# Discrete Adduct Clustering Model for MS Data

February 9, 2015

## 1 Introduction

To annotate precursor masses of peak features in an input file against some database of molecules, we have to do some crude form of discretisation anyway (when matching the mass values). In this report, we introduce the discretisation process early early on.

## 2 Discretisation

We can bin the data along the mass dimension. We index peak features by $n = 1, ..., N$ and the bins (i.e. precursor mass clusters) by $k = 1, ..., K$.. Given any peak feature $n$ with observed mass $m_n$, we want to compute the precursor mass $q_k$, associated to a cluster $k$, by applying the $M + H$ adduct inverse-transformation:

$$q_k = \frac{m_n|c| + ce - \sum_i h_i G_i}{n} \tag{1}$$

where $c$ is the charge, $e$ is the mass of the electron, $h_i$ and $G_i$ are the adduct parts. Specifically for the M+H adduct, $n = 1.0$ and $ce - \sum_i h_i G_i = 1.00727645199076$. Further details can be found in Adduct_notes.pdf.

Given $q_k$, we then create a mass bin centered at $q_k$

$$q_k \pm b_k \tag{2}$$

where $b_k = q_k \cdot w \cdot (1e - 6)$ is the bin width The value of $w$ is specified by the user. This process is repeated for all $N$ observed peak features, resulting in $K$ bins, where $N = K$.

Each bin now corresponds to a 'valid' potential assignment of an observed peak to a precursor mass – based on the crucial assumption that an acceptable precursor mass clustering should have the $M + H$ adduct peak inside. Note that in this binning scheme, there are usually overlapping bins. It's a good idea to

explore other binning approach, e.g. non-overlapping bin at fixed interval, etc, and see if this affects the results (maybe not?).

Note from email:

- How do you discretise? We need to ensure that everything that can be transformed to a particular precursor mass is in the same bin. It should make discretisation actually pretty easy. In fact, you could after doing the transformations in a continuous space... this would not necessarily work once we go to multiple files...

# 3 Model

Denote the peak feature by $d_n = (x_n, y_n)$ where $x_n$ is the mass value and $y_n$ the RT value. We use the variable $z_n = k$ to denote the assignment of peak feature $n$ to bin $k$.

Given the data, we want to infer the assignment of the $z_n$ variables to the precursor mass clusters. Assume a fixed number of clusters based on the known number of 'valid' precursor masses $K$, each $z_1, ..., z_n$ is therefore a categorical variable independently drawn from a categorical distribution with parameter $\boldsymbol{\theta}$. In turn, the parameter vector $\boldsymbol{\theta}$ of length $K$ is drawn from a Dirichlet distribution with parameter $\alpha$. So, the likelihood of a peak $n$ to be assigned into a cluster $k$ depends on **(1)** whether there's a possible transformation from the observed mass $x_n$ to any precursor mass $q_k$ given the list of adducts, and **(2)** based on the RT values. The model is therefore

$$\begin{aligned} \boldsymbol{\theta} &\sim Dir(\alpha) && (3)\\ z_n = k &\sim Cat(\boldsymbol{\theta}) && (4)\\ d_n &\sim L(d_n | z_n = k, ...) && (5) \end{aligned}$$

The likelihood $L(d_n|z_n = k, ...)$ can be factorised into its mass and RT terms

$$L(d_n|z_n = k, ...) = p(x_n|z_n = k, ...) \cdot p(y_n|z_n = k, ...) \quad (6)$$

For the mass term $p(x_n|z_n = k, ...)$, let $I_k(x_n)$ to be the indicator function defined as

$$I_k(x_n) \begin{cases} 1 & \text{there is an adduct transformation from } x_n \text{ to bin k} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

In short, $I_k(x_n) = 1$ if there is an adduct transformation that lets us reach the cluster $k$ from $x_n$ ($q_k - b_k \leq x_n \leq q_k + b_k$ for a precursor mass $q_k$ and its interval $b_k$). For each $x_n$, define $k^*$ to be the set of all such valid transformations. Then the mass term is

$$p(x_n|z_n = k, ...) = \frac{1}{|k^*|} \sum_k I_k(x_n) \tag{8}$$

For the RT term $p(y_n|z_n = k, ...)$, $y_n$ is normally distributed with mean $\mu_k$ and some precision (inverse variance) $\delta$. The cluster mean $\mu_k$ is in turn drawn from another normal distribution with mean $\mu_0$ and precision $\tau_0$

$$\begin{aligned} p(y_n|z_n = k, \mu_k, \delta, ...) &= \mathcal{N}(y_n|\mu_k, \delta^{-1}) & (9) \\ p(\mu_k|\mu_0, \tau_0) &= \mathcal{N}(\mu_k|\mu_0, \tau_0^{-1}) & (10) \end{aligned}$$

For Gibbs sampling, we need the conditional distribution

$$\begin{aligned} p(z_n = k|\boldsymbol{\theta}, ...) &\propto (\alpha_k + z_k) \cdot L(d_n|z_n = k) & (11) \\ &= (\alpha_k + z_k) \cdot p(x_n|z_n = k, ...) \cdot p(y_n|z_n = k, ...) & (12) \end{aligned}$$

where $p(x_n|z_n = k, ...)$ is as defined in eq. (8). For the RT term, we marginalise over all values of $\mu_k$ and get:

$$p(y_n|z_n = k, ...) = \mathcal{N}(y_n|\beta_k, \lambda_k^{-1}) \tag{13}$$

where $\lambda_k = ((\tau_0 + \sigma c_k)^{-1} + \delta^{-1})^{-1}$ and $\beta_k = \frac{1}{\lambda_k}[(\mu_0\tau_0) + (\delta \sum_n y_{n \in k})]$. Here, $y_{n \in k}$ denotes the RT values of any peak $n$ currently assigned to cluster $k$, and $c_k$ the count of such peaks.

Full derivations to follow in later sections.

# 4 Results

Results can be found in the Python notebook at `http://nbviewer.ipython.org/github/sdrogers/metabolomics_tools/blob/master/discretisation/notebooks/Test_discrete_clustering.ipynb`

- To make the model extendable to isotopes, you need the final column from the mulsub file too. In fact, remind me to give you mulsub2 which is the one I've been using, which has the various isotopes in as adducts. The file you have has only zeros in the final column, which is why you've ignored it, I guess.
- We should settle on a way of storing peaks and files and both use it. Yours is probably neater than mine.
- We should settle on the output we want from the clustering models, and both use it too. Perhaps this should be discussed tomorrow.

# 5 Derivations

Some standard derivations for Dirichlet-categorical model.

The joint distribution of the data is

$$p(z_1, ..., z_n, \boldsymbol{\theta}, \alpha) = \prod_n p(z_n = k|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\alpha) \tag{14}$$

$$p(z_n = k|\boldsymbol{\theta}) = Cat(\boldsymbol{\theta}) = \prod_k (\boldsymbol{\theta}_k)^{z_k} \tag{15}$$

$$p(\boldsymbol{\theta}|\alpha) = Dir(\alpha) = \frac{1}{B(\alpha)} \prod_k (\boldsymbol{\theta}_k)^{\alpha_k - 1} \tag{16}$$

For Gibbs sampling, we need $p(z_n|\boldsymbol{\theta}, ...)$. This is

$$p(z_n = k|\boldsymbol{\theta}, ...) \propto p(z_n = k, \boldsymbol{\theta}, ...) \tag{17}$$

$$= p(z_n = k|\boldsymbol{\theta}, ...) \cdot p(\boldsymbol{\theta}|\alpha) \tag{18}$$

$$= \int [p(z_n = k|\boldsymbol{\theta}, ...) \cdot p(\boldsymbol{\theta}|\alpha)] \, d\theta \tag{19}$$

$$= \int \left[ \prod_k (\boldsymbol{\theta}_k)^{z_k} \cdot \frac{1}{B(\alpha)} \prod_k (\boldsymbol{\theta}_k)^{\alpha_k - 1} \right] d\theta \tag{20}$$

$$= \frac{1}{B(\alpha)} \int \left[ \prod_k (\boldsymbol{\theta}_k)^{z_k + \alpha_k - 1} \right] d\theta \tag{21}$$

$$= \frac{B(\alpha + z)}{B(\alpha)} \tag{22}$$

Eq (21) is obtained as such: we know

$$B(\alpha) = \int \left[ \prod_k (\boldsymbol{\theta}_k)^{z_k + \alpha_k - 1} \right] d\theta \tag{23}$$

as it's the normalising constant in the multinomial pdf, so

$$B(\alpha + z) = \int \left[ \prod_k (\boldsymbol{\theta}_k)^{z_k + \alpha_k - 1} \right] d\theta \tag{24}$$

Continuing from eq (22), the beta function is defined as $B(\alpha) = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$.

Expand everything, and we get

$$
\begin{aligned}
p(z_n = k|\boldsymbol{\theta}, ...) &\propto \left(\frac{\prod_k \Gamma(\alpha_k + z_k)}{\Gamma(\sum_k \alpha_k + z_k)}\right)\left(\frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)}\right) && (25)\\
&= \left(\frac{\Gamma(\sum_k \alpha_k)}{\Gamma(\sum_k \alpha_k + z_k)}\right)\left(\frac{\prod_k \Gamma(\alpha_k + z_k)}{\prod_k \Gamma(\alpha_k)}\right) && (26)\\
&\propto \prod_k \frac{\Gamma(\alpha_k + z_k)}{\Gamma(\alpha_k)} && (27)\\
&\propto \prod_k \Gamma(\alpha_k + z_k) && (28)\\
&\propto ...? && (29)\\
&\propto \alpha_k + z_k && (30)
\end{aligned}
$$

# References