Chapter 4

Incorporating Clustering Information into Peak Alignment

4.1 Introduction

None of the alignment tools surveyed in Section 2.4.2 take into account the structural dependencies between related peaks produced by the same metabolite when solving the correspondence problem. Such information could potentially be used to improve the alignment process since a set of related peaks in one run should generally be aligned to another set of related peaks in the other run. As described in Section 2.4.1, related peaks are defined to be all those peaks that appear in a run due to the presence of one compound (peptide/metabolite) in the sample being analysed. Examples of related peaks are isotope peaks, multiple adduct and deduct peaks, and fragment peaks, elaborated further in Section 2.4.1. Such peaks should co-elute from the column and have similar chromatographic shapes and RT values. The related peak information can come from any peak grouping methods, of which clustering via RT is one instance, but one key assumption is that groups of co-eluting peaks corresponding to the same metabolite are generally preserved across runs.

In this chapter, we propose clustering the related peaks sharing similar RT values together, and using the information from the clustering process to modify the similarity score matrix used for the alignment (matching) of peaks across runs. This idea is illustrated in Figure 4.1 and further introduced in Sections 4.2 and 4.3. As shown in Figure 4.1, initial weights are computed between pairs of peaks in the two runs that are within m/z and RT tolerances (e.g. W_{AE} and W_{AJ}). When related peak information is added, the similarity between peaks A and E is increased due to peak A being related to another peak (B) that is similar to a peak (G) related to E. On the other hand, the similarity between A and J is not increased as J does not have any related peaks that could potentially be matched to peaks related to A. In other words, we are proposing using the structural dependencies present between peaks

in each run to modify the similarity scores and improve alignment performance: the more peaks related to A that could be matched to peaks related to E, the more likely it becomes that A should be matched to E.

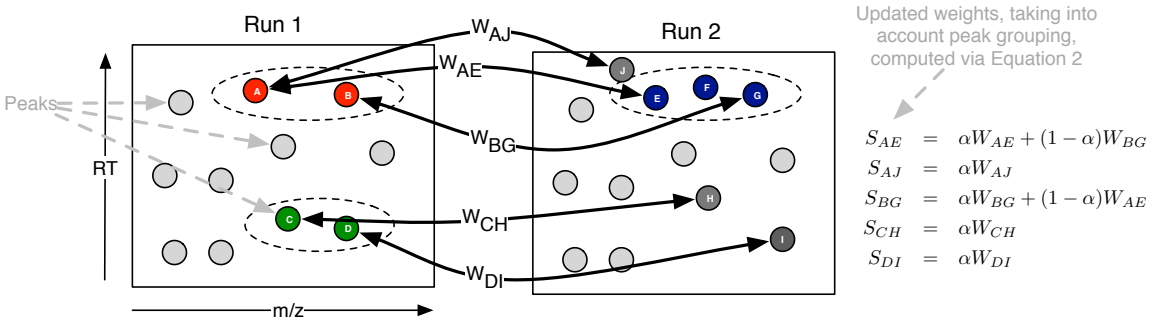


Figure 4.1: Illustrative example of the incorporation of grouping information into the similarity score. Each node in the figure is a peak feature, and dotted ovals represent groups of related peaks, e.g. isotopes, fragments, etc. Initially weights (e.g. W_{AE}) are computed for pairs of peaks (one from each run) with m/z and RT within pre-defined thresholds. These weights are converted into an overall score by incorporating grouping information. For example, peak pairs (A, E) and (B, G) are both within the threshold. Because A and B are in the same group, and E and G are in the same group, the weights between pairs (A, E) and (B, G) are upweighted. Peak J is not related to any peaks that could be matched with A's related peaks and the similarity between A and J is therefore downweighted (because $\alpha \leq 1$). The same applies to similarities between pairs (C, H) and (D, I).

Statement of Original Work

The idea of constructing alignment via approximate maximum weighted matching was proposed by the author. Simon Rogers conceived the idea of using the clustering information of related peaks to modify the similarity matrix used for matching. Code implementation and performance evaluation was carried out by the author.

4.2 Direct Matching

Our proposed alignment method combines a novel similarity score with maximum weighted bipartite matching. This results in pairwise alignments which can be, if desired, extended to multiple alignments with hierarchical merging strategy. In such merging strategies, having an accurate initial pairwise alignments is important because of its influence on the final multiple alignment results. In the following sections, we describe each step in more detail.

4.2.1 Feature Matching

A peak feature refers to a tuple of $(m/z, RT)$ produced as output after the initial peak detection stage of LC-MS data. Here, m/z is the mass-to-charge value and RT the retention time value of a peak feature. Suppose we wish to align run A containing N_A peaks with run B containing N_B peaks. Alignment between two runs can be represented as a matching problem on a bipartite graph G, where nodes in the graph are the features, edges are the potential correspondence between features and the weights on the edges are the similarity scores (entries in S) between features. In SIMA [27], the Gale-Shapley algorithm [54] is used to find a stable matching in G. A matching is stable if there are no two features in different runs that would prefer to be matched to each other than to their currently matched partners. Since the stable matching is computed based on ranked preference, valuable information could be discarded as distances between features are converted to ranks. As such, we prefer to use a method that maximises the total sum of similarity scores of matched features (maximum weighted matching).

The benefit of maximum weighted bipartite matching in solving the peak correspondence problem has been studied in [29] in their LWBMatch tool. LWBMatch shows that such matching method, coupled to a local regression method, is able to align runs having large and systematic drifts in RT values. The well-known Hungarian algorithm [55] attributed to Kuhn and Munkres is used in LWBMatch to solve this problem. The time complexity of the Hungarian algorithm is $O(n^3)$, where n is the number of peaks in the larger set. While the Hungarian algorithm's implementation can be improved to $O(n^2 log\, n)$ by using Fibonacci heaps for the shortest path computation, the polynomial time complexity required in this scheme is often too slow to be practical for alignments of the large number of runs produced in large-scale untargeted LC-MS studies. Consequently, we compute an approximation of the maximum weighted matching using a simple greedy algorithm that runs in $O(m\, log\, n)$ time, where n and m denote the number of vertices and edges in the bipartite graph G to be solved. The greedy algorithm is straightforward to describe: pick the heaviest edge e in G, where e represents a potential match between nodes (features). Add e to the matching solution M and remove all other edges adjacent to e from G. Repeat until all edges in G have been exhausted. This simple greedy algorithm is known to provide a lower bound of at least 1/2 of the maximum weight in the matching [56].

4.2.2 Feature Similarity

To define a similarity measure between peak features, we follow SIMA [27] in using the Mahalanobis distance between two peaks $\boldsymbol{p}_i \in A$, $\boldsymbol{p}_j \in B$ where each peak is a vector of its

m/z and RT values $\boldsymbol{p}_i = [m_i, t_i]^\mathsf{T}$ and $\boldsymbol{p}_j = [m_j, t_j]^\mathsf{T}$. The distance is given as:

$$D(\boldsymbol{p}_i, \boldsymbol{p}_j) = \sqrt{(\boldsymbol{p}_i - \boldsymbol{p}_j)^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{p}_i - \boldsymbol{p}_j)},$$

where the covariance matrix $\boldsymbol{\Sigma}$ is a diagonal matrix of mass-to-charge tolerance σ_m^2 and retention time tolerance σ_t^2. The diagonal covariance matrix $\boldsymbol{\Sigma}$ assumes independence between the σ_m^2 and σ_t^2 components. To reduce the computational burden, entries in $\boldsymbol{D}$ are only computed when the peaks' m/z and RT values are within σ_m and σ_t. We now define the similarity score between two peaks as one minus their normalised distance:

$$W(\boldsymbol{p}_i, \boldsymbol{p}_j) = 1 - \frac{D(\boldsymbol{p}_i, \boldsymbol{p}_j)}{D_{max}}, \tag{4.1}$$

where D_{max} is the maximum computed distance between peaks in the two runs being aligned. Collectively, we call the $N_A \times N_B$ matrix of similarity scores between all peaks in run A and B to be $\boldsymbol{W}$.

4.3 Incorporating Related Peak Groups

4.3.1 Combining Scores

The similarity score matrix $\mathbf{W}$ can now be combined with related peak information to obtain a final score, $\mathbf{S}$:

$$\boldsymbol{S} = \alpha \boldsymbol{W} + (1 - \alpha) \boldsymbol{L} \tag{4.2}$$

where $\boldsymbol{L}$ is the cluster similarity score between the two peaks in a single run (described below), and α ($0 \leq \alpha \leq 1$) is a parameter controlling the relative influence of the two components. To compute $\boldsymbol{L}$, we require related peak groupings from the two runs being aligned. This takes the form of an $N_A \times N_A$ matrix $\mathbf{C}^A$ for run A and an $N_B \times N_B$ matrix $\mathbf{C}^B$ for run B. Entries in $\boldsymbol{C}^A$ and $\boldsymbol{C}^B$ can be either binary (0, 1) or probability values, depending on the peak grouping algorithm used. For example, if a greedy clustering approach is applied to the features in run A, the ij-th element of $\mathbf{C}^A$ will be either 1 or 0, depending on whether the i-th and j-th features (peaks) in A are clustered together (1) or not (0). Note that in the following, we define the diagonal components of both matrices to be zero to avoid double counting. We then compute $\mathbf{L}$ as follows:

$$\boldsymbol{L} = \boldsymbol{C}^A \cdot \boldsymbol{W} \cdot \boldsymbol{C}^B. \tag{4.3}$$

The resulting matrix gives cluster similarity scores such that each element L_{ij} of $\boldsymbol{L}$ is the sum of weight from peaks in the same cluster as i in run A to peaks in the same cluster as

j in run B. This allows us to use the matrix $\boldsymbol{L}$ to upweight the similarity scores between peaks in the same cluster in one run that also have more potential matches to peaks in the same cluster in the other run of the matching. Computation of Equation 4.3 is illustrated in Figure 4.1. The ratio parameter α controls how much clustering information we bring into the overall similarity score matrix $\boldsymbol{S}$, with its value bounded in $0 \leq \alpha \leq 1$. Setting $\alpha = 1$ results in a matching that uses only information from $\boldsymbol{W}$, the similarity score matrix. Setting $\alpha = 0$ means that the matching is performed based only on the cluster similarity score $\boldsymbol{L}$. From our experience, a reasonable range of values for α lies between 0.2 to 0.4.

Our proposed approach is independent of the method used to group related peaks in each run. For comparison, we call our method that does not use the cluster similarity score ($\alpha = 1$) to be Maximum-Weighted (MW). We then demonstrate the performance improvement from incorporating related peaks information using two different clustering algorithms: a greedy RT clustering approach (described in Section 4.4) and a statistical mixture model (Section 4.5). The combination of matching with the greedy clustering is called MWG, while the alternative approach that uses the probabilities coming from the mixture model is called MWM.

4.4 Greedy Clustering of Related Peaks

In the greedy clustering method, the most intense peak in the dataset is selected and clustered with other candidate peaks inside a retention time window g_{tol}. The next most intense peak that has not already been clustered is processed, and the grouping process is repeated until all peaks are exhausted. If chromatographic peak shapes information is available (such as for the Metabolomic dataset used in section 4.7.2), the Pearson correlation coefficient between the chromatographic peak signals of the most intense peak and the candidate peaks are computed. Only candidate peaks with Pearson correlation values greater than some threshold c are accepted into the newly-formed cluster. This greedy clustering process results in binary grouping matrices $\mathbf{C}^A$ and $\mathbf{C}^B$ that can be used in eq. 4.3.

4.5 Mixture Model Clustering of Related Peaks

We can also group related peaks together by their RT values using a mixture model. Our observation consists of a vector of N observed peak's RT values $\mathbf{y} = (y_1, y_2, ..., y_n)$, and our aim is to partition each set of peaks into K groups of related peaks (clusters) by their RT values. We used a Gaussian mixture model with Dirichlet Process prior (described further in Section 3.4) to model the data. A peak is indexed by the variable $n = 1, ..., N$ and a cluster indexed by the variable $k = 1, ..., K$. Each Gaussian mixture component has some mean

μ_k are assumed to have a fixed precision (inverse variance) δ, corresponding to the fixed retention time tolerance for each group of related peaks. Let the indicator $\boldsymbol{z}_{nk} = 1$ denotes the assignment of peak n to RT cluster k. Then:

$$\begin{aligned} \boldsymbol{\pi}|\alpha &\sim GEM(\gamma) && (4.4)\\ \boldsymbol{z}_{nk} = 1|\boldsymbol{\pi}_k &\sim \boldsymbol{\pi}_k && (4.5)\\ \mu_k|\mu_0, \tau_0 &\sim \mathcal{N}(\mu_k|\mu_0, \tau_0^{-1}) && (4.6)\\ y_n|\boldsymbol{z}_{nk} = 1, \mu_k &\sim \mathcal{N}(\mu_k, \delta^{-1}) && (4.7) \end{aligned}$$

where $\boldsymbol{\pi}$ is the mixing proportions, distributed according to the GEM (Griffiths, Engen and McCloskey) distribution. The GEM distribution over $\boldsymbol{\pi}$ is parameterised by the concentration parameter γ and is described through the stick-breaking construction:

$$\begin{aligned} \beta_k &\sim Beta(1, \gamma) && (4.8)\\ \boldsymbol{\pi}_k &= \beta_k \prod_{l=1}^{k-1}(1 - \beta_l) && (4.9) \end{aligned}$$

The mixture component mean μ_k is drawn from a base Gaussian distribution with mean μ_0 and precision τ_0. We set μ_0 to the mean of the observed data, while τ_0 is set to a broad value of 5E-3. Analytical inference is not tractable here, so we use the Gibbs sampling scheme for inference. To do this, we need the conditional probability of $p(\boldsymbol{z}_{nk} = 1, ...)$ of peak n to be in an existing cluster k (or k^* if a new cluster is to be created), given any other parameters in the model. This conditional probability is given by:

$$P(\boldsymbol{z}_{nk} = 1|\mathbf{y}_n, \ldots) \propto \begin{cases} c_k \cdot p(\mathbf{y}_n|\boldsymbol{z}_{nk} = 1, ...) \\ \gamma \cdot p(\mathbf{y}_n|\boldsymbol{z}_{nk^*} = 1, ...) \end{cases} \quad (4.10)$$

where c_k is the current number of members (peaks) in an existing cluster k. $p(\mathbf{y}_n|\boldsymbol{z}_{nk} = 1, ...)$ is the likelihood of peak $\mathbf{y}_n$ in an existing cluster k. We can marginalise over all mixture components and get:

$$p(\mathbf{y}_n|\boldsymbol{z}_{nk} = 1...) = \mathcal{N}(\mathbf{y}_n|\mu_k, \lambda_k^{-1}) \quad (4.11)$$

where $\lambda_k = ((\tau_0 + \sigma c_k)^{-1} + \delta^{-1})^{-1}$ and $\mu_k = \frac{1}{\lambda_k}\left[(\mu_0\tau_0) + (\delta \sum_n \mathbf{y}_{n\in k})\right]$. Here, $\mathbf{y}_{n\in k}$ denotes the RT values of any peak n currently assigned to cluster k, and c_k the count of such peaks. The conditional probability of peak n to be in a new cluster k^* is:

$$p(\mathbf{y}_n|\boldsymbol{z}_{nk^*} = 1...) = \mathcal{N}(\mathbf{y}_n|\mu_0, \lambda_{k^*}^{-1}) \quad (4.12)$$

where $\lambda_{k^*} = (\tau_0^{-1} + \sigma^{-1})^{-1}$. In a step of the Gibbs sampling procedure, we perform the assignment of peak n to cluster k, creating new cluster k^* if necessary. Using the posterior summaries across all samples drawn $S^* = \frac{1}{R}\sum_{r=1}^{R} s_r$, where s_r is the r-th posterior sample collected after a suitable burn-in period and R is the total number of samples taken (excluding burn-in samples), we can obtain the marginal posterior of the probability of two features (peaks) to be in the same cluster k averaged across all samples. These probabilities comprise the elements of $\mathbf{C}^A$ and $\mathbf{C}^B$ (i.e. the ij-th element of $\mathbf{C}^A$ is the proportion of samples from run A in which peaks i and j were in the same cluster), which can be used in eq. 4.3.

4.6 Evaluation Study

Performance evaluation of alignment methods is difficult due to the lack of gold standard and evaluation criteria for benchmarking [8, 57]. Relatively few works, such as [23], exists that provide a comprehensive ground truth for evaluation. In fact, despite the numerous alignment methods that exist, most methods remain unevaluated, evaluated against a small number of alternatives or evaluated based on highly subjective criteria [18]. In particular, evaluation of alignment quality through manual visual inspection of superimposed profile images and some selected chromatograms is problematic and is not a systematic approach towards performance evaluation. While straightforward, the visual inspection of alignment quality is tedious and do not work for evaluation of a large number of aligned peaksets produced by the alignment of a large number of samples. It is also often subjective and might suffer from dissimilar interpretations across different experiments and datasets.

In this chapter, the performance of the proposed methods and other benchmark methods is evaluated using precision and recall on LC-MS datasets from proteomic, metabolomic and glycomic experiments. The proteomic datasets are obtained from [23] while the glycomic dataset comes from [1]. These datasets provide the ground truth for alignment and have used to benchmark alignment performance in other evaluation studies [23, 24, 26, 27, 1]. Additionally, we also introduce a metabolomic dataset generated from the standard runs used for the calibration of chromatographic columns [33]. The runs were produced from different LC-MS analyses separated by weeks, representing a challenging alignment scenario.

Many direct matching methods work in a pairwise fashion and produce an overall results via some merging strategies of intermediate results. Pairwise performance therefore limits overall performance, and as such, in this chapter, we focus on evaluation using only pairs of runs. Some (P2, metabolomic and glycomic) of the datasets selected for evaluation in our experiments have more than 2 runs, so we select only 2 runs each to form a training and testing set. The procedure for doing so is described in the respective following sections for each dataset.

4.6.1 Proteomic Datasets

[23] introduces two benchmark LC-MS proteomic sets (P1, P2) constructed to evaluate the ability of alignment tools in dealing with large retention time drifts. Both the P1 and P2 datasets were analysed using an automated LC-LC/MS-MS platform. Each dataset comes in multiple chromatography salt-step fractions, obtained by bumping the salt level at every 10 minutes interval during chromatographic separation. P1 was produced from *E. coli* samples digested by trypsin, and comes in 2 runs for each fraction. P2 was obtained from *M. smegatis* protein extracts, similarly digested by trypsin, and contains 3 runs for each fraction. P2 was constructed to be a greater challenge to align with runs separated by weeks. Alignment ground truth is established in [23] by means of peptides that can be reliably identified during the identification stage. Only identification annotations with SEQUEST Xcorr score >1.2 is included. Annotations are then filtered by their retention times and matched across runs.

For the proteomic datasets, each fraction in P1 has two runs used for alignment, while each fraction in P2 has three runs (we use only the first two to establish pairwise alignments). Tables 4.1 shows the number of features for each run of the P1 and P2 datasets used for evaluations. Both P1 and P2 represent challenging alignment cases, with large deviations in RT values across runs. This is especially true for P2 with LC-MS runs separated by weeks and large differences in the number of features per run. Further details on the nature of the datasets can be found in [23].

Fraction	# runs	# features per run (P1)	# features per run (P2)
000	2	5824	5054
		4782	5100
020	2	1114	3271
		1021	529
040	2	1230	1483
		958	678
060	2	1902	-
		1440	-
080	2	1183	474
		903	438
100	2	745	401
		581	429

Table 4.1: No. of features in the proteomic (P1 and P2) datasets. Note that fraction 060 is not present in P2.

4.6.2 Metabolomic Datasets

We use a metabolomic dataset generated from a mixture of 104 standard metabolites used for the calibration of chromatographic columns (details in [33]). These runs were produced by ZIC-HILIC chromatography (Merck Sequant, Darmstadt, DE) on an UltiMate 3000 RSLC system (Thermo, Hemel Hempstead, UK), coupled to an Orbitrap Exactive mass spectrometer (Thermo, Hemel Hempstead, UK) in positive mode. The metabolomic dataset is available in different 11 runs, produced from different LC-MS analyses separated by weeks. While these runs are not true technical replicates, they are similar enough to be treated as replicates for the purpose of performance evaluation, and they represent a realistic and fairly challenging alignment scenario. The output from each of these runs is available in PeakML format, which were then converted into a suitable format using the mzMatch suite [58]. Both the original PeakML files and the converted text files can be found in our site. To generate the actual training and testing sets, 30 randomly pairs of runs were extracted as training sets, and another 30 pairs of runs extracted for testing sets. Table 4.2 shows the number of features in each run of the metabolomic dataset.

Metabolomic Run	# features	Metabolomic Run	# features
1	4999	7	6319
2	4986	8	4101
3	6836	9	5485
4	9752	10	5034
5	7076	11	5317
6	4146		

Table 4.2: No. of features in the full metabolomic dataset

Alignment ground truth was constructed from the putative identification of peaks in each of the 11 runs separately at 3 ppm using mzMatch's Identify module, taking as additional input a database of 104 compounds known to be present and a list of common adducts in positive ionisation mode (Table 4.3). This is followed by matching of features that share same annotations across runs to construct the alignment ground truth. Only peaks unambiguously identified with exactly one annotation are used for this purpose, as peaks with more than one annotations per run are discarded from the ground truth construction. The results from this process is an alignment ground truth for a smaller subset of peaks in the runs that can be reliably identified at high mass precision. Note that constructing alignment ground truth in this manner does not introduce bias to the ground truth as the identification information is not used during the alignment stage.

Adduct Types			
M+2H	M+H	M+ACN+H	M+H+NH4
M+NH4	M+ACN+Na	2M+ACN+H	M+ACN+2H
M+Na	M+2ACN+H	M+2ACN+2H	M+CH3OH+H
2M+H			

Table 4.3: List of common adduct types in positive ionisation mode for ESI.

4.6.3 Glycomic Dataset

[1] provides a glycomic dataset containing 23 runs, produced from untargeted LC-MS study for identifying N-glycan disease biomarkers in glyomics studies. LC-MS data were acquired from a Dionex 3000 Ultimate nano-LC system, coupled to an LTQ-Orbitrap Velos mass spectrometer on positive mode. Alignment ground truth is established in [1] based on a manual comparison of measured mass values with theoretical values (taking into account hydrogen adducts) and visual inspection of potentially incorrect assignments. We randomly extracted 30 pairs of runs for training and another 30 pairs of runs for testing performance evaluation from the full glyocomic dataset provided by [1], which comes in 23 runs in total. The following tables show the number of features in each run and the indices of the pairs of files randomly selected as training and testing sets in our Glycomic experiment.

Glycomic Run	# features	Glycomic Run	# features
1	856	13	911
2	1088	14	1144
3	922	15	932
4	808	16	1541
5	886	17	1022
6	850	18	1051
7	979	19	1119
8	1008	20	1047
9	904	21	1017
10	1043	22	990
11	1041	23	977
12	885		

Table 4.4: No. of features in the full glycomic dataset from [1]

4.6.4 Experimental setup

The alignment tools evaluated have in common user-defined mass-to-charge ratio (m/z) and RT window parameters. These parameters act as hard thresholds that determine the solution space to be explored in the m/z and RT dimensions when matching features. Performance

of all alignment procedures is highly dependent on the assumptions and choice of parameter values that underpin them [18]. For example, warping methods must make assumptions regarding the mathematical form of the warping function and are dependent on a good choice of reference run. Direct matching approaches typically need to decide on the form of peak similarity function, and define some m/z and RT windows, outside of which, peaks cannot be matched. Whilst the m/z window and parameters can often be determined based on the mass accuracy of the measurement equipment, there is no obvious way to determine the RT window and associated parameters. The optimal choice of such parameters could have a significant influence on the final results [18], and there is no reason to believe that these parameters should remain constant across different experiments.

Previous studies that use the proteomic dataset presented here [23, 26, 27] varied the window parameters and reported the best performance achieved. Whilst informative, this procedure is unrealistic due to the role of the ground truth in choosing the optimal parameter values. To provide a more realistic estimate of performance, we also present the performance on a separate testing set. In other words, we optimise the window parameters on one alignment task and report the performance when using these optimised parameters on a second task (distinct from the first task). This reflects the scenario where the parameters are set based on performance on a previous dataset or due to information supplied from the instrument manufacturer and tells us how critical setting these parameters is for each method.

In this chapter, *training set* refers to the data on which alignment parameters are optimised and *testing set* refers to the independent set on which alignment performance is evaluated. We believe that this represents a more realistic measure of alignment performance and provides us with some information as to how the different algorithms generalise to new datasets. We addressed the lack of comparative evaluation of alignment tools as discussed in [18] by independently reproducing key results from [23] and [27] for the Join and SIMA alignment methods. Our evaluation studies were performed on proteomic, metabolomic and glycomic datasets introduced before to validate the hypothesis that using related-peak information can improve alignment performance. Since most direct matching algorithms work in a pairwise fashion (pairs of runs are matched and the results combined), pairwise performance therefore limits overall performance, justifying the choice for our experiments. For the proteomic datasets, each fraction in P1 has two runs used for alignment, while each fraction in P2 has three runs (we use only the first two to establish pairwise alignments). Similarly for the metabolomic and glycomic datasets, we randomly extracted 30 pairs of runs for training and another 30 pairs of runs for testing performance evaluation.

Performance is evaluated in terms of precision, recall and F_1-score. Looking at pairwise matching, we can define the following positive and negative instances with respect to some pairwise alignment ground truth:

- True Positive ($\boldsymbol{TP}$): pairs of peaks that should be aligned and are aligned.
- False Positive ($\boldsymbol{FP}$): pairs of peaks that should not be aligned but are aligned.
- True Negative ($\boldsymbol{TN}$): pairs of peaks that should not be aligned and are not aligned.
- False Negative ($\boldsymbol{FN}$): pairs of peaks that should be aligned but are not aligned.

In the context of alignment performance, precision ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$) is the fraction of aligned pairs in the output that are correct with respect to the ground truth, while recall ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$) is the fraction of aligned pairs in the ground truth that are aligned in the output. A perfect alignment would have both precision and recall to be 1. In addition, we also computed the F_1 score (the harmonic mean of precision and recall) such that $F_1 = 2(precision \cdot recall)/(precision + recall)$. Only feature pairs present in the ground truth are considered for evaluation. The idea of using pairwise matching to define alignment performance evaluation is not new, and has also been done in [29]. Collectively for the purpose of performance evaluation, the set of Precision, Recall and F_1 values is referred to as a 'measurement'.

4.6.5 Other Alignment Tools For Comparison

Our proposed approach was benchmarked against MZmine2's Join Aligner [24] and SIMA [27]. Our own matching method (MW) also serves as a useful baseline to demonstrate any difference in performance with or without using clustering information. The two benchmark tools employ different approaches towards alignment. Join Aligner is a greedy direct-matching method, while SIMA is a combinatorial direct-matching method, with an optional warping step to correct RT shifts after an initial matching has been established. Users of the MZmine2's toolkit may have good reasons to prefer Join Aligner to the more recent RANSAC Aligner due to its simplicity and speed. Join Aligner produces a deterministic alignment output (so running it each time on the same input and parameters gives the same result), in contrast to the RANSAC aligner, which is non-deterministic. Join Aligner has relatively few parameters to configure, the most important ones being the *m/z tolerance* and *retention time tolerance* parameters. These parameters are used for thresholding and score calculations, and they were varied within reasonable ranges during our experiments. Similarly, the two most important parameters used in SIMA for thresholding and computing feature similarities are the $T_{(m/z)}$ and T_{rt} parameters (equivalent to our σ_m and σ_t). We let these two parameters vary in our experiments. SIMA also offers an optional step to correct for retention time distortion by constructing a smooth and monotonic warping function for the maximum likelihood alignment path after the initial matching has been done. The utility of this optional step is not obvious to end-users, since it requires additional parameters to

configure and relies on having an initial correspondence established. Therefore, we chose to test only the core matching functionality in SIMA.

4.6.6 Parameter Optimisation

For every evaluated method in our experiments, we performed grid-search on the m/z and RT windows parameters using the training set. We then used those optimal parameters to perform alignment on the testing set, giving us the respective performance measures (Precision, Recall, F_1) on the testing set. For testing set consisting of multiple fractions, we report the average performance measures on the testing fractions.

For training using the P1 and P2 datasets in the proteomic experiments, the m/z and RT tolerances were varied within: $\{1.0, 1.2, ..., 2.0\}$ for the m/z tolerance, and $\{5, 10, .., 300\}$ seconds for the RT tolerance. The parameter ranges were chosen based on reasonable estimates of the instrument's precision and prior RT tolerance values as reported by [23]. We kept all the default values for the remaining parameters in each evaluated tool, if any. For MWG, we also varied the ratio parameter α from $\{0.1, 0.2, ..., 1.0\}$ and the grouping parameter g_{tol} from $\{1, 2, ..., 10\}$ seconds and uses the combination that results in the best performance. For MWM, the ratio parameter α was varied from $\{0.1, 0.2, ..., 1.0\}$ but mixture model parameters were kept the same for clustering of all fractions in P1 and P2. When clustering all fractions in a dataset, a broad Gaussian prior was set for the component mean μ_j of each cluster j. The component precision s_j was set to 5 seconds, while the DP concentration parameter γ is set to 1. We drew 2000 posterior samples (with 1000 initial burn-in samples) for each run during the Gibbs sampling steps to construct the probability matrix of peak-vs-peak to be in the same cluster.

For the Metabolomic and Glygomic experiments, 30 pairs of run were randomly extracted from the M1 metabolomic dataset in [23] and from the glycomic dataset in [1]. These were assigned to be the training sets. Another 30 pairs of runs were extracted from each dataset to be the testing sets. Each pair of runs in the training set is assigned a partner pair of runs in the testing set. Parameters were optimised on pairwise runs in the training set and performance evaluated on the assigned partner runs in the testing set. For both datasets, the m/z tolerances used were $\{0.05, 0.1, 0.25\}$ and RT $\{5, 10, 15, ...100\}$ seconds. These ranges of parameters were selected in view of instrument accuracy and RT noise level of the LC-MS instruments that generate our metabolomic data and in [1]. The ratio parameter α was from $\{0.1, 0.2, ..., 1.0\}$ and the grouping parameter g_{tol} from $\{2, 4, ..., 10\}$ seconds for both datasets, and for the metabolomic dataset where chromatographic peak shapes information is available and used for greedy clustering in MWG, the threshold for the Pearson correlation coefficient between peak shape signals was varied from $c = \{0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$.

4.7 Results and Discussions

4.7.1 Proteomics Experiments

Single-fraction Experiment

Both P1 and P2 data consist of multiple fractions. In the first experiment, we investigate the best possible performance by using the same fraction as training and testing sets. As described in Section 4.6.6, on each training set (a fraction), we optimised the m/z and RT window parameters for alignments. The m/z parameters are in parts per million, normally notated 'ppm' and the range of m/z parameters used were $\{1.0, 1.1, ..., 2.0\}$ and RT $\{5, 10, ..., 300\}$ seconds. Parameters that control the grouping and influence of the cluster similarity score for our MWG and MWM methods were also optimised. The ratio parameter α was set to $\{0.1, 0.2, ..., 1\}$ for both MWG and MWM. The grouping tolerance g_{tol} was set to $\{1, 2, ..., 10\}$ seconds for greedy clustering, while the same hyperparameters were used for clustering of all fractions in case of mixture-model clustering (further details on parameter range selections are in Section 4.6.6).

The results are shown in Tables 4.5 and 4.6. We see that approximate maximum weighted matching (MW) alone performs competitively to other tools. On the P1 data (Table 4.5), incorporating grouping information (MWG, MWM) improves F_1 score performance over MW. MWG outperforms MWM, which may be due to the fact that the greedy approach is easier to optimise. For the P2 data (Table 4.6), which contains features with significantly higher RT drift across runs, again we find that MW is competitive and clustering information (MWG) improves performance for all fractions. The results here show the potential of our proposed approach: any peak grouping results expressed in a suitable matrix format can be incorporated into our method, and used as additional information during the matching stage. Figure 4.2 shows how the benefit of incorporating clustering information is realised during matching: it allows the matching methods to explore regimes in the solution space having higher precision and recall values. On some training fractions, both methods that incorporate clustering information show significant increases in the best possible F_1 score. For dataset P1 fraction 000, this is an 11%-improvement for MWG and a 7.5%-improvement for MWM. For dataset P2 fraction 100, this is a 51%-improvement for MWG and 25%-improvement for MWM. Smaller improvements can be observed from other fractions in the Proteomic datasets too.

Multiple-fractions Experiment

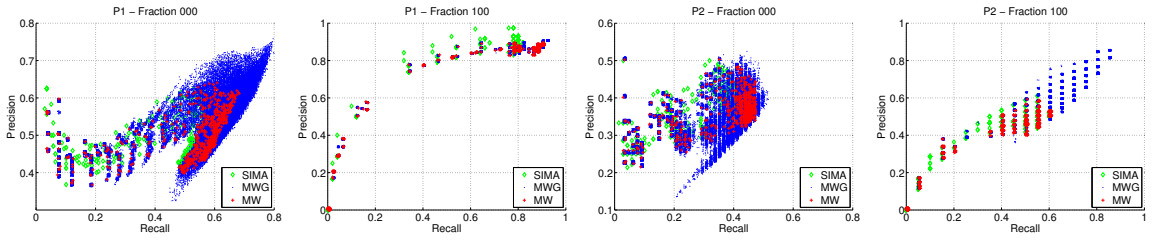
The single-fraction experiment does not represent a very realistic scenario as the optimal parameters were determined with respect to an alignment ground truth; practitioners might

Fraction	Join	SIMA	MW	MWG	MWM
000	0.63	0.64	0.64	**0.77**	0.71
020	0.88	0.88	0.88	**0.95**	0.90
040	0.82	0.83	0.85	**0.87**	0.86
060	0.76	0.78	0.78	**0.88**	0.83
080	0.90	0.89	0.88	**0.92**	0.90
100	0.89	0.89	0.89	**0.91**	0.91
Mean	0.81	0.82	0.82	**0.88**	0.85

Table 4.5: F_1 scores for the single-fraction experiment results on the P1 dataset. The tool with the highest F_1 score for each fraction is highlighted in bold. The results for 'All' show the average F_1 scores of individual fractions.

Fraction	Join	SIMA	MW	MWG	MWM
000	0.45	0.45	0.45	**0.49**	0.45
020	0.77	0.78	0.79	**0.80**	0.79
040	0.77	0.78	0.77	**0.80**	0.77
080	0.66	0.68	0.67	0.67	**0.72**
100	0.55	0.58	0.56	**0.85**	0.70
Mean	0.64	0.65	0.65	**0.72**	0.69

Table 4.6: F_1 scores for the single-fraction experiment results on the P2 dataset. The tool with the highest F_1 score for each fraction is highlighted in bold. The results for 'All' show the average F_1 scores of individual fractions.



(a) P1 Fraction 000 (b) P1 Fraction 100 (c) P2 Fraction 000 (d) P2 Fraction 100

Figure 4.2: Precision and recall training performance for all parameters (m/z, RT tolerance, α and g_{tol}) varied in the experiment for the fractions containing the most (Fig. 2a & 2c) and least (Fig. 2b & 2d) number of features in the P1 and P2 datasets. **REDRAW IN MATPLOTLIB**

not possess that information in real analytical situations. In this experiment, we improved upon the single-fraction experiments by using each fraction in each dataset as the training set and the remaining fractions as the testing set. Parameters were optimised on the training set and performance evaluations were performed on the testing set. This training-testing procedure produces 6 measurements for P1 and 5 measurements for P2, corresponding to the number of training fractions in each dataset. The overall F_1 score reported for each measurement is the average F_1 scores from individual testing fractions. The aim of this experiment is to investigate how well the different methods generalise to data that may have slightly different characteristics from that used to optimise the parameters – i.e. how critical the particular parameter values are.

Tables 4.7 and 4.8 show the F_1 score across measurements. On P1, the best overall performance is achieved by our methods that incorporate clustering information into alignment (MWG, MWM). On P2, the results are less homogeneous, with no method consistently performing best on all the different testing fractions. In the case of the noisiest data (dataset P2 fraction 000), our proposed approach incorporating greedy clustering (MWG) shows a decrease in overall testing performance instead. This is because the greedy clustering method used is sensitive to the choice of parameters and do not generalise well across the different fractions of P2. For instance, the best MWG's grouping tolerance parameter for fraction 000 is 5 seconds, while it is 1 second for fraction 080. The results suggest the dependence of our methods on the quality of groupings of related peaks in order to generalise well on different runs. The heterogeneous testing performance in the multiple-fractions experiment of P2 shows that no method performs best and the choice of optimal parameters that work for certain runs do not generalise well to others on datasets with very high RT variability.

Training Frac.	**Testing Performance**				
	Join	**SIMA**	**MW**	**MWG**	**MWM**
000	0.82	0.85	0.82	**0.86**	**0.86**
020	0.78	0.76	0.78	**0.79**	0.75
040	0.78	0.76	0.77	0.79	**0.81**
060	0.78	0.78	0.77	**0.84**	0.83
080	0.71	0.73	0.72	0.77	**0.78**
100	0.75	0.77	0.74	0.76	**0.78**

Table 4.7: Multiple-fractions experiment results for the P1 dataset. For each training fraction, the reported testing performance is the average of individual F_1 scores from the testing fractions. The top-performing method (highest F_1 score) is highlighted in bold.

Training Frac.	Testing Performance				
	Join	SIMA	MW	MWG	MWM
000	0.62	**0.64**	0.61	0.48	0.61
020	**0.58**	0.56	0.55	0.43	0.55
040	0.52	**0.56**	**0.56**	0.41	**0.56**
080	0.56	0.50	0.50	0.50	**0.57**
100	**0.63**	0.57	0.56	0.44	0.57

Table 4.8: Multiple-fractions experiment results for the P2 dataset. For each training fraction, the reported testing performance is the average of individual F_1 scores from the testing fractions. The top-performing method (highest F_1 score) is highlighted in bold.

4.7.2 Metabolomic and Glycomic Datasets

We further explore the performance of our proposed methods on the metabolomic and glycomic datasets. From the full dataset, we randomly extracted 30 pairs of runs as the training sets and another 30 pairs of runs as the testing sets. Each training set is paired to a testing set. Parameters were optimised on the training set and the best attainable performance reported as the training performance. Generalisation performance is evaluated on testing sets using the optimal parameters from the training stage.

Figures 4.3 and 4.4 summarise the results from the experiments. We see that all methods perform better on the glycomic set than on the metabolomic set. This is explained by the fact that the metabolomic runs represent a generally more challenging alignment scenario with significantly more features to align. MW performs identically to SIMA on both datasets due to the similar form of Mahalanobis distance function used. This is despite the differences in the actual matching method that establishes feature correspondences in SIMA and MW, emphasising the fact that the actual choice of matching function might be less important than other factors, such as the determination of similarity scores between peaks. On the glycomic dataset, adding clustering information improves the training performance, with an increase in the mean of the F_1 scores across 30 measurements from 0.89 (MW) to 0.93 (MWG) and 0.92 (MWM). This also translates into statistically significant improvements on the testing sets for both MWG ($p=0.01$, paired t-test) and MWM ($p=0.002$, paired t-test) over MW.

On the metabolomic dataset, where it is potentially harder to produce good clustering results due to the larger number of peaks and the more complex elution profile, we observe improvements in the mean of the F_1 scores from 0.83 (MW) to 0.90 (MWG) and 0.85 (MWM) on the training sets. These are also statistically significant improvements for both MWG ($p<0.001$, paired t-test) and MWM ($p<0.001$, paired t-test) over MW. The training results confirm our hypothesis that indeed incorporating clustering information (by modifying the similarity matrix used for matching in the proposed manner) can be used to help improve matching results over the case when such information is not used. However, this does not

translate into any statistically significant improvements on the testing sets, suggesting that for the metabolomic dataset evaluated here, our proposed methods are also sensitive to parameter choices, and the choices of particular parameters (especially for the clustering step) that work on some runs may not generalise well to others. The results shown for running MWG on the metabolomic data in Figures 4.3 and 4.4 takes into account the Pearson correlations of the chromatographic shapes between peak features during the clustering process, since that information is available and straightforward to incorporate into the greedy clustering process. Results for MWG that consider only the RT values are presented and discussed in the following paragraph.

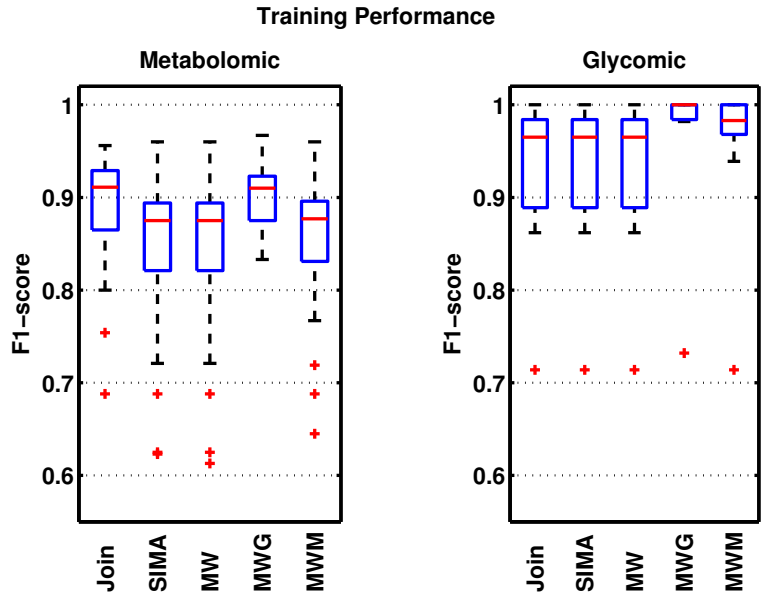


Figure 4.3: Training performance shows the best F_1 scores obtained by each method on 30 pairs of randomly-selected metabolomic and glycomic training sets.

We also compared the results for MWG on both the training and testing sets on the standard metabolomic dataset when the greedy grouping is performed using only RT information (MWG (RT)) and when chromatographic peak shape correlations are also considered (MWG(RT+PS)) during the grouping process. Statistically significant differences can be observed on the training performance of Figure 4.5, with the mean of F_1 scores for MW 0.83, MWG(RT) 0.88 and MWG(RT+PS) 0.90. However, this does not translate to any improvements on the testing sets, with the mean of F_1 scores for MW 0.86, MWG(RT) 0.83 and MWG(RT+PS) 0.85. Introducing clustering information when only RT information is used during the clustering process (MWG(RT)) reduces testing performance. The training results suggest that where additional information such as chromatographic peak shapes are available, they should be used for the clustering step in the proposed methods. However, the lack of any statistically significant testing improvements between MW and MWG (RT+PS), suggest that the optimal parameters from training runs do not generalise well to different testing

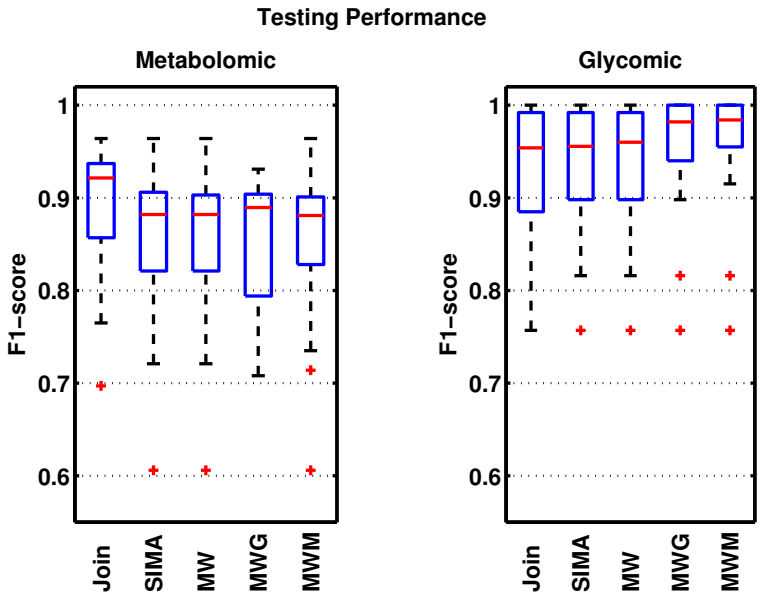


Figure 4.4: Testing performance shows how well each method generalise on the 30 different testing sets, each evaluated using the optimal training parameters from its corresponding training set.

runs for the greedy clustering approach in general, especially for complex metabolomic runs, with large number of features that tend to closely co-elute with each other.

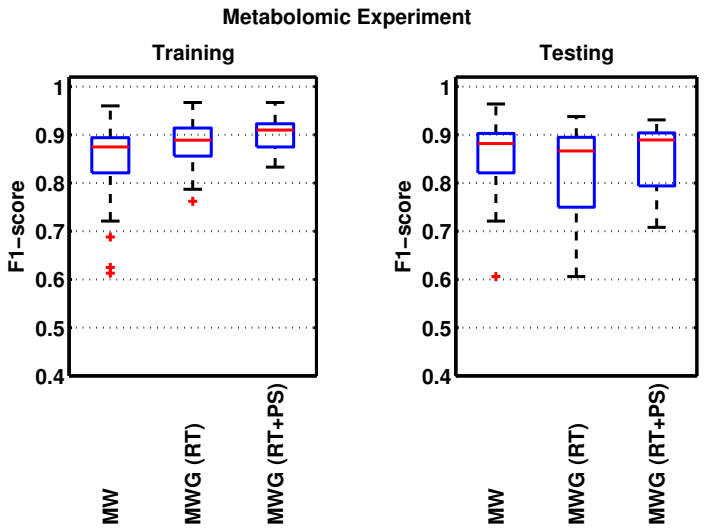


Figure 4.5: Comparisons in matching performance when greedy clustering with retention time (MWG(RT)) and peak shape correlations (MWG(RT+PS)) are used.

4.7.3 Running Time

Computational times of the proposed methods are primarily affected by the number of features in the runs being aligned and to some extent, the thresholding parameters used during

similarity score computation and feature matching. Table 4.9 reports the measured running time for each proposed method using the parameters that give the best training performance). For each fraction being aligned, the running times were measured three times on a standard laptop with Intel Core i5 CPU running at 2.5 GHz, and the average value reported for matching only (MW), matching incorporating greedy clustering (MWG) and matching incorporating mixture model clustering (MWM). The time complexity of the mixture-model clustering step in MWM is $O(N)$ where N is the number of features in the run being clustered. We took 2000 posterior samples, discarding the first 1000 samples during the burn-in period. The number of samples were chosen to ensure convergence to the stationary distribution during inference.

Fraction	Total Features	MW	MWG	MWM
000	10606	9	12	2700
020	2135	1	2	524
040	2188	2	2	540
060	3342	2	3	825
080	2086	2	2	505
100	1326	1	2	321

Table 4.9: Example measured execution time in seconds on fractions of the P1 dataset

4.8 Conclusion

In this chapter, we have proposed a novel peak matching method that incorporates related peak information to improve alignment performance. The method takes related peak information in the form of peak-by-peak binary or real-valued similarity matrices and as such is independent of the particular method used to compute these. The method fits into the category of *direct matching* approaches – those alignment approaches that do not perform an explicit time-warping phase. Our experimental results demonstrate the potential of this approach. From the training results, we see evidence of performance improvement across all evaluated datasets by incorporating grouping information into the matching process in the proposed manner. With the exception of the metabolomic dataset, both the greedy and model-based clustering approaches evaluated in our experiments rely only on the RT information for grouping related peaks. By looking at the testing performance, our results also explore the ability of the evaluated methods to generalise on different runs using less than optimal parameters. This is important because in the actual analytical situation of LC-MS data, neither the optimal parameters nor the alignment ground truth is known.

Note that our method relies on grouping of related peaks, and this introduces additional user-defined parameters. However, as our experiments have shown, in some settings, it may be

much easier to produce good groupings of related peaks than accurately determine RT window parameters (the same grouping parameters were used for all evaluation datasets in the case of mixture-model clustering). Depending on the nature of the data, parameters relating to within-run characteristics (e.g. RT window for grouping related peaks) may be more likely to generalise across runs and experiments than parameters relating to between-run characteristics (particularly RT). For example, changes in the liquid chromatography (LC) column would likely result in related-peaks still co-eluting but could significantly change the absolute RT.

It would be interesting to investigate in greater detail any performance improvements that can be obtained from using other peak grouping methods, such as [59] that uses a mixture model of peak shape correlations or [38] that considers the dependencies between adduct and isotopic peaks when clustering. Exploring alternative approximate matching algorithms (such as the scaling algorithm in [56], which provides a $(1 - \epsilon)$ approximation of the maximum weighted matching in optimal linear time for any ϵ) and evaluating the benefits of incorporating different clustering approaches into our proposed alignment method are avenues for future work. Finally, the different alignment methods evaluated in this chapter also suffer from variable behaviours depending on the order of the runs being aligned [7]. This is particularly true in the case of alignment of multiple runs (typical in large-scale LC-MS studies), where the final alignment results are often constructed through merging of intermediate alignments of pairwise runs. Different alignment methods may employ a different merging approach, for example, Join merges the intermediate results towards a reference run while SIMA allows the possibility of using a greedy hierarchical merging scheme. Systematic evaluation on how the chosen merging scheme may influence alignment performance is beyond the scope of this chapter and is an item for future work.

The related-peak based similarity score that underpins our approach could be applied to many other direct matching approaches, e.g. SIMA: [27] and similar ideas could also be incorporated into recently developed methods that take into account the presence of internal standards [1]. The evaluation datasets and pipeline developed over the course of our experiments in this chapter serves as the foundation for performance evaluations in subsequent chapters that follow.

Chapter 5

Providing Confidence Values in Alignment Results

5.1 Introduction

The goal of establishing the matching of peaks across multiple runs at once can be viewed as a clustering problem, where a set of peaks can be grouped (by their m/z, RT and other suitable features) into local clusters within each run (representing all of the peaks from an individual compound), which are further grouped into global clusters shared across runs. A preliminary form of this idea has been explored in [60], where hierarchical clustering is performed on the total ion chromatogram data to group peaks into within-run local clusters, which are further grouped into across-run super clusters. The highly accurate mass information available from modern LC-MS platforms is not used in [60], although it is highlighted as a possible future work. The choice of using a hierarchical clustering method in [60] also requires choosing various user-defined parameters, such as determining a suitable cut-off for the dendogram produced, deciding on a suitable linkage method and defining an appropriate distance measure between groups of peaks.

According to [18], the common shortcomings shared by many alignment methods include the incorrect modelling assumption that elution order of peaks is preserved across runs and the abundance of user-defined parameters, which can dramatically influence alignment results. Further uncertainties can be introduced due to the selection of a reference run and the construction of a guide tree in hierarchical methods. Since alignment is such an important part of the data preprocessing steps, it would be useful to be able to robustly identify the uncertainty or confidence in the alignment results. The subject of identifying and quantifying uncertainty has been extensively investigated in the problem of multiple sequence alignment (MSA) for genomics and transcriptomics. [61] attempt to quantify the alignment uncertainty of the popular MSA tool ClustalW [62], based on evaluations using synthetic

data, and concludes that between half to all columns in their benchmark MSA results contain alignment errors. [63] construct a score that reflects the consensus between all possible pairwise alignments in T-COFFEE, while [64] propose GUIDANCE, a confidence measure obtained from pertubations of guide trees. Statistical approaches that provide a measure of confidence in alignment results have also been explored by [65] and [66], where the MSA results and phylogeny are constructed simultaneously, thus eliminating the need for a guide tree.

Additionally, it is also desirable for alignment methods to provide some measure of confidence in the quality of its alignment. In the absence of ground truth information, the user typically measures alignment quality through manual inspection or by comparing and visualising the summary statistics (e.g. median, standard deviation of retention time) across different replicates. Alignment methods with confidence values is a big research gap that, to our knowledge, has not been addressed at all by any of the alignment tools surveyed earlier. Some interactive analysis tools like MAVEN [67] can assign quality scores to individual peaks. This is accomplished by training a neural network (or a decision tree) on training data that have been manually annotated using metrics of peak quality. Other approach like [68] computes the Pearson correlations between intensity profiles of all peaks across replicates. Moving from these approaches towards a robust method that can provide confidence values for groups of aligned peaks across many label-free experiments is challenging research problem.

Despite the clear benefits of alignment uncertainty quantification in the sequence domain, the challenge of quantifying alignment results remains relatively unaddressed for the alignment of multiple runs in LC-MS-based-omics. Bayesian methods operating on profile data (e.g. [69, 70, 1]) and feature-based alignment methods (e.g. [71, 24, 27]) exist to correct RT drift, but in such methods, uncertainties are not propagated from the RT regression stage to the necessary peak matching stage that follows. Several recent feature-based alignment methods incorporate probabilistic modelling as part of their workflow, making it possible to extract some form of scores or probabilities on the alignment results. These methods are often limited to the alignment of two runs, which is not a realistic assumption in actual LC-MS experiments. For example, [72] propose an empirical Bayes model for pairwise peak matching. Matching confidence can be obtained from the model in form of posterior probability for any peak pair in two runs, however constructing multiple alignment results in [72] still requires a greedy search to find candidate features within m/z and RT-RT tolerances to a predetermined set of 'landmark' peaks. [73] describe PeakLink, a workflow for alignment that performs an initial warping using a fourth-degree polynomial. PeakLink poses the pairwise matching problem as a binary classification problem, where a Support Vector Machine (SVM) is trained based on an alignment ground truth derived from MS-MS information and used to differentiate matching and non-matching candidate pairs to produce the actual align-

ment results. While not explicitly included in the output of PeakLink, a matching score can be extracted from the SVM that represents how far each candidate pair is from the decision boundary. Note that these scores are not well-calibrated in the probabilistic sense, thus making comparisons of matching scores less straightforward. PeakLink is also not extended to the problem of aligning multiple runs, although [73] state that it would be possible to do so with the choice of a suitable reference run.

In this work, we expand upon the idea of viewing direct matching as a hierarchical clustering problem by proposing **HDP-Align**, a Bayesian non-parametric model that groups related-peaks within runs and assigns them to global clusters shared across runs. Within each global cluster, peaks are further grouped by their m/z values into mass clusters, which represent the various ionisation products (IPs) derived from the global compound. The proposed model allows us to infer the matching of peaks across all runs at once, without the need for any intermediate merging of pairwise runs, and the resulting posterior summaries provide us with a confidence score in the matching quality of aligned peaksets. This introduces the possibility of allowing the user to trade recall for precision from the alignment results by returning a smaller subset of the results having a higher confidence score of being correctly aligned. Figure 5.1 shows an illustration of the clustering process in HDP-Align. Additionally, the latent variables of clustering structure inferred in the model can potentially have physically meaningful identities that can be used for further analysis, and using a metabolomic dataset, we demonstrate the usefulness of such clustering objects by using the mass clusters derived from the model to perform putative annotations of features based on their potential adduct types and metabolite identities.

5.2 Hierarchical Dirichlet Process Mixture Model for Alignment

5.2.1 Model Description

The proposed model for HDP-Align is framed as a Hierarchical Dirichlet Process (HDP) mixture model [74], described further in the background in Section 3.5. Essential modifications to the basic HDP model were performed to suit the nature of the multiple peak alignment problem. Our input consists of J input files, indexed by $j = 1, ..., J$, corresponding to the J LC-MS runs to be aligned. Each j-th input file contains N_j peak features in total, which can be separated into K_j local clusters of related-peak features. In a j-th file, peak features are indexed by $n = 1, ..., N_j$ and local clusters are indexed by $k = 1, ..., K_j$. Across all files, we assign each local cluster k in file j to a global cluster $i = 1, ..., I$, where I is the total number of global clusters, using the indicator variable v, as described in the fol-

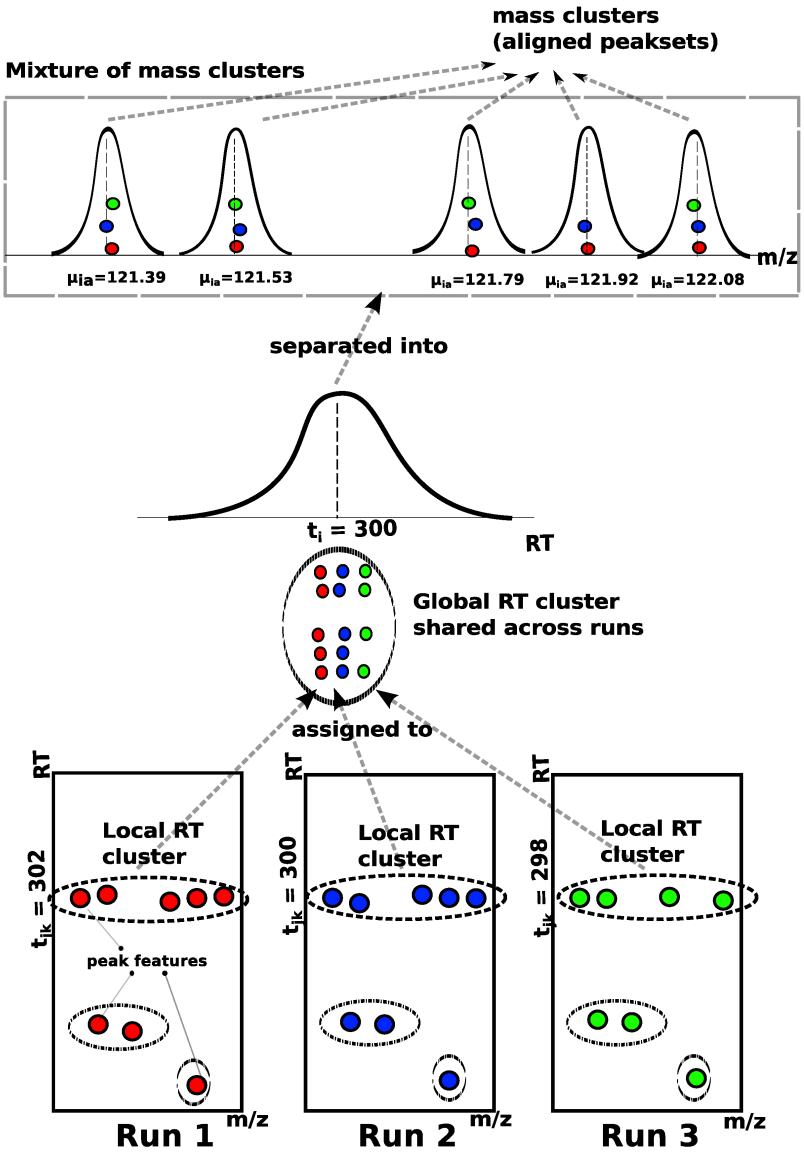


Figure 5.1: An illustrative example of how the proposed model in HDP-Align simultaneously **(1)** performs the clustering of related peak features into within-run local clusters by their RT values, **(2)** assigns the peak features to global RT clusters shared across runs, and **(3)** separates peak features into mass clusters, which correspond to aligned peaksets.

lowing paragraph. A global cluster corresponds to the compound of interest during LC-MS analysis, e.g. metabolite or peptide fragment, that is present across runs, while local clusters are realisations of the global clusters in a specific run. Finally, within each global cluster i, we can further group peak features by their m/z values into A mass clusters (indexed by $a = 1, ..., A$) corresponding to the ions produced by different adduct-isotope combinations of the global compound during the MS process. To be specific, we call these peaks the ionisation products (IPs) from here on, rather than using the term 'related peaks' as in the previous chapter. Related/IP peaks have previously been elaborated further in Section 2.4.1.

We use the indicator variable $\boldsymbol{z}_{jnk} = 1$ to denote the assignment of peak n in file j to local cluster k in that file. Similarly, $\boldsymbol{v}_{jni} = 1$ if peak n in file j is assigned to global cluster i, and $\boldsymbol{v}_{jnia} = 1$ if peak n in file j is assigned to mass cluster a linked to metabolite i. Let d_j be the list of observed data of peak features in file j, $d_j = (\mathbf{d}_{j1}, \mathbf{d}_{j2}, ..., \mathbf{d}_{jn})$ where $\mathbf{d}_{jn} = (x_{jn}, y_{jn})$ with x_{jn} the RT value and y_{jn} the log m/z value of the peak feature. $\boldsymbol{\theta}$ denotes the global mixing proportions and $\boldsymbol{\pi}_j$ the local mixing proportions for file j. The global mixing proportions $\boldsymbol{\theta}$ are distributed according to the Griffiths, Engen and McCloskey (GEM) distribution:

$$\boldsymbol{\theta}|\alpha' \sim GEM(\alpha') \tag{5.1}$$

where the GEM distribution over $\boldsymbol{\theta}$ is described through the stick-breaking construction:

$$\beta_i \sim Beta(1, \alpha') \tag{5.2}$$

$$\theta_i = \beta_i \prod_{l=1}^{i-1}(1 - \beta_l) \tag{5.3}$$

The local mixing proportions $\boldsymbol{\pi}_j$ are distributed according to a Dirichlet Process (DP) prior with the base measure $\boldsymbol{\theta}$ and concentration parameter α_t.

$$\boldsymbol{\pi}_j|\alpha_t, \boldsymbol{\theta} \sim DP(\alpha_t, \boldsymbol{\theta}) \tag{5.4}$$

Within each file j, the indicator variable $\boldsymbol{z}_{jnk} = 1$ denotes the assignment of peak n in file j to local RT cluster k in that file. This follows the local mixing proportions for that file.

$$\boldsymbol{z}_{jnk} = 1|\boldsymbol{\pi}_j \sim \boldsymbol{\pi}_j \tag{5.5}$$

The RT value t_i of a global mixture component is drawn from a base Gaussian distribution with mean μ_0 and precision (inverse variance) σ_0.

$$t_i|\mu_0, \sigma_0 \sim \mathcal{N}(\mu_0, \sigma_0^{-1}) \tag{5.6}$$

The RT value t_{ij} of a local mixture component in file j is normally distributed with mean t_i

and precision δ. The precision controls how much RT values of related-peak groups across runs are allowed to deviate from the parent global compound's RT.

$$t_{jk}|t_i, \delta \sim \mathcal{N}(t_i, \delta^{-1}) \tag{5.7}$$

Finally, the observed peak RT value is normally distributed with mean t_{jk} and precision γ. The precision controls how much RT values of peaks can deviate from their related-peak group.

$$x_{jn}|\boldsymbol{z}_{jnk} = 1, t_{jk}, \gamma \sim \mathcal{N}(t_{jk}, \gamma^{-1}) \tag{5.8}$$

The m/z value produced through high-precision MS instrument is highly accurate, and its correspondence is often preserved across runs. Once peaks have been assigned to their respective global clusters, we need to further separate peaks within each global cluster into mass clusters to obtain the actual alignment. These mass cluster corresponds to ionisation products. We do this by incorporating an internal DP mixture model on the m/z values (y_{jn}) within each global cluster i. Let the indicator $\boldsymbol{v}_{jnia} = 1$ denotes the assignment of peak n in file j to mass cluster a in the i-th global cluster. Then:

$$\begin{aligned}
\boldsymbol{\lambda}_i|\alpha_m &\sim GEM(\alpha_m) && (5.9)\\
\boldsymbol{v}_{jnia} = 1|\boldsymbol{\lambda}_i &\sim \boldsymbol{\lambda}_i && (5.10)\\
\mu_{ia}|\psi_0, \rho_0 &\sim \mathcal{N}(\mu_{ia}|\psi_0, \rho_0^{-1}) && (5.11)\\
y_{jn}|\boldsymbol{v}_{jnia} = 1, \mu_{ia} &\sim \mathcal{N}(\mu_{ia}, \rho^{-1}) \cdot I(\mathbf{d}_{jn}) && (5.12)
\end{aligned}$$

where the index ia refers to the a-th mass cluster of the i-th global cluster. $\boldsymbol{\lambda}_i$ is the mixing proportions of the i-th internal DP mixture for the masses, with α_m the concentration parameter. μ_{ia} is the mass cluster mean, drawn from the Gaussian base distribution with mean ψ_0 and precision ρ_0. The observed mass value is drawn from a Gaussian distribution with the component mean μ_{ia} and precision ρ, for which the value is set based on the MS instrument's resolution. Additionally, we add an additional constraint on the likelihood of y_{jn} using the indicator function $I(\cdot)$ such that $I(\mathbf{d}_{jn}) = 1$ if there are no other peaks inside the mass cluster that come from the same file as the current $\mathbf{d}_{jn}$ peak, and 0 otherwise. This constraint captures the restriction that a peak feature can only be matched to other peaks from different files, reflecting the assumption that within each LC-MS run, compounds produce ionisation products with distinct mass-to-charge fingerprints that can be used for matching to other runs.

5.2.2 Inference

Inference within the model is performed via a Gibbs sampling scheme, allowing us to compute posterior probabilities over the alignment of any set of peaks across the J files via the proportion of posterior samples in which they are assigned to the same mass component (a) in the same top-level cluster. In each iteration of the sampling procedure, we instantiate the mixture component parameters for the local RT cluster (t_{jk}) and global RT cluster (t_i) in the mixture model. In the internal DP mixture linked to each global cluster i, we marginalise out the mass cluster parameters (μ_{ia}). The initialisation step of the sampler is performed by assigning all peaks in each run into a single local RT cluster. Across runs, these local clusters are assigned under a global cluster shared across runs. Within a global cluster, peak features coming from different runs are assigned to a single mass cluster. The sampler than iterates through each peak feature, removing it from the model, updating the assignment of peak features to clusters and performing the necessary book-keeping on any instantiated mixture components. Further details on the specific Gibbs update statements can be found in following sections.

Updating peak assignments

We use the following variables to denote the count of items in any clustering object: c_{jk} is the number of peaks in a local cluster k of file j. c_i is the number of local clusters in a global cluster i, and c_{ia} is the number of peaks in a mass cluster a inside a global RT cluster i. To update the assignment of a peak $\mathbf{d}_{jn}$ to local RT cluster k during Gibbs sampling, we need the conditional probability of $P(\boldsymbol{z}_{jnk} = 1)$ given every other parameters, denoted as $P(\boldsymbol{z}_{jnk} = 1|\mathbf{d}_{jn}, \ldots)$.

$$P(\boldsymbol{z}_{jnk} = 1|\mathbf{d}_{jn}, \ldots) \propto \begin{cases} c_{jk} \cdot p(\mathbf{d}_{jn}|\boldsymbol{z}_{jnk} = 1, ...) \\ \alpha_t \cdot p(\mathbf{d}_{jn}|\boldsymbol{z}_{jnk^*} = 1, ...) \end{cases} \tag{5.13}$$

We will consider the top and bottom terms of eq. 5.13 separately in the following.

1. The likelihood of the peak $\mathbf{d}_{jn}$ to be in an existing local RT cluster k, $p(\mathbf{d}_{jn}|\boldsymbol{z}_{jnk} = 1, ...)$ is proportional to c_{jk}. This is assumed to factorise across the RT (x_{jn}) and mass (y_{jn}) terms

$$p(\mathbf{d}_{jn}|\boldsymbol{z}_{jnk} = 1, ...) = p(x_{jn}|\boldsymbol{z}_{jnk} = 1, ...) \cdot p(y_{jn}|\boldsymbol{z}_{jnk} = 1, ...) \tag{5.14}$$

The RT term $p(x_{jn}|\boldsymbol{z}_{jnk} = 1, ...)$ in eq. 5.14 is normally distributed with mean t_{jk} and precision γ, while the mass term $p(y_{jn}|\boldsymbol{z}_{jnk} = 1, ...)$ is an internal DP mixture of mass

components linked to the parent global cluster i of an existing local cluster k. We then marginalise over all mass clusters in i to get $p(y_{jn}|\boldsymbol{z}_{jnk}=1,\boldsymbol{v}_{jni}=1...)$

$$\begin{aligned} p(y_{jn}|\boldsymbol{z}_{jnk}=1,\boldsymbol{v}_{jni}=1...) &= \sum_a \frac{c_{ia}}{\alpha_m+\sum_a c_{ia}} p(y_{jn}|\boldsymbol{v}_{jnia}=1,...) \\ &+ \frac{\alpha_m}{\alpha_m+\sum_a c_{ia}} p(y_{jn}|\boldsymbol{v}_{jnia^*}=1,...) \end{aligned} \tag{5.15}$$

To compute the terms in eq. 5.15, first we consider the case for an existing mass cluster a in the global RT cluster i. Then,

$$p(y_{jn}|\boldsymbol{v}_{jnia}=1,...)=\mathcal{N}(\mu_{ia},\rho^{-1}) \tag{5.16}$$

For a new mass cluster a^* in the global RT cluster i, we marginalise out μ_{ia} to obtain

$$p(y_{jn}|\boldsymbol{v}_{jnia^*}=1,...)=\mathcal{N}(\psi_0,\rho^{-1}+\rho_0^{-1}) \tag{5.17}$$

2. The likelihood of the peak $\mathbf{d}_{jn}$ to be in a new local cluster k^* is proportional to α_t. Marginalising over all global clusters, we get

$$\begin{aligned} p(\mathbf{d}_{jn}|\boldsymbol{z}_{jnk^*}=1,...) &= \sum_i \left[\frac{c_i}{\alpha'+\sum_i c_i} p(\mathbf{d}_{jn}|\boldsymbol{v}_{jni}=1,...)\right] \\ &+ \frac{\alpha'}{\alpha'+\sum_i c_i} p(\mathbf{d}_{jn}|\boldsymbol{v}_{jni^*}=1,...) \end{aligned} \tag{5.18}$$

There are two terms to compute in eq. 5.18: whether peak $\mathbf{d}_{jn}$ is in an existing global cluster i or a new global cluster i^*. For an existing global RT cluster i in eq. 5.18, $p(\mathbf{d}_{jn}|\boldsymbol{v}_{jni}=1,...)$ is assumed to factorise into its RT and mass terms, so $p(\mathbf{d}_{jn}|\boldsymbol{v}_{jni}=1,...)=p(x_{jn}|\boldsymbol{v}_{jni}=1,...)\cdot p(y_{jn}|\boldsymbol{v}_{jni}=1,...)$. We marginalise over all local RT clusters to obtain

$$p(x_{jn}|\boldsymbol{v}_{jni}=1,...)=\mathcal{N}(x_{jn}|t_i,\gamma^{-1}+\delta^{-1}) \tag{5.19}$$

and marginalise over all possible mass clusters in the internal DP linked to global cluster i to obtain $p(y_{jn}|\boldsymbol{v}_{jni}=1,...)$. This is defined in eq. 5.15). Finally, for a new global RT cluster i^* in eq. 5.18, $p(\mathbf{d}_{jn}|\boldsymbol{v}_{jni^*}=1,...)$ is also assumed to factorise into its RT and mass terms. Then, we marginalise over t_{jk} and t_i to obtain

$$p(x_{jn}|\boldsymbol{v}_{jni^*}=1,...)=\mathcal{N}(x_{jn}|\mu_0,\sigma_0^{-1}+\gamma^{-1}+\delta^{-1}) \tag{5.20}$$

and marginalise over μ_{ia} to get

$$p(y_{jn}|\boldsymbol{v}_{jni^*}=1,...)=\mathcal{N}(y_{jn}|\psi_0,\rho^{-1}+\rho_0^{-1}) \tag{5.21}$$

Updating instantiated variables

The following expressions are used to update the instantiated mixture component parameters in the model during Gibbs sampling.

1. Updating global cluster's RT t_i: here, $t_{jk\in i}$ refers only to local RT clusters currently assigned to the global cluster i, and c_i is the count of such peaks. Then

$$p(t_i|\ldots) \propto p(t_i|\mu_0, \sigma_0^{-1}) \prod_j^J \prod_k^K p(t_{jk\in i}|t_i, \delta) = \mathcal{N}(\mu_i, \gamma_i^{-1}) \qquad (5.22)$$

   where $\mu_i = \frac{1}{\gamma_i}\left[\mu_0\sigma_0 + \delta \sum_j \sum_k t_{jk\in i}\right]$ and $\gamma_i = \sigma_0 + \delta c_i$.

2. Updating local cluster's RT t_{jk}: here, $x_{jn\in k}$ refers only to peaks currently assigned to the local RT cluster k, and c_{jk} is the count of such peaks.

$$p(t_{jk}|\ldots) \propto p(t_{jk}|t_i, \delta^{-1}) \prod_j^J \prod_n^N p(x_{jn\in k}|t_{jk}, \gamma) = \mathcal{N}(\mu_k, \gamma_k^{-1}) \quad (5.23)$$

   where $\mu_k = \frac{1}{\gamma_k}\left[t_i\delta + \gamma \sum_j \sum_n x_{jn\in k}\right]$ and $\gamma_k = \delta + \gamma c_{jk}$.

5.2.3 Using the Inference Results

Feature Matching

The Gibbs sampling procedure produces a collection of samples from the posterior distribution over all parameters of the HDP-Align model. We can use these samples to compute various posterior summaries and more interestingly, extract the alignment of peaks from the inference results (since features assigned into the same mass cluster with the same global RT cluster are considered to be aligned). For each sample from the posterior distribution, we record the aligned peaksets of peak features put into the same mass cluster. Averaging over all samples provides a distribution over these aligned peaksets.

Note that across the returned aligned peaksets, it is possible for the same peak to be matched to different partners with varying probabilities, depending on how often they co-occur together in the same mass cluster. To allow the possibility of controlling precision and recall from the results, we provide another user-defined threshold t, where peak feature combinations are included in the output from the model only when they occur with matching probability $>t$. Varying this threshold allows user to trade precision for recall: a low value for t gives a larger set of results that are potentially less precise, while conversely a high t provides

a smaller, more precise set of aligned peaksets. This is an output not available from other alignment methods and can potentially be useful in problem domains where high precision is required from the alignment results.

Isotopic Product and Metabolite Identity Annotations

As described in Section 2.4.1, in metabolomics studies using electrospray ionisation, a single metabolite can generate multiple ionisation products peaks, (such as isotopic variants, adducts, fragment peaks), alongside other peaks resulting from noise and artifacts introduced during mass spectrometry [2]. Determining and annotating these IP peaks are desirable to remove extraneous peaks and reduce the burden of subsequent downstream analysis. Additionally, deducing the precursor molecular masses that generate the IPs is often essential in order to query compound library databases before assigning putative metabolite identities.

The resulting clustering objects inferred from HDP-Align lend themselves to further analysis in a natural fashion, as global RT clusters in HDP-Align may correspond to metabolites, while local RT clusters may correspond to the noisy realisations of these metabolites within each run. Mass clusters in the internal mixture of each global cluster could correspond to the IPs. To demonstrate the possibility of obtaining additional information beyond alignment from the output of HDP-Align, we follow the workflow in [2] that performs IPs and metabolite annotations of peak features. This workflow is composed of multiple key steps: peak matching, ionisation product clustering and metabolite mass matching. A key difference of HDP-Align to the workflow in [2] lies in the fact that HDP-Align is able to perform the two separate steps of peak alignment and potential IP clustering simultaneously, as shown in Figure 5.2.

Given the set of potential IP clusters, we can perform IP annotation on the peaks. To do this using the metabolomic dataset, first we take the set of clustering objects produced in a single posterior sample. For each mass cluster, we assign its m/z value to be the average m/z values of features assigned to it, denoted by m. A list of common adducts (Table 4.3) in positive ionisation mode is used to compute the inverse transformation $t^{-1}(m, d, e, u) = ((e * m) - d)/u$ for a precursor mass c that generates m. Here, d is the adduct mass, e is the charge and u the number of metabolite molecules in the IP type. Following [2], any two mass clusters sharing the same precursor mass c (within tolerance) provide a vote on the presence of that consensus precursor mass. The respective pair of mass clusters and features within can then be annotated with the adduct type that produces the transformation t^{-1} to the shared precursor mass c. The set of precursor masses deduced in this manner can also be used to query KEGG (a database of metabolite compounds) in order to assign putative identities to global compounds.

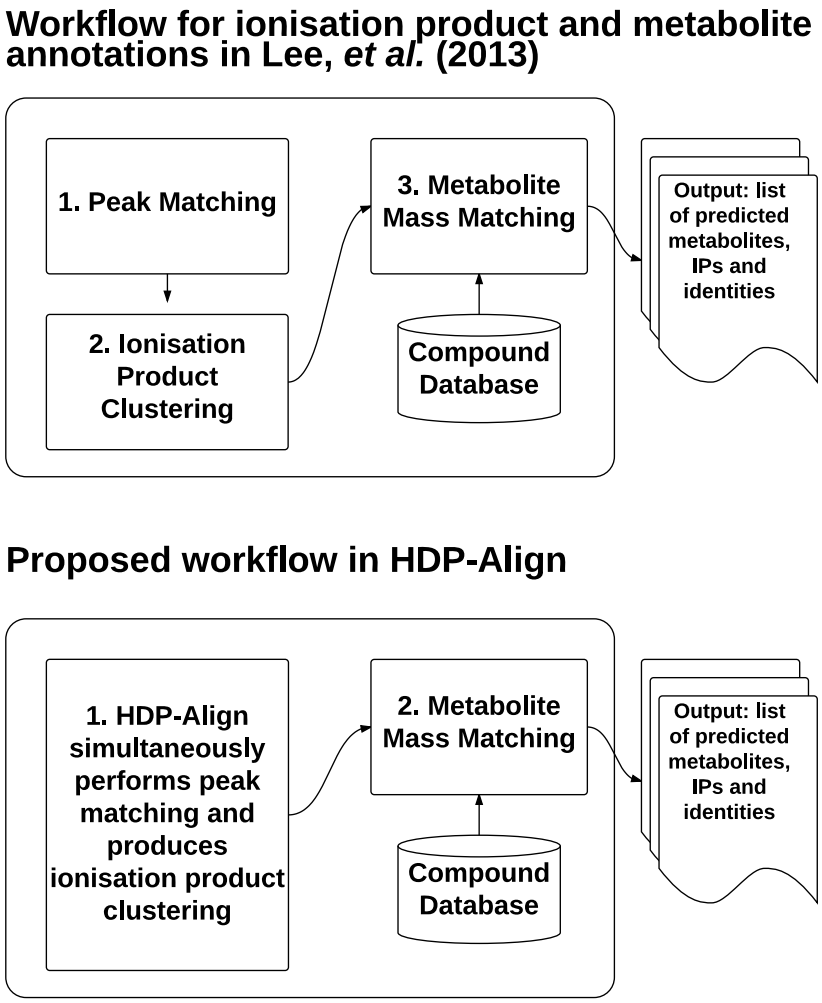


Figure 5.2: Comparisons on the workflow to assign putative annotations on isotopic products and metabolites described in [2] and in HDP-Align.

5.3 Evaluation Study

5.3.1 Evaluation Datasets

Performance of the proposed methods and other benchmark methods is evaluated on the LC-MS datasets of proteomic, glycomic and metabolomic experiments first introduced in Section 4.6. As before, all 6 fractions from the P1 Proteomic dataset in [23] are used. Each fraction contains 2 runs of features having high RT variations across runs are used in our experiments. Unlike Section **??** where only pairs of runs used, here we use the first 10 runs of the Glycomic dataset provided by [1] for our multiple-runs experiment. Additionally, the Standard metabolomic dataset, first introduced in Section **??**, is also used. Here, we selected 6 runs for our experiment. Table 5.1 summarises the different evaluation datasets and the number of features each has.

Dataset	No. runs	Total Features
P1 Frac 000	2	10606
P1 Frac 020	2	2135
P1 Frac 040	2	2188
P1 Frac 060	2	3342
P1 Frac 080	2	2086
P1 Frac 100	2	1326
Glycomic	10	9344
Metabolomic	6	7477

Table 5.1: Total number of runs and features of the selected evaluation datasets.

5.3.2 Performance Measures

While a definition of precision and recall in the context of alignment performance has been proposed and used in Chapter 4, the performance measures defined there applies only to pairwise alignment, i.e. an aligned peakset can only consist of two matched peak features, at most. Here, we propose a generalisation of the performance measures defined in Section 4.6.4 to apply to the alignment of multiple runs.

To provide a definition of 'precision' and 'recall' suitable for evaluating alignment performance of multiple runs, we first enumerate all the possible q-size combinations for every aligned peakset in both the method's output and the ground truth list. For example, an alignment method returns a list of two aligned peaksets $\{a, b, c, d, \}, \{e, f, g\}$ as output. When $q = 2$, this output can be enumerated into a list of 9 'alignment items' of all the pairwise combinations of features: $\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{e, f\}, \{e, g\}$, $\{f, g\}$. Let M and G be the results from such enumeration from a method's output and the

ground truth respectively. Each distinct combination of features in M and G can be considered as an item during performance evaluation. Intuitively, the choice of q reflects the strictness of what is considered to be a true positive item, with larger values of q demanding an alignment method that produces results spanning more runs correctly.

For a given q, the following positive and negative instances of alignment item can now be defined for the purpose of performance evaluation:

- True Positive ($\boldsymbol{TP}$): items that should be aligned (present in G) and are aligned (present in M).
- False Positive ($\boldsymbol{FP}$): items that should not be aligned (absent from G) but are aligned (present in M).
- True Negative ($\boldsymbol{TN}$): items that should not be aligned (absent from G) and are not aligned (absent from M).
- False Negative ($\boldsymbol{FN}$): items that should be aligned (present in G) but are not aligned (absent from M).

In the context of alignment performance, precision ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$) is therefore the fraction of items in M that are correct with respect to G, while recall ($\frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$) is the fraction of items in G that are aligned in M. A method with a perfect alignment output would have both precision and recall values of 1.0.

5.3.3 Benchmarking Method

Following Chapter 4, we benchmark HDP-Align against two established alignment methods: SIMA [27] and MZmine2's Join Aligner [24]. The selection of SIMA and Join as the benchmark methods is motivated by the fact that both methods are direct matching methods (thus easily comparable to HDP-Align) but still differ sufficiently in how they establish the final alignment results, in particular when it comes to the alignment of multiple runs. This is primarily due to the differences between both methods in the form of the distance/similarity function between peak features, the actual matching algorithm itself and the merging order of pairwise results to construct the full alignment results.

The two most important parameters to configure in both methods are the mass and RT tolerance parameters, used for thresholding and computing feature similarities during matching. We label these common parameters as the $T_{(m/z)}$ and T_{rt} parameters. Note that despite the common label, each method may use the parameter values differently during the alignment process. In our experiments, we let $T_{(m/z)}$ and T_{rt} vary within reasonable ranges (details in

Section 5.3.4) and report all performance values generated by each combination of the two parameters.

5.3.4 Parameter Optimisations

Tables 5.2 and 5.3 describe the parameter ranges of each method during performance evaluation. For HDP-Align (Table 5.2), we perform the experiments based on our initial choices on the appropriate parameter values. These are almost certainly less than optimal and can be optimised further. The mass cluster standard deviation $\sqrt{\rho^{-1}}$ for HDP-ALign is set to the equivalent value in parts-per-million (ppm). These are 500 ppm for the Proteomic dataset and 3 ppm for the Glycomic and Metabolomic datasets. The local (within-run) cluster RT standard deviation $\sqrt{\gamma^{-1}}$ is assumed to be fairly constant and set to 2 seconds for all datasets, while the global cluster standard deviation $\sqrt{\delta^{-1}}$ is set in the following dataset-specific manner: 50 seconds for the Proteomic dataset and 20 seconds for the remaining datasets. The larger standard deviation value is required for the Proteomic dataset to accomodate for greater RT drifts across runs. Other hyperparameters in HDP-Align are fixed to the following values: $\alpha' = 10$, $\alpha_t = 10$, $\alpha_m = 100$. The values of the precision hyperparameters for global cluster RT (σ_0) and mass cluster (ρ_0) are set to a broad value of 1/5E6. No significant changes were found to the results when these hyperparameters for the DP concentrations and cluster precisions were varied. The mean hyperparameters μ_0 and ψ_0 are set to the means of the RT and m/z values of the input data respectively. During inference for the Glycomic and Metabolomic datasets, 500 posterior samples were collected for the Gibbs sampling procedure, discarding the first 500 during the burn-in period. For the Proteomic dataset with larger RT deviations, 5000 posterior samples were obtained after discarding the first 5000 samples during burn-in. The number of samples is selected to ensure convergence during inference.

Dataset	HDP
P1 Frac 000	$\sqrt{\rho^{-1}} = 500$ ppm, $\sqrt{\gamma^{-1}} = 2$ s, $\sqrt{\delta^{-1}} = 50$ s
P1 Frac 020	
P1 Frac 040	
P1 Frac 060	
P1 Frac 080	
P1 Frac 100	
Glycomic	$\sqrt{\rho^{-1}} = 3$ ppm, $\sqrt{\gamma^{-1}} = 2$ s, $\sqrt{\delta^{-1}} = 20$ s
Metabolomic	$\sqrt{\rho^{-1}} = 3$ ppm, $\sqrt{\gamma^{-1}} = 2$ s, $\sqrt{\delta^{-1}} = 20$ s

Table 5.2: Parameters used for HDP-Align

For SIMA and Join, we report the results from all combinations of the mass and RT tolerance parameters within reasonable ranges listed in Table 5.3. This follows from the range of parameters selected for evaluation experiments in the previous Chapter 4. The ranges of

$T_{(m/z)}$ and T_{rt} parameters used are based values reported on [23] for the Proteomic dataset and [1] for the Glycomic dataset. For the Metabolomic dataset, they were chosen in light of the mass accuracy and RT deviations of the data.

Dataset	Benchmark (SIMA, Join)
P1 Frac 000	$T_{(m/z)} = \{1.0, 1.1, ..., 2.0\}$, $T_{rt} = \{10, 20, ..., 180\}$ s
P1 Frac 020	
P1 Frac 040	
P1 Frac 060	
P1 Frac 080	
P1 Frac 100	
Glycomic	$T_{(m/z)} = \{0.05, 0.1, 0.25\}$,$T_{rt} = \{5, 10, ..., 120\}$ s
Metabolomic	$T_{(m/z)} = \{0.001, 0.01, 0.1\}$, $T_{rt} = \{5, 10, ..., 120\}$ s

Table 5.3: Parameters used for the benchmark methods (SIMA, Join).

5.4 Results

Precision and recall values for the evaluated methods methods on the different datasets are shown in Sections 5.4.1 and 5.4.2. Additionally, an example of the further annotations for the putative adduct type and metabolite identity that can be produced by HDP-Align is also shown in Section 5.4.2. Running time of the evaluated methods are reported in Section 5.4.3.

5.4.1 Proteomic (P1) Results

Figure 5.3 shows the results from performance evaluation on the Proteomic (P1) dataset. We see that both benchmark methods (SIMA and Join) produce a wide range of performance depending on the parameter values for $(T_{(m/z)}, T_{rt})$ chosen. Sensitivity to parameter values is expected on this dataset due to the low mass accuracy in the MS instrument that produces the data and the high RT drifts present across runs (further details in [23]). HDP-Align performs well on several fractions (particularly fractions 040, 060, 080, 100) with precision-recall performance close to the optimal performance attainable by the benchmark methods. On all fractions, HDP-Align is also able to produce higher-precision results compared to the benchmark methods by reducing recall through setting the appropriate values for the threshold t. The primary benefits of quantifying alignment uncertainties is realised here as the well-calibrated probability scores on the matching confidence of aligned peak features produced HDP-Align allows the user to choose which point along the PR curve to operate on. It is less obvious how this can be accomplished in the benchmark methods by varying the RT (T_{rt}) and m/z ($T_{m/z}$) thresholding parameters, if at all possible.

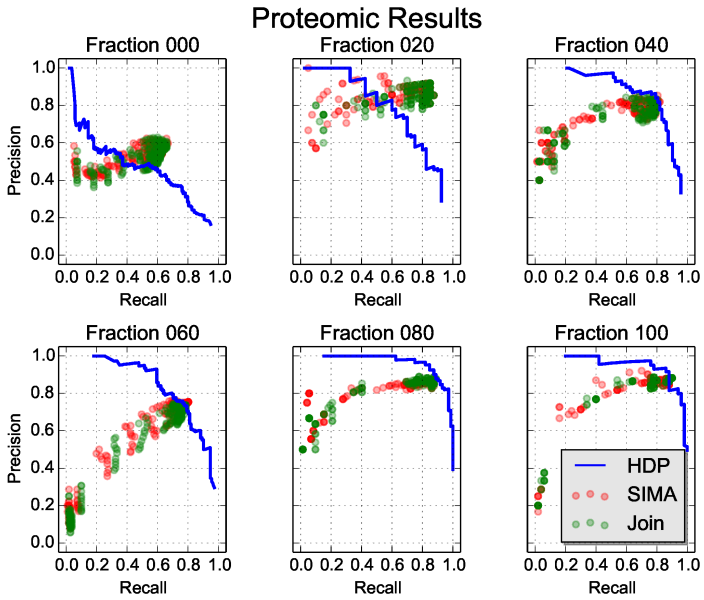


Figure 5.3: Precision-recall values on the different fractions of the Proteomic (P1) dataset.

5.4.2 Glycomic and Metabolomic Results

Figures 5.4 and 5.5 show the results from experiments on the Glycomic and Metabolomic datasets. Similar to the Proteomic dataset, a wide range of precision-recall values can be observed in the results for the benchmark methods on the two datasets. The performance of HDP-Align, using the same set of parameters on both datasets, come close to the optimal results from the benchmark methods, while still allowing the user to control the desired point along the precision-recall curve to operate on.

The results for the Glycomic dataset (Figure 5.4) also show some additional results on how the measured precision-recall values might change depending on the strictness of what constitutes an 'item' during performance evaluation. This is accomplished by gradually increasing the value for q (described in detail in Section 5.3.2) that determines the size of the feature combinations enumerated from a method's output. For example, q=2 considers all pairwise combinations of features from the method's output during performance evaluation, while $q = 4$ considers all combinations of size 4, and so on. Figure 5.4 shows that as q is increased, parameter sensitivity seems to become more of an issue for the benchmark methods, with more parameter sets having lower precisions in the results. Across all qs evaluated, parameter pairs that produce the best alignment performance (points with high precision and recall values) are generally small $T_{(m/z)}$ and large T_{rt} values. Examples of parameter pairs that produce the best and worse performance for SIMA are shown in Figure 5.5. The results here appear to suggest the importance of having high mass precision during matching. Importantly, we see from Figure 5.4 that the performance of HDP-Align remains fairly consistent as q is increased.

The Metabolomic dataset also provides us with additional results in form of annotations of putative adduct type and metabolite identities. A thorough evaluation on the quality of such annotations, in comparison to e.g. the workflow proposed in [2], is beyond the scope of this chapter and would likely necessitate using a different and more appropriate evaluation dataset. Instead, we present an example of the further analysis performed by HDP-Align (as proposed in Section 5.2.3) on the resulting clustering objects after inference. Figure 5.6 shows a global RT cluster where peak features across runs have been grouped by their RT and m/z values. Within this global cluster, peak features are further separated into 6 mass clusters – corresponding to ionisation products produced by the global cluster during mass spectometry. In Figure 5.6, mass cluster A and B contain features aligned from several runs but they do not have any other mass cluster sharing a possible precursor mass. Mass cluster C and D share a common precursor mass (292.12696) and can thus be annotated by the adduct type that produce the transformation. Similarly, mass cluster E and F share a common precursor mass at 383.14278. Queries to a local KEGG database are issued based on the precursor mass values, producing several compound identities that can be putatively assigned to the global RT cluster. It is a great strength of our approach that this putative identification step appears very naturally from the alignment results.

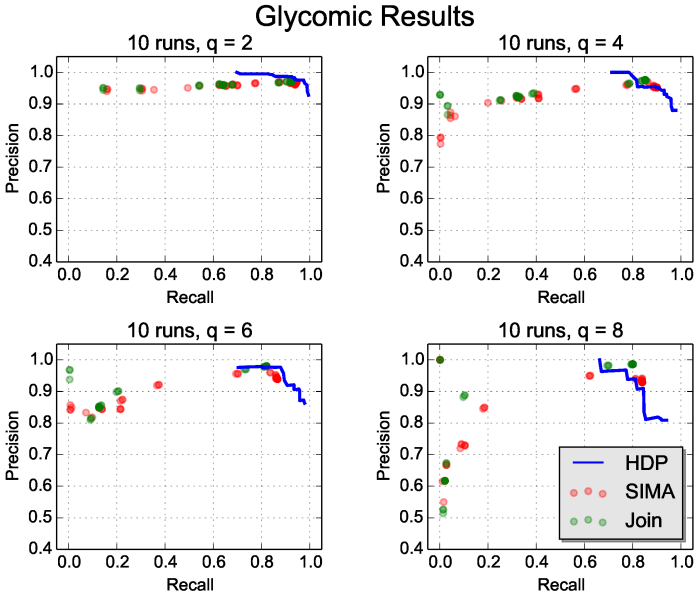


Figure 5.4: Precision-recall values on the alignment of 10 runs from the Glycomic dataset when q (the strictness of performance evaluation as described in Section 5.3.2) is gradually increased.

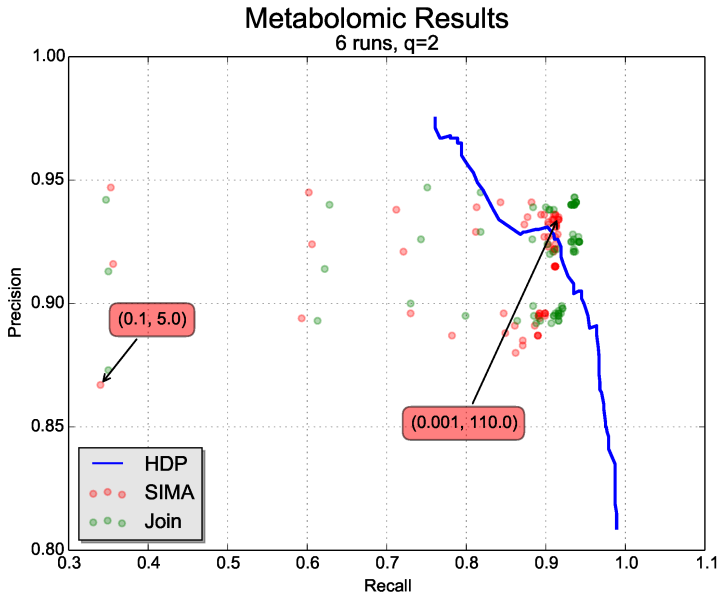


Figure 5.5: Precision-recall values on the alignment of 6 runs from the Metabolomic dataset. The parameter values $(T_{m/z}, T_{rt})$ that produce the best and worst performance in SIMA are also annotated in the Figure (red boxes).

5.4.3 Running Time

The main factor affecting the running time of HDP-Align is the total number of peaks across all runs to be processed and the number of samples produced during Gibbs sampling. In each iteration of Gibbs sampling, HDP-Align removes a peak from the model, updates parameters of the model conditioned on every other parameters, and reassigns a peak into RT and mass clusters. The time complexity of this operation is $O(N)$, where N is the total number of peaks across all runs. In practice, additional time will also be spent on various necessary book-keeping operations, such as deleting empty clusters that are no longer required, updating internal data structures, etc. A representative running time is given as $N = 9344$ for the Glycomic dataset. HDP-Align requires approximately 5 hours to collect 1000 samples. In comparison, both SIMA and Join perform alignment within 5 to 10 seconds. Similarly, for $N = 7477$ for the Metabolomic dataset, HDP-Align produces the results in approximately 4 hours after collecting 1000 samples, while SIMA and Join complete within seconds. The running time of HDP-Align, while being significantly longer than these two benchmark methods, is comparable to other computationally-intensive steps (e.g. peak detection) in a typical LC-MS pipeline.

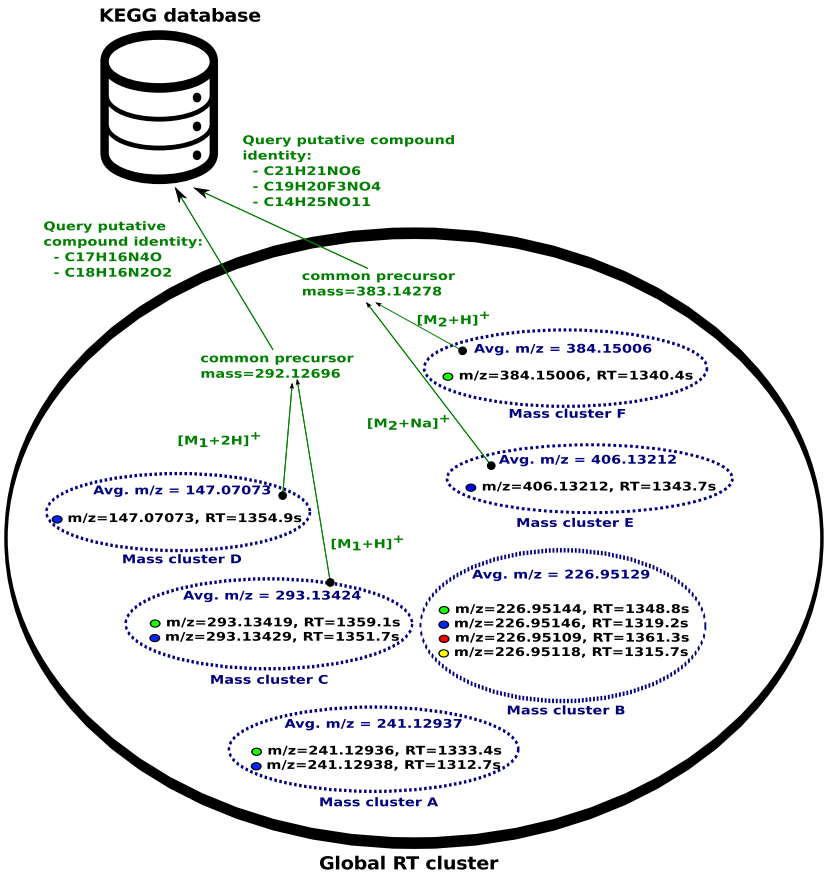


Figure 5.6: Example analysis that can be performed on the clustering objects inferred in a Gibbs sample from HDP-Align. The outer black oval denotes a global RT cluster (generally corresponding to a metabolite compound), while the smaller dotted ovals within denote mass clusters (labelled as mass cluster A, B, C, D, E, F). Peak features are denoted by the filled circle, with the fill colour indicating the originating run of a peak. Green colours denote additional analysis steps that can be performed on the mass cluster objects. **REDRAW TO LOOK NICER?**

5.5 Discussion and Conclusion

We have presented a hierarchical non-parametric Bayesian model that performs direct matching of peak features, a problem of significant importance in the data pre-processing pipeline of large untargeted LC-MS datasets. Unlike other direct matching methods, the novelty of our proposed approach lies in its ability of to produce well-calibrated probability scores on the matching confidence of aligned peak features (evidenced by the increasing precision and decreasing recall as the threshold t is increased). This is accomplished by casting the multiple alignment problem of LC-MS peak features as a hierarchical clustering problem. Matching confidence can then be obtained based on the probabilities of co-eluting peak features to be assigned under the same mass component of the same global cluster. Experiments based on datasets from real proteomic, glycomic and metabolomic experiments show that HDP-Align is able to produce alignment results competitive to the benchmark alignment methods, with the added benefit of being able to provide a measure of confidence in the alignment quality. This can be useful in real analytical situations, where neither the optimal parameters nor the alignment ground truth is known to the user.

Through comparisons against benchmark methods, our studies have also investigated the effect of sub-optimal parameter choices on alignment performance. While beyond the scope of our paper, we agree with [18, 57] that thorough investigations into the influence of numerous configurable parameters (prevalent in nearly all LC-MS data processing pipeline) on the resulting biological conclusions are of utmost importance. This should be followed by the development of methods to minimise or automatically-tune such configurable parameters. Despite the abundance of new methods proposed for LC-MS data pre-processing, relatively few studies have been done on the subject of quantifying uncertainties and alleviating the burden of parameter optimisations during actual data analysis. One way to minimise the number of parameters is through the integration of multiple steps in the typical LC-MS pipeline into fewer steps. Our proposed model in HDP-Align can potentially be extended in this manner, as evidenced by the metabolomic dataset results where we directly use the clustering objects inferred from the model to perform further analysis on putative adduct and metabolite type annotations. While the proposed annotation approach in Section 5.2.3 is fairly simple, it can be easily extended to more sophisticated annotation strategies, such as in CAMERA [37].

A primary weakness of HDP-Align lies in the long computational time required to produce results. Additional work will be required to reduce the computational burden of the model through various optimisation tricks and potentially by parallelising the Gibbs inference step using e.g. the method described in [**?**]. Another possibility is to adopt a different non-sampling-based inferential approach or perhaps even a simpler model altogether, while still retaining the essence and benefits of the HDP-Align model. The key insight here lies in modelling related peaks as within-file clusters in a single run but also allowing these within-

file clusters to be generated by globally-shared clusters spanning across multiple runs. The results presented in the current chapter suggest the method shows enough promise to warrant the effort to speed it up, and indeed that is what we will discuss in the next chapter.

Another aspect worthy of investigation is determining the most effective way to present and visualise the alignment probabilities produced by HDP-Align. Additional sources of information present in the LC-MS data, such as chromatographic peak shapes, can also be used to improve alignment performance and subsequent analyses that follow.

Finally, replacing or enhancing the mixture of mass components used in HDP-Align with a more appropriate mass model, such as that in MetAssign [38] that specifically takes into account the inter-dependency structure of peaks, is an avenue for future work. This will be particularly useful when extending the proposed model in HDP-Align into a single inferential model that encompasses many intermediate steps in a typical LC-MS data processing pipeline.