HDP for Alignment of Multiple LC-MS Data in Metabolomics

May 23, 2014

1 Introduction

Peak alignment is an important step in the pre-processing stages of LC-MS data. Errors produced during alignment will have a potentially significant impact in the later analysis stage (e.g. when performing differential analysis on the abundance of metabolites of interests). Existing methods do not take into account the dependencies of related peaks during alignments. We do that here by proposing a hierarchical Bayesian model that assigns related peaks to metabolite within and across files.

2 Hierarchical Dirichlet Process (HDP) Mixture Model

Add description about mixture model, DP, HDP, Chinese Restaurant Franchise and the stick breaking stuff

See [2] too.

3 HDP Model - RT Only

Metabolites generates multiple clusters across files. In turn, clusters generates multiple peaks within files. We can model this process with a hierarchical mixture model of Gaussian components. Within each file, related peaks are grouped together. These groups are shared across files through top-level latent variables that represents the metabolites. We use Dirichlet Process (DP) prior to avoid setting the number of clusters a priori. The overall model fits into the Hierarchical Dirichlet Process (HDP) framework [6].

The observed data is the peak RT values $\mathbf{d} = (x_n^j)$ for all peaks across files. Within each file j, the peak RT value x_n^j is drawn from its parent cluster's RT t_k^j, which is normally distributed.

$$x_n^j | t_k^j, \gamma \sim \mathcal{N}(t_k^j, \gamma^{-1}) \tag{1}$$

The cluster RT t_{ij} in file j is drawn from the metabolite RT t_i that is shared across all files.

$$t_k^j | t_i, \delta \sim \mathcal{N}(t_i, \delta^{-1}) \tag{2}$$

The metabolite RT t_i is drawn from a base distribution of predicted retention times of metabolites. This is a Gaussian for now.

$$t_i | \mu_0, \sigma_0 \sim \mathcal{N}(\mu_0, \sigma_0^{-1}) \tag{3}$$

3.1 Modelling Assumptions

Some current assumptions to think about in the model:

1. Every peak must be generated by a metabolite. How about signals that are just noise (not coming from any metabolite) ?
2. Top-level component (metabolites) can produce multiple clusters in the same file. This seems to be a reasonable assumption ?
3. Not considering the mass information yet (including adducts, isotope etc) ..
4. We model RT drifts across replicates using the clusters (which are basically the noisy realisation of metabolite in each file). Assume clusters are IID given the parent metabolite, i.e. no correlation in the drift over time.

3.2 Inference

Denote the assignment of peak n to cluster k in file j by $z_{jnk} = 1$. The matrix $\boldsymbol{Z}$ contains this assignment for all indices j,k and n in the model. We also need another indicator value to set the assignment of clusters within each file to the top-level component (metabolite) shared across files. Denote by $v_{jki} = 1$ if cluster k in file j is assigned to metabolite i. Collectively, we store the assignments in the matrix $\boldsymbol{V}$ for all i, j and k.

Given the observed data $\mathbf{x}$, our task is the infer the parameters $\boldsymbol{\theta} = (\boldsymbol{t}, \boldsymbol{t}^j, \boldsymbol{Z}, \boldsymbol{V})$. We use Gibbs sampling to sample from the posterior distribution $P(\boldsymbol{\theta}|\boldsymbol{x}) \propto P(\boldsymbol{x}|\boldsymbol{\theta})P(\boldsymbol{\theta})$. The likelihood term $P(\boldsymbol{x}|\boldsymbol{\theta})$ is

$$P(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_i \sum_j \sum_n \Big[p(x_n^j | t_k^j) \cdot p(t_k^j | t_i) \cdot p(z_{jnk} = 1) \cdot p(v_{jki} = 1) \Big] \tag{4}$$

We set a DP prior with concentration parameter α on the assignment of peaks to clusters $p(z_{jnk} = 1)$, and another DP prior with concentration parameter α' on the assignment of clusters to metabolites $p(v_{jki} = 1)$. We use MCMC method to sample from the posterior distribution $P(\boldsymbol{\theta}|\boldsymbol{x})$. Specifically, we follow the posterior sampling in the Chinese restaurant franchise, as described in section 5.1 in [6]. Here, we explicitly assign the peaks to their parent clusters, which are then assigned to the parent metabolites.

The next sub-sections describe the update steps during Gibbs sampling of the posterior distribution

3.2.1 Initialisation

1. For every file, assign all peaks under 1 cluster.
2. Across files, all clusters are assigned under 1 global metabolite.
3. The metabolite's RT is sampled from the base distribution (eq. 3), and the cluster's RT is sampled conditioned on it's parent metabolite RT (eq. 2).

3.2.2 Useful identities

The integral of the product of two pdf $p(x|y,c) \cdot p(y|a,b)$ where both are normal densities

$$\int \left[\mathcal{N}(x|y, c^{-1}) \cdot \mathcal{N}(y|a, b^{-1})\right] \, dy \quad \propto \quad \mathcal{N}(x|a, (c^{-1} + b^{-1})) \tag{5}$$

Proof to follow ...

3.2.3 Updating peak assignments

To update the assignment of peak n to cluster k in file j, we need the conditional probability of $p(z_{jnk} = 1)$ given other parameters. This is

$$p(z_{jnk} = 1|\ldots) \propto \begin{cases} c_{jk} \cdot L(x_n^j | z_{jnk} = 1) \\ \alpha_t \cdot L(x_n^j | z_{jnk^*} = 1) \end{cases} \tag{6}$$

For existing cluster k, the likelihood of the peak is proportional to c_{jk}, the number of peaks inside that cluster.

$$L(x_n^j | z_{jnk} = 1) = \mathcal{N}(x_n^j | t_k^j, \gamma^{-1}) \tag{7}$$

For a new cluster k^*, this is proportional to α_t. The likelihood then depends on whether the peak is assigned to an existing metabolite i or a new metabolite i^*.

$$L(x_n^j|z_{jnk^*}=1) = \sum_i \left[\frac{c_i}{\alpha' + \sum_i c_i} p(x_i^j|v_{jni}=1) \right] + \frac{\alpha'}{\alpha' + \sum_i c_i} p(x_i^j|v_{jni^*}=1) \tag{8}$$

Here α' is the concentration parameter for metabolite DP mixture. Let the indicator $v_{jni} = 1$ denotes the assignment of peak n in file j to metabolite i. Then for existing metabolite i, $p(x_i^j|v_{jk^*i} = 1)$, the likelihood of the peak under that metabolite, can be computed by marginalising over all possible values of clusters RT t_k^j. Note that we use the identity in equation 5 to go from equation 10 to equation 11.

$$\begin{aligned}
p(x_n^j|v_{jni}=1) &\propto \int \left[p(x_n^j|t_k^j,\gamma) p(t_k^j|t_i,\delta) \right] \, dt_k^j \tag{9} \\
&\propto \int \left[\mathcal{N}(x_n^j|t_k^j,\gamma^{-1}) \cdot \mathcal{N}(t_k^j|t_i,\delta^{-1}) \right] \, dt_k^j \tag{10} \\
&\propto \mathcal{N}(x_n^j|t_i,(\gamma^{-1}+\delta^{-1})) \tag{11}
\end{aligned}$$

For new metabolite i^*, the likelihood is obtained by marginalising over all possible values of clusters RT t_k^j and metabolite RT t_i

$$\begin{aligned}
p(x_n^j|v_{jni^*}=1) &\propto \int\int \left[p(x_n^j|t_k^j,\gamma) p(t_k^j|t_i,\delta) \right] \, dt_k^j \, dt_i \tag{12} \\
&\propto \int\int \left[\mathcal{N}(x_n^j|t_k^j,\gamma^{-1}) \cdot \mathcal{N}(t_k^j|t_i,\delta^{-1}) \cdot \mathcal{N}(t_i|\mu_0,\sigma_0^{-1}) \right] \, dt_k^j \, dt_i \tag{13} \\
&\propto \mathcal{N}(x_n^j|\mu_0,(\sigma_0^{-1}+\gamma^{-1}+\delta^{-1})) \tag{14}
\end{aligned}$$

The above assignment steps can be illustrated using the Chinese Restaurant Franchise metaphor. Here, a peak is a customer, a file is a restaurant, a cluster is a table, and a metabolite is a dish. During the Gibbs sampling step, a new customer $n + 1$ arrives in the restaurant. We then need to determine whether this new customer is to be seated on a new or existing table. For an existing table, the likelihood proportional to the number of people already sitting on that table (c_{jk}) times the likelihood of the new customer belonging on the table (eq. 7). For a new table, the likelihood is proportional to the DP concentration parameter α times the likelihood of ordering any of the existing or new dishes (eq. 8). When deciding which dish to order, we marginalise over all possible tables in eq. 9 and eq. 12 (so we don't have to care about which table is actually selected).

Having computed both likelihood values of the customer sitting on existing tables or a new table (eq. 6), we do a coin toss to decide which table k is actually selected. At this point, we should have maintained a restaurant-level variable K, which is the counter of all tables that exist in the restaurant. If $k \leq K$, we send the customer to sit on table k and share the dish already ordered by his friends sitting on table k. Otherwise if $k > K$, we increment K by 1, create a new table $k+1$ and offers the customer the menu of dishes to order. At this point, the new table $k+1$ is created but not assigned any retention time value (mixture parameter) yet. We will return to this later.

Since the customer is the first to sit on the table, he would order the dish to be shared on that table. We reuse the computation done for eq. 8 to compute the likelihood of the new customer ordering existing dishes or new dish, and do a coin toss to select the dish i. If $i \leq I$, where I is the global counter of all dishes that exist *across restaurants*, we serve the customer the dish i. Otherwise, we get our chef to devise a new dish $i+1$ for him. The new dish $i+1$ can subsequently be ordered by any new customer that arrives in any of the restaurant in our franchise. The mixture parameter (retention time value) t_i of the new dish $i+1$ can be computed given the data x_n^j using Bayes' rule:

$$\begin{aligned}
p(t_i|x_n^j) &\propto p(x_n^j|t_i)p(t_i) && (15)\\
&\propto \left[\int p(x_n^j|t_k^j)p(t_k^j|t_i)\, dt_k^j\right] \cdot p(t_i) && (16)\\
&\propto \left[\int \mathcal{N}(x_n^j|t_k^j,\gamma^{-1}) \cdot \mathcal{N}(t_k^j|t_i,\delta^{-1})\, dt_k^j\right] \cdot \mathcal{N}(t_i|\mu_0,\sigma_0^{-1}) && (17)\\
&\propto \mathcal{N}(x_n^j|t_i,\gamma^{-1}+\delta^{-1}) \cdot \mathcal{N}(t_i|\mu_0,\sigma_0^{-1}) && (18)
\end{aligned}$$

In the LHS of the equation above, $p(t_i|x_n^j)$ is a Normal distribution. Let's say it is parameterised by $\mathcal{N}(t_i|\mu_i,\gamma_i^{-1})$.

$$\begin{aligned}
\mathcal{N}(t_i|\mu_i,\gamma_i^{-1}) &\propto \mathcal{N}(x_n^j|t_i,\gamma^{-1}+\delta^{-1}) \cdot \mathcal{N}(t_i|\mu_0,\sigma_0^{-1}) && (19)\\
exp\left(\frac{-\gamma_i}{2}(t_i-\mu_i)^2\right) &\propto exp\left(\frac{-(\gamma^{-1}+\delta^{-1})^{-1}}{2}(x_n^j-t_i)^2+\frac{-\sigma_0}{2}(t_i-\mu_0)^2\right) && (20)
\end{aligned}$$

Collecting the quadratic term containing $(t_i)^2$, we can solve for γ_i

$$\begin{aligned}
\frac{-\gamma_i}{2}(t_i)^2 &= \frac{(\gamma^{-1}+\delta^{-1})^{-1}}{2}(t_i)^2+\frac{-\sigma_0}{2}(t_i)^2 && (21)\\
\gamma_i &= (\gamma^{-1}+\delta^{-1})^{-1}+\sigma_0 && (22)
\end{aligned}$$

Similarly, collecting the linear term containing t_i, we can solve for μ_i

$$\begin{aligned}
\frac{-\gamma_i}{2}(2t_i\mu_i) &= \frac{-(\gamma^{-1}+\delta^{-1})^{-1}}{2}(2x_n^j t_i)+\frac{-\sigma_0}{2}(2t_i\mu_0) && (23)\\
-\gamma_i\mu_i &= -(\gamma^{-1}+\delta^{-1})^{-1}x_n^j-\sigma_0\mu_0 && (24)\\
\mu_i &= \frac{1}{\gamma_i}\left[(\gamma^{-1}+\delta^{-1})^{-1}x_n^j+\sigma_0\mu_0\right] && (25)
\end{aligned}$$

So, when creating a new dish t_i, we sample its value from a Normal distribution $\mathcal{N}(t_i|\mu_i, \gamma_i^{-1})$ with mean μ_i and precision γ_i given above. Similarly, we can sample from the posterior distribution of table value t_k^j given its parent t_i and the data x_n^j:

$$p(t_k^j|x_n^j) \quad \propto \quad p(x_n^j|t_k^j, \gamma)p(t_k^j|t_i, \delta) = \mathcal{N}(t_k^j|\mu_k, \gamma_k^{-1}) \tag{26}$$

where

$$\gamma_k = \gamma + \delta \tag{27}$$

$$\mu_k = \frac{1}{\gamma_k}\left[\gamma x_n^j + \delta t_i\right] \tag{28}$$

The specific steps to update $p(z_{jnk} = 1)$ is given in algorithm 1.

Algorithm 1 Updating peak assignments (assign_peaks.m)

```
I = 0, K = 0
j = loop randomly over files
    n = loop randomly over peaks

        remove peak n from model
        if this results in empty cluster k
            delete cluster k from model
            if this results in empty metabolite i
                delete metabolite i from model
                I = I-1
            end
        end

        P1 = compute cjk * L(x_n|existing cluster, eq 7)
        P2 = compute alpha * L(x_n|new cluster, eq 8)
        new_k = pick_cluster(P1, P2)
        if new_k > K

            K = K+1
            P3 = compute ci * L(x_n|t_i, eq 11)
            P4 = compute alpha' * L(x_n|t_i, eq 14)
            new_i = pick_metabolite(P3, P4)

            if new_i > I
                I = I+1
                ti = sample new metabolite RT given peak n (eq. 18)
                set metabolite RT = t_i
            end

            assign v_jki = 1 for cluster k in file j to metabolite i
            t_jk = sample new cluster RT given t_i and peak n (eq. 26)
            set cluster RT = t_jk

        end
        assign Z_jnk = 1 for peak n to cluster k in file j

    end
end
```

3.2.4 Updating cluster assignments

Given some existing cluster k, we would like to assign this cluster (a whole block of peaks $\{x_n^j\}$ under k) into a new metabolite i.

The probability is proportional to c_i for existing metabolite and α' for new metabolite. The likelihood term for existing table is

$$p(t_k^j|t_i,\delta) = \mathcal{N}(t_k^j|t_i,\delta^{-1}) \tag{29}$$

For new table, it is

$$\begin{aligned} p(t_k^j|t_{i^*},...) &\propto \int \Big[p(t_k^j|t_i,\delta) \cdot p(t_i|\mu_0,\sigma_0) \Big] \, dt_i && (30) \\ &\propto \mathcal{N}(t_k^j|\mu_0,(\delta^{-1}+\sigma_0^{-1})) && (31) \end{aligned}$$

Algorithm 2 Updating peak assignments (assign_peaks.m)

```
j = loop randomly over files
    k = loop randomly over clusters

        remove cluster k from model
        if this results in empty metabolite i
            delete cluster k from model
            if this results in empty metabolite i
                delete metabolite i from model
                I = I-1
            end
        end

        P1 = compute ci * L(t_jk|existing metabolite, eq 29)
        P2 = compute alpha' * L(t_jk|new metabolite, eq 32)
        new_i = pick_metabolite(P1, P2)
        if new_i > I
            I = I+1
            ti = sample new metabolite RT given ? (eq. ?)
            set metabolite RT = ti
        end
        assign v_jki = 1 for cluster k in file j to metabolite i

    end
end
```

3.2.5 Updating metabolite RTs

The conditional probability of metabolite RTs $p(t_i|\ldots)$ is given by the product of the prior probability on t_i multiplied by the likelihood of the RT values of metabolite i's child clusters. $t^j_{v_{jki}=1}$ denotes the retention time of cluster k in file j that is assigned to metabolite i. There is a total of J files altogether. So, for every metabolite i, we update its RT:

$$p(t_i|\ldots) \propto p(t_i|\mu_0,\sigma_0^{-1}) \prod_j^J \prod_k^K p(t^j_{v_{jki}=1}|t_i,\delta^{-1}) = \mathcal{N}(\mu_a,\gamma_a^{-1}) \qquad (32)$$

where

$$\mu_a = \frac{1}{\gamma_a} \left[\mu_0\sigma_0 + \delta \sum_j \sum_k t^j_{v_{jki}=1} \right] \qquad (33)$$

$$\gamma_a = \sigma_0 + c_{v_{jki=1}}\delta \tag{34}$$

$$c_{v_{jki}=1} = \sum_j \sum_k I(v_{jki} = 1) \tag{35}$$

3.2.6 Updating cluster RTs

Similar to above, for every j file, there is a total of N peaks inside. $x^j_{z_{jnk}=1}$ denotes the retention time of peak n that is assigned to cluster k in file j. So, for every cluster k in file j, we update its RT:

$$p(t^j_k|\ldots) \propto p(t^j_k|t_i, \delta^{-1}) \prod_n^N p(x^j_{z_{jnk}=1}|t^j_k, \gamma^{-1}) = \mathcal{N}(\mu_b, \gamma_b^{-1}) \tag{36}$$

where

$$\mu_b = \frac{1}{\gamma_b}\left[t_i\delta + \gamma \sum_n x^j_{z_{jnk}=1}\right] \tag{37}$$

$$\gamma_b = \delta + c_{z_{jnk}=1}\gamma \tag{38}$$

$$c_{z_{jnk}=1} = \sum_n I(z_{jnk} = 1) \tag{39}$$

3.3 Marginalising out the clusters

If we marginalise out t^j_k, the metabolite RT t_i is known but the new cluster RT t^j_k is unknown, so we can no longer assume conditional independence between the peaks. To compute the probability of this block of peaks under the new metabolite i, we have to apply the chain rule:

$$p(x^j_1, x^j_2, \ldots, x^j_n|t_i) = p(x^j_1|t_i) \cdot p(x^j_2|t_i, x^j_1) \cdot p(x^j_3|t_i, x^j_1, x^j_2) \ldots p(x^j_n|t_i, x^j_1, x^j_2, \ldots, x^j_{n-1}) \tag{40}$$

where

$$p(x^j_n|t_i, x^j_1, x^j_2, \ldots, x^j_{n-1}) = \int \left[p(x^j_n|t^j_k) \cdot p(t^j_k|t_i, x^j_1, x^j_2, \ldots, x^j_{n-1})\right] dt^j_k \tag{41}$$

More to come ...

3.4 Matching

Quick note.

Given the grouping (clustering) of peaks within the top components, we have to perform another additional step of actually matching the peak. We use the samples produced during the Gibbs sampling to compute the posterior 'similarity' of peaks. Following is a procedure for a quick test using the HDP result for matching. Steps involved:

1. Cluster all peaks across files at once using HDP
    a) HDP score = posterior 'similarity' of two peaks being assigned to the same metabolite
2. Perform greedy matching
    a) Distance = Mahalanobis(p1, p2), but using HDP posterior similarity score above for the time component.
    b) Peaks outside mass tolerance will not be matched

Parameters:

1. # samples = 200 (100 burn-in)
2. mu_0 = mean of RT values from input files
3. sigma_0 = 1/5000
4. top_alpha = 1
5. alpha = 1
6. delta_prec = 1/30
7. gamma_prec = 1/60

Test on the P1 data: 2 replicates per fraction. Fraction 080 has ~ 1200 peaks, and fraction 100 has ~750 peaks.

OOops ... forgot to restrict the matching so that we match only peaks within the same component !

3.5 Results

Not that great ?

3.6 Discussion

1. Are the probabilities well-calibrated ? We want to plot some ROC curve. Say, for each matching, assign the posterior similarity as its score. Make a ROC curve out of this. What do we get ?

Fraction	Join	OpenMS	SIMA	MW	HDP
080	0.944	0.882	0.944	0.944	0.917
100	0.950	0.920	0.950	0.950	0.940

Table 1: F1 scores

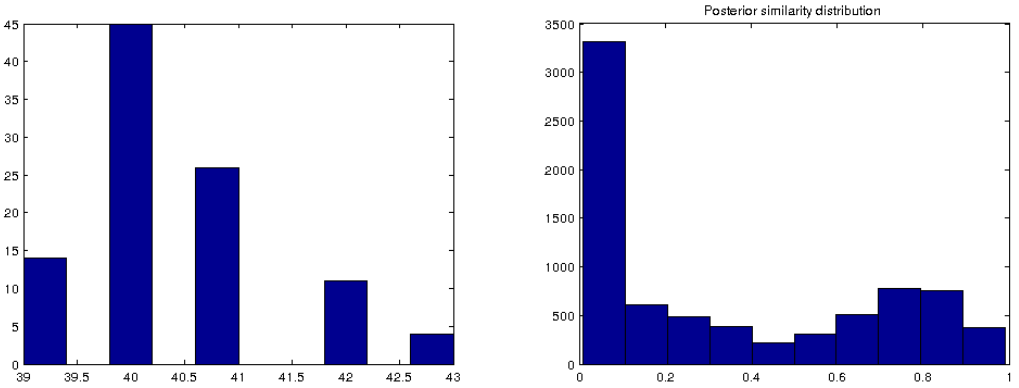

Figure 1: P1_100. Left is posterior distribution of the number of top components inferred, right is the distribution on the 'similarity' of peaks. Peak p1 is similar to p2 if they are placed together under the same top component in a sample.

2. Is there any relationship between the cluster retention time t_{ij} and the metabolite retention time t_i within each file. For each posterior sample within each file, plot the t_{ij} and t_i, what do you observe / expect to see ?

4 HDP Model - RT & Mass

Each metabolite m has A_m possible adducts / deducts, where adduct a in metabolite m has I_{am} isotopic peaks. Adduct is a product of direct addition of of $>=$ 2 distinct molecules that results in a single reaction product containing all the atoms of the component molecules. In LC-MS, the term adduct ion refers refers to the ions formed by adduction of an ionic species to the molecule. In positive ion analysis, the most common metal adduct ions are single-charged sodium ($[M + Na]^+$) and potassium ($[M + K]^+$) adducts. Adduct ions can be recognized from the mass spectrum by observing the difference from the analyte molecule, e.g. the adduct ion composition $[M + H]^+$ has the observed m/z value $M + 1$, where M is the molecular mass of the analyte molecule (i.e. metabolite). Most naturally occuring elements in nature consist of isotopes. Isotopes are variants of a particular chemical element that have the same number of protons but differ in the number of neutrons (they have different atomic mass).

Describe ESI ionisation here ..

There are two modelling approaches when we want to bring in the mass information: one where we consider the sum formula of the metabolite and one where we don't.

4.1 Modelling mass with knowledge of formulae

The approach in MetAssign requires us to have the formula of the metabolite to begin with (since we're computing the isotopic profile from the formula, see [5]). This is because peaks are related by their isotopic distribution along the m/z dimension (in the mass spectrum). Given observed peaks, we can attempt to predict their possible formulae as such:

1. Remove mass of adduct from the observed mass. This gives us the neutral mass m
2. Generate $F = \{$all formulae possible given the mass constraint of $m + \epsilon$, where ϵ is the MS instrument accuracy$\}$.
3. Filter F by various heuristics such as the 7 golden rule [3], element counts, ring double bond etc. This should reduce $|F|$ a lot from say, 50000 entries to 1000.
4. For every $f \in F$, we can assign it some isotopic pattern score, which is the goodness of fit between the predicted isotopic peaks (generated by f versus the observed peaks). Predicted peaks can be efficiently generated according to [5]. Methods that compute this score are for e.g. in MzMine2 [4] and SIRIUS [1].

Then what ... ?

4.2 Modelling mass without knowledge of formulae

Another way is if we do not explicitly model the formula. The observed data is now a vector of mass and retention time $\mathbf{d}_n^j = (x_n^j, y_n^j)$ where x_n^j is the retention time value and y_n^j is the mass value of peak n in file j. We perform the assignment of $\mathbf{d}_n^j$ based on its retention time value to RT cluster k and metabolite i (as in section 3.2) and also based on the mass information to separate peaks further into mass clusters. To do this, each metabolite i is linked to an infinite mixture model (using DP prior) where components in this mixture model correspond to groupings of peaks by their masses. The Normal distribution is used as the component in the linked DP mixture. Individual peaks are now assigned to an RT cluster (and its respective parent metabolite) and put into a mass cluster.

[draw the plate diagram here]

Table 2 shows the correspondence between the actual biological entities and their correspondence in the CRF analogy. In the CRF story, a customer n walks into the restaurant j and is seated at table k. He is then served the dish i which is shared across the restaurant franchise. Additionally, the dish now also comes in seasoned varieties (e.g. less salt, spicier).

Indexing variable	Biological entity	CRF entity
n	Peak	Customer
j	File (replicate)	Restaurant
k	RT cluster	Table
i	Metabolite	Dish
a	Mass cluster	Seasoned dish

Table 2: CRF analogy for the RT + mass HDP model

Let the indicator $v_{jnia} = 1$ denotes the assignment of peak n in file j to metabolite i and the mass cluster a within i. Then following equation 6, the likelihood of peak data point $\mathbf{d}_n^j$ going into an existing cluster is proportional to c_{jk}, the number of peaks inside that cluster.

$$L(\mathbf{d}_n^j | z_{jnk} = 1) \quad = \quad p(x_n^j | z_{jnk} = 1) p(y_n^j | v_{jnia} = 1) \tag{42}$$

where the RT term is

$$p(x_n^j | z_{jnk} = 1) = \mathcal{N}(x_n^j | t_k^j, \gamma^{-1}) \tag{43}$$

and the mass term is

$$p(y_n^j | v_{jnia} = 1) = \sum_a \left[\frac{c_{ia}}{\alpha'_m + \sum_a c_{ia}} p(y_n^j | v_{jnia} = 1) \right] + \frac{\alpha_m}{\alpha_m + \sum_a c_{ia}} p(y_n^j | v_{jnia^*} = 1) \tag{44}$$

Then

$$p(y_n^j | v_{jnia} = 1) = \mathcal{N}(y_n^j | \theta_{ia}, \rho^{-1}) \tag{45}$$

where θ_{ia} is the mean of the component a, and ρ^{-1} the precision, which should a priori be set to be a small value by the user (or maybe we can draw from some Gamma distribution ?). The component mean θ_{ia} is drawn from the base distribution

$$p(\theta_{ia} | \theta_0, \rho_0) = \mathcal{N}(\theta_{ia} | \psi_0, \rho_0^{-1}) \tag{46}$$

Similar to how it's done for RT clustering previously, once a customer has been offered a dish i, we compute the likelihood for the customer to be assigned existing or new seasoned dishes. The seasoned dish a is selected via a cointoss. For each dish i, the variable A_i maintains the number of seasoned dish variations that exist. If the new seasoned dish is required ($a > A_i$), we create one and serves that the customer n. When creating a new seasoned dish, we need to set its mixture parameter θ_{ia}

$$p(\theta_{ia}|y_n^j) \quad \propto \quad p(y_n^j|\theta_{ia}, \rho)p(\theta_{ia}|\psi_0, \rho_0) = \mathcal{N}(\theta_{ia}|\mu_a, \sigma_a^{-1}) \tag{47}$$

where

$$\sigma_a = \rho + \rho_0 \tag{48}$$

$$\mu_a = \frac{1}{\sigma_a}\left[\rho y_n^j + \rho_0\psi_0\right] \tag{49}$$

Then for every mass cluster a in the mixture associated to metabolite i, we update its mixture parameter:

$$p(\theta_{ia}|\ldots) \propto p(\theta_{ia}|\psi_0, \rho_0)\prod_n^N p(y_n^j|\theta_{ia}, \rho) = \mathcal{N}(\mu_b, \sigma_b^{-1}) \tag{50}$$

$$\sigma_b = \rho_0 + N\rho \tag{51}$$

$$\mu_b = \frac{1}{\sigma_b}\left[\rho_0\psi_0 + \rho\sum_n y_n^j\right] \tag{52}$$

It is also possible for a customer or a set of customers to reject the seasoned dish served. This happens when a customer n is switching table from k to k', or when the whole table k switches the dish assigned to it from i to i'. In this case, we have to perform the necessary book-keeping such as recomputing the mixture parameter θ_{ia} when its customer membership changes, and deleting the whole lot of dish variations a when their linked dish i is deleted.

References

[1] Sebastian Böcker, Matthias C Letzel, Zsuzsanna Lipták, and Anton Pervukhin. SIRIUS: decomposing isotope patterns for metabolite identification. *Bioinformatics (Oxford, England)*, 25(2):218–24, January 2009.

[2] Seyoung Kim and P Smyth. Hierarchical Dirichlet processes with random effects. *NIPS*, 2006.

[3] Tobias Kind and Oliver Fiehn. Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC bioinformatics*, 8:105, January 2007.

[4] Tomáš Pluskal, Taisuke Uehara, and Mitsuhiro Yanagida. Highly accurate chemical formula prediction tool utilizing high-resolution mass spectra, MS/MS fragmentation, heuristic rules, and isotope pattern matching. *Analytical chemistry*, 84(10):4396–403, May 2012.

[5] Ross K Snider. Efficient calculation of exact mass isotopic distributions. *Journal of the American Society for Mass Spectrometry*, 18(8):1511–5, August 2007.

[6] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet Processes. pages 1–30, 2005.