$

Having derived the marginal distributions, it is now possible to use these to carry out the Gibbs sampling process using these as discussed in the previous section. However, it is possible to simplify these further.

The above marginal distribution for z_{nk} in can be simplified 5.7 by integrating out both the π and and μ terms. As shown in A.1.1 of teh Appendix, using the fact that the pdf of a Dirichlet distribution with parameter vector $\boldsymbol{\alpha}$ is of the form

$$\frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k x_k^{\alpha_k - 1} \text{ (where } \Gamma(.) \text{ is the gamma function),}$$

the π term may be integrated out and re-expressed as:

$$\frac{(\alpha_k + c_k^{-n})}{\sum_j (\alpha_j + c_j^{-n})}. \tag{5.15}$$

The c_j^{-n} used in the above expression gives the the number of peaks (excluding peak n) which are in cluster j and is known as the **cluster count** and represents an important step in the Gibbs sampling algorithm.

The $\boldsymbol{\mu_k}$ term can also be replaced in a similar way by considering its posterior distribution conditioned on all values of $\boldsymbol{X} = \boldsymbol{x}$ excluding the data point under consideration. This can be derived by following the same steps as for the derivation of (4.12). After following this through, the individual mass and RT parameters may be shown to be:

$$\hat{\sigma}^2_{k,M} = \frac{\sigma^2_M \sigma^2_{0,M}}{\sigma^2_M + \sigma^2_{0,M} c_k^{-n}}, \hat{\mu}_{k,M} = \hat{\sigma}^2_M \left(\frac{\mu_{0,M}}{\sigma^2_{0,M}} + \frac{\sum_{j \neq n} z_{jk} x^k_{M,j}}{\sigma^2_M} \right) \tag{5.16}$$

and

$$\hat{\sigma}^2_{k,RT} = \frac{\sigma^2_{RT} \sigma^2_{0,RT}}{\sigma^2_{RT} + \sigma^2_{0,RT} c_k^{-n}}, \hat{\mu}_{k,RT} = \hat{\sigma}^2_{RT} \left(\frac{\mu_{0,RT}}{\sigma^2_{0,RT}} + \frac{\sum_{j \neq n} z_{jk} x_{RT,j}}{\sigma^2_{RT}} \right). \tag{5.17}$$

Using 5.16 and 5.17, the μ term in 5.7 may now be removed by conditioning $\boldsymbol{X}_n$ on all other values of $\boldsymbol{X}$ and then using then applying 5.8 along with properties of normally distributed random variables as follows:

$$\begin{aligned} \boldsymbol{X}_n | \boldsymbol{X}_{-\boldsymbol{n}} &= ((\boldsymbol{X} - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k) | \boldsymbol{X}_{-\boldsymbol{n}} && (5.18) \\ &= [N(\hat{\mu}^k_M, \sigma^k_M + \hat{\sigma}^k_M), N(\hat{\mu}^k_{RT}, \sigma^k_{RT} + \hat{\sigma}^k_{RT})]^T && (5.19) \\ &= N(\hat{\boldsymbol{\mu}}_k, \Sigma + \hat{\Sigma}_k). && (5.20) \end{aligned}$$

Step (4.25) follows from the result that if $X \sim N(a, b^2)$ and $Y \sim N(c, d^2)$ then $X + Y \sim N(a + b, c^2 + d^2)$. (See [5] for details.) Hence, 4.6 can now be written as:

$$p(z_{nk} = 1 | \boldsymbol{X}, \boldsymbol{\mu}, \boldsymbol{z}_{-n}, \boldsymbol{\pi}) = \frac{\alpha_k + c_k^{-n}}{\sum_{j=1}^K (\alpha_j + c_j^{-n})} * N(\boldsymbol{x}_n | \hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k). \tag{5.21}$$

Using 5.21 greatly reduces the number steps required in the Gibbs sampler since it is no longer necessary to sample from the marginal distributions for $\boldsymbol{\mu}_k$ and $\boldsymbol{\pi}_k$ (4.8 and 4.11). As an overview, at each iteration of the Gibbs sampling algorithm, first the cluster counts c_k^{-n} are updated and these are then used to calculate the terms in 5.21 in order to obtain the probabilities $p(z_{nk} = 1 | ...)$. These can then be used to draw a sample of $\boldsymbol{z}_n$ from its corresponding multinomial distribution.

5.2.2 Peak Cluster Model Gibbs Sampling Algorithm

Based on 5.21, the full Gibbs sampling algorithm is as follows:

1. Initialise the z_{nk} and c_k^{-n} for each $n = 1, ..., N$ and $k = 1, ..., K$ with initial estimates.
2. For each iteration i
   2.1. For each $n \leq N$:
      2.1.1. For each $k \leq K$:
         2.1.1.1. Remove z_{nk} from c_k^{-n}
         2.1.1.2. Calculate each probability $p_{nk} := p(z_{nk} = 1|...)$ using 5.21
      2.1.2. Sample $\boldsymbol{z}_n$ from $Multinomial(p_{n1}, p_{n2}, ..., p_{nK})$
      2.1.3. Update the c_k^{-n} using the new values of z_{nk} for each k
3. While i less that total number of iterations, repeat step 2
4. Terminate

5.3 Variational Bayes

The second inference method considered in this dissertation is the Variational Bayes algorithm. The aim of this method is to fit 5.6 to a given data set by first approximating it by a function $Q(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}) = Q_z(\boldsymbol{Z})Q_\pi(\boldsymbol{\pi})Q_\mu(\boldsymbol{\mu})$. In this sense, it is an approximation to the Gibbs sampler and will be used to provide a second independent implementation of the clustering model. In general, the Variational Bayes also offers faster convergence that the Gibbs sampler.

The motivation for this method comes from maximising the log-likelihood function for the model. By taking natural log of 5.6, this can be expressed as:

$$\ln(p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\pi})) \propto \sum_n \sum_k z_{nk}[\ln(\pi_k) + \ln(N(\boldsymbol{x}_n^k|\boldsymbol{\mu}_k, \Sigma))] + \sum_k \ln(p(\boldsymbol{\mu}_k|\boldsymbol{\mu}_0, \Sigma_0)) + \ln(p(\boldsymbol{\pi}|\boldsymbol{\alpha})). \tag{5.22}$$

As shown in section A.1.2 of the Appendix, a lower bound, on $\ln p(\boldsymbol{x})$ can be derived in terms of an arbitrary distribution $Q(\boldsymbol{\theta})$ and the **Kullback-Leibler** (KL) divergence between $Q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\boldsymbol{x})$, where the model parameters $\mathbf{z}, \boldsymbol{\mu}$ and $\boldsymbol{\pi}$ into single vector $\boldsymbol{\theta}$. The KL divergence measures the similarity between two distributions and takes the value zero if the two distributions are identical and is negative otherwise. The aim is to choose Q so as to maximise the KL bound by varying and hence obtain an approximation to the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$. (See [7] for details.)

5.3.1 Derivation of Variational Bayes Algorithm for Peak Clustering

The aim is choose a function of the peak clustering model parameters, $Q(\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\pi})$, which minimises the KL bound and therefore approximates the model's posterior distribution. In this dissertation, $Q(.)$ will be taken to be of the form $Q_\pi(\boldsymbol{\pi})Q_z(\mathbf{z})Q_\mu(\boldsymbol{\mu})$. It should be noted, however, that this introduces an independence assumption between the model parameters which is unlikely to be

completely accurate. This form will however offer a close enough approximation and will make the algorithm more computationally straight-forward to carry out.

As stated in [7], it can be shown that the $Q_i(.)$ $(i = z, \pi, \mu)$ which minimise the KL divergence and give the closest approximation to the posterior distribution are of the form:

$$Q_i(i) \propto \exp\{\ \mathbf{E}_{Q_j(j)Q_k(k)}[\ln(p(\boldsymbol{X}, \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\pi}))]\}\ (\ i, j, k \in \{\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\pi}\},\ i \neq j, k) \tag{5.23}$$

Now using 5.23 and 5.22 it is now possible to derive each of $Q_\pi(\boldsymbol{\pi})Q_z(\mathbf{z})Q_\mu(\boldsymbol{\mu})$ in turn.

For Q_π, the expression becomes:

$$\begin{aligned} Q_\pi(\boldsymbol{\pi}) &\propto \exp\{\mathbf{E}_{Q_z(z)Q_\mu(\mu)}[\sum_n \sum_k z_{nk}(\ln(\pi_k)) + \ln(p(\boldsymbol{\pi}|\boldsymbol{\alpha})]\} \\ &= \exp\{\sum_n \sum_k \langle z_{nk}\rangle \ln(\pi_k))\} * p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \\ &= \prod_k \pi_k^{\alpha_k + \sum_N \langle z_{nk}\rangle - 1} \end{aligned}$$

Hence $Q_\pi(\boldsymbol{\pi})$ is a Dirichlet distribution with parameter vector $\tilde{\boldsymbol{\alpha}} = [\tilde{\alpha_1}, \tilde{\alpha_2}, ..., \tilde{\alpha_K}]^T$ where each $\tilde{\alpha_k}$ is of the form:

$$\tilde{\alpha_k} = \alpha_k + \sum_n \langle z_{nk}\rangle. \tag{5.24}$$

Similarly, for Q_μ:

$$\begin{aligned} Q_\mu(\boldsymbol{\mu_k}) &\propto \exp\{\mathbf{E}_{Q_z(z)Q_\pi(\pi)}[\sum_n z_{nk} \ln(N(\boldsymbol{x}_n^k|\boldsymbol{\mu}_k, \Sigma)) + \ln(p(\boldsymbol{\mu}_k|\boldsymbol{\mu}_0, \Sigma_0))]\} \\ &= \prod_n N(\boldsymbol{x}_n^k|\boldsymbol{\mu}_k, \Sigma)^{\langle z_{nk}\rangle} * p(\boldsymbol{\mu}_k|\boldsymbol{\mu}_0, \Sigma_0) \end{aligned}$$

Using the same method as in the previous section for the Gibbs sampler of equating coefficients with the standard form of the pdf of a normal distribution, it can be shown that $Q_\mu(\boldsymbol{\mu_k})$ can be written as a bivariate normal distribution with mean and variance parameters $\tilde{\boldsymbol{\mu}}_{\boldsymbol{k}}$ and $\tilde{\Sigma}_k$ respectively. The individual mass and RT parameters are:

$$\tilde{\sigma}^2_{k,M} = \frac{\sigma^2_M \sigma^2_{0,M}}{\sigma^2_M + \sigma^2_{0,M} \sum_j \langle z_{jk}\rangle x^k_{M,j}},\ \tilde{\mu}_{k,M} = \tilde{\sigma}^2_M \left(\frac{\mu_{0,M}}{\sigma^2_{0,M}} + \frac{\sum_j \langle z_{jk}\rangle x^k_{M,j}}{\sigma^2_M} \right) \tag{5.25}$$

and

$$\tilde{\sigma}^2_{k,RT} = \frac{\sigma^2_{RT} \sigma^2_{0,RT}}{\sigma^2_{RT} + \sigma^2_{0,RT} \sum_j \langle z_{jk}\rangle x_{RT,j}},\ \tilde{\mu}_{k,RT} = \tilde{\sigma}^2_{RT} \left(\frac{\mu_{0,RT}}{\sigma^2_{0,RT}} + \frac{\sum_j \langle z_{jk}\rangle x_{RT,j}}{\sigma^2_{RT}} \right). \tag{5.26}$$

Lastly, for Q_z:

$$\begin{aligned} Q_z(\mathbf{z}) &\propto \exp\{\mathbf{E}_{Q_\pi(\pi)Q_\mu(\mu)}[\sum_n \sum_k z_{nk}(\ln(\pi_k) + \ln(N(\boldsymbol{x}_n^k|\boldsymbol{\mu}_k, \Sigma)))]\} \\ &= \exp\{\sum_n \sum_k z_{nk}(\langle \ln(\pi_k)\rangle + \langle \ln(N(\boldsymbol{x}_n^k|\boldsymbol{\mu}_k, \Sigma))\rangle)\} \end{aligned}$$

As shown in A.1.3, this can be rearrange to show that $Q(\boldsymbol{z}_n)$ follows a multinomial distribution with parameters $\gamma_{nk}/\sum_j \gamma_{nj}$ where

$$\ln(\gamma_{nk}) = \psi(\tilde{\alpha_k}) - \psi(\sum_j \tilde{\alpha_j}) - \frac{(x_n^M)^2 - 2x_n^M\tilde{\mu}_{M,k} + \tilde{\mu}_{M,k}^2 + \tilde{\sigma}_{M,k}^2}{2\sigma_M^2} - \frac{(x_n^{RT})^2 - 2x_n^{RT}\tilde{\mu}_{RT,k} + \tilde{\mu}_{RT,k}^2 + \tilde{\sigma}_{RT,k}^2}{2\sigma_{RT}^2} - \ln(2\pi\sigma_M\sigma_{RT}).$$

(In the above $\psi(.)$ is the digamma function, see [9] for details.) The expected value for each is $z_n k$ is then:

$$\langle z_{nk}\rangle = \frac{\gamma_{nk}}{\sum_j \gamma_{nj}} \ (n = 1, 2, ..., N \text{ and } k = 1, 2, ..., K). \tag{5.27}$$

It should be noted that the $\langle z_{nk}\rangle$ are dependent on the parameter values for both $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ and vice versa. Hence, the algorithm involves first calculating updated values of the $\tilde{\alpha}_k$, μ_k^M and μ_k^{RT} terms and the using these to update the $\langle z_{nk}\rangle$. Having done this, the new $\langle z_{nk}\rangle$ can now be used to update the $\tilde{\alpha}_k$, μ_k^M and μ_k^{RT} terms. This process can then be repeated until the parameters each converge.

5.3.2 Variational Bayes Peak Clustering Algorithm

Having derived expressions for all of the key terms of the variational Bayes algorithm, the key steps are as follows:

1. Estimate initial values for the $\tilde{\alpha}_k$, μ_k^M and μ_k^{RT} terms using 5.24, 5.25 and 5.26 respectively.
2. Use the current values of $\tilde{\alpha}_k$, μ_k^M and μ_k^{RT} to calculate the $\langle z_{nk}\rangle$ using equation A.11.
3. Use the $\langle z_{nk}\rangle$ to calculate updated values for $\tilde{\alpha}_k$, μ_k^M and μ_k^{RT}.
4. If not converged yet repeat steps 2. and 3.
5. Terminate

5.3.3 Clustering

As stated previously, the peaks are to be clustered using their precursor mass, retention time and intensity values. A transformation is available which will be used to obtain the precursor masses from the observed mass to charge ratios. However, while this is computationally straight-forward to implement, one issue with this is that its parameters depend on the particular adduct to which the peak being considered corresponds. This is not known, and there will therefore be a potentially very large number of different clusters to which each peak could belong to depending on what transformation is being used to obtain its precursor mass. Fortunately, this number can be greatly reduced by using the restriction that the M+H adduct **must be present** in each cluster. The M+H adduct is by far the most frequently observed and will always be observed for each molecule. Hence, it would not make sense for a cluster to not contain the M+H adduct.

Making use of this restriction, an initial list of N potential clusters with initial cluster means

$\boldsymbol{\mu}_0^k = [\mu_{0,M}^k, \mu_{0,RT}^k]$ can be constructed. This can be done by applying the M+H adduct transform to the observed mass to charge ratio of the k^{th} peak and setting this equal to $\mu_{0,M}^k$. The value of $\mu_{0,RT}^k$ will also be set equal to the k^{th} peak's retention time. A list of possible clusters to which each peak can now be obtained by considering each peak and cluster allocation in turn.

Another assumption which further reduces the number of clusters to which a peak can belong is that the M+H peak in each cluster must also have the largest intensity value. This assumption means that each peak can only be allocated to a given cluster (that is not its own M+H cluster) if its intensity value is less than that of the M+H peak. Further restrictions are that a peak's precursor mass and retention time must be within fixed intervals $[\mu_{0,M}^k - \delta_k^M, \mu_{0,M}^k - \delta_k^M]$ and $[\mu_{0,RT}^k - \delta_k^{RT}, \mu_{0,RT}^k - \delta_k^{RT}]$. In carrying this the cluster algorithms in practice, is assumed that a peak's precursor mass must be within 5 parts per million of the $\mu_{0,M}^k$, that is if:

$$\frac{\text{precursor mass} - \mu_{0,M}^k}{\mu_{0,M}^k} \leq 5 \times 10^{-6}, \tag{5.28}$$

and its retention time must be within 10 seconds of $\mu_{0,RT}^k$.

5.3.4 Possible Cluster Identification Algorithm

Based on these assumptions, the following algorithm for identifying the possible clusters to which peak can belong has been constructed:

1. For each of the N possible peaks:
    1.1. For each peak, add each cluster it belongs to under the M+H transform to its list of possible clusters.
    1.2. For each cluster where the peak is not the corresponding M+H adduct peak:
        1.2.1. Compare the peak's intensity to that of the cluster
        1.2.2. If it is greater then go back to 1.2 and move to the next cluster
        1.2.3. Compare the peak's retention time to that of the cluster
        1.2.4. If it is outside the cluster's retention time window go back to 1.2 and move to the next cluster
        1.2.5. For each precursor mass transform except the M+H transform:
            1.2.5.1. Apply the transform to the peak's mass to charge ratio
            1.2.5.2. If the transformed mass is out with the cluster's mass acceptable mass window go back to 1.2.5 and move to the next transform
            1.2.5.3. Else, add the cluster to the list of the peak's possible cluster and record the corresponding transform
2. Terminate

Applying the above algorithm will greatly reduce the number of possible clusters to which it will belong. For each peak, the corresponding cluster constructed by applying the M+H transform to its observed mass to charge value will always be in its list of possible clusters. There may also be a small number of there clusters to which a given peak may also belong if it meets each of the steps

in part 1.1 of the above algorithm. However, there will be many peaks which do not have any other possible clusters and it is now known now known to which cluster they must belong without having to apply any clustering algorithm. This also means the matrix of the $p(z_{nk} = 1|...)$ probabilities will be sparse and that a much smaller subset of peak and cluster combinations need now only be considered when implementing either the Gibbs sampling or variational Bayes algorithms. This can be used to improve the performance of each algorithm's implementation.

5.3.5 Identifying Cluster Masses

The motivation for clustering the observed peak data is to be able to match each peak to a molecule. This will be done by assigning a mass value to each cluster and then comparing this to a database of mass values for the sample's known constituent molecules. Hence, it is necessary to assign a mass to each cluster following peak allocation.

For variational Bayes, a clear choice is to use $\tilde{\mu}_{k,M}$, the expected value of $Q_\mu(\boldsymbol{\mu}_k)$ as shown in 5.25. For Gibbs sampling there are a few possible alternatives. For example, the precursor mass of the M+H adduct could be used. However, here the posterior distribution of the mass mean, $\tilde{\mu}_{k,M}$ will be used. This value is calculated at for each cluster at each iteration. On the final iteration, the peak's cluster is set to be the most probable cluster (i.e. the cluster to which the peak has been allocated most often over all of the iterations). The cluster mass is set to be the average over the posterior mass values recorded for this cluster over all of the iterations.

5.4 Implementation

In implementing the algorithms described above, it was necessary to think carefully about which data structures should be used. As described in Chapter 4, an object orientated approach was taken with the main classes being used in developing the cluster model being:

- Peak: Used to represent each peak in the input data
- Cluster: Used to represent each cluster
- PossibleCluster: Used to represent clusters to which a peak can possibly belong to. This class is used to connect Peak objects with Cluster objects.
- Transform: Used to represent a transform which may be applied to a peak in order to calculate its precursor mass

The first step in the implementation was to read in the data for each of the input text files and create all of the Peak and Transform objects and then storing the Peak objects in a list (**peaks**) and the transforms in a dictionary (**transforms**). The transforms dictionary takes a string representing the corresponding adduct name, hence the M+H transform object can be extracted it by passing it the string "M+H". The Cluster objects can now be created from each of the Peak objects in turn, using applying the M+H transform to obtain the values for each Cluster's mass mean, and these were then stored in a list called **clusters**.

Algorithm 5.3.4, for identifying each peak's possible clusters, could then be implemented using the lists **peaks** and **clusters** along with the dictionary **transforms**. This was done by looping over **peaks** and checking this against each element of **clusters** as described in 5.3.4. Once it was identified that a Peak could belong to a Cluster under a particular Transform, a PossibleCluster object was created and added to Peak object's list of possible clusters (a property of each Peak object of type list called **possible_clusters**).

Having identified all of the PossibleCluster objects for each Peak, the Peak objects were then separated into those with only one PossibleCluster and those with more than one PossibleCluster. This was done by checking whether the length of their **possible_clusters** list was equal to 1 or greater than 1 and then allocating them to one of two further lists, **only_one_cluster** and **more_than_one_cluster**. As discussed above, if a Peak only has one PossibleCluster then there is no need to go through the steps in the either in clustering algorithms for it. Hence, both the implementations of the Gibbs sampling and variational Bayes algorithms could be made more efficient by focussing only on the Peak objects in **more_than_one_cluster**.

The next stage in the implementation was to construct the Gibbs sampling and variational Bayes algorithms. As noted above, it was important to take advantage of the fact that many peaks can only belong to one possible cluster and, even for those with more than possible cluster, the list of possible clusters will be very sparse in each case.

In a first attempt at implementing these algorithms, an array was used to store the values of the z_{nk} and $\langle z_{nk} \rangle$ pramaters for the Gibbs sampler and variational Bayes methods respectively. However, this was very memory inefficient and led to the algorithms running very slowly. This issue is addressed by introducing the PossibleCluster class and adding the list **possible_clusters** as a property of the Peak class. With this structure, it is possible to only loop through each peak and its possible clusters rather than going through every peak/cluster combination in ether clustering algorithm.

The implementations of the Gibbs sampling and variational Bayes algorithm are very similar in their overall structure. They both begin by allocating peaks with only one possible cluster to their single cluster and then applying the steps set out in 5.2.2 and 5.3.2 to peaks with more than one cluster and their corresponding lists of possible clusters.

The key step in each algorithm involves looping over each Peak with more than on cluster and its corresponding PossibleCluster objects. The probabilities needed to determine the cluster allocation at each iteration are then stored in a dictionary. This dictionary takes references to each PossibleCluster object as its keys and the corresponding probabilities that each Peak belongs to each PossibleCluster as its values. Creating this dictionary allows the probabilities for each PossibleCluster object to be stored without holding a large number of zero entries (as would be the case if a matrix was used).

5.5 Testing

The main test for the clustering algorithms developed was to compare the results produced with those from an independent implementation of the Gibbs sampler which had been developed. This implementation had been run on a test data set of peak data and the results of its cluster allocations

were available. In order to test the implementations of the Gibbs sampler and variational Bayes algorithm, a program was written to compare the cluster number assigned to each peak in both implementations and then computes the an overall percentage of the total number of peaks that the two files agree on for the test file as a whole. The results for the Gibbs sampler and Variational Bayes algorithms are shown in B.2.1 of Appendix B. As can be seen, the percentages matches are approximately 98% and 95% for the Gibbs and variational Bayes algorithms. This suggests that the implementations offer a strong level of agreement with this implementation with differences likely to be mainly attributable to stochastic variation in the Gibbs algorithm.

Assertions were also added to the probabilities calculated in the two clustering algorithms for the probabilities calculated in each. These were added to verify that none of these values are less than zero. Both algorithms run without throwing an exception relating to these assertions.

Chapter 6

Evaluation

6.1 Overview

This section evaluates the cluster model described in the previous section by assessing the output it produces from running raw peak data produced from a mass spectrometer.

Peak data has been provided for two standards, that is, chemical solutions for which the constituent molecules are known, and this has been run through the clustering model. The peak data for each standard is spread across multiple files with each file corresponding to a individual run through the mass spectrometer. Hence, each standard has been processed through the mass spectrometer several times and each file represents an independent sample of the peak data produced from the mass spectrometry process.

The process implemented by the peak clustering software tool that has been developed has two main stages. The first stage takes the raw peak data and, using either of the Gibbs sampling or Variational Bayes algorithms, clusters the peaks using their precursor mass, retention time and intensity values. The output from this step is a list containing each peak along with its allocated cluster and the associated adduct transform which places it in its allocated cluster. Also produced is a list of each cluster along with its associated mass, retention time and list of adducts. Each cluster corresponds to an individual molecule.

The second stage in the process is now to match each cluster to a molecule. This is done by matching the cluster masses output from the first stage to a list of each standard's constituent molecules and their known mass. This comparison is done by calculating the percentage difference between cluster mass and known molecule. A match has been found if this difference is within 5 parts per million (PPM). This is if

$$\frac{\text{Cluster Mass} - \text{Known Mass}}{\text{Known Mass}} \leq 5 \times 10^{-6}. \tag{6.1}$$

It should be noted that not every cluster will be allocated to a molecule under this method. This is in part because of noise in the raw peak data. As discussed previously, the mass spectrometry process is highly sensitive to the experimental conditions in which it is carried out. Some of the peaks produced may, for example, correspond to impurities present in the sample or some other factor affecting with the experiment conditions. Hence, it is not unusual for a large number of the

clusters to be matched to a molecule.

Having matched clusters to molecules, it is now possible to analyse the adduct patterns for each molecule. A question of particular interest is whether a given adduct pattern has any predictive capability for molecule identification, particularly between **isomers**. Two molecules are isotopes if they have the same constituent molecules (and hence the same mass) but have a different chemical structure.

The output from each of these two stages in terms of their ability to provide meaningful insight into peak data produced by a mass spectrometer.

6.2 Evaluation of the Peak Clustering Process

The first part of the evaluation will focus on assessing the output from the initial peak clustering process.

6.2.1 Comparison of Peak Clustering Results for the Gibbs Sampling and Variational Bayes Algorithms

The raw peak data has been run using both the Gibbs Sampling and variational Bayes algorithms. Having done this, it is now possible to compare the output from the two algorithms. As the variational Bayes algorithm is essentially an approximation of the Gibbs sampler, it would be expected that the results of the two algorithms should be very similar if both models have been correctly implemented.

To test this, a collection of five standard 1 and five standard 2 files have been run through both algorithms. A program has been written to compare the cluster number assigned to each peak using each of the two methods. For each file, it counts each time a peak has been allocated to the same cluster using both algorithms and then computes an overall percentage of the total number of peaks that the two methods agree on for the file as a whole. The results are shown in B.2.2 of Appendix B.

As shown by these results, the two methods agree on 95% of the peaks in each file, which indicates a strong level of agreement. Some difference between methods is expected due to stochastic variation in the Gibbs sampler and also due to fact that variational Bayes is an approximation.

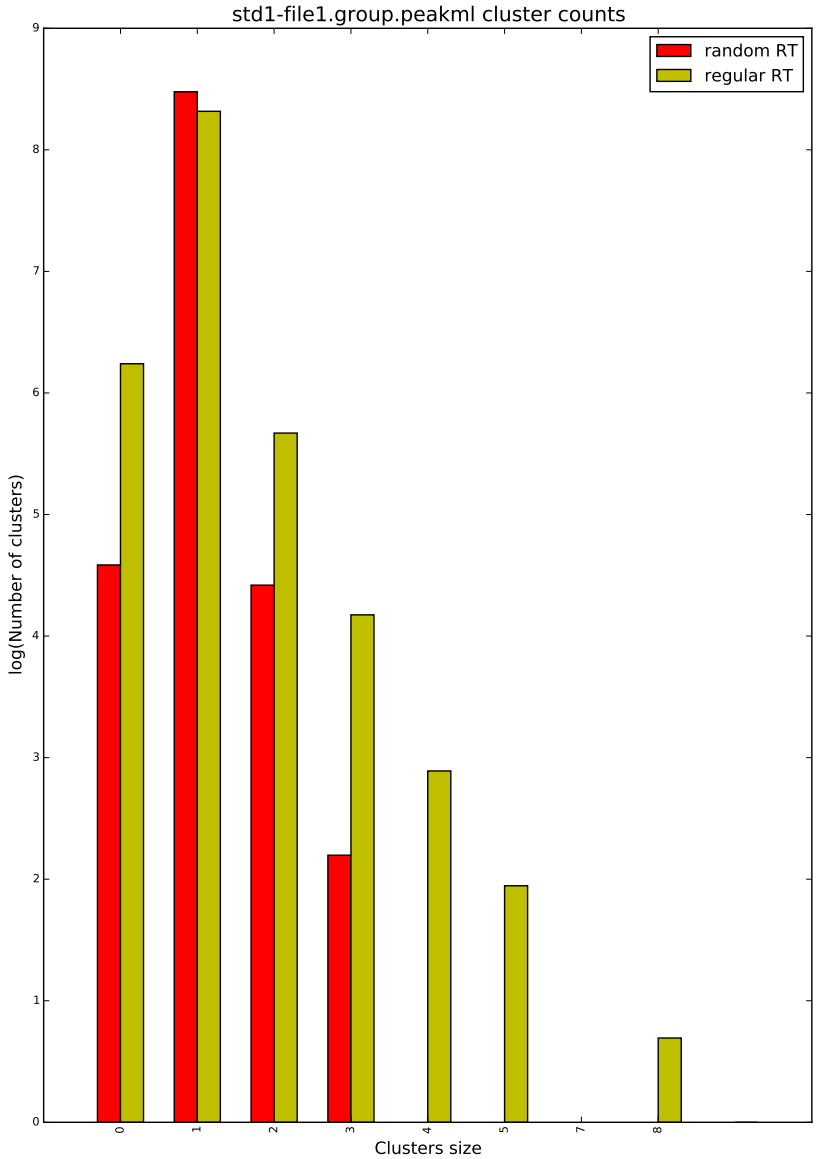
The fact that both methods give very similar output helps to cross-validate their output. Both models have different implementations and the fact that they agree provides a string indication that they have been implemented correctly. In light of the fact that their outputs are very similar, from this point on the evaluation will focus only on the output from using the Gibbs sampling algorithm.

6.2.2 Identifying Presence of Underlying structure in the Data

In order to further assess the effectiveness of the Gibbs sampling algorithm, a further test was carried out to check whether the clusters it identifies are representative of an underlying structure in

the peak data (due to peaks belonging to the same molecule) or whether they were due to chance or an error in the algorithm's implementation. If there was no structure in the data then it would be expected that the number of peaks clustered together would be significantly lower than if such a structure was present. One way to remove any structure in the peak data is to randomise it. This randomised data can then be run through the Gibbs sampler and the output compared with the standard data.

In order to create a mix of the of the peak data, the retention time values have been randomly permuted for all of the peaks. This has been done for a number of standard 1 and standard 2 files and this data has then been run through the Gibbs sampling clustering algorithm. For each cluster obtained in the output from processing this data, a count of the number of peaks that have been allocated has been made. The same counts have been made for the clusters produced from the non-randomised data. A plot of cluster size against the natural logarithm of the number of clusters observed to be of this size was made for each file. (The natural logarithm has been taken in order aid comparison in light of the large number of peaks belonging to each file.) Shown below is a plot for the first file - the plots for the other files are similar and are shown in B.2.3 of Appendix B.



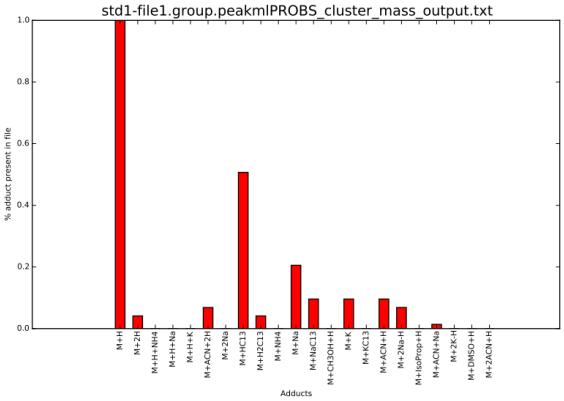
As can be seen from the above plots, the largest cluster size for the the randomised data is three compared with seven for that of the regular peak data. Also, the size of the bar plots for the regular data is larger than that of the randomised data for all cluster sizes except one. It should be again noted that this chart has been plotted on a log scale and the difference between the two bar sizes at a cluster size of one is much larger than it appear on the chart - both data sets contain the same number of clusters and this difference accounts for the excess shown in the plots for the regular data over those for the randomised data at all other cluster sizes. Also of note is the difference between the plots at a cluster size of zero. Given each peak must be allocated to exactly one cluster, a larger number of empty clusters must then correspond to a larger number of clusters with more than one peak.

Therefore the above plots indicate that randomising the data significantly reduces the number of peaks being allocated to the same cluster by the Gibbs sampling algorithm. This suggests that there is an underlying structure present in the regular peak data and that this structure is being detected by the clustering algorithm.

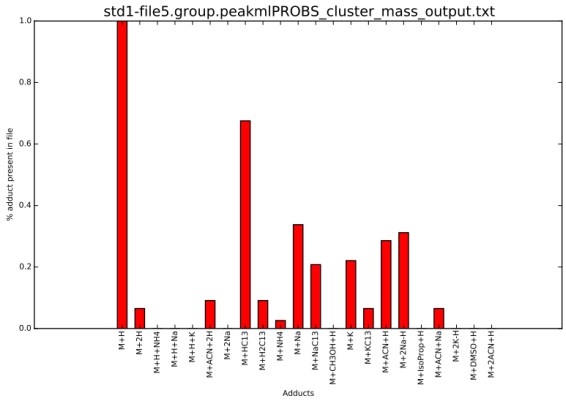
6.3 Assessing Adduct Consistency Across the Files

Having allocated peaks to clusters, the next stage in the process is to allocate clusters to molecules and then plot the adduct patterns for each molecule. Before doing so, however, the files were checked for consistency. It may have been the case that one file was corrupted due to, say, the presence of an external substance. Such an error would affect the adduct patterns observed. This would impact the ability to draw any conclusions from adduct patterns produced across all of the files. In order to check this, the frequency with which each adduct was observed was plotted for each file. Any deviation between these plots would indicate a potential issue with either the experimental set up or the clustering model used to produce the analysis. These plots are shown in B.2.3 of Appendix B. Whilst some degree of variation is expected, it can be seen from these plots that there is a strong degree of consistency across the files.

For example, shown below are plots of the counts for two separate standard 1 files:



(a) Plots of each adduct's frequency from the first file for standard 1.

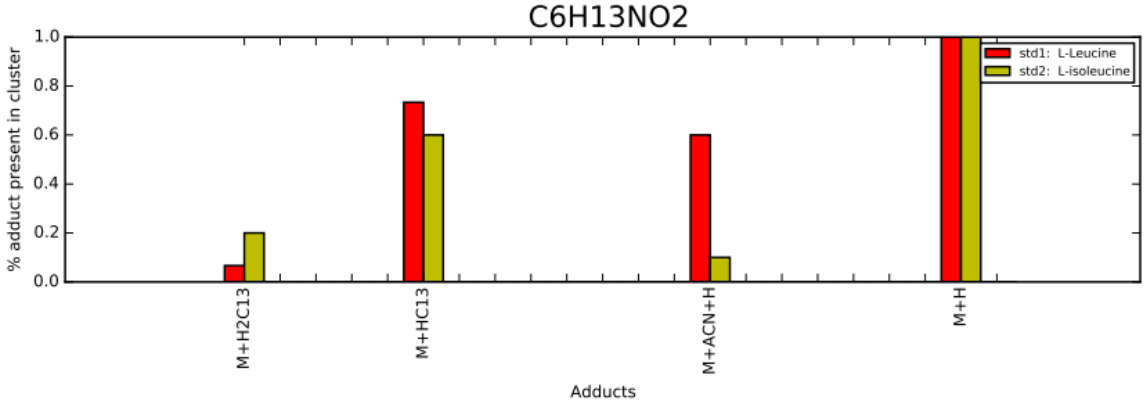


(b) Plots of each adduct's frequency from the fith file for standard 1.

6.4 Assessing Ability to Identify Isotopes

A question of much interest is whether the adduct pattern observed for a molecule has any predictive power. In particular, given only the adduct patterns for two molecules which are isotopes, is it possible correctly identify each molecule using only their adduct patterns? With a view to answering this question, various plots associated with the adduct patterns of isotopes in the two standards have been plotted. Each of the two standards contain different molecules however there are sixteen molecules in standard 1 which have a corresponding isotope in standard 2.

Plots showing the percentage frequency that each adduct is present for a given molecule have been produced and are shown in B.2.3 of Appendix B. Also, plots of the intensities each isotope's standard 1 and standard 2 molecule were also made - each point plotted has an x-value and y-value corresponding to the standard 1 and standard 2 molecule's mean intensity across all files. These are as shown in B.2.3 of Appendix B. (Blank plots shown for a particular molecule indicate that neither isomer was identified in the samples.)Each of the plots produced can now be studied in order to assess whether there are any significant difference between the adduct patterns produced for isomers. For example, shown below are the plots for leucine and isoleucine (chemcical formula C6H13NO2):



Firstly, as can be seen in the above figure the M+H adduct is always present in both isomers as expected. Also looking further at the above adduct pattern, there is a significant difference between the peaks observed for M+ACN+H, and it would appear that this adduct is much more common for L-leucine than for L-isoleucine. This type of significant difference is of interest as it may suggest that a significant presence of the M+ACN+H is an indicator that the molecule observed is L-leucine rather than L-isoleucine. The rest of the plots can also be studied in a similar manner with a view to identifying substantial differences in adduct patterns such as this.

In order to further test the potential for using adduct patterns to distinguish between isomers, a further experiment was carried out. First the peak data files were subdivided into 18 training files and 13 test files. First the probabilities of the presence of each adduct in each isomer were calculated across all of the training files. The test was then to use these probabilities to determine whether each molecule was the standard 1 or the standard 2 isomer based on the presence or absence of each adduct observed in the test data. That is, say for a given molecule in a given test data file, a binary string for each adduct i was observed:

$$\boldsymbol{b} = (b_1, b_2, ...).$$

Suppose also that the probabilities computed from the test data that each adduct is present for a standard 1 and standard 2 molecule are

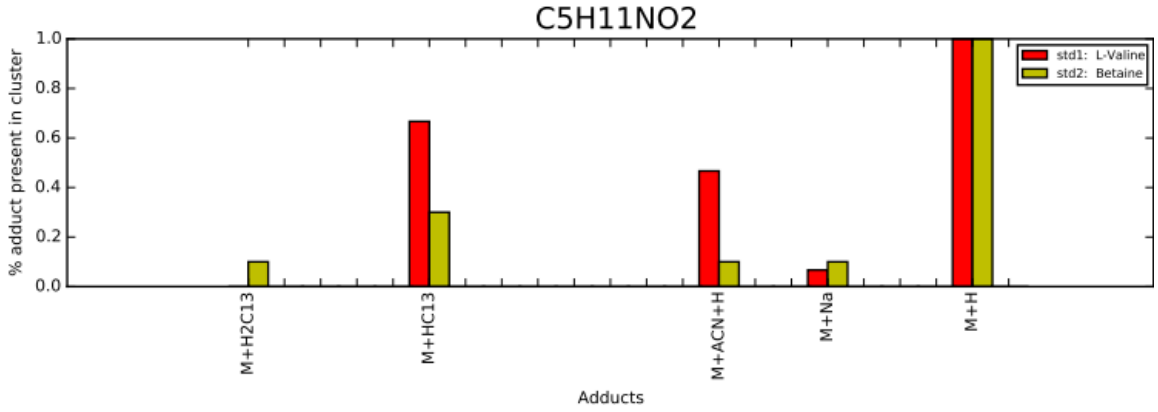
$$\boldsymbol{p}^1 = (p_{1,1}, p_{1,2}, ...) \text{ and } \boldsymbol{p}^2 = (p_{2,1}, p_{2,2}, ...)$$

respectively. Then two scores can be computed, each using $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$, as follows:

$$S1 = \sum_i p_{1,i}^{b_i}(1 - p_{1,i})^{1-b_i} \text{ and } S2 = \sum_i p_{2,i}^{b_i}(1 - p_{2,i})^{1-b_i}, \tag{6.2}$$

with $S1$ and $S2$ giving respective measures of how likely the test molecule is to be the standard 1 or standard 2 isotope (the larger score indicating which isomer the molecule is).

Running the above test across all of the test files for both L-leucine and L-isoleucine leads to leucine being correctly identified as standard 1 on 80% of the standard 1 test files and L-isoleucine being correctly identified as standard 2 on 71% of the standard 2 test files. By comparison, carrying out the same test for other standard 1 and 2 isomers such as L-Valine and Betaine (chemical formula C5H11NO2) leads to correct classification in 60% and 57% of test files respectively. Shown bellow are the plots showing the percentage of time each adduct is present for C5H11NO2:



As can be seen from the above plot, it is notable that there appears to be a larger presence of the M+HC13 and M+ACN+H adducts in L-Valine than in betaine. However, these differences are not quite as pronounced as for that of the M+HC13 peaks plotted for L-leucine and L-isoleucine however - this may provide an explanation as to why identification has been more successful for these two molecules.

The plots and test carried out go some way to suggest that the adduct patterns may be of use in identifying isomers. However, it should be noted that significant amount of further work would need to be carried out first. One issue with carrying out the above tests was that there were a significant number of molecules in the standard 2 files which were not matched to any particular cluster. Hence, the plots showing the frequency of each adduct's presence for each molecule must be handled with care since. For example, there may be cases where the standard 1 molecule was matched to a cluster across all files but standard 2 was only matched in one of its files. This would mean that the standard 2 frequencies were only based on a single observation and hence it would be difficult to draw meaningful comparisons between the isomers adduct patterns in this situation. Given extra time, it would be desirable to run the model on more data files with a view to obtaining

a more even number of cluster matches in order to be able to better compare the adduct patterns more effectively. In short, the results produced indicate that the predictive ability of adduct patterns is an area of potential interest where there is much scope for further work to be carried out.

Chapter 7

Conclusion

7.1 Current Status

The Gibbs sampling and Variational Bayes algorithms have both been implemented for the peak clustering model and both run on a number of samples of MS peak data for two standards. Having then compared the output from running these two algorithms, they were observed to that they produce very similar results. This was as expected as the variational Bayes algorithm is essentially an approximation of the Gibbs sampler. Following this comparison, it was decided to focus on the Gibbs sampler for the remainder of the analysis.

Having now implemented the clustering algorithm, the next step was to fit the clusters generated to the known constituent molecules for each standard. Having done this, the adduct patterns associated with each molecule could now be examined.

A question of particular interest was whether these adduct patterns could be used to correctly identify each molecule in a given pair of isomers. That is, given only the adduct patterns for two molecules known to belong to a particular isomer pair, can each molecule correctly be identified from this information alone? The work carried out provides an indication that this is indeed possible. However, further is required in order obtain a more definitive analysis to this question. The next section sets out some suggestions for further work that could be carried out in the future in order to make further progress towards this.

7.2 Suggestions for further work

7.2.1 Further Work for Improving the Clustering Algorithms

There is scope for some further work to be carried out with a view to improving the implementations of the Gibbs sampling and variational Bayes algorithms.

In order to further evaluate the variational Bayes algorithm, it is possible to explicitly derive the

lower bound used in its construction. This can then be calculated at for each iteration of the algorithm (see A.1.2 in Appendix A). If the algorithm has been implemented correctly, then this should increase at each iteration until it converges - indicating that the algorithm itself has converged to a solution. Plotting this will provide further confirmation that the algorithm has been implemented correctly.

For the Gibbs sampler, the burn-in period has been chosen to be 500 iterations as it is believed that this is more than sufficient for the algorithm to converge to its stationary distribution. However, this could be determined more precisely. For example, multiple Gibbs samplers with different initial cluster allocations could be initialised. Each of these could then be run for a first block of 100 iterations. Then, select one or more peak which could belong to more than one cluster (i.e. peaks which don't only belong to a single cluster with probability one) and compare the probabilities that they belong to each cluster across each of the runs. The probabilities for each peak and cluster cluster are calculated as the number of times that the peak is allocated to a cluster divided by a count of the total number of iterations run. If the probabilities across the runs across the runs are similar then this suggests that the Gibbs sampler has converged to its stationary distribution after this first block of 100 iterations. If they have not converged, then the algorithms can be re-started from their current position and re-run for a further 100 iterations but with the counts (the iteration count and the number of times each peak is allocated to each cluster) used in calculation of the probabilities reset to zero. This process can then be repeated until convergence is observed.

7.2.2 Further Work for Assessing Predictive Ability of Adduct Patterns

As discussed in the previous chapter, the work carried out in this dissertation suggests that the adduct patterns observed may be of use in identifying isomers. The next stage now would be to identify isomers where the adduct patterns have been particularly successful in their identification and then obtain extra experimental mass spectrometry data on these in order to further analyse them. However, there is a significant financial cost associate with processing samples through the mass spectrometer. In light of this, further work would be needed here in order to further establish which molecules are should be assessed further.

It was discussed in the previous chapter how the scores set out in equation 6.2 could be used to assess the predictive power of adduct patterns in distinguishing between isomer pairs. This involves first computing probabilities of the presence or absence of each adduct for each molecule using a collection of training files. Then, from the adduct patterns observed in the test files, the scores can now be calculated for each molecule to measure whether they more closely resemble the standard 1 or standard 2 isomer. It can then be assessed whether the scores obtained from the adduct patterns correctly identify as the standard 1 and standard 2 molecule in each isomer pair.

In the data used to compute these scores there was imbalance between the number of molecules allocated to clusters between the standard 1 and standard 2 files. For standard 2 files, there were significantly fewer molecules matched to clusters and hence there was a lack of adduct patterns available for the standard 2 molecules in carrying out this analysis. It is suspected that this is due to an error in the standard 2 file data. Hence, one potential area for further work is in obtaining further data sets of standard 2 molecules which provide a greater degree of molecule identification and then using these as a basis for carrying out an analysis of the scores. This would allow a greater number of molecule's adduct patterns to be tested and would provide a more thorough analysis of their predictive ability.

In calculating the scores, the selection of training and test files from the available data files was done arbitrarily. However, there will be variations between the files in the number of molecules identified as well as in the adduct patterns associated with each molecule. Therefore, another area for further work is to calculate the scores using different combinations of files being used as training and test data. For example, the proportion of the total files used that are to be used as test and training data could first be decided on (e.g. 60% training and 40% test) and then files could be randomly assigned to either the training or test group in keeping with these proportions. This could be repeated several times with the classification results for each molecule, over the current allocation of test files, being recorded each time. This would help eliminate the effect that different combinations of training and test files may have on the analysis.

Also, at present, the scores for each molecule are being computed for individual test files. However, it is also desirable to be able to be assess the classification of each molecule across all of the test files as a whole. A further area for future work is therefore to construct a score which can be used for the purpose. Rather using the binary presence or absence of each adduct, such a score would need to use the frequency with which each adduct is observed across the test files for a given molecule.

In the analysis carried out only the presence or absence of each adduct for a given molecule has been considered. For example, if a given adduct was observed for a molecule in 3 out of 5 files, then the probability, p, of observing the adduct for this molecule is taken as 0.6. Hence, a binomial model is being fitted to each molecule where the probability of observing a given adduct in x out of n files is proportional to $p^x(1-p)^{n-x}$. Another area for further work would be to extend this model by incorporating the intensity peak data. The intensity data observed for a molecule in a given file could be normalised by dividing each intensity by the M+H adduct intensity (the largest intensity for each molecule). These normalised values could now be used as the probability values for each adduct's presence. That is, for each adduct, the intensity information will now give probabilities p_i for each adduct i which sum to one. Using these, a multinomial distribution could instead be fitted for each molecule with the probability of observing the vector $[x_1, x_2, ...]^T$ proportional to $p_1^{x_1} \times p_2^{x_2} \times ...$,where x_i equals 1 or 0 for adduct i and indicates its presence or absence.

Appendix A

First appendix

A.1 Key terms and Derivations

A.1.1 Gibbs Sampler - Marginalisation Step

The π term may be integrated out by proceeding as follows:

$$\pi_k = p(z_{nk} = 1|\boldsymbol{z}_{-n}, \alpha) \tag{A.1}$$

$$= \int p(z_{nk} = 1|\pi) p(\boldsymbol{\pi}|\boldsymbol{z}_{-n}, \alpha) d\boldsymbol{\pi} \tag{A.2}$$

$$= \int \pi_k \frac{\Gamma(\sum_k (\alpha_k + c_k^{-n}))}{\prod_k \Gamma(\alpha_k + c_k^{-n})} \prod_j \pi_j^{\alpha_j + c_j^{-n} - 1} d\boldsymbol{\pi} \text{ (where } c_j^{-n} = \sum_{i \neq n} z_{ik}) \tag{A.3}$$

$$= \frac{\Gamma(\sum_k (\alpha_k + c_k^{-n}))}{\prod_k \Gamma(\alpha_k + c_k^{-n})} \int \prod_j \pi_j^{\alpha_j + c_j^{-n} + \delta_{jk} - 1} d\boldsymbol{\pi} \tag{A.4}$$

(where $\delta_{jk} = 1$ for $j = k$ and is zero otherwise) (A.5)

The term inside the integral is itself a Dirichlet distribution with parameter $\alpha_j + c_j^{-n} + \delta_{jk}$. Hence this can be evaluated by comparing it to the normalisation constant term in Dirichlet pdf:

$$= \frac{\Gamma(\sum_k (\alpha_k + c_k^{-n}))}{\prod_k \Gamma(\alpha_k + c_k^{-n})} * \frac{\prod_j (\Gamma(\alpha_j + c_j^{-n} + \delta_{jk}))}{\Gamma(\sum_j (\alpha_j + c_j^{-n} + \delta_{jk}))} \tag{A.6}$$

$$= \frac{\Gamma(\sum_k (\alpha_k + c_k^{-n}))}{\Gamma(\sum_j (\alpha_j + c_j^{-n} + \delta_{jk}))} * \frac{\prod_j (\Gamma(\alpha_j + c_j^{-n} + \delta_{jk}))}{\prod_k \Gamma(\alpha_k + c_k^{-n})} \tag{A.7}$$

Finally, this expression may be further simplified by using the property of the gamma function that $\Gamma(\alpha + 1) = \Gamma(\alpha)$:

$$= \frac{\Gamma(\sum_j (\alpha_j + c_j^{-n}))}{\sum_j (\alpha_j + c_j^{-n}) \Gamma(\sum_j (\alpha_j + c_j^{-n}))} * \frac{\alpha_k \Gamma(\alpha_k + c_k^{-n})}{\Gamma(\alpha_k + c_k^{-n})} \qquad = \frac{(\alpha_k + c_k^{-n})}{\sum_j (\alpha_j + c_j^{-n})}. \tag{A.8}$$

A.1.2 Variational Bayes Lower Bounnd

The log-likelihood function may be written as:

$$\begin{aligned}\ln p(\boldsymbol{x}) &= \ln \int p(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \ln \int Q(\boldsymbol{\theta}) \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{Q(\boldsymbol{\theta})} d\boldsymbol{\theta}\end{aligned}$$

where $Q(\boldsymbol{\theta})$ is an arbitrary distribution. Now applying Jensen's inequality (see [7] for details):

$$\ln(\boldsymbol{E}_{p(z)}[f(z)]) \geq \boldsymbol{E}_{p(z)}[\ln(f(z))],$$

a lower bound on $\ln p(\boldsymbol{x})$ can be obtained as:

$$\begin{aligned}\ln p(\boldsymbol{x}) &\geq \int Q(\boldsymbol{\theta}) \ln(\frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{Q(\boldsymbol{\theta})}) d\boldsymbol{\theta} \\ &= L(\boldsymbol{\theta})\end{aligned}$$

Hence

$$\ln p(\boldsymbol{x}) - L(\boldsymbol{\theta}) \geq 0.$$

Expanding the left hand side of the above inequality gives:

$$\begin{aligned}\ln p(\boldsymbol{x}) - L(\boldsymbol{\theta}) &= \ln p(\boldsymbol{x}) - \int Q(\boldsymbol{\theta}) \ln(\frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{Q(\boldsymbol{\theta})}) d\boldsymbol{\theta} \\ &= \ln p(\boldsymbol{x}) - \int Q(\boldsymbol{\theta}) \ln(\frac{p(\boldsymbol{\theta}|\boldsymbol{x}) p(\boldsymbol{x})}{Q(\boldsymbol{\theta})}) d\boldsymbol{\theta} \\ &= \ln p(\boldsymbol{x}) - \int Q(\boldsymbol{\theta}) \ln(\frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{Q(\boldsymbol{\theta})}) d\boldsymbol{\theta} - \int Q(\boldsymbol{\theta}) \ln(p(\boldsymbol{x})) d\boldsymbol{\theta} \\ &= -\int Q(\boldsymbol{\theta}) \ln(\frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{Q(\boldsymbol{\theta})}) d\boldsymbol{\theta} \\ &= -KL[Q(\boldsymbol{\theta})][p(\boldsymbol{\theta}|\boldsymbol{x})] \qquad \text{(A.9)}\end{aligned}$$

where $KL[Q(\boldsymbol{\theta})][p(\boldsymbol{\theta}|\boldsymbol{x})]$ is the **Kullback-Leibler** (KL) divergence between $Q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\boldsymbol{x})$. (See [7] for details.)

A.1.3 Derivation of Q_z

$$\begin{aligned}Q_z(\mathbf{z}) &\propto \exp\{\mathbf{E}_{Q_\pi(\pi)Q_\mu(\mu)}[\sum_n \sum_k z_{nk}(\ln(\pi_k) + \ln(N(\boldsymbol{x}_n^k|\boldsymbol{\mu}_k, \Sigma)))]\} \\ &= \exp\{\sum_n \sum_k z_{nk}(\langle \ln(\pi_k) \rangle + \langle \ln(N(\boldsymbol{x}_n^k|\boldsymbol{\mu}_k, \Sigma)) \rangle)\} \\ &= \exp\{\sum_n \sum_k z_{nk}(\langle \ln(\pi_k) \rangle + \langle \ln(N(x_{n,M}^k|\mu_k^M, \sigma_{RT}^2)) \rangle + \langle \ln(N(x_n^{RT}|\mu_k^{RT}, \sigma_{RT}^2)) \rangle)\}\end{aligned}$$

Writing

$$\begin{aligned}\ln \gamma_{nk} &= \langle \ln(\pi_k) \rangle + \langle \ln(N(x^k_{n,M}|\mu^M_k, \sigma^2_M)) \rangle + \langle \ln(N(x^{RT}_n|\mu^{RT}_k, \sigma^2_{RT})) \rangle \\ &= \langle \ln(\pi_k) \rangle - \frac{\langle (x^k_{n,M} - \mu^M_k)^2 \rangle}{2\sigma^2_M} - \frac{\langle (x^{RT}_n - \mu^{RT}_k)^2 \rangle}{2\sigma^2_{RT}} - \ln(2\pi\sigma_M\sigma_{RT})\end{aligned}$$

As $\boldsymbol{\pi}$ follows a distribution, it can be shown that

$$\langle \ln(\pi_k) \rangle = \psi(\tilde{\alpha_k}) - \psi(\sum_j \tilde{\alpha_j}) \ (k = 1, 2, ..., K)$$

where $\psi(.)$ is the digamma function. (See for details.) We then have that

$$\begin{aligned}\ln \gamma_{nk} &= \psi(\tilde{\alpha_k}) - \psi(\sum_j \tilde{\alpha_j}) - \frac{(x^k_{n,M})^2 - 2x^k_{n,M}\langle \mu^M_k \rangle + \langle (\mu^M_k)^2 \rangle}{2\sigma^2_M} \\ &\quad - \frac{(x^{RT}_n)^2 - 2x^{RT}_n \langle \mu^{RT}_k \rangle + \langle (\mu^{RT}_k)^2 \rangle}{2\sigma^2_{RT}} - \ln(2\pi\sigma_M\sigma_{RT}) \\ &= \psi(\tilde{\alpha_k}) - \psi(\sum_j \tilde{\alpha_j}) - \frac{(x^k_{n,M})^2 - 2x^k_{n,M}\tilde{\mu}_{M,k} + \tilde{\mu}^2_{M,k} + \tilde{\sigma}^2_{M,k}}{2\sigma^2_M} \\ &\quad - \frac{(x^{RT}_n)^2 - 2x^{RT}_n \tilde{\mu}_{RT,k} + \tilde{\mu}^2_{RT,k} + \tilde{\sigma}^2_{RT,k}}{2\sigma^2_{RT}} - \ln(2\pi\sigma_M\sigma_{RT}). \qquad \text{(A.10)}\end{aligned}$$

Substituting this into the above equation for Q_z gives

$$\begin{aligned}Q_z(\mathbf{z}) &\propto \exp(\sum_n \sum_k z_{nk} \ln \gamma_{nk}) \\ &= \prod_n \prod_k \gamma^{z_{nk}}_{nk}\end{aligned}$$

After normalising, it follows $Q(\boldsymbol{z}_n)$ follows a multinomial distribution with parameters $\gamma_{nk} / \sum_j \gamma_{nj}$ and:

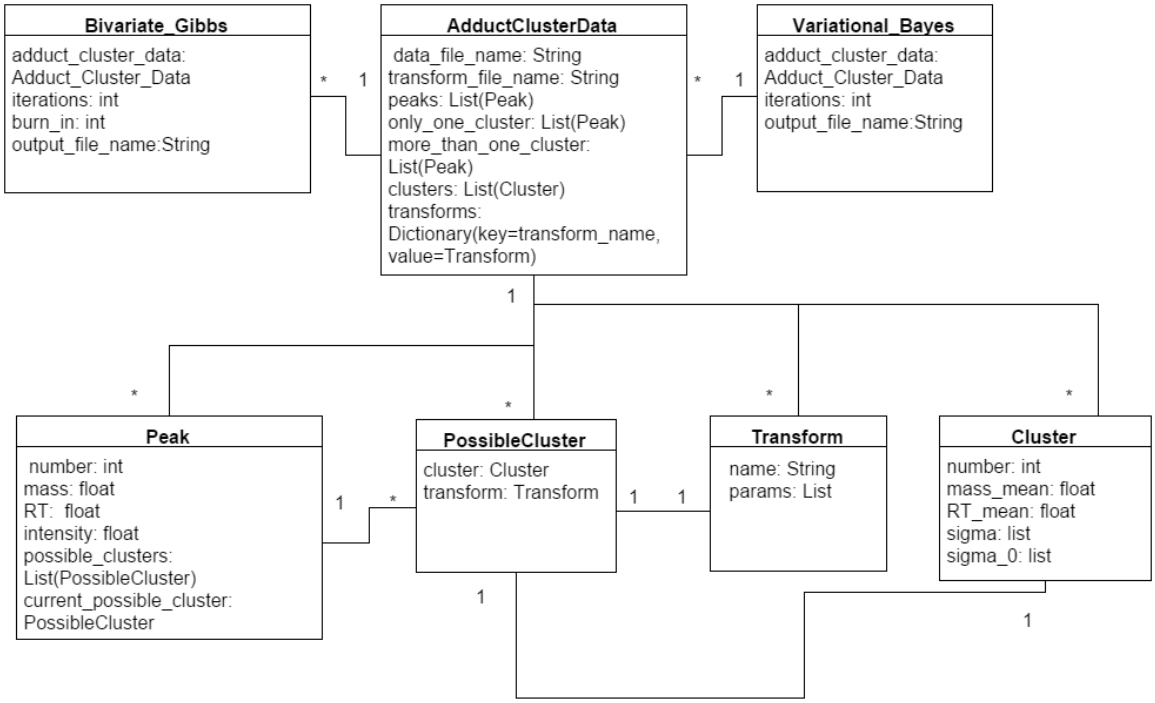
$$\langle z_{nk} \rangle = \frac{\gamma_{nk}}{\sum_j \gamma_{nj}} \ (n = 1, 2, ..., N \ and \ k = 1, 2, ..., K). \qquad \text{(A.11)}$$

Appendix B

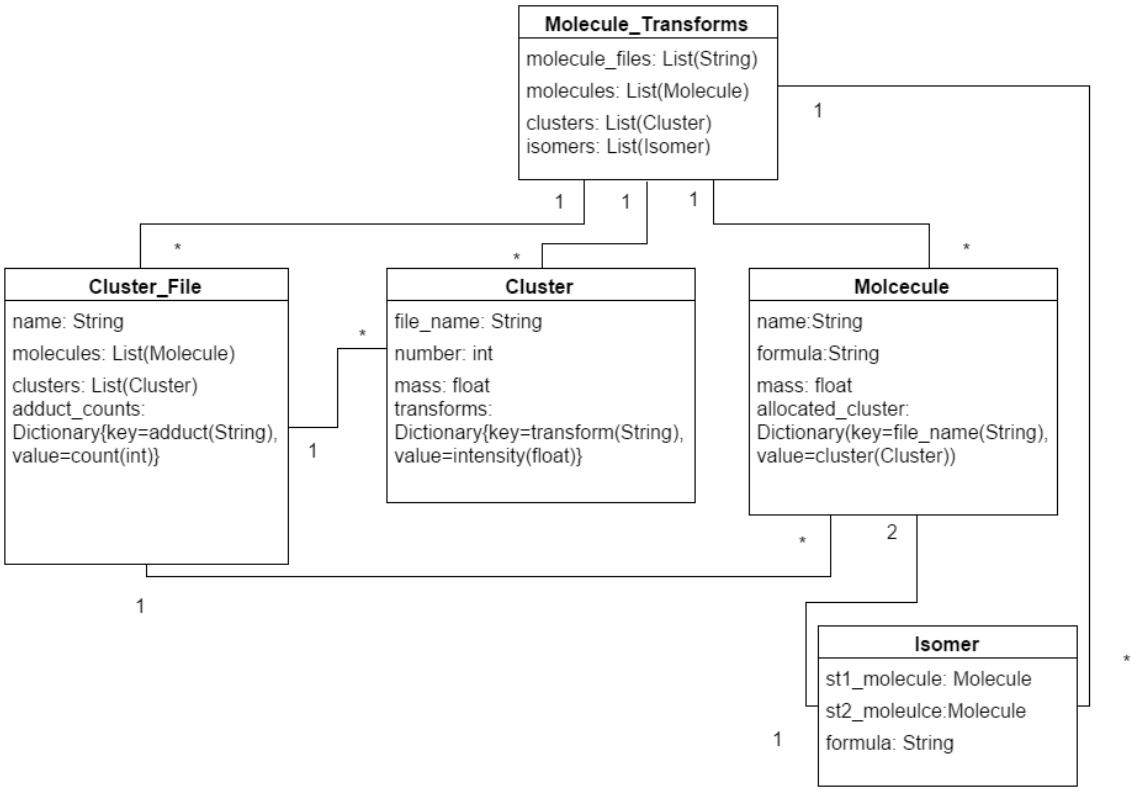
Second appendix

B.1 Class Diagrams

B.1.1 Clustering Step Class Diagram



B.1.2 Molecule Allocation Step



B.1.3 Read Me

ReadMe

The document sets out each of the files for the peak clustering tool developed.

1. **Peak_Cluster_Model.py**

   This is the main file used to run the program. To run the program type “python Peak_Cluster_Model.py” in the command line.

2. **Adduct_Details.py**

   Contains classes used in implementing the Gibbs sampling and vibrational Bayes algorithms.

3. **Adduct_Cluster_Data.py**

   Used to read in peak data from the raw MS files. Stores creates the Peak, Cluster, PossibleCluster and Transform objects needed in the Gibbs sampling and Variational Bayes clustering algorithms.

4. **Gibbs_Sampling.py and Variational_Bayes.py**
   Used to run the Gibbs sampling and variational Bayes algorithms respectively. Ensure peak data files to be run are contained in the Data folder. The output from the algorithm will be saved to the folder Data/Output. Two output files should be saved per folder- ending “RUN_JF” showing the cluster each peak is allocated to and another ending “cluster_mass_output” showing the details of each cluster’s mass.

5. **Molecule_Matching/Adduct_Molecules_all_files_UPDATED.py**
   Contains classes used to allocate clusters to molecules and identify and analyse isomers.

6. **Molecule_Matching/get_adducts_and_molecules_all_files_UPDATED.py**
   Used to allocate clusters to molecules and the produce details of each molecule's adduct frequencies across the files and its average intensities. Also used to identify and analyse isomer pairs. Various outputs summarising adduct details for isomers saved to folder Molecule_Matching/Output.

7. **Random_RT_Test.py**
   Used to run the Gibbs sampler for randomised RT values. Output charts saved to folder Data/Output_random_RT_test.

8. **Randomised_RT_Values.py**
   Reads in peak data from input files and creates peaks with randomised RT files for use in the Gibbs sampler.

9. **Molecule_Matching/verification/get_adducts_and_molecules_VERIFICATION.py**
   Computes scores for each isomer in order to assess predictive ability of adduct patterns.

10. **Test_Clustering/compare_output.py**
    Compare the cluster allocations between the outputs of two different runs of a clustering algorithm.

B.2 Data and Reports

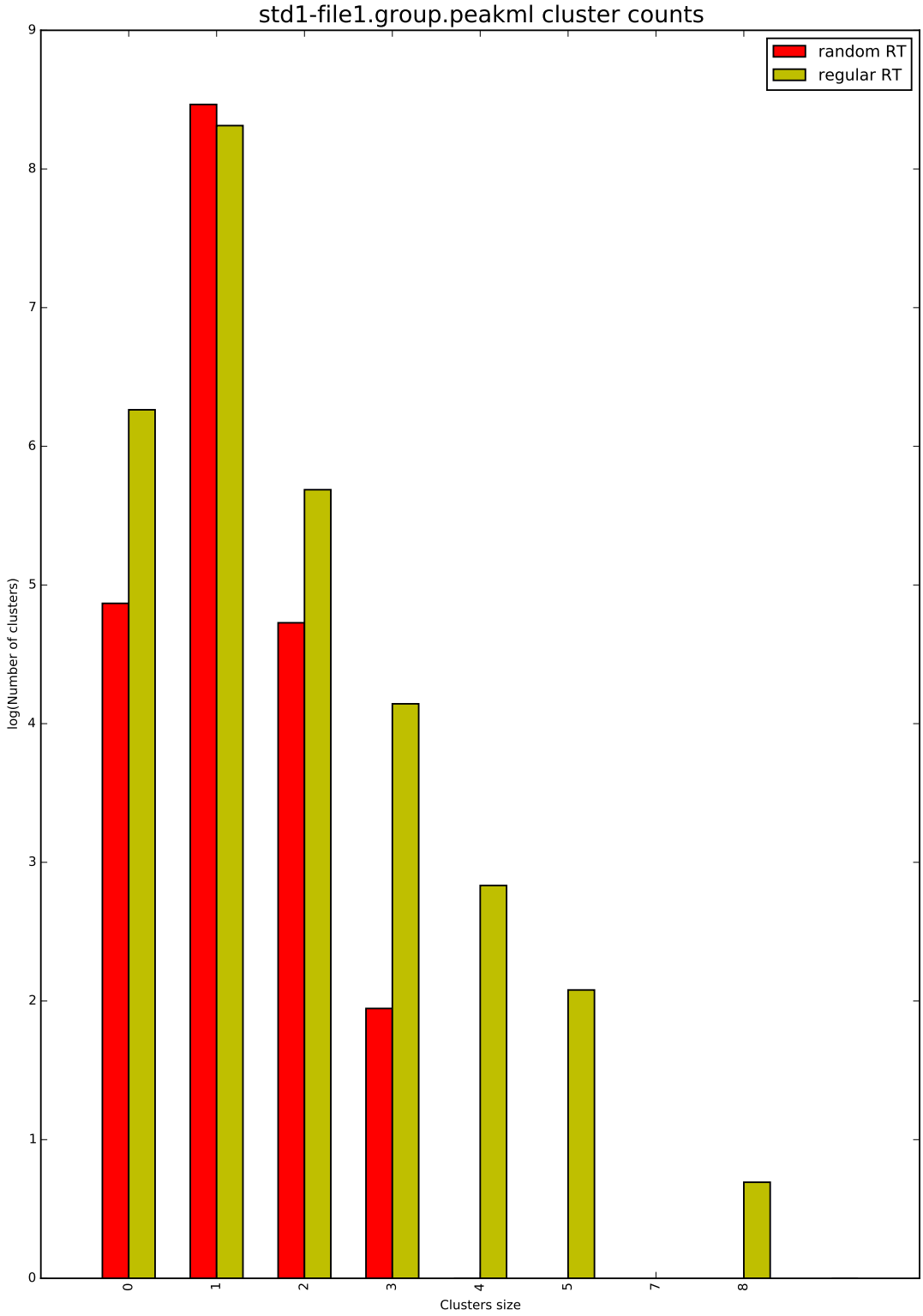
B.2.1 Comparison of Gibbs Sampler and Variational Bayes Algorithms with an Independently Developed Clustering Algorithm

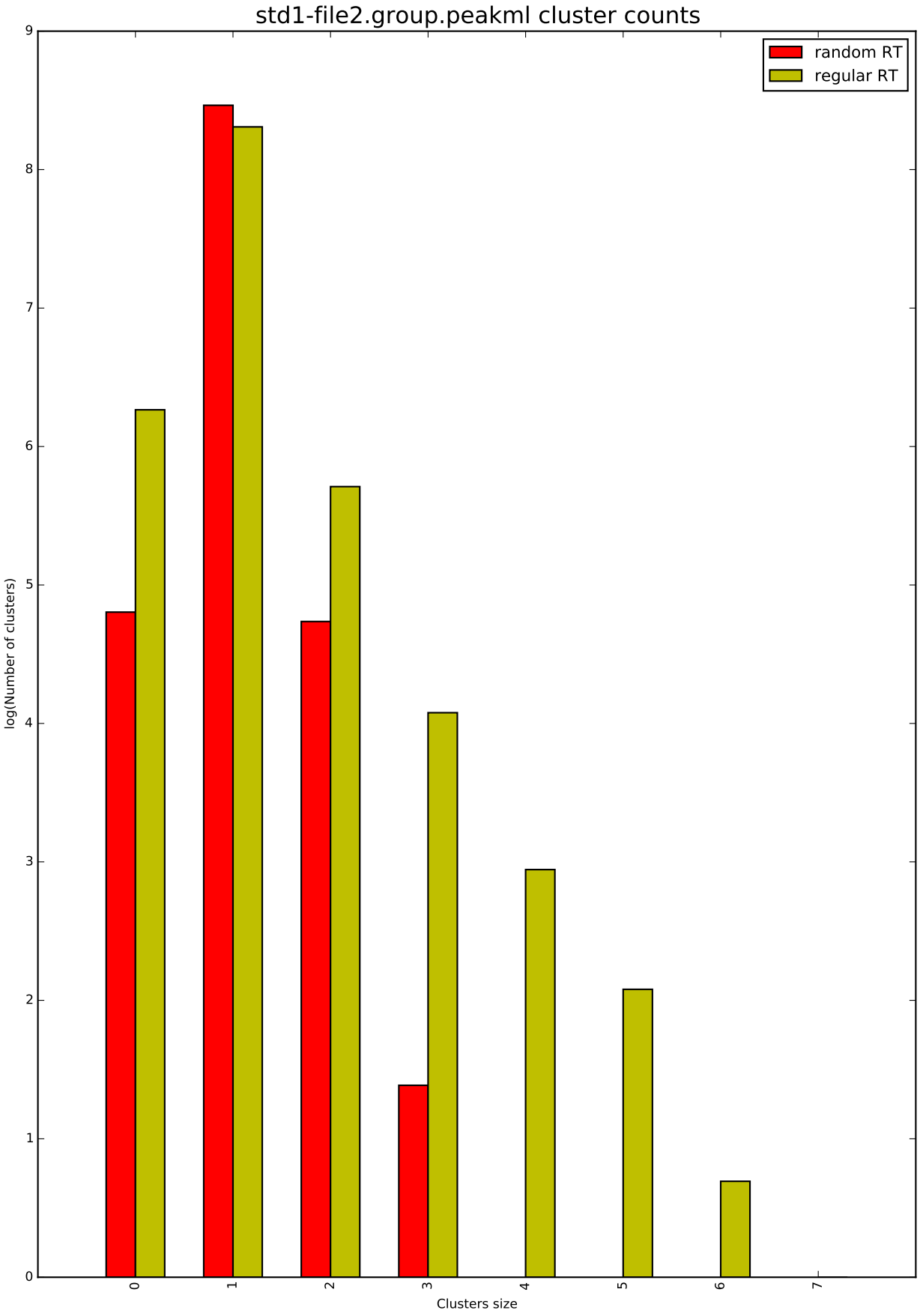
```
Gibbs Sampler File Name Percentage Match

testtxt_RUN_JF.txt     0.989397879576


Gibbs Sampler File Name Percentage Match

testtxt_VB_RUN_JF.txt     0.944788957792
```

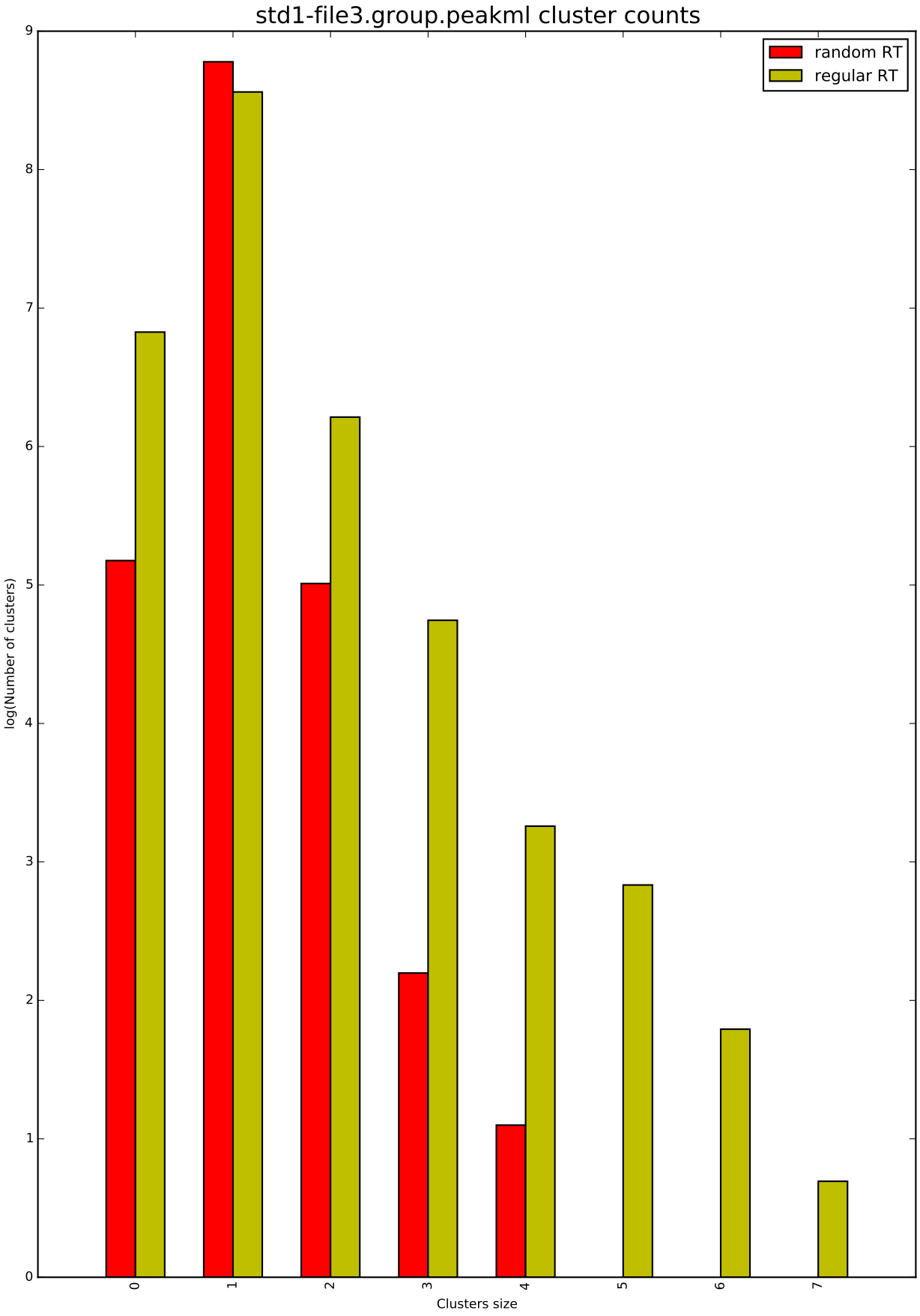
B.2.2 Comparison of Gibbs Sampler and Variational Bayes Peak Clustering

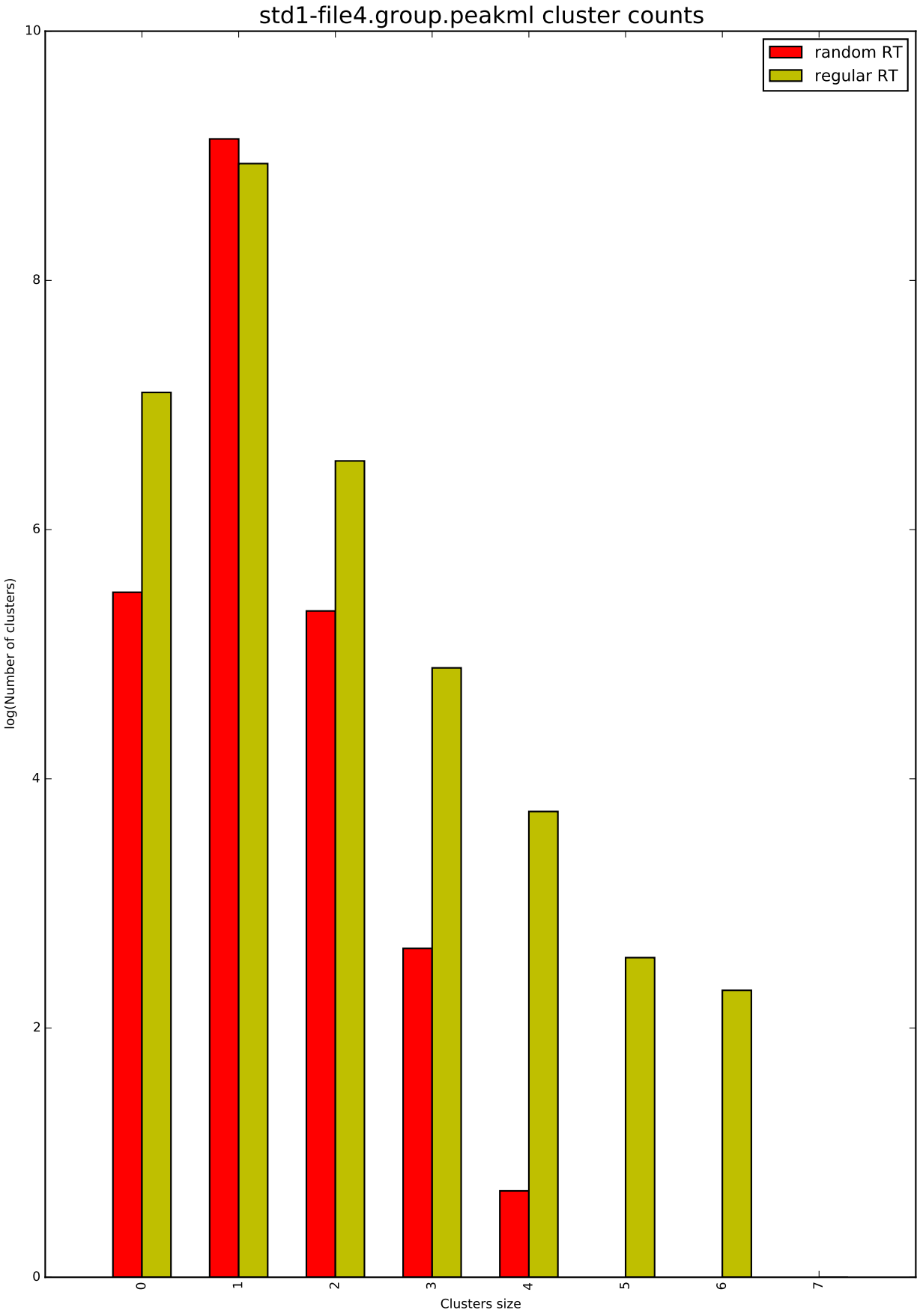
Gibbs Sampler File Name	Percentage Match with VB File
std1-file1.group.peakml_RUN_JF.txt	0.945189037808
std1-file2.group.peakml_RUN_JF.txt	0.947252306458
std1-file3.group.peakml_RUN_JF.txt	0.958894090111
std1-file4.group.peakml_RUN_JF.txt	0.954163248564
std1-file5.group.peakml_RUN_JF.txt	0.957179197287
std2-file1.group.peakml_RUN_JF.txt	0.947902385522
std2-file2.group.peakml_RUN_JF.txt	0.958153347732
std2-file3.group.peakml_RUN_JF.txt	0.942173479561
std2-file4.group.peakml_RUN_JF.txt	0.940278521693
std2-file5.group.peakml_RUN_JF.txt	0.958007459993

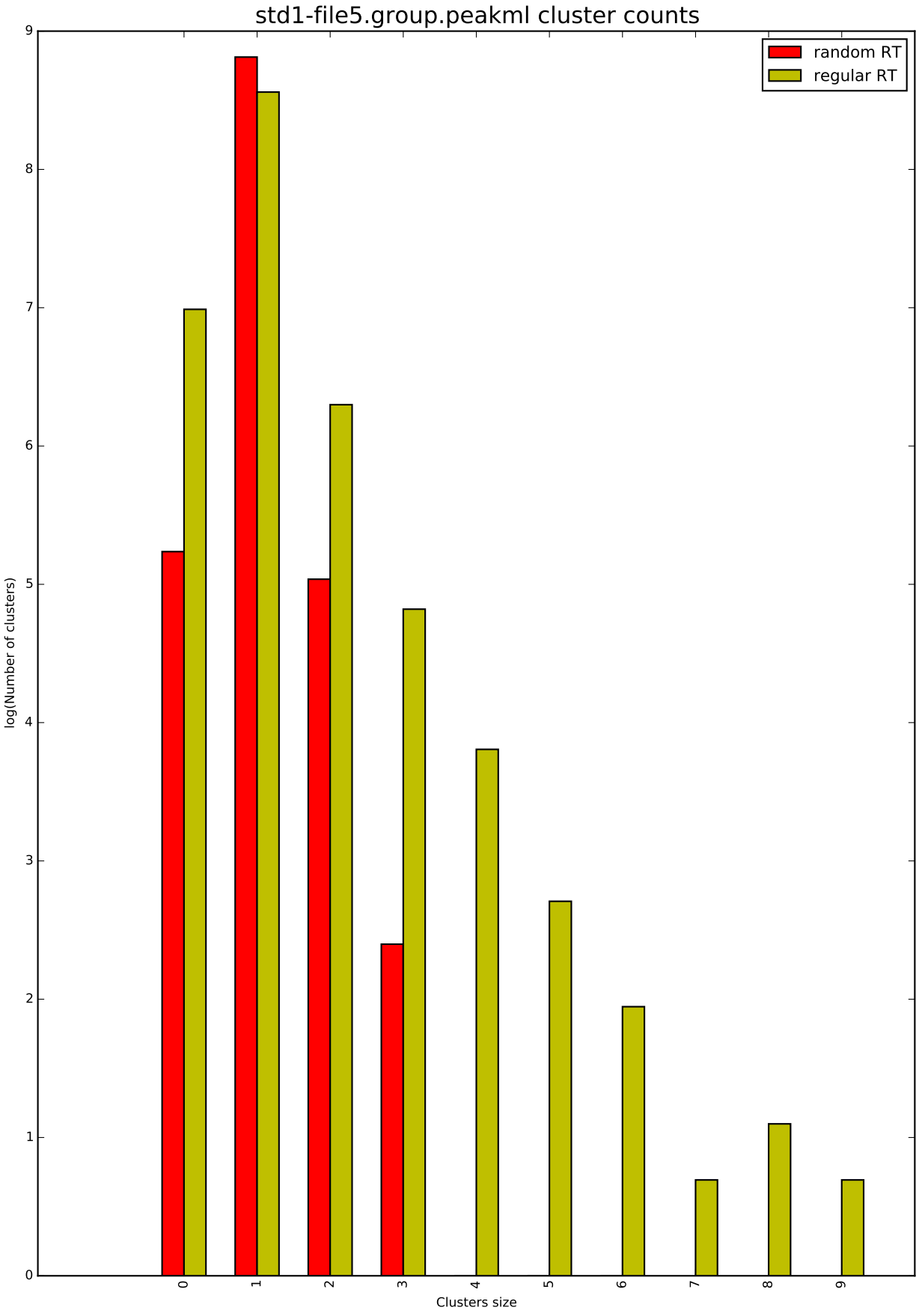
B.2.3 Plots Showing Counts of Each Cluster Size for Randomised and Regular Peak Data

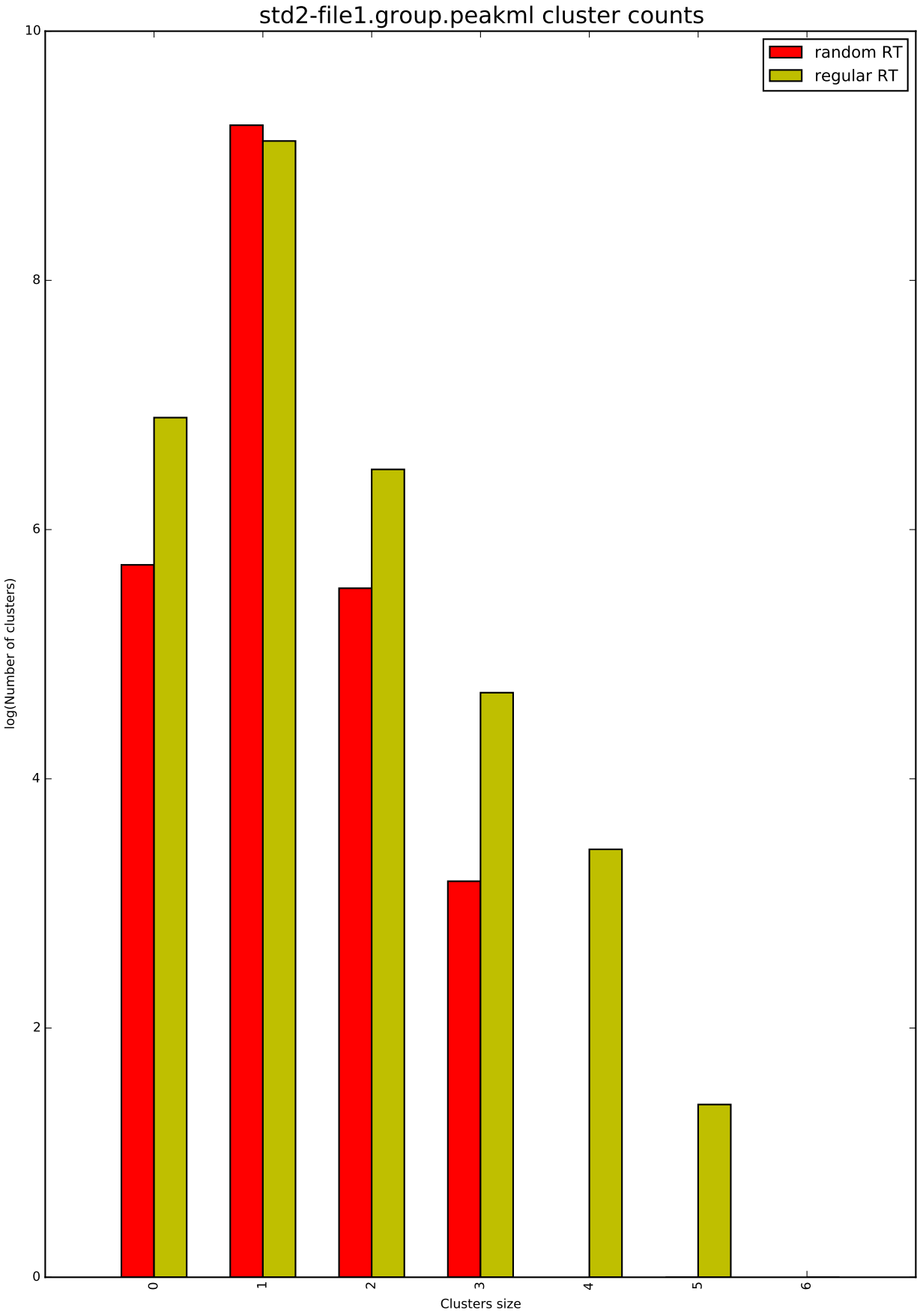


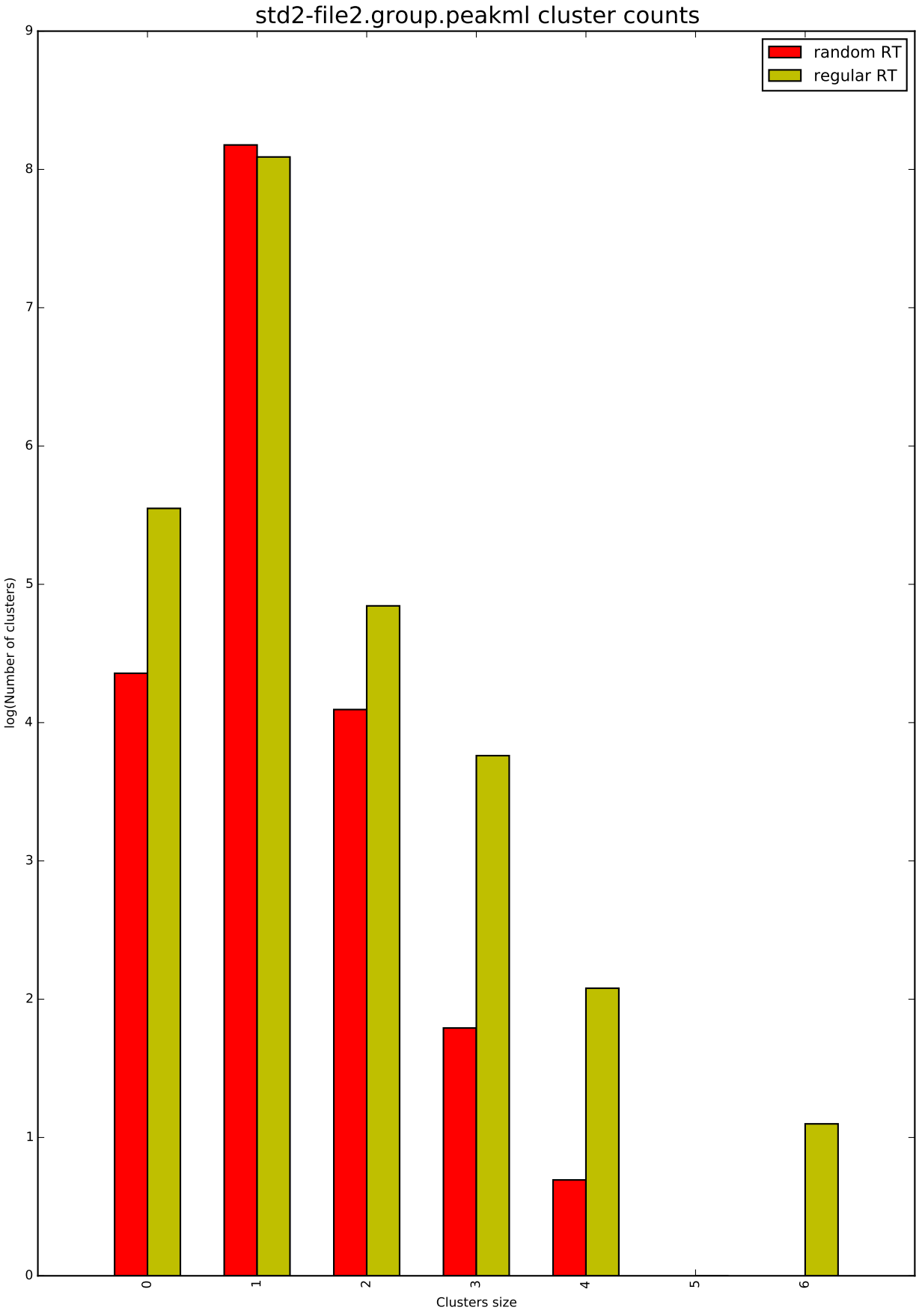

std1-file2.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
1
2
3
4
5
6
7
8
9

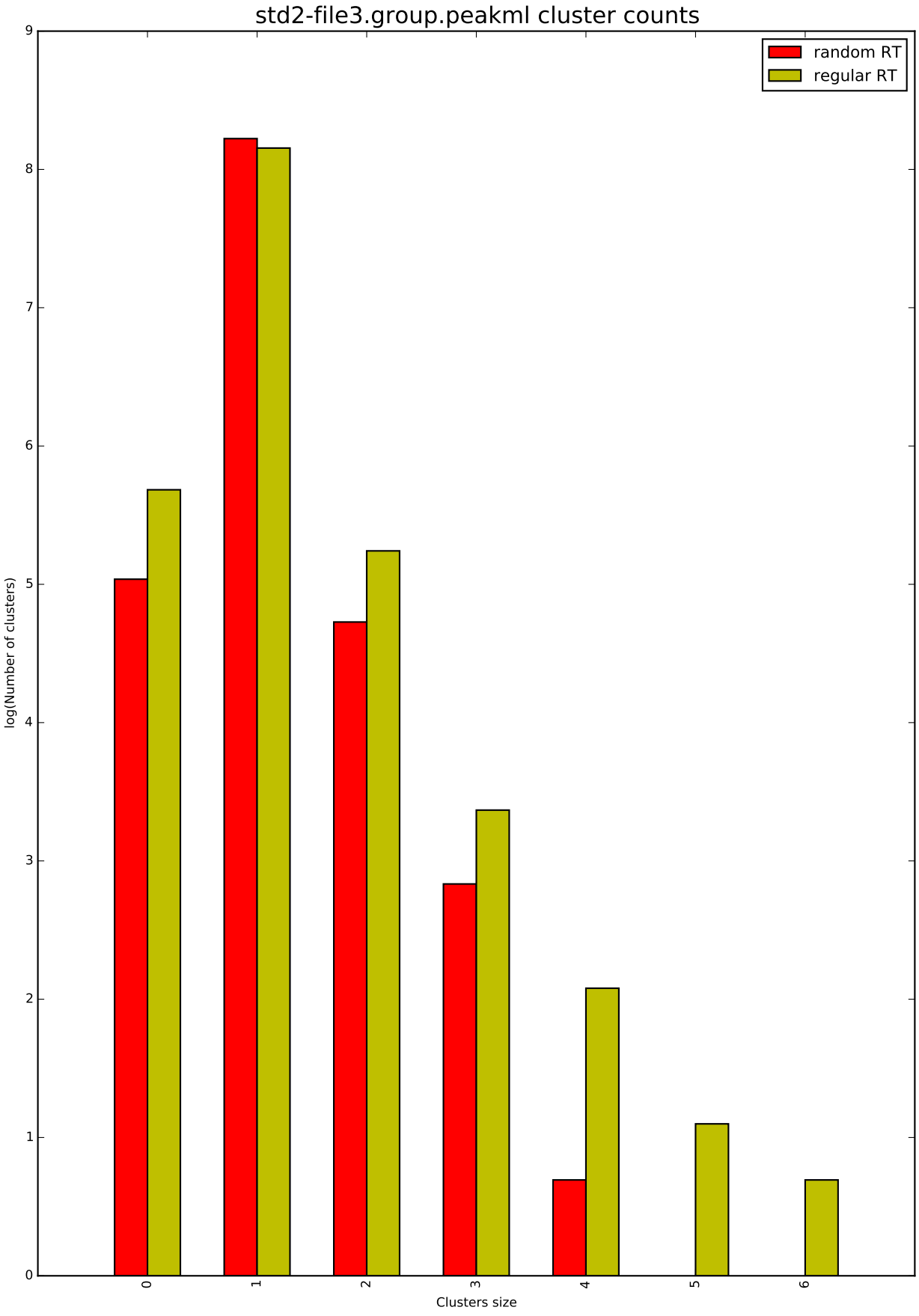

std1-file3.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
8
9

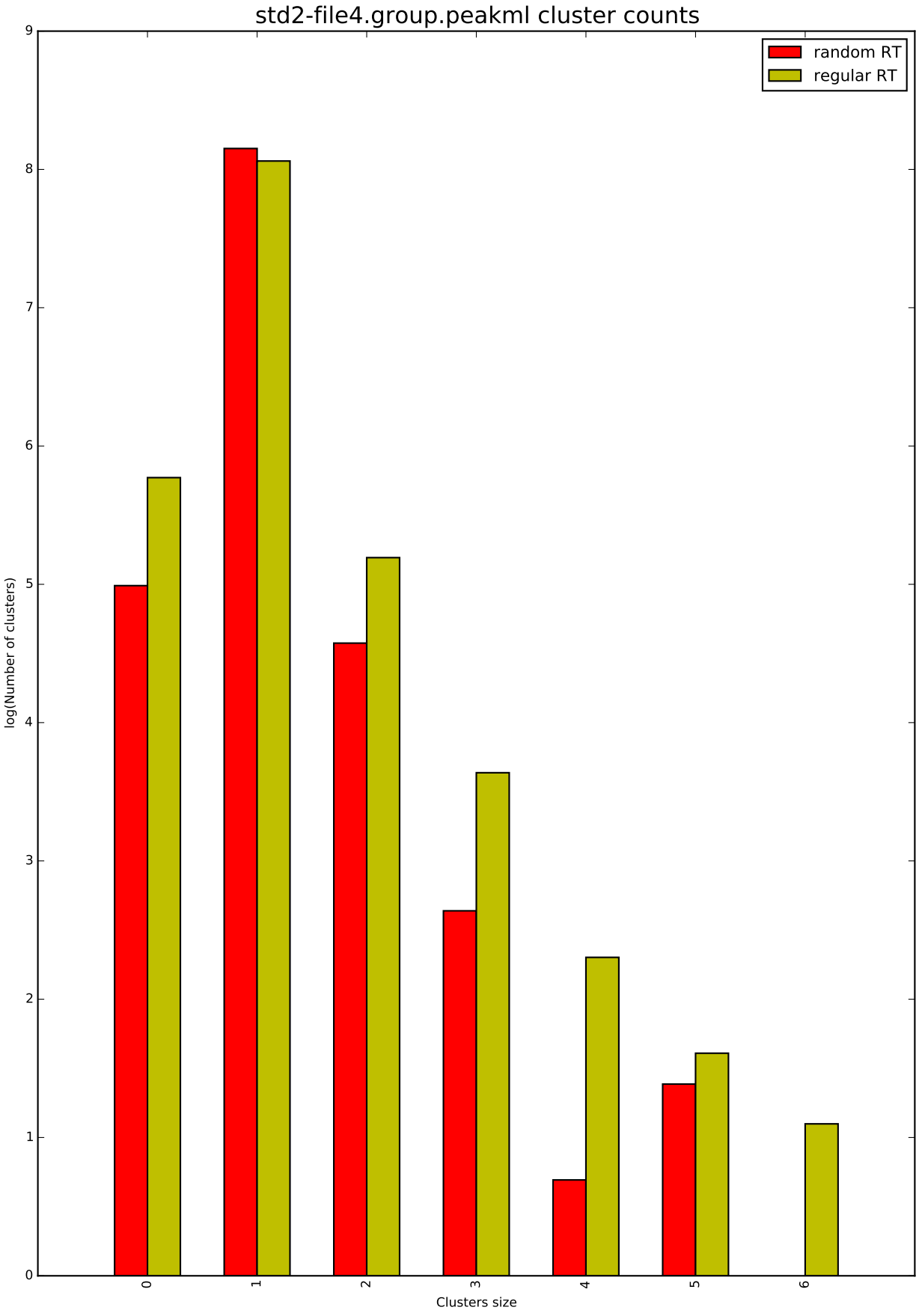

std1-file4.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
2
4
6
8
10
0
1
2
3
4
5
6
7

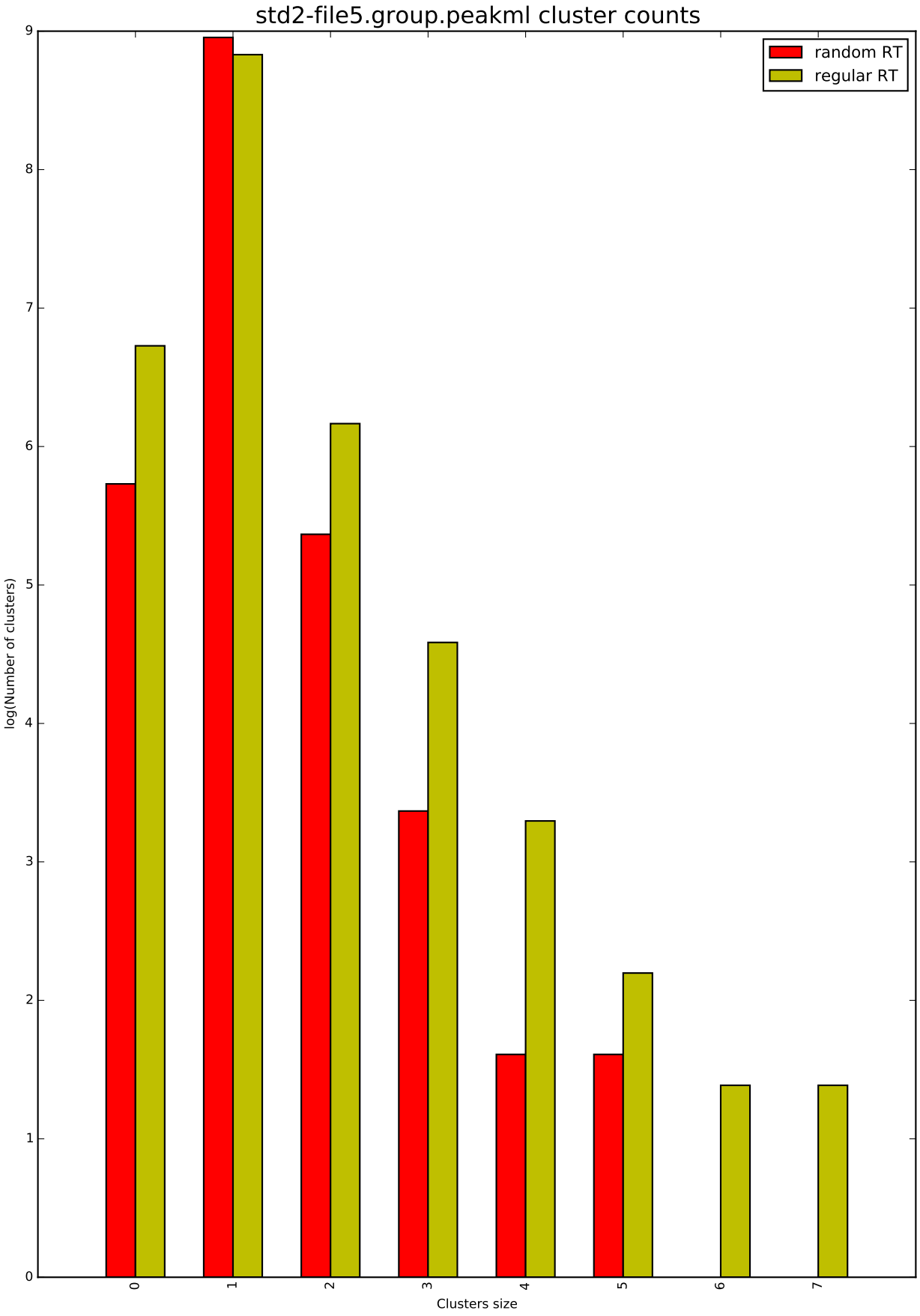

std1-file5.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
1
2
3
4
5
6
7
8
9


std2-file1.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
2
4
6
8
10
0
1
2
3
4
5
6

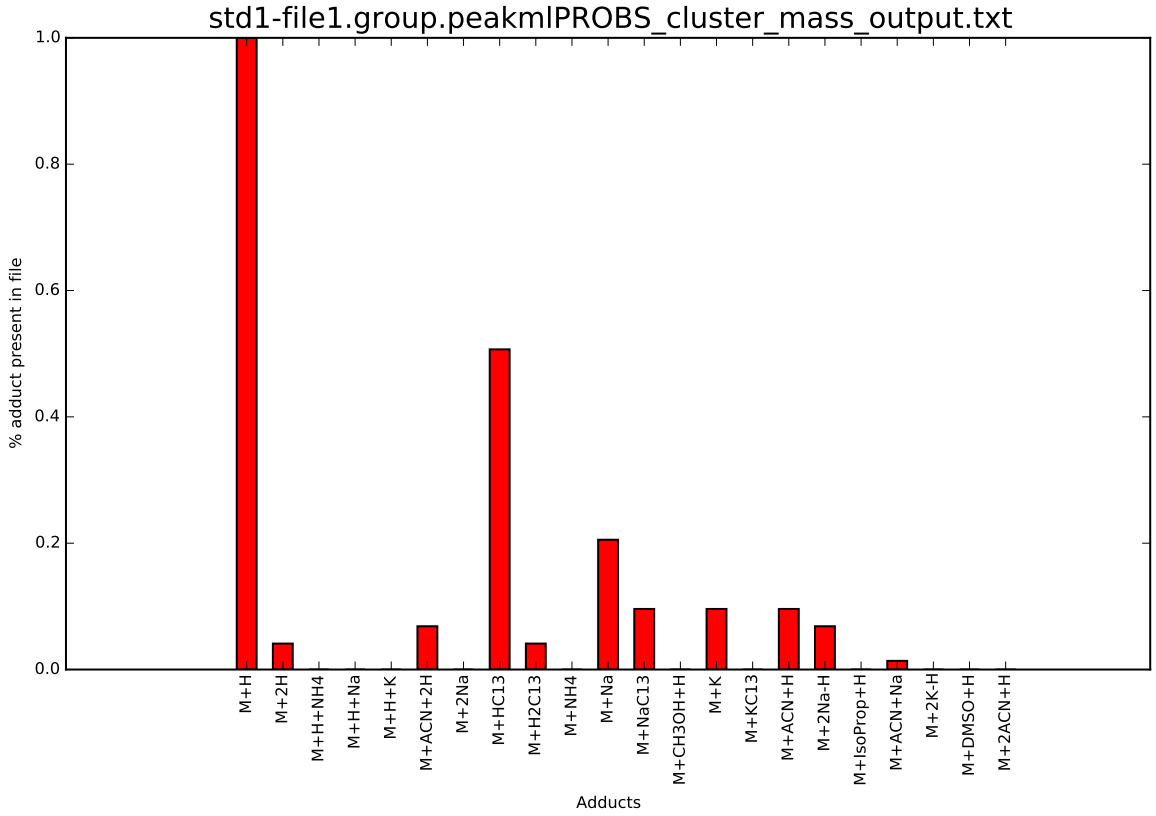

std2-file2.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8
9

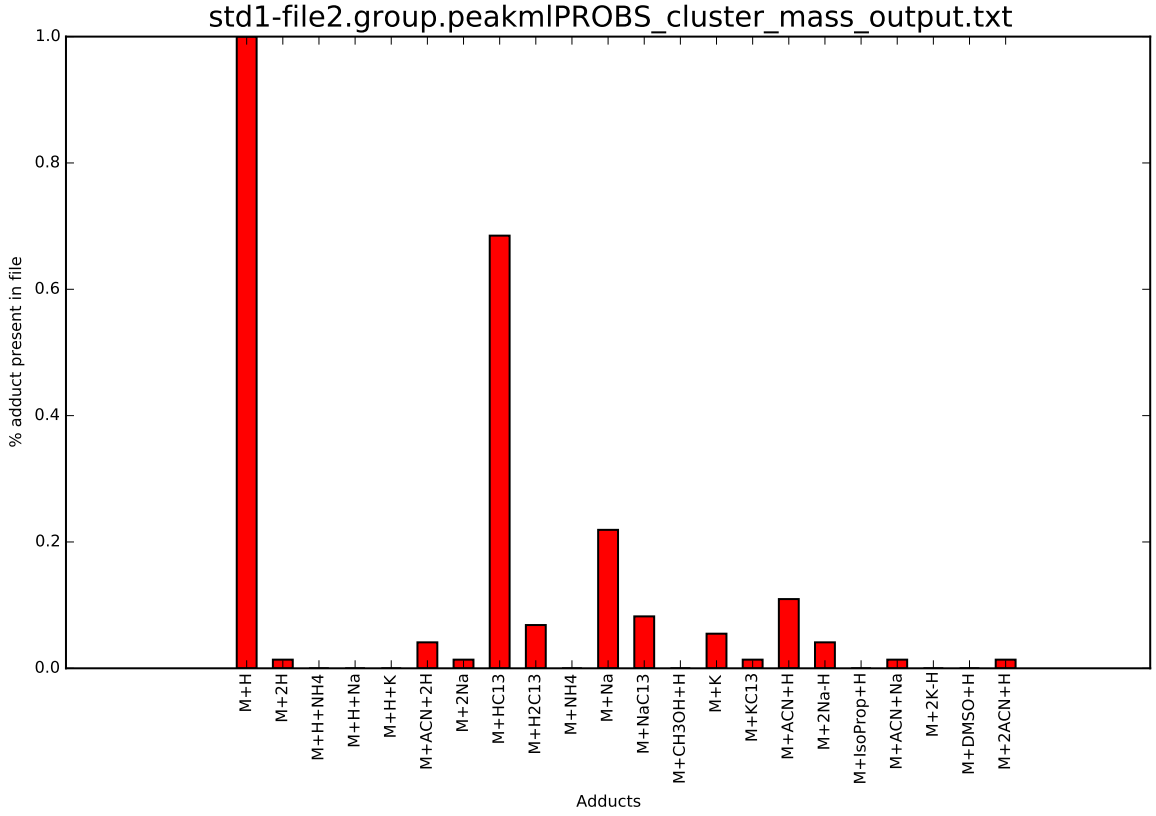

std2-file3.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8
9

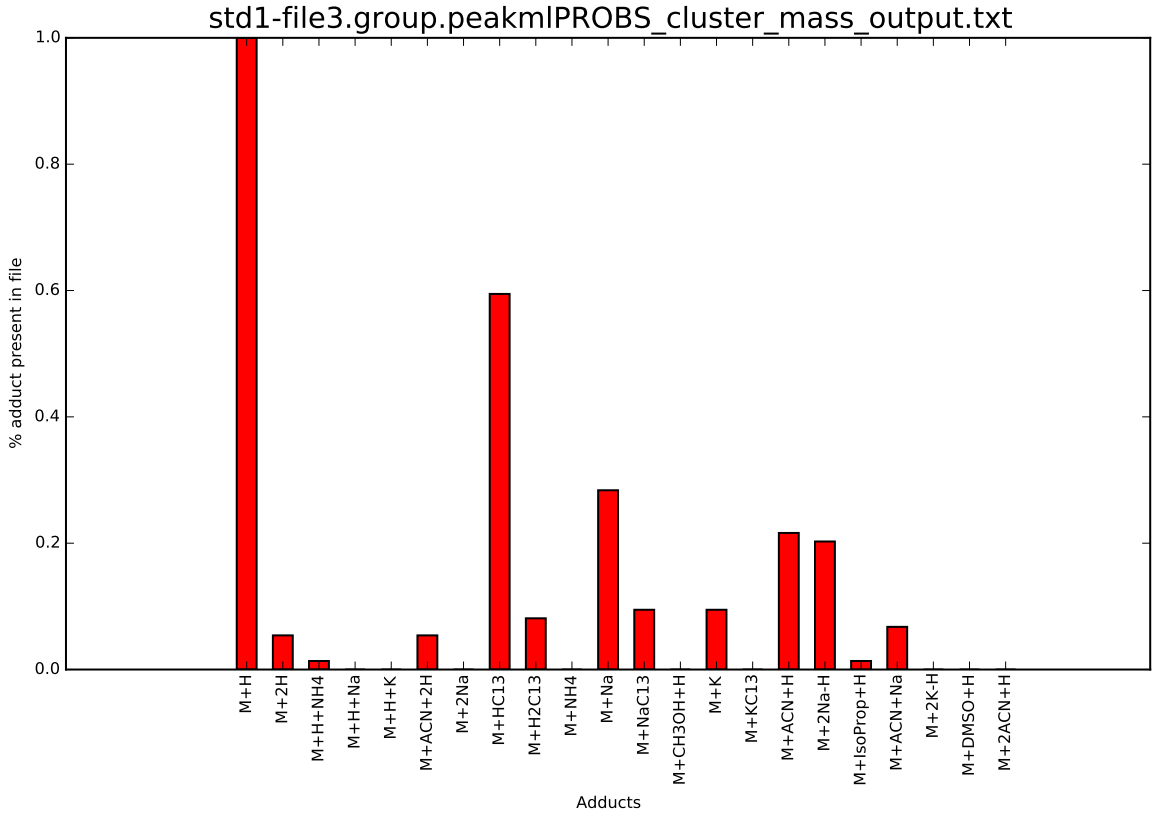

std2-file4.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8
9

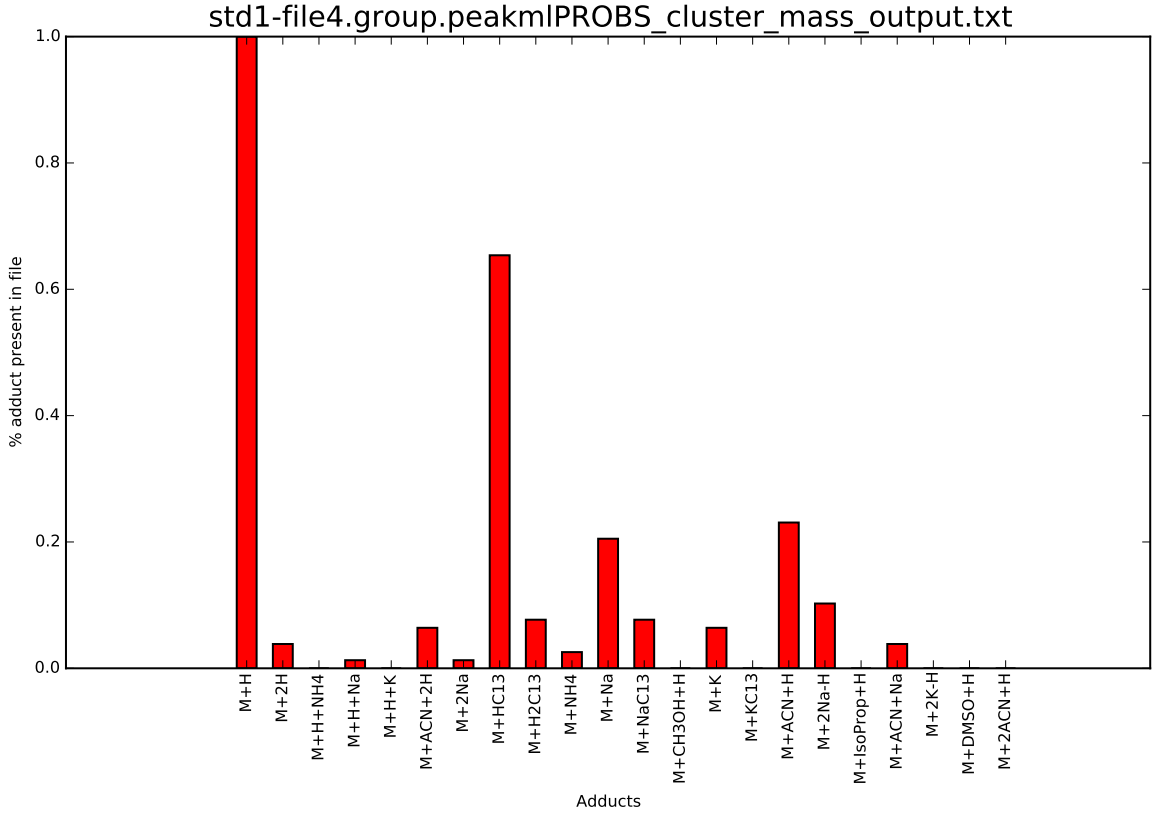

std2-file5.group.peakml cluster counts
random RT
regular RT
log(Number of clusters)
Clusters size
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
8
9

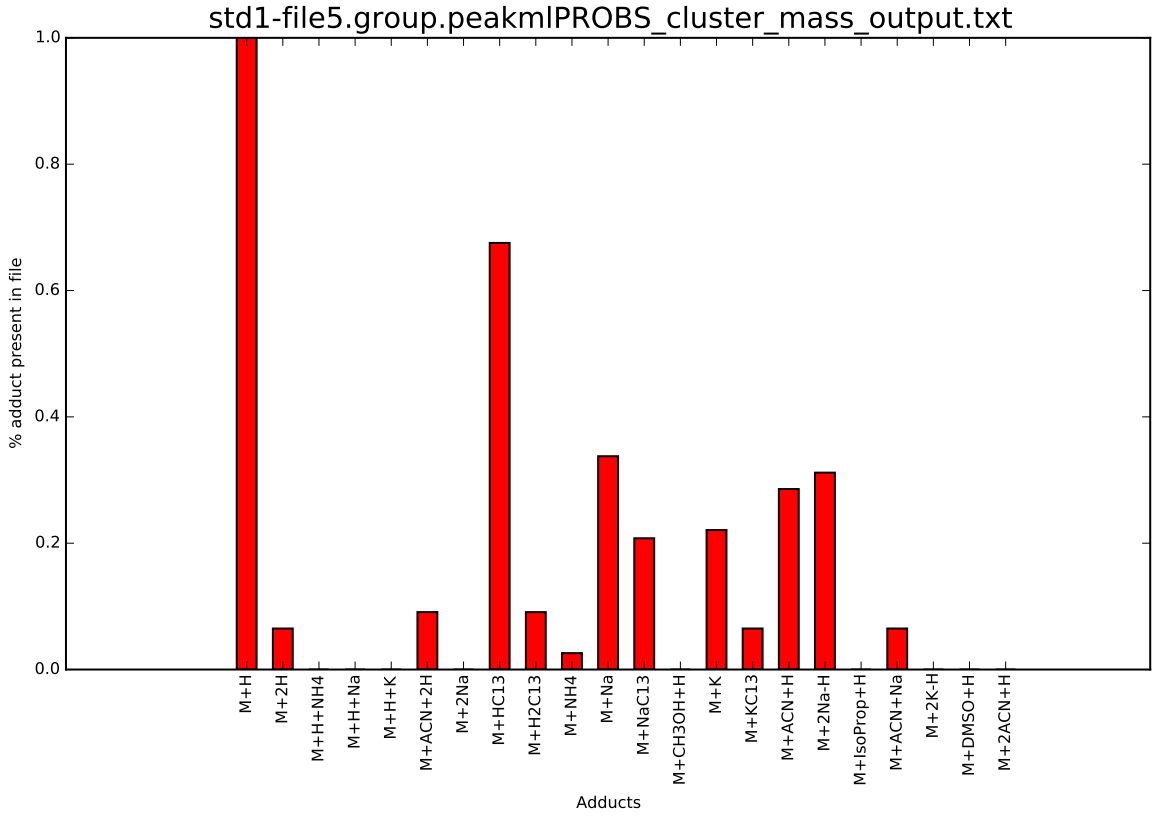
B.2.4 Plots Showing Adduct Frequencies Across Files

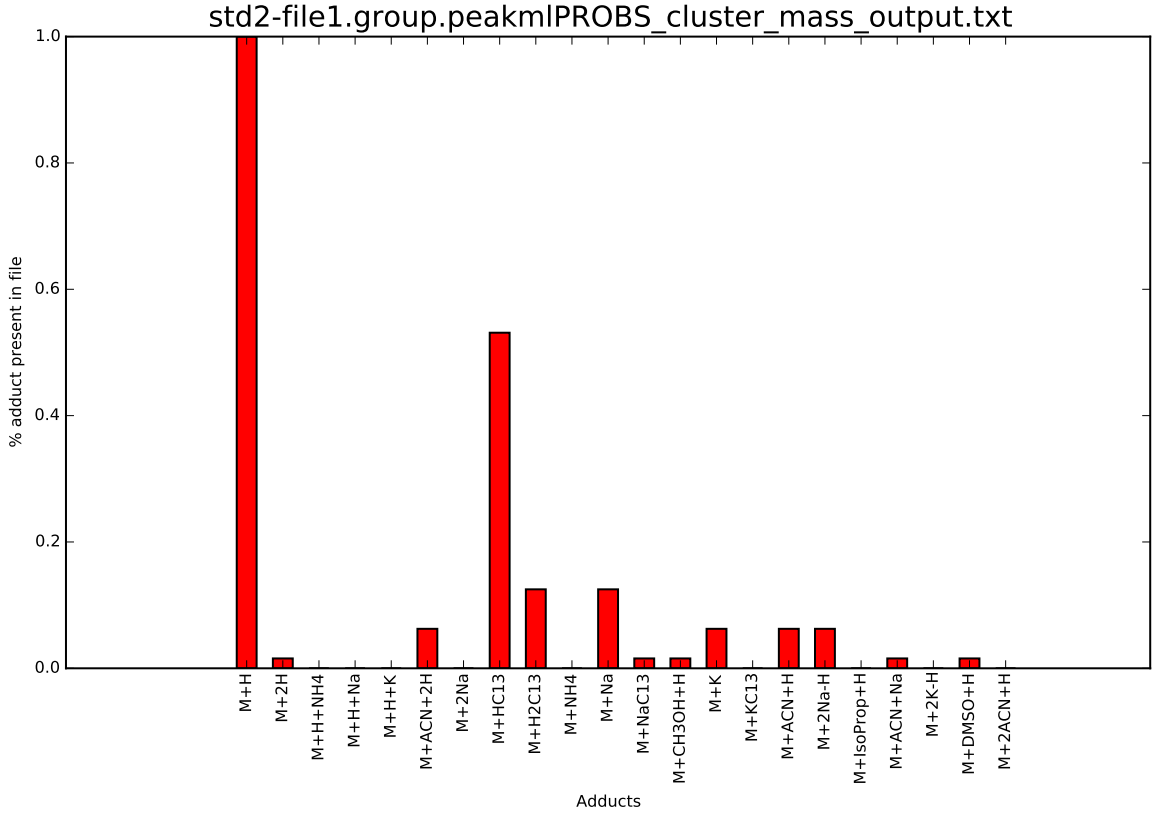


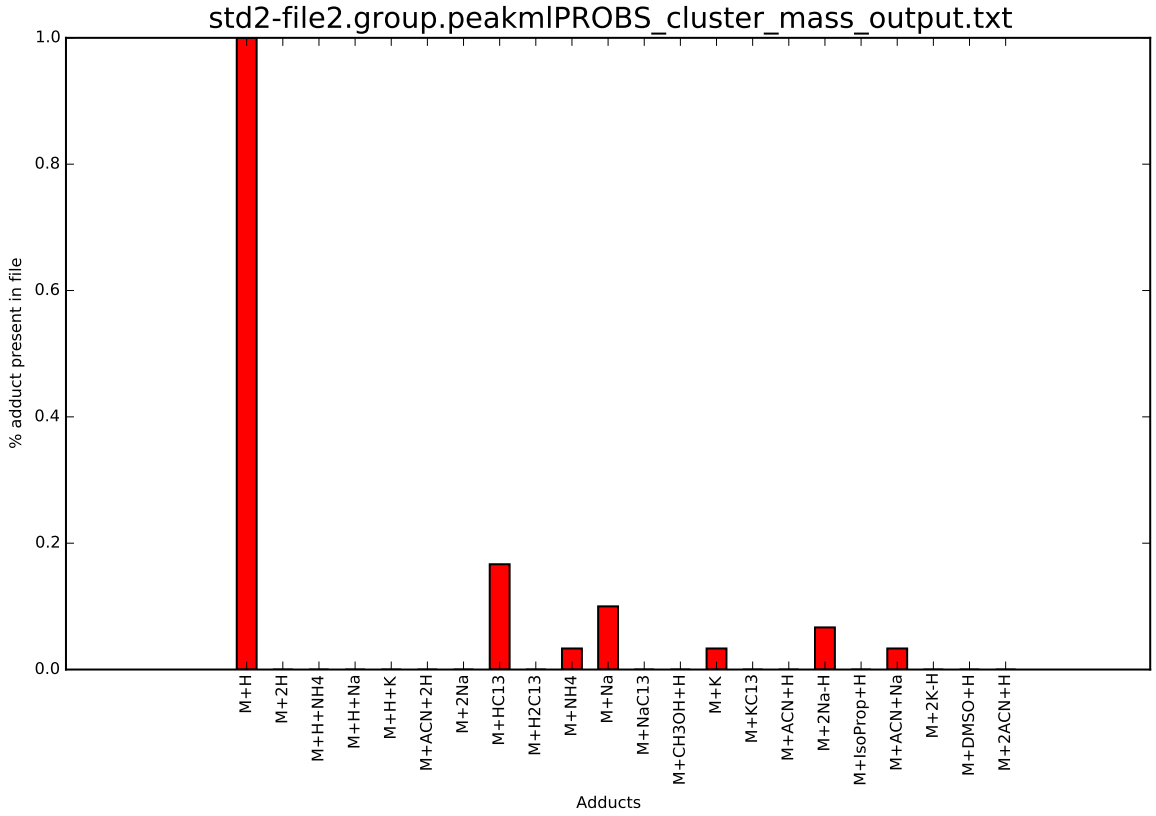


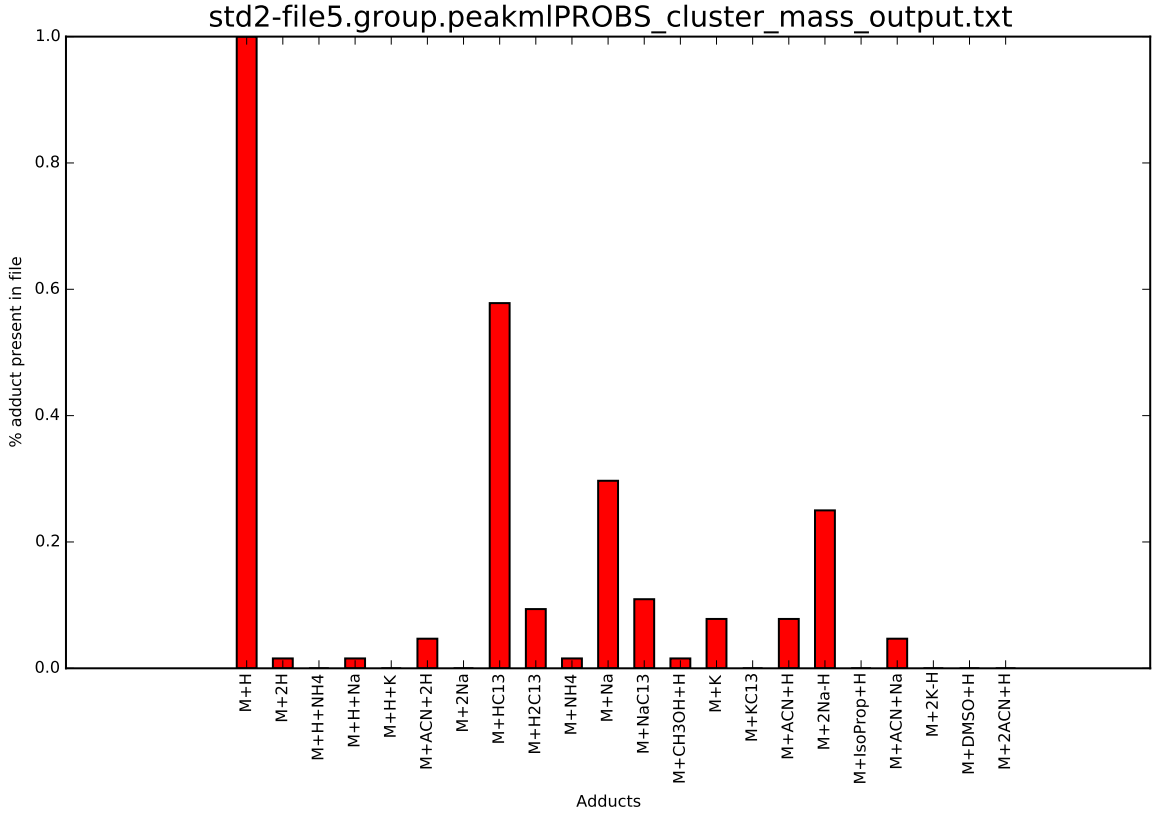

std1-file3.group.peakmlPROBS_cluster_mass_output.txt
% adduct present in file
1.0
0.8
0.6
0.4
0.2
0.0
M+H
M+2H
M+H+NH4
M+H+Na
M+H+K
M+ACN+2H
M+2Na
M+HC13
M+H2C13
M+NH4
M+Na
M+NaC13
M+CH3OH+H
M+K
M+KC13
M+ACN+H
M+2Na-H
M+IsoProp+H
M+ACN+Na
M+2K-H
M+DMSO+H
M+2ACN+H
Adducts


std1-file4.group.peakmlPROBS_cluster_mass_output.txt
% adduct present in file
1.0
0.8
0.6
0.4
0.2
0.0
M+H
M+2H
M+H+NH4
M+H+Na
M+H+K
M+ACN+2H
M+2Na
M+HC13
M+H2C13
M+NH4
M+Na
M+NaC13
M+CH3OH+H
M+K
M+KC13
M+ACN+H
M+2Na-H
M+IsoProp+H
M+ACN+Na
M+2K-H
M+DMSO+H
M+2ACN+H
Adducts

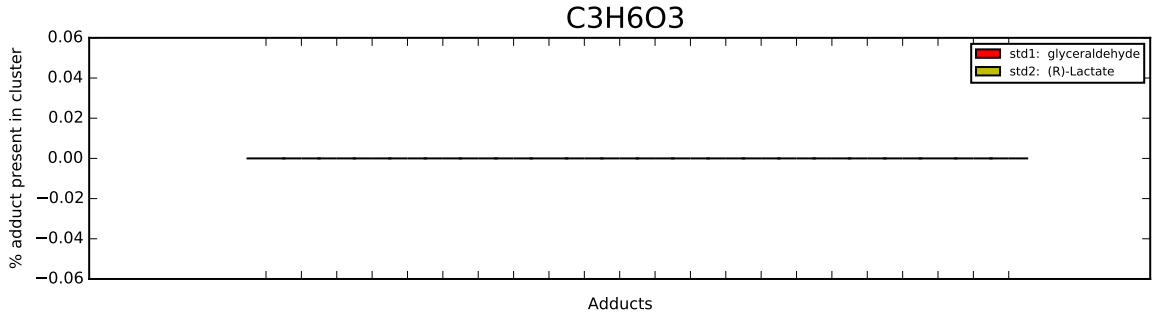

std1-file5.group.peakmlPROBS_cluster_mass_output.txt
% adduct present in file
1.0
0.8
0.6
0.4
0.2
0.0
M+H
M+2H
M+H+NH4
M+H+Na
M+H+K
M+ACN+2H
M+2Na
M+HC13
M+H2C13
M+NH4
M+Na
M+NaC13
M+CH3OH+H
M+K
M+KC13
M+ACN+H
M+2Na-H
M+IsoProp+H
M+ACN+Na
M+2K-H
M+DMSO+H
M+2ACN+H
Adducts

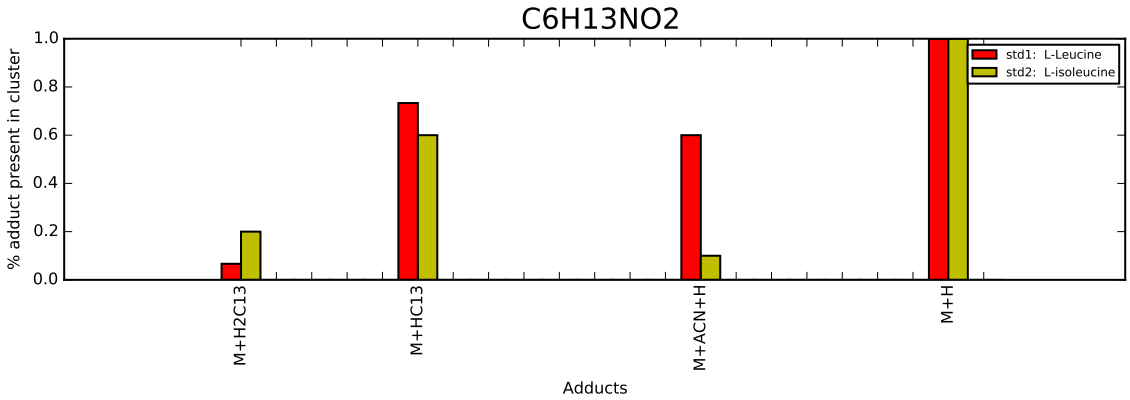

std2-file1.group.peakmlPROBS_cluster_mass_output.txt
% adduct present in file
1.0
0.8
0.6
0.4
0.2
0.0
M+H
M+2H
M+H+NH4
M+H+Na
M+H+K
M+ACN+2H
M+2Na
M+HC13
M+H2C13
M+NH4
M+Na
M+NaC13
M+CH3OH+H
M+K
M+KC13
M+ACN+H
M+2Na-H
M+IsoProp+H
M+ACN+Na
M+2K-H
M+DMSO+H
M+2ACN+H
Adducts

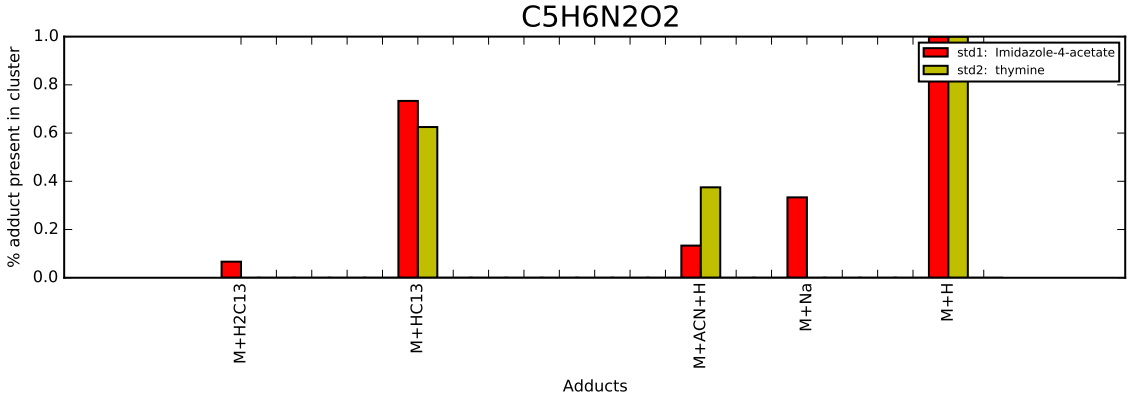

std2-file2.group.peakmlPROBS_cluster_mass_output.txt
% adduct present in file
1.0
0.8
0.6
0.4
0.2
0.0
M+H
M+2H
M+H+NH4
M+H+Na
M+H+K
M+ACN+2H
M+2Na
M+HC13
M+H2C13
M+NH4
M+Na
M+NaC13
M+CH3OH+H
M+K
M+KC13
M+ACN+H
M+2Na-H
M+IsoProp+H
M+ACN+Na
M+2K-H
M+DMSO+H
M+2ACN+H
Adducts

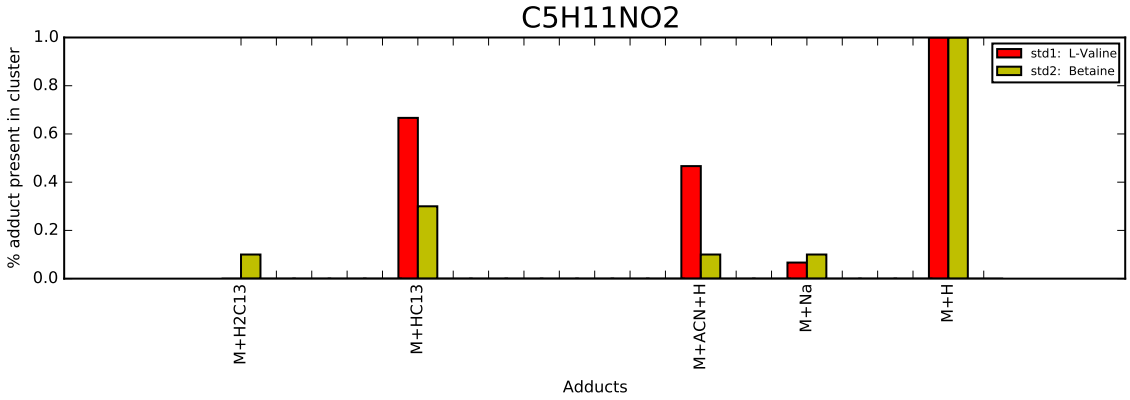

std2-file5.group.peakmlPROBS_cluster_mass_output.txt
% adduct present in file
1.0
0.8
0.6
0.4
0.2
0.0
M+H
M+2H
M+H+NH4
M+H+Na
M+H+K
M+ACN+2H
M+2Na
M+HC13
M+H2C13
M+NH4
M+Na
M+NaC13
M+CH3OH+H
M+K
M+KC13
M+ACN+H
M+2Na-H
M+IsoProp+H
M+ACN+Na
M+2K-H
M+DMSO+H
M+2ACN+H
Adducts

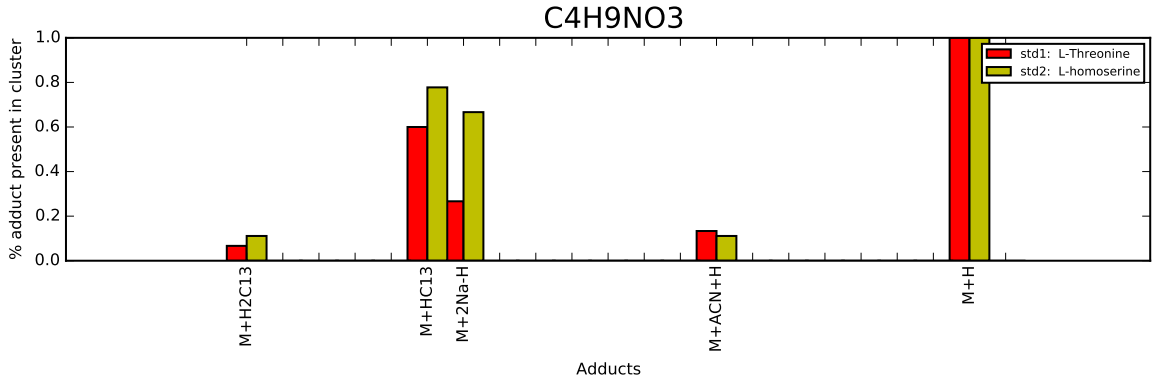
B.2.5 Plots Showing How Frequently Each Adduct is Present in Pairs of Isomers

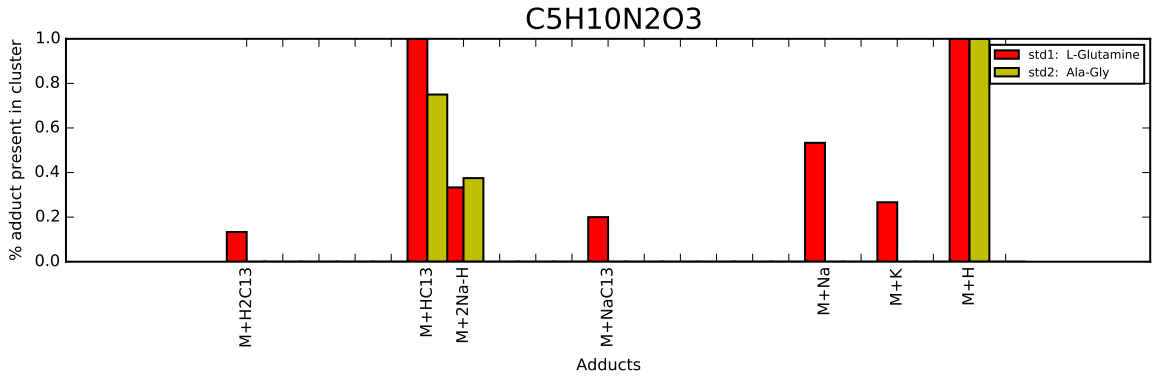


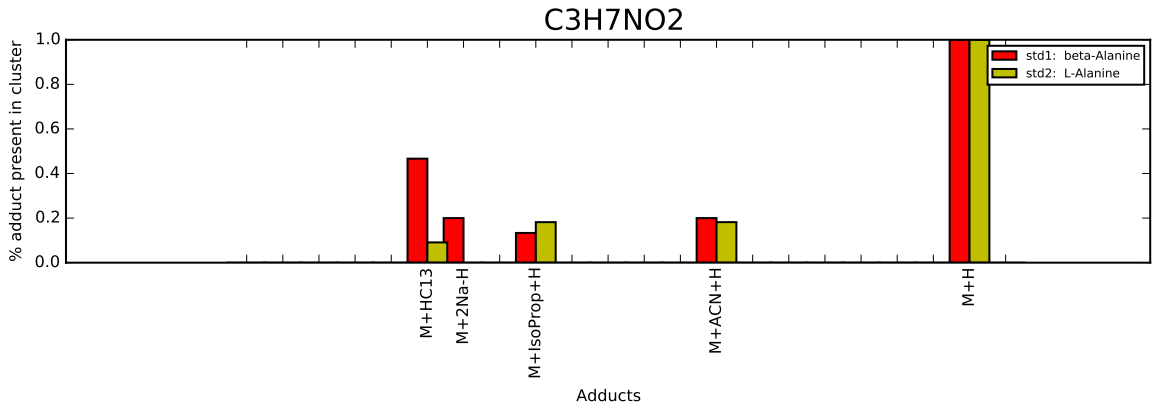


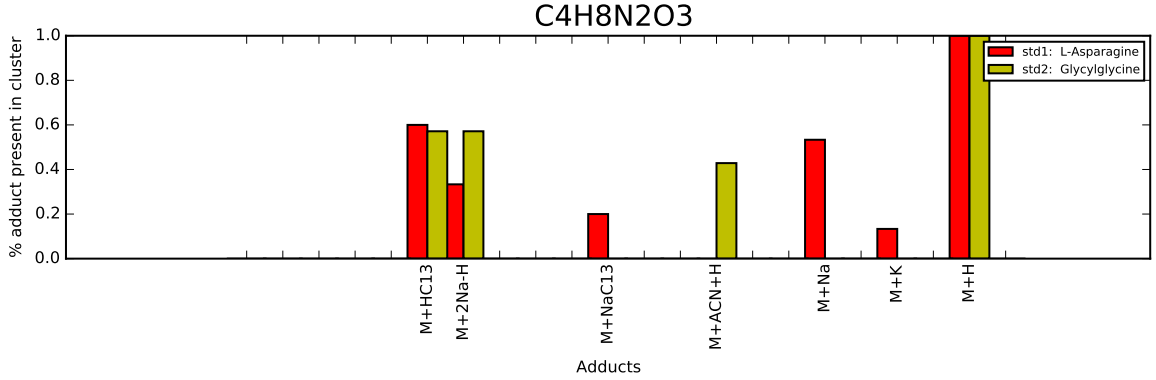


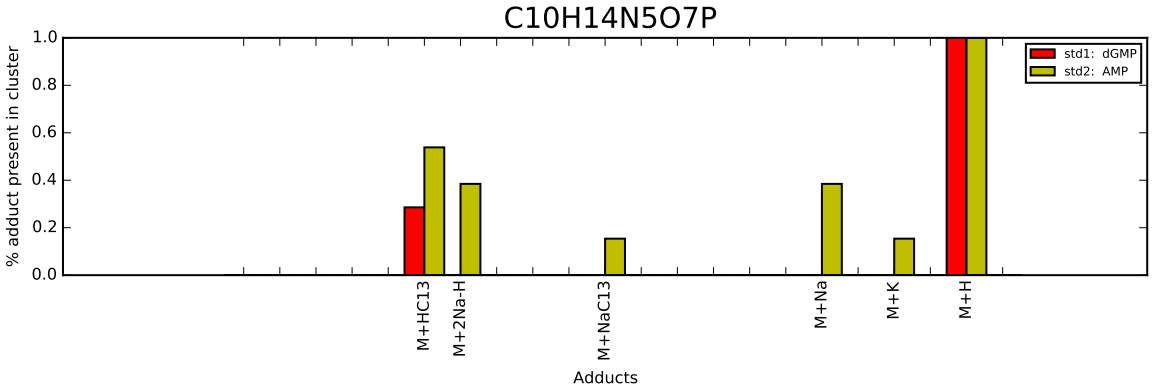


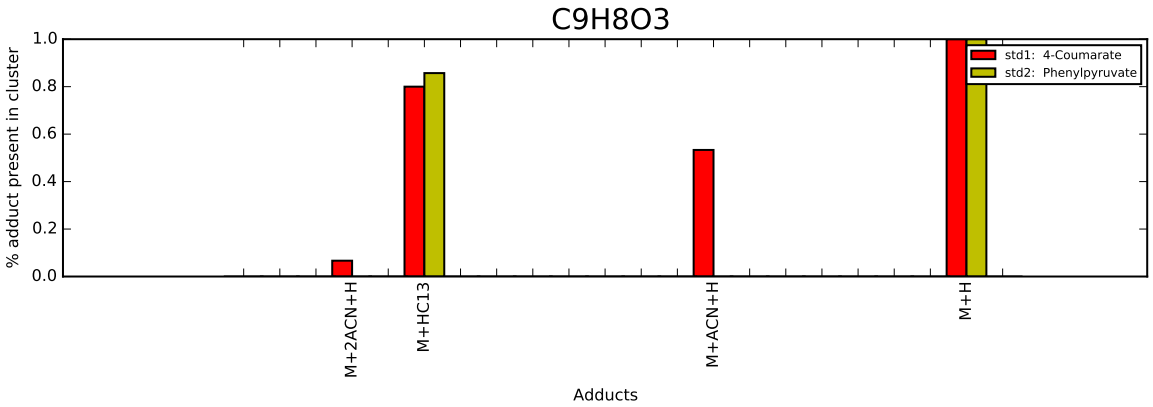

C4H9NO3
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: L-Threonine
std2: L-homoserine
M+H2C13
M+HC13
M+2Na-H
M+ACN+H
M+H
Adducts

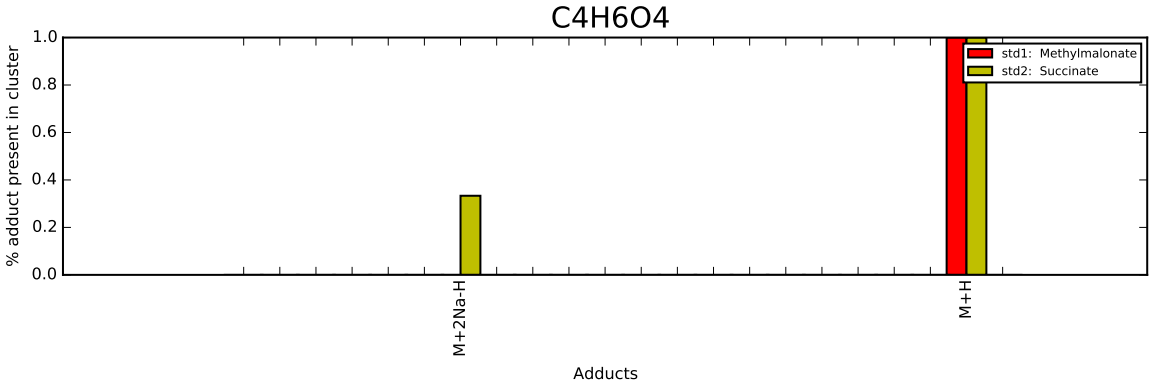

C5H10N2O3
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: L-Glutamine
std2: Ala-Gly
M+H2C13
M+HC13
M+2Na-H
M+NaC13
M+Na
M+K
M+H
Adducts

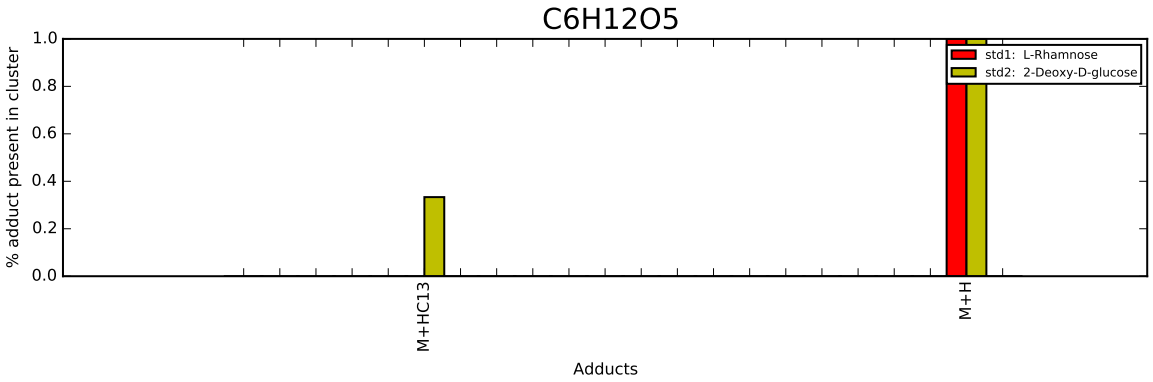

C3H7NO2
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: beta-Alanine
std2: L-Alanine
M+HC13
M+2Na-H
M+IsoProp+H
M+ACN+H
M+H
Adducts

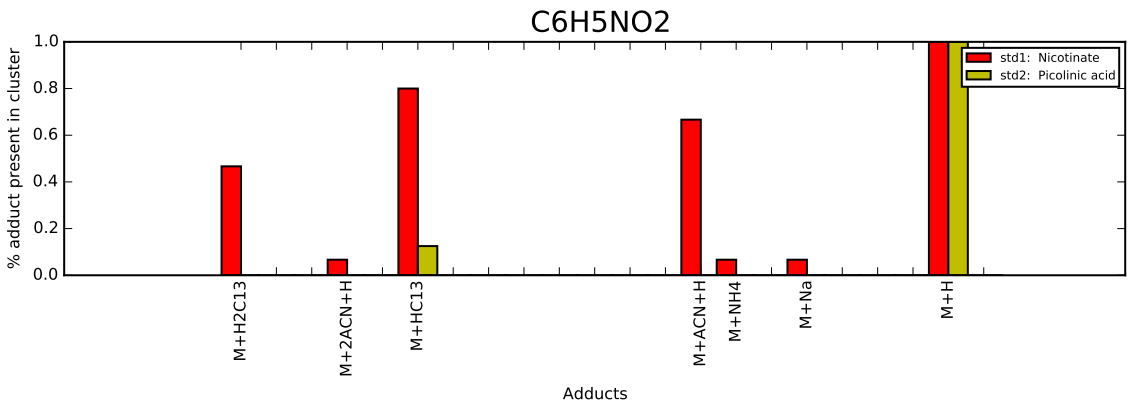

C4H8N2O3
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: L-Asparagine
std2: Glycylglycine
M+HC13
M+2Na-H
M+NaC13
M+ACN+H
M+Na
M+K
M+H
Adducts

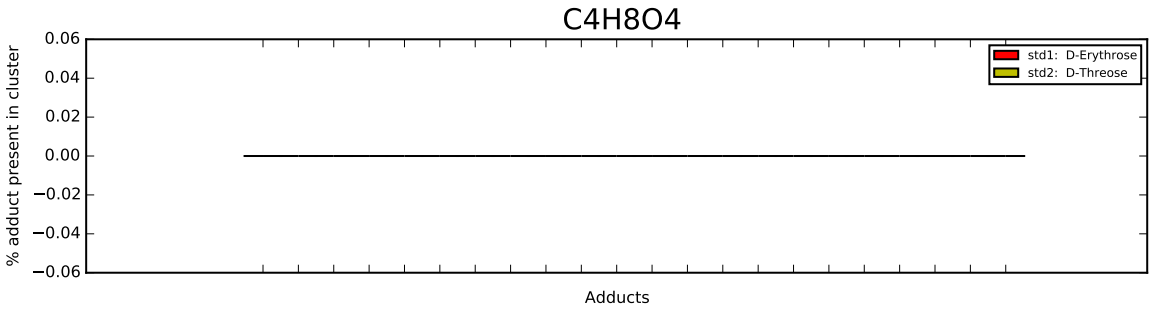

C10H14N5O7P
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: dGMP
std2: AMP
M+HC13
M+2Na-H
M+NaC13
M+Na
M+K
M+H
Adducts

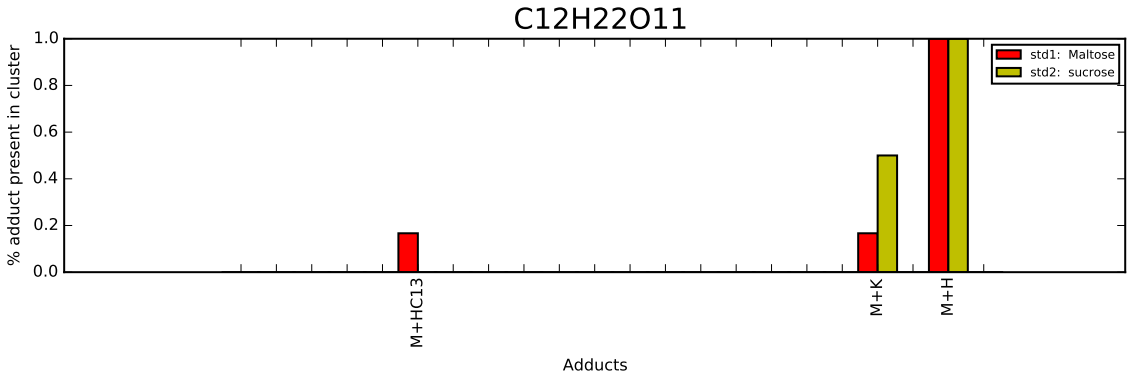

C9H8O3
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: 4-Coumarate
std2: Phenylpyruvate
M+2ACN+H
M+HC13
M+ACN+H
M+H
Adducts

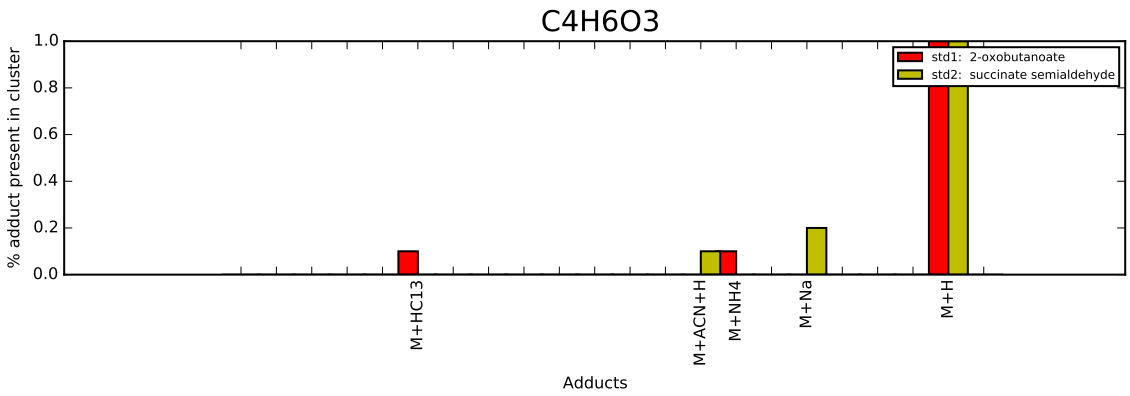

C4H6O4
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: Methylmalonate
std2: Succinate
M+2Na-H
M+H
Adducts


C6H12O5
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: L-Rhamnose
std2: 2-Deoxy-D-glucose
M+HC13
M+H
Adducts

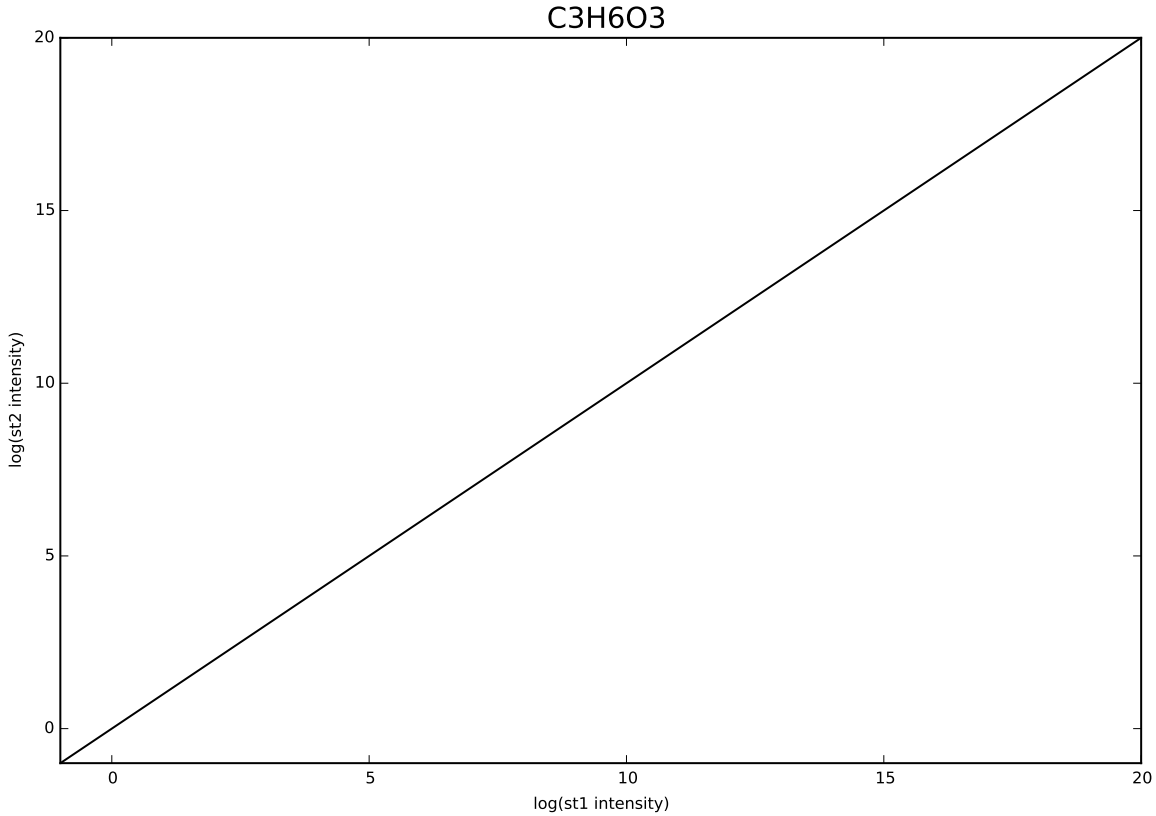

C6H5NO2
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: Nicotinate
std2: Picolinic acid
M+H2C13
M+2ACN+H
M+HC13
M+ACN+H
M+NH4
M+Na
M+H
Adducts

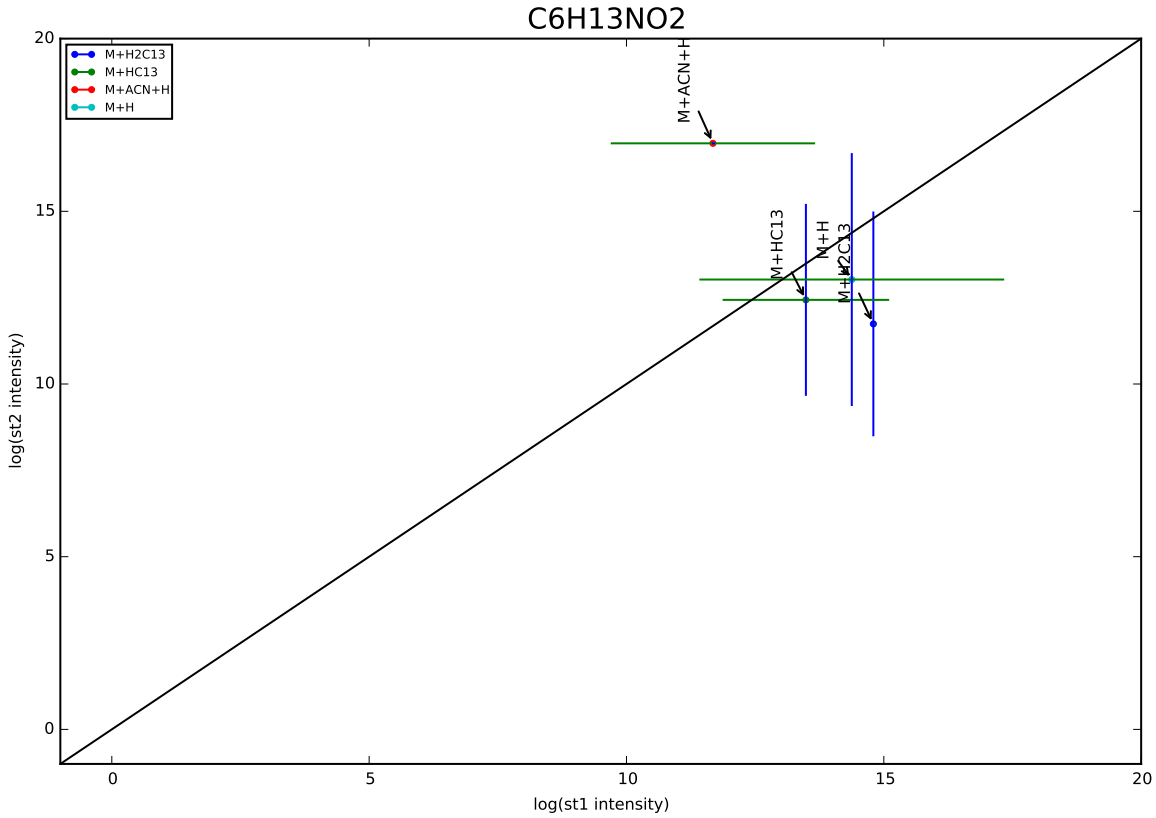

C4H8O4
% adduct present in cluster
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
std1: D-Erythrose
std2: D-Threose
Adducts

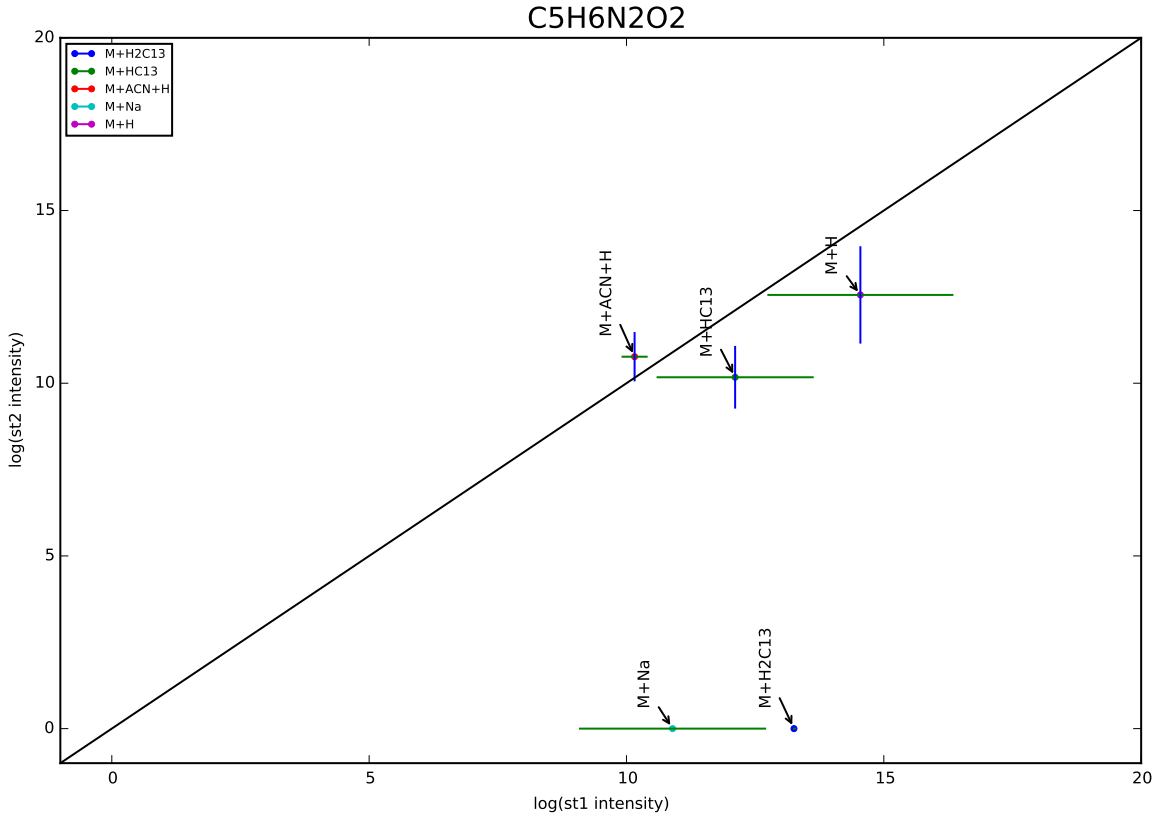

C12H22O11
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: Maltose
std2: sucrose
M+HC13
M+K
M+H
Adducts

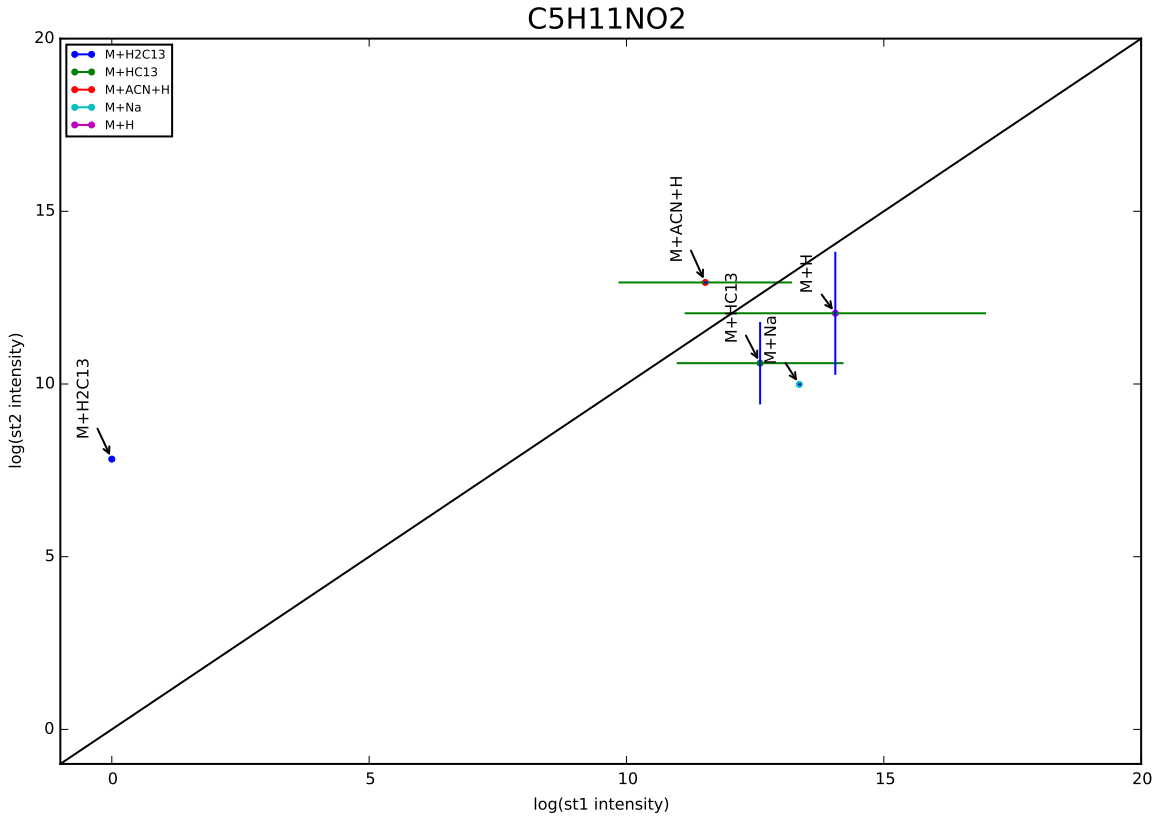

C4H6O3
% adduct present in cluster
1.0
0.8
0.6
0.4
0.2
0.0
std1: 2-oxobutanoate
std2: succinate semialdehyde
M+HC13
M+ACN+H
M+NH4
M+Na
M+H
Adducts

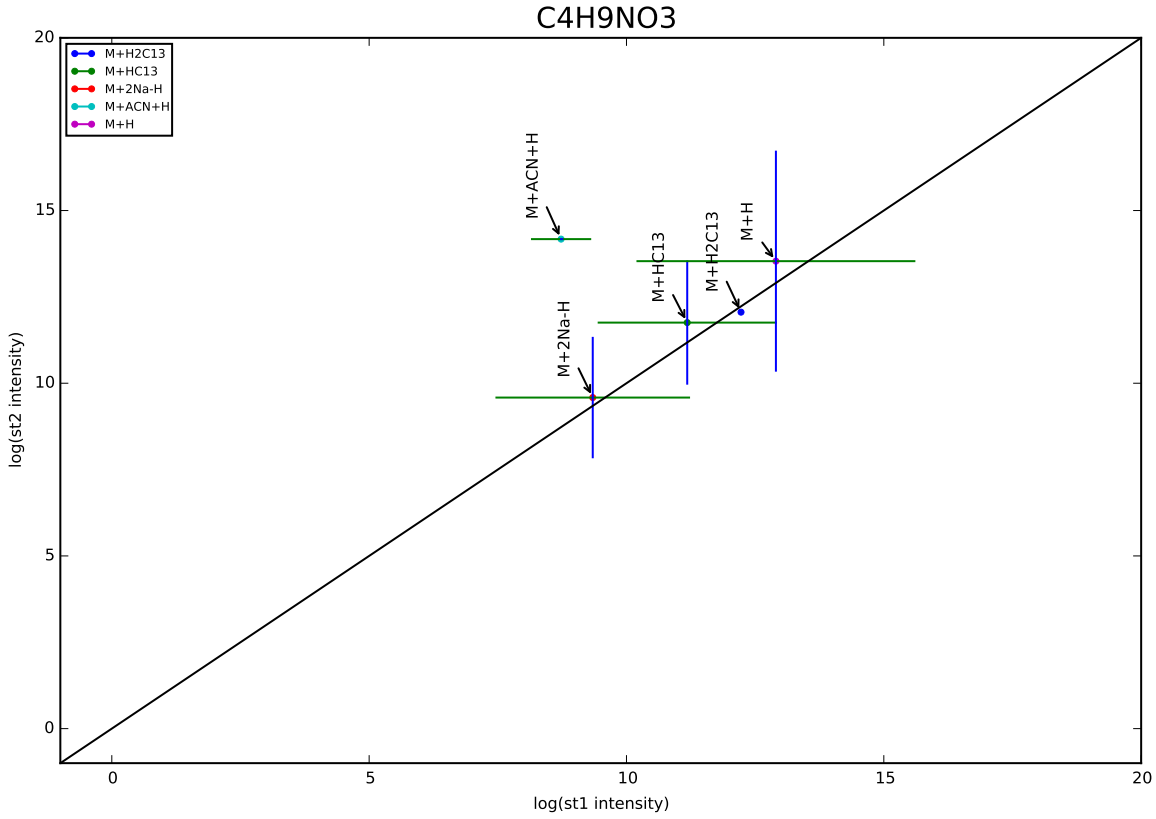
B.2.6 Plots of Isotope's Standard 1 and Standard 2 Adduct Intensities

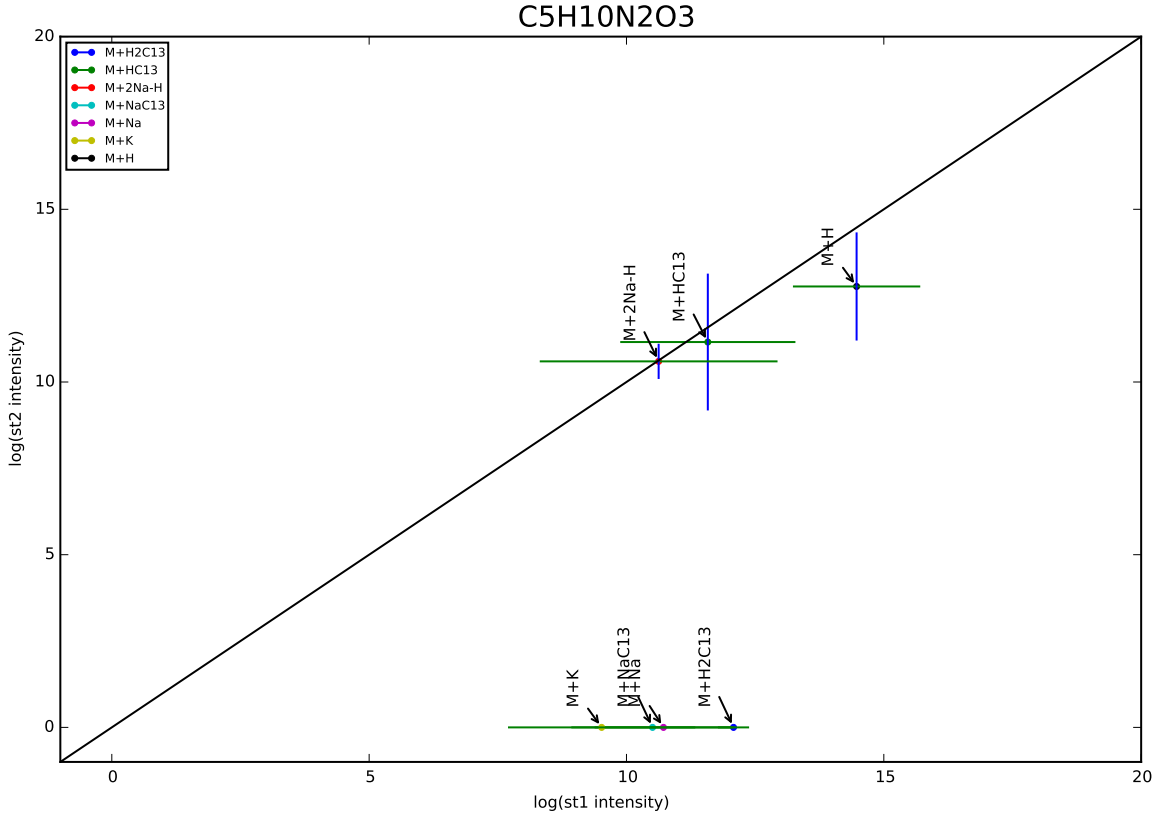


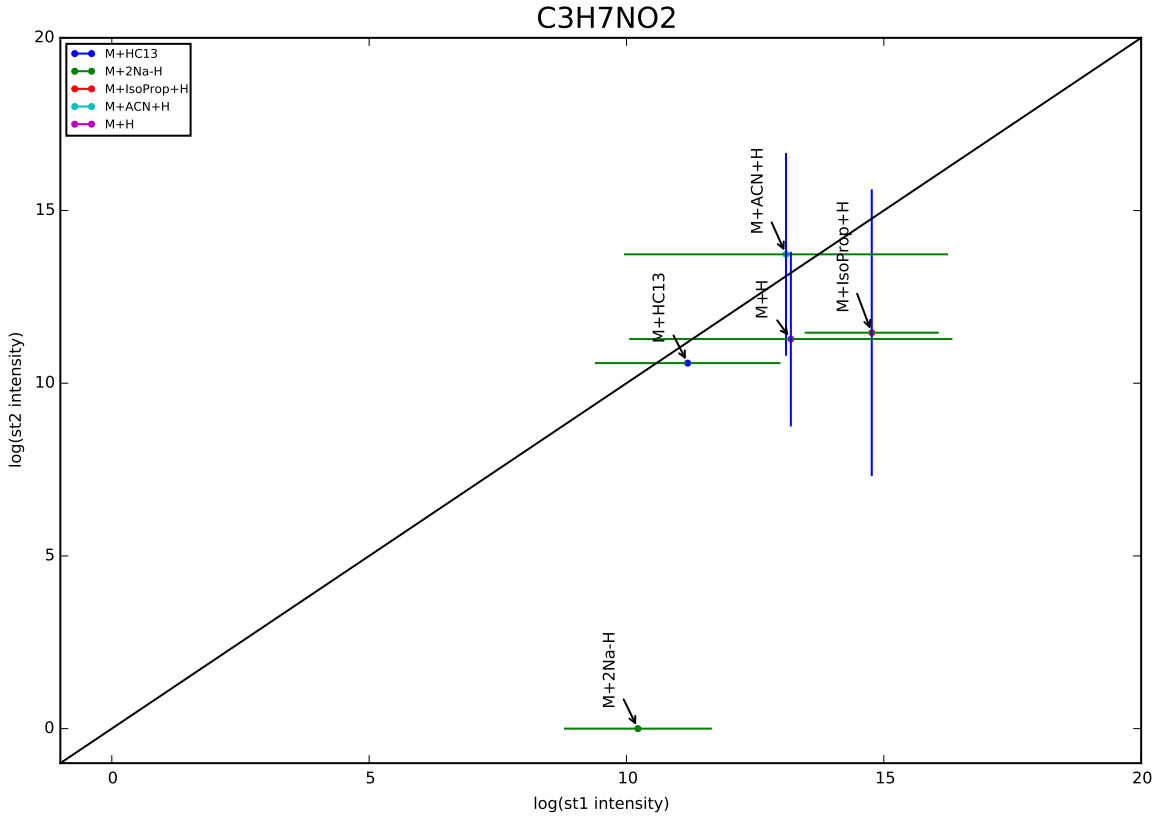


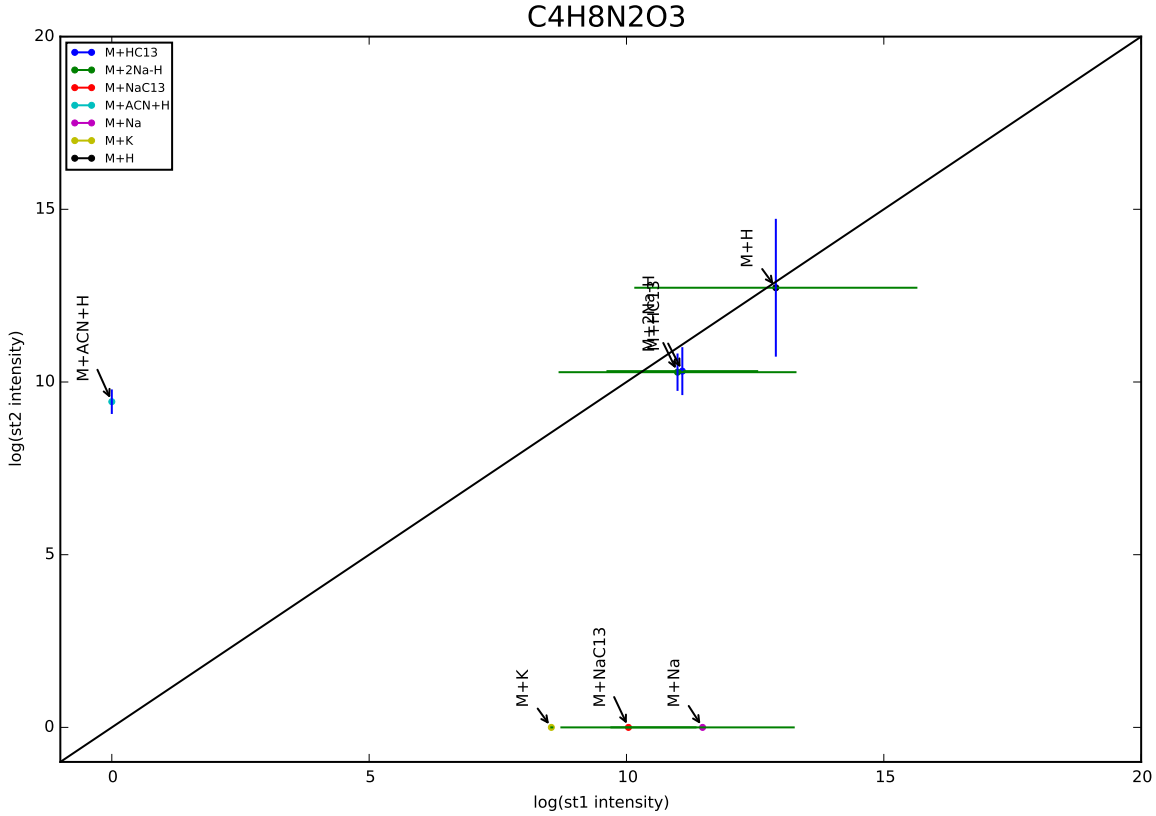

C5H6N2O2
M+H2C13
M+HC13
M+ACN+H
M+Na
M+H
M+ACN+H
M+HC13
M+H
M+Na
M+H2C13
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

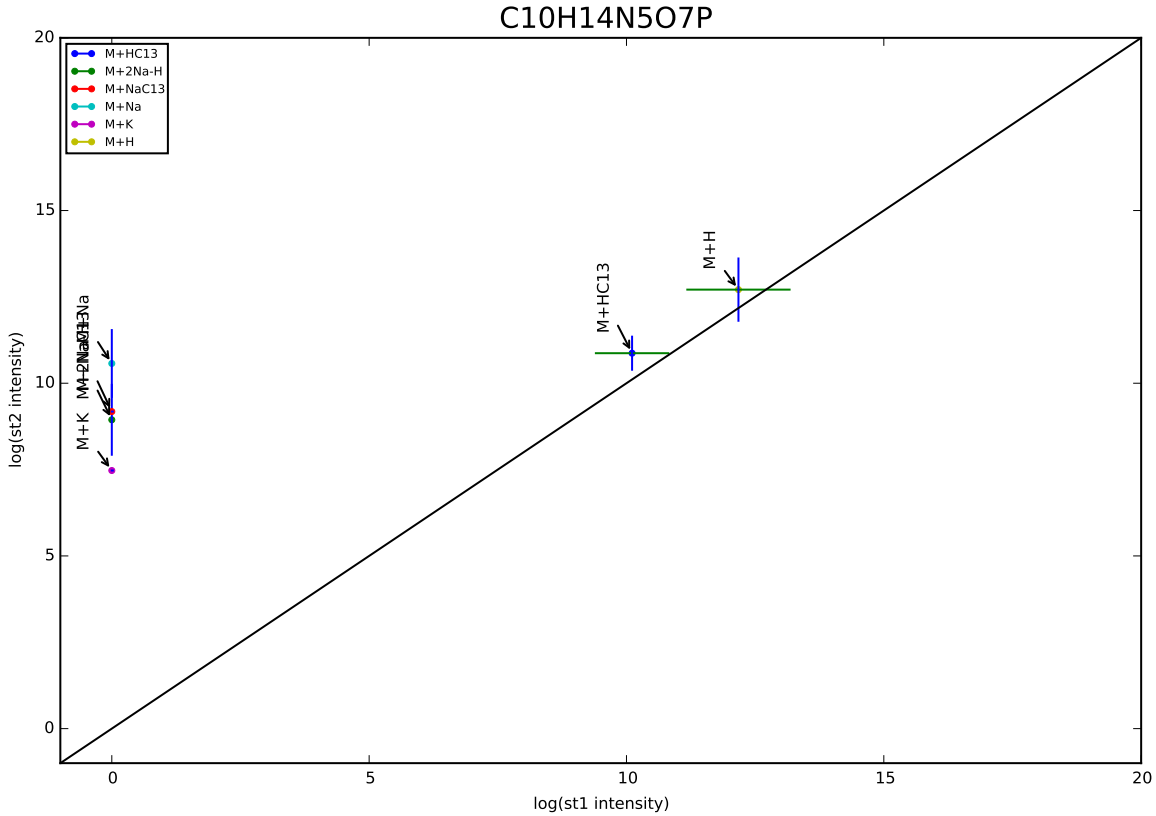

C5H11NO2
M+H2C13
M+HC13
M+ACN+H
M+Na
M+H
M+ACN+H
M+HC13
M+H
M+Na
M+H2C13
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

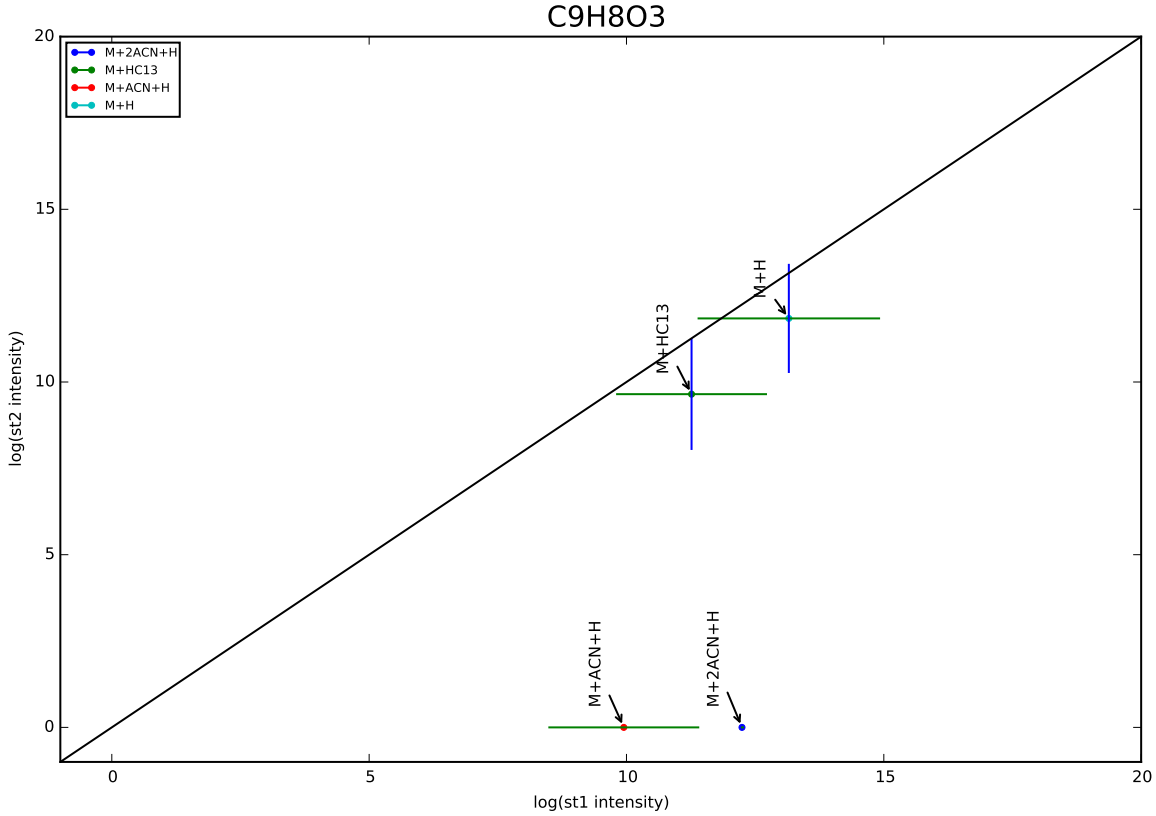

C4H9NO3
M+H2C13
M+HC13
M+2Na-H
M+ACN+H
M+H
M+ACN+H
M+HC13
M+H2C13
M+H
M+2Na-H
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

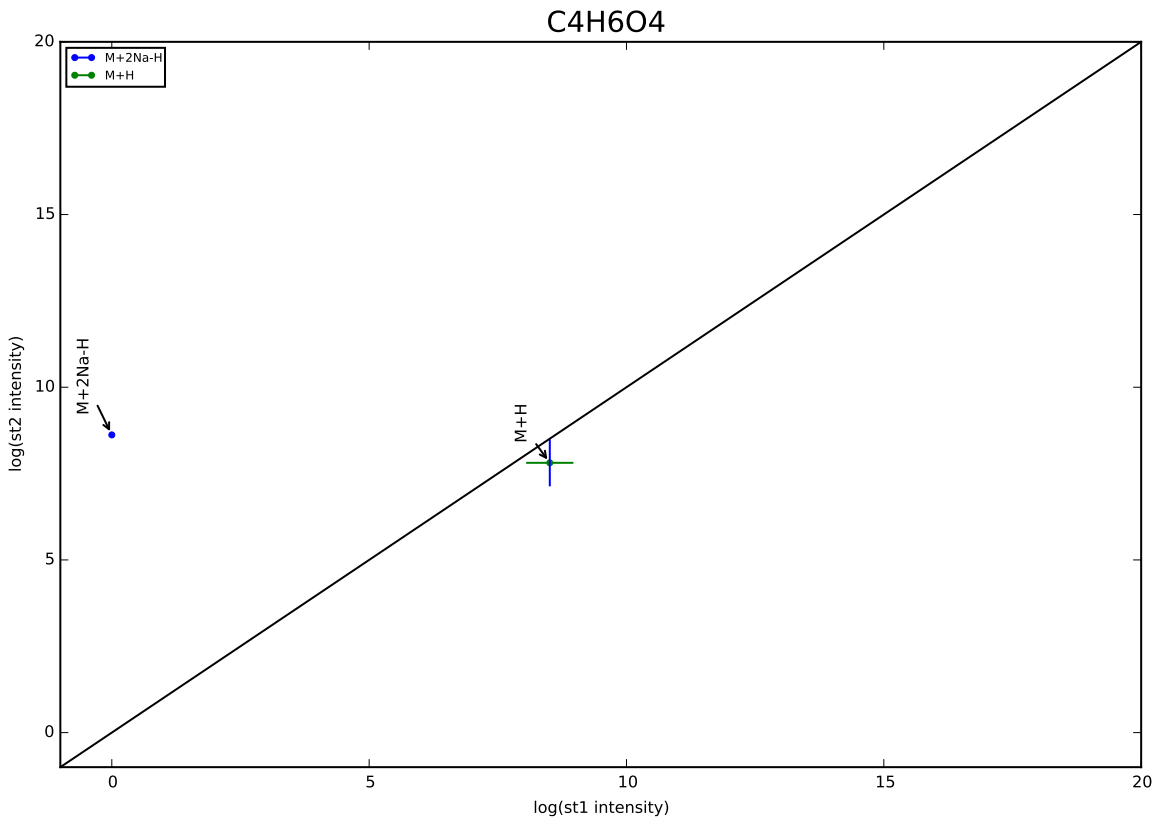

C5H10N2O3
M+H2C13
M+HC13
M+2Na-H
M+NaC13
M+Na
M+K
M+H
M+H
M+2Na-H
M+HC13
M+K
M+NaC13
M+Na
M+H2C13
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

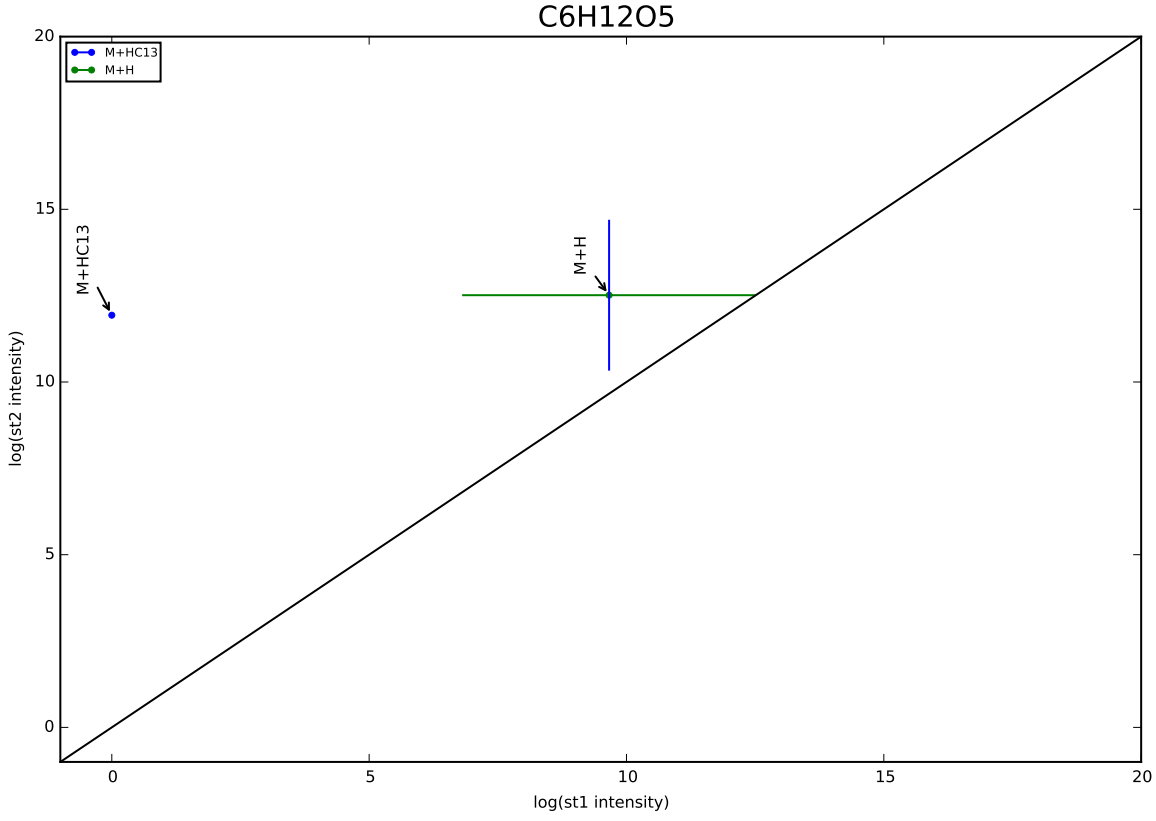

C3H7NO2
M+HC13
M+2Na-H
M+IsoProp+H
M+ACN+H
M+H
M+ACN+H
M+IsoProp+H
M+H
M+HC13
M+2Na-H
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

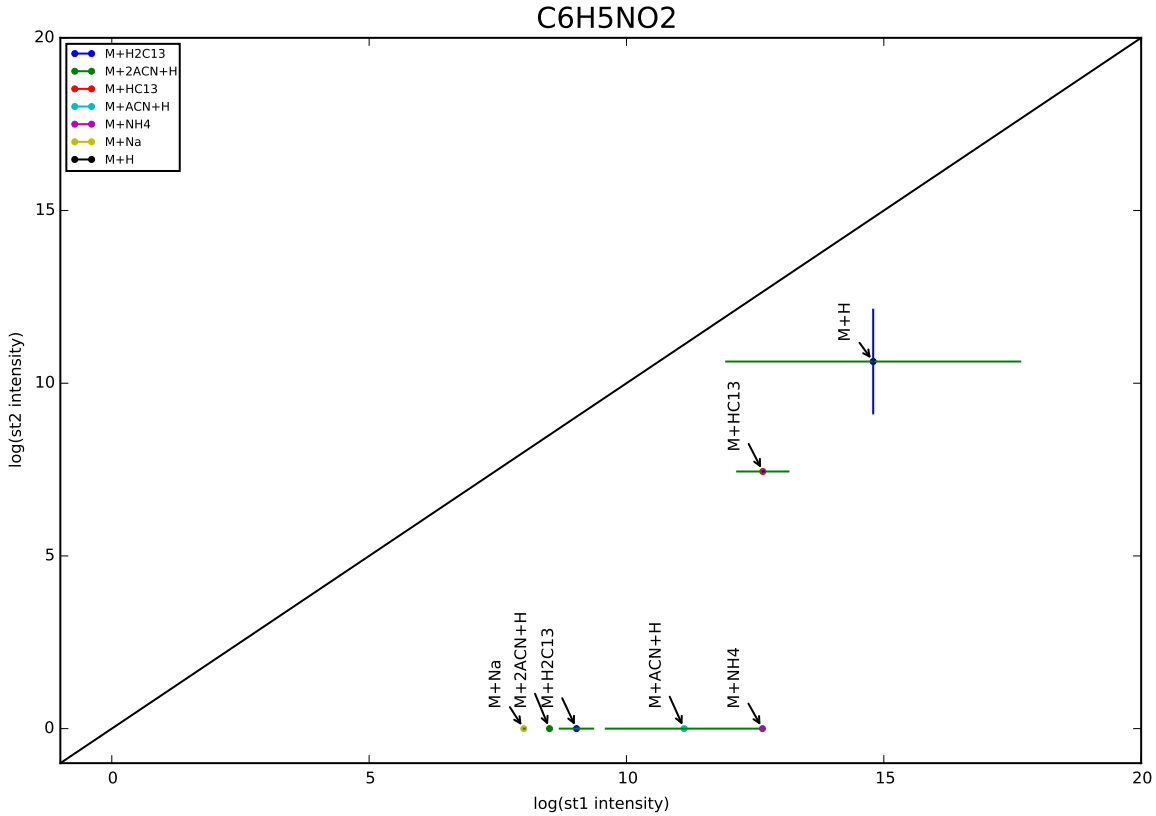

C4H8N2O3
M+HC13
M+2Na-H
M+NaC13
M+ACN+H
M+Na
M+K
M+H
M+ACN+H
M+H
M+2Na-H
M+HC13
M+K
M+NaC13
M+Na
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

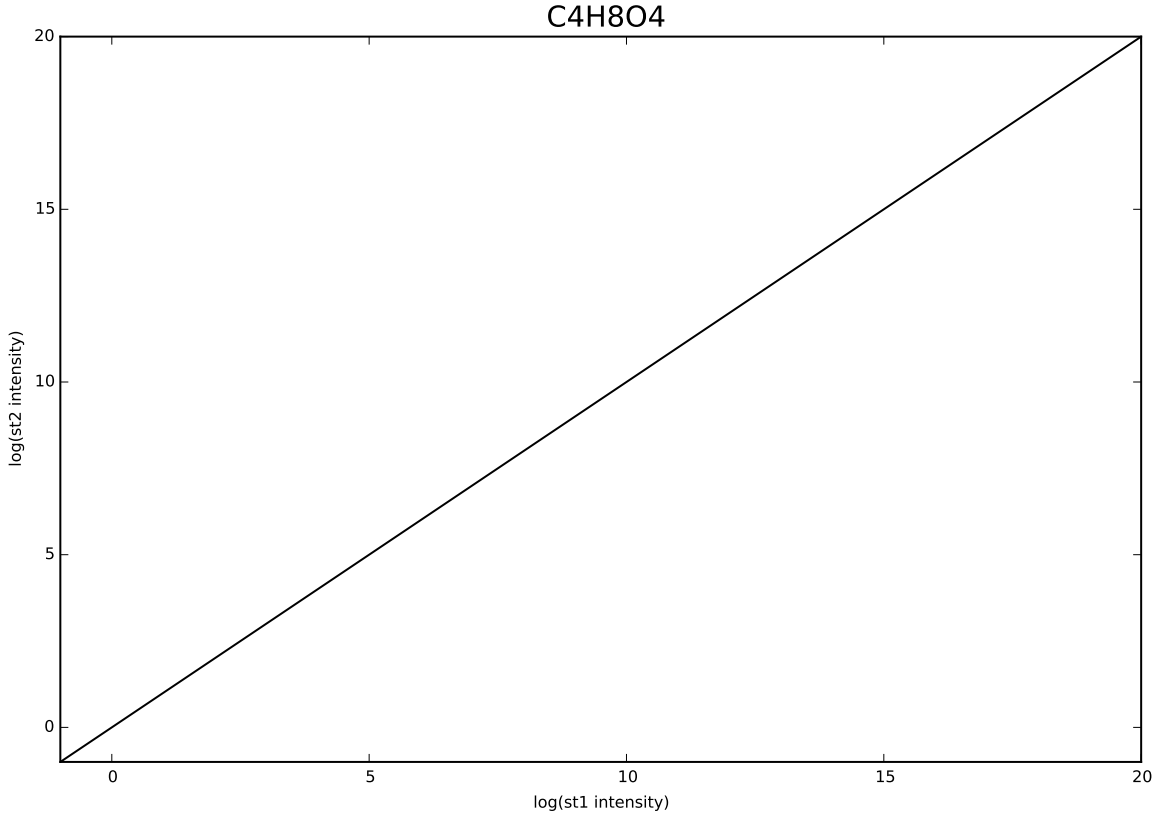

C10H14N5O7P
M+HC13
M+2Na-H
M+NaC13
M+Na
M+K
M+H
M+H
M+HC13
M+K
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

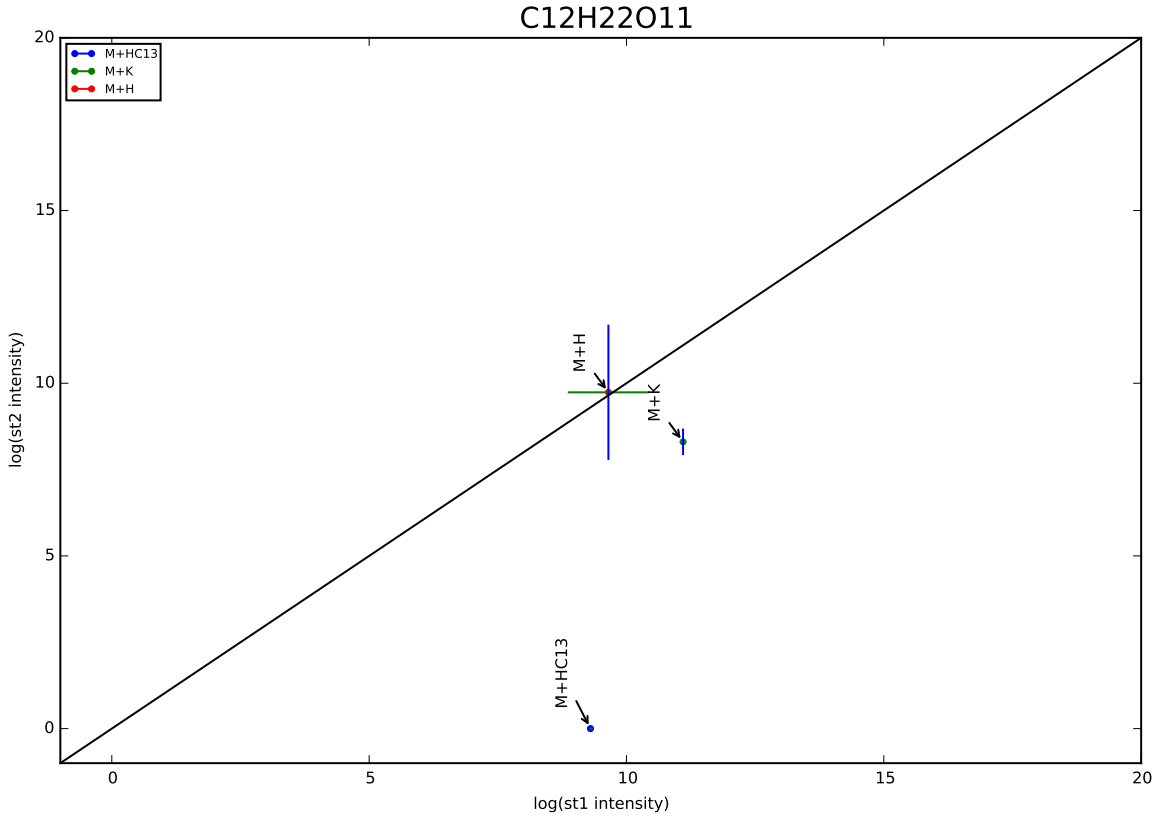

C9H8O3
M+2ACN+H
M+HC13
M+ACN+H
M+H
M+H
M+HC13
M+ACN+H
M+2ACN+H
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

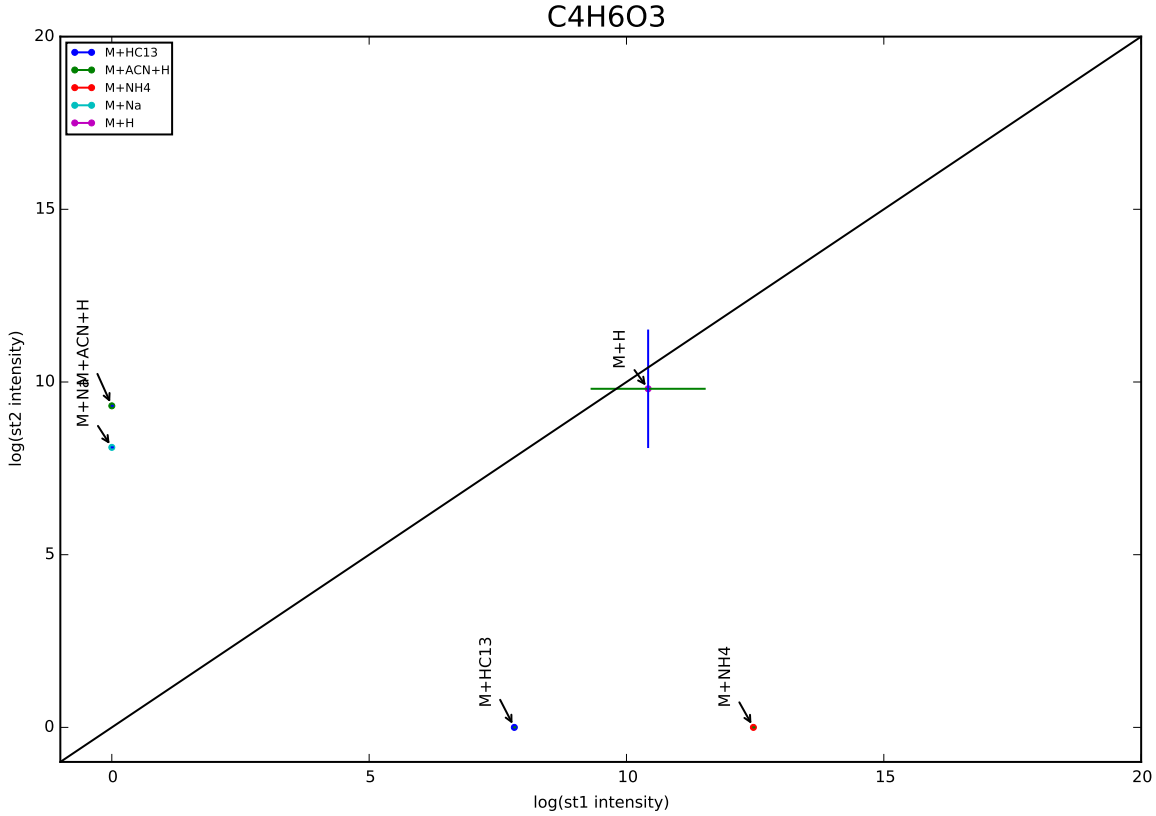

C4H6O4
M+2Na-H
M+H
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20


C6H12O5
M+HC13
M+H
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20


C6H5NO2
M+H2C13
M+2ACN+H
M+HC13
M+ACN+H
M+NH4
M+Na
M+H
M+H
M+HC13
M+Na
M+2ACN+H
M+H2C13
M+ACN+H
M+NH4
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20


C4H8O4
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20


C12H22O11
M+HC13
M+K
M+H
M+H
M+K
M+HC13
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20


C4H6O3
M+HC13
M+ACN+H
M+NH4
M+Na
M+H
M+Na
M+ACN+H
M+H
M+HC13
M+NH4
log(st2 intensity)
log(st1 intensity)
0
5
10
15
20

Bibliography

[1] Ronan Daly Simon Rogers Joe Wandy Andris Jankevics Karl E. V. Burgess and Rainer Breitling. Metassign: probabilistic annotation of metabolites from lcms data using a bayesian clustering approach. *Bioinformatics*, 2014.

[2] Simon Rogers Ronan Daly and Rainer Breitling. Mixture model clustering for peak filtering in metabolomics. *http://www.researchgate.net/*, 2014.

[3] Terk Shuen Lee Ying Swan Ho Hock Chuan Yeo Joyce Pei Yu Lin Dong-Yup Lee. Precursor mass prediction by clustering ionization products in lc-ms-based metabolomics. *Metabolomics*, 2013.

[4] Rob Smith Andrew D. Mathis Dan Ventura John T Prince. Proteomics, lipidomics, metabolomics: a mass spectrometry tutorial from a computer scientists point of view. In *The 10th Annual Biotechnology and Bioinformatics Symposium*, 2013.

[5] Sheldon M. Ross. *Introduction to Probability Models*. Elsevier, 2007.

[6] Andrew Gelman John B. Carlin Hal S. Stern Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2004.

[7] Mark Girolami Simon Rogers. *A First Course in Machine Learning*. Chapman and Hall/CRC, 2011.

[8] O. David Sparkman. *Mass Spectrometry Desk Reference*. Pittsburgh: Global View Pub., 2000.

[9] Eric W. Weisstein. Digamma function. *From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/DigammaFunction.html*.