Supplementary Information

Targeted Realignment of LC-MS Profiles by Neighbor-wise Compound-specific Graphical Time Warping with Misalignment Detection

Chiung-Ting Wu, David M. Herrington, Yizhi Wang, Yinxue Wang, Timothy Ebbels, Ibrahim Karaman, Gonçalo Graça, Yue Wang, and Guoqiang Yu

Contents

Detection of Misaligned Features

In this section, we define a test statistic and derive the p-value for each initially-aligned feature from XCMS. Assume there are N samples in total. For a feature detected in n samples, we denote the indices (run-order) of these samples as a set $\{l_1, l_2, \ldots, l_n\}$, where $1 \leq l_i \leq N$ and $l_1 < l_2 < \cdots < l_n$. For this feature, we define the test statistic

$$t = \max_{1 \leq i, j \leq n} |l_i - l_j|, \tag{S1}$$

which is the same as the *range* of the n indices. It is clear that $n - 1 \leq t \leq N - 1$. Under the null hypothesis, these n samples are randomly drawn from the N samples. Therefore, each index in this feature can be viewed as a random variable and we assume it follows a discrete uniform distribution. Our goal is to find the null distribution of t, which is obtained by subtracting the smallest of these n random variables from the largest one. To achieve this, we sort these n random variables from small to large. Each *sorted* random variable is no longer uniformly distributed. Instead, its distribution need to be calculated using order statistics (Arnold, et al., 1992). For l_i, the ith smallest index in the feature, the probability mass function (pmf) is

$$f_{l_i}(k) = \frac{\binom{k-1}{i-1}\binom{N-k}{n-i}}{\binom{N}{n}}, i \leq k \leq N - n + i, \tag{S2}$$

and the joint pmf of l_i and l_j is given by

$$f_{l_i, l_j}(k, l) = \frac{\binom{k-1}{i-1}\binom{l-k-1}{j-i-1}\binom{N-l}{n-j}}{\binom{N}{n}}, i \leq k < l \leq N - n + j, l - k \geq j - i. \tag{S3}$$

To obtain the null distribution of the test statistics, we let $i = 1, j = n, l = k + t$. Thus, equation (S3) becomes

$$f_{l_1, l_n}(k, k+t) = \frac{\binom{t-1}{n-2}}{\binom{N}{n}}, 1 \le k < k+t \le N, t \ge n-1. \qquad (S4)$$

Therefore, under the null hypothesis, the pmf of t is

$$f_T(t) = \sum_{k=1}^{N-t} f_{l_1, l_n}(k, k+t) = (N-t)\frac{\binom{t-1}{n-2}}{\binom{N}{n}}, n-1 \le t \le N-1. \qquad (S5)$$

For an observed feature, we can calculate its *p*-value:

$$p\text{-}value = \Pr\{t \le t_{obs}\} = \sum_{t=n-1}^{t_{obs}} (N-t)\frac{\binom{t-1}{n-2}}{\binom{N}{n}}, \qquad (S6)$$

where t_{obs} is the test statistic calculated in that feature (Connor, 1969). The summation is over all ts that are smaller than t_{obs}, which reflects the assumption that the more concentrated the indices are, the more likely misalignment exists. The smaller the p-value, the more unlikely the observed feature follows the null distribution, and the more likely misalignment occurs.

Details on the alignment module of ncGTW

ncGTW contains a multiple alignment module that is designed to flexibly model and incorporate any structural information among samples, as long as their relationship can be represented as a weighted graph. This module aims to find a set of warping functions, through which each sample can be aligned to a reference. The main challenge in multiple alignment is the lack of a priori reference. Indeed, various strategies have been proposed to select a reference sample or estimate a reference using all samples. However, when the samples have complex patterns such as missing signals at some time points, or contain significant noise, no single sample merits a good reference while the estimation of an ensemble reference requires a set of pre-aligned samples.

Instead of explicitly picking a certain reference, ncGTW extracts the needed information from all possible sample pairs to perform multiple alignment. To deal with missing signals and noise, ncGTW borrows the idea from graphical time warping (GTW) (Wang, et al., 2016) to incorporate the structural information in the dataset, which makes the estimation of the pairwise warping functions more accurate. The flowchart of ncGTW is shown in **Fig. 5** in the main article. We first find all pairwise warping functions under the constraints of structural prior knowledge. Then we use these pairwise warping functions as constraints to estimate the final warping functions, which align all samples to a reference. These two subproblems are formulated as network flow problems and each can be solved efficiently with a global optimal solution.

Problem modeling and methods

Given N samples (curves) $\{x_1, \ldots, x_N\}$, multiple alignment problem aims to find a set of warping functions $\{\Phi_{i,c}\}, i \in \{1 \ldots N\}$, through which each sample can be aligned to a reference x_c. For the sake of clarity, all samples have the same number of points P. The subscript "i, c" means this function maps a set of points $\{x_{ip}\}, p \in \{1 \dots P\}$ in curve x_i to a corresponding set of points in curve x_c. That is, for any point x_{ip} in x_i, $\Phi_{i,c}$ can always map x_{ip} to at least one point in x_c, where p is from 1 to P.

Definition 1 – valid warping function

A valid warping function for the pair of curves (x_i, x_j) is a set of integer pairs $\Phi_{i,j} = \{(p,q)\}$, such that the following conditions are satisfied: (a) boundary conditions: $(1,1) \in \Phi_{i,j}$ and $(P,P) \in \Phi_{i,j}$; (b) continuity and monotonicity conditions: if $(p,q) \in \Phi_{i,j}$, then $(p-1,q) \in \Phi_{i,j}$ or $(p,q-1) \in \Phi_{i,j}$ or $(p-1,q-1) \in \Phi_{i,j}$. An example is shown in **Fig. S1a**.

Definition 2 – inverse of a valid warping function

Given a valid warping function $\Phi_{i,j} = \{(p,q)\}$, the inverse of $\Phi_{i,j}$ is ${\Phi_{i,j}}^{-1} = \{(q,p)\} = \Phi_{j,i}$

Definition 3 – alignment cost

For any given valid warping function $\Phi_{i,j}$ and its corresponding pair of curves (x_i, x_j), the associated alignment cost is defined as follows:

$$\mathrm{cost}(\Phi_{i,j}) = \sum_{(p,q)\in\Phi_{i,j}} g(\mathrm{x}_{ip}, \mathrm{x}_{\mathrm{j}q}), \qquad (S7)$$

where $g(\mathrm{x}_{ip}, \mathrm{x}_{\mathrm{j}q})$ is any nonnegative function which computes the distance between x_{ip} and $\mathrm{x}_{\mathrm{j}q}$.

Stage 1: jointly aligning all pairs with the structural prior incorporated

To estimate $\{\Phi_{i,j}\}$ jointly, ncGTW considers all possible sample pairs. For each pair, one sample is set as the reference. In other words, in the first stage, ncGTW tries to align each sample to the other samples. For N samples, the number of alignment pairs is $N(N-1)$. However, $\Phi_{i,j}$ is the inverse of $\Phi_{j,i}$, so only $N(N-1)/2$ pairs need to be considered in the real implement. That is, only $N(N-1)/2$ pairwise warping functions are needed.

In order to incorporate the structural information, as GTW, we need to convert the given structural information in the dataset into the warping function neighborhood information. Suppose (x_i, x_j) are neighbors by structural information, we consider the pair of warping functions, $(\Phi_{i,k}, \Phi_{j,k})$, as well as $(\Phi_{k,i}, \Phi_{k,j})$, as neighbors for all *k*. Also, if (x_i, x_j) are neighbors and (x_k, x_l) are also neighbors respectively, we can consider the pair of $\Phi_{i,k}$ and $\Phi_{j,l}$ are neighbors. **Fig. S1c** gives an example of the warping function neighborhood information of five samples. The run orders of the five samples are exactly 1, 2, 3, 4, and 5. Thus, these five samples are expected to have a pattern of continuous changing, so (x_1, x_2), (x_2, x_3), (x_3, x_4), and (x_4, x_5) are considered as neighbors respectively. More examples about the structures are shown in the experiment section.

Definition 4 – neighboring warping functions

Suppose the neighboring structure for a set of *M* valid warping functions is given by the graph $G_{struct} = \{V_s, E_s\}$, where V_s is the set of nodes, with each node corresponding to a warping function, and E_s is the set of undirected edges between nodes. If $v_{ij}, v_{kl} \in V_s$ and $(v_{ij}, v_{kl}) \in E_s$, we call $\Phi_{i,j}$ and $\Phi_{k,l}$ neighbors, denoted by $((i,j),(k,l)) \in \ Neib$.

As an improvement of GTW, ncGTW also adapts the idea of dynamic time warping (DTW). DTW aligns two curves x_i and x_j by finding a warping function $\Phi_{i,j}$ that minimize the alignment cost (S7). The correspondence represented by $\Phi_{i,j}$ can be visualized as a path in a DTW grid, from bottom left to top right, and the weight of each edge is decided by the distance function $g(\cdot)$ with all possible point pairs (x_{ip}, x_{jq}), where $x_{ip} \in x_i$ and $x_{jq} \in x_j$. DTW estimates Φ in the

DTW grid using dynamic programming and various additional constraints can be employed, such as the direction of the path.

Definition 5 – DTW grid for a single pair of curves

For each pair of curves, consistent with the cost function (S7), there is an induced directed planar graph (Korte, et al., 2012), $G_{ij} \coloneqq \{V_{ij}, E_{ij}\}, 1 \leq i < j \leq N$, where V_{ij} and E_{ij} are the nodes and directed edges respectively. Each point pair $\left(x_{ip}, x_{jq}\right)$ is corresponding to a node $V_{ij,pq} \in V_{ij}$, where $1 \leq p, q \leq P$. The weight of $\left(V_{ij,p_1q_1}, V_{ij,p_2q_2}\right) \in E_{ij}$ is the distance between the two points (x_{ip_1}, x_{jq_1}), measured by $g\left(x_{ip_1}, x_{jq_1}\right)$. An example is shown in **Fig. S1a**. Any directed path from the bottom-left corner to the upper-right corner is corresponding to a valid warping function $\Phi_{i,j}$.

Once the structure for warping functions is obtained, the joint alignment of all pairs of samples can be readily solved by the recently developed model – Graphical Time Warping (GTW). When we jointly align multiple pairs of curves with structural information, our goal is to minimize both the overall alignment cost and the distance between neighboring warping functions.

Definition 6 – distance between two valid warping functions

For any two given valid warping functions $\Phi_{i,j}$ and $\Phi_{k,l}$, the distance between them is defined as follows:

$$\mathrm{dist}\left(\Phi_{i,j}, \Phi_{k,l}\right) = \frac{1}{2} \sum_{1 \leq n \leq P} \left| \max_{(p_i,n) \in \Phi_{i,j}} p_i - \max_{(p_k,n) \in \Phi_{k,l}} p_k \right| + \left| \min_{(p_i,n) \in \Phi_{i,j}} p_i - \min_{(p_k,n) \in \Phi_{k,l}} p_k \right|, \qquad (S8)$$

which is equivalent to the area of the region bounded by the two corresponding paths in a DTW grid as shown in **Fig. S1a**.

Mathematically, to balance the alignment cost (S7) and the distance between warping functions (S8), we want to minimize the following cost function for GTW problem:

$$\min_{\Phi} f\,(\Phi) = \min_{\Phi=\{\Phi_{i,j}|1\leq i<j\leq N\}} \sum_{1\leq i<j\leq N} \mathrm{cost}\big(\Phi_{i,j}\big) + \kappa_1 \sum_{((\mathrm{i,j}),(\mathrm{k,l}))\in \mathrm{Neib}} \mathrm{dist}\big(\Phi_{i,j}, \Phi_{k,l}\big)\,, \qquad (S9)$$

where κ_1 is the parameter which balance the two terms. The first term is the overall alignment cost of the warping functions. The second term could be considered as the sum of the "dissimilarity" between each pair of the neighboring warping functions. Again, the neighboring warping functions should be similar. In the other words, their distance should be small.

There are three major steps to solve the GTW problem. Firstly, GTW transforms the DTW grid for each pair of curves to an equivalent minimum cut problem as a DTW graph. Supposing there are *M* pairs of curves, where each pair contains two curves to be aligned with each other, then we will have *M* DTW grids, of which there are also *M* DTW graphs. Secondly, GTW adds in extra edges between DTW graphs, if two pairs of curves are considered as neighbors. Note that if edges are to be added between two DTW graphs, all the corresponding vertices in the two graphs need to be connected. Thus, the *M* separate DTW graphs become an extended graph (GTW graph). Thirdly, GTW proved that the joint alignment of multiple pairs with smoothness constraints imposed could be formulated as a minimum cut problem in the GTW graph. Hence, efficient network flow algorithms can be used to find a global optimal solution.

Definition 7 – DTW graph

Define $G'_{ij} \coloneqq \{V'_{ij}, E'_{ij}\}$ as the DTW graph of the DTW grid G_{ij}, where nodes V'_{ij} are all faces of G_{ij}. That is, for each $V_{ij,pq} \in V_{ij}$, where $2 \leq p \leq P-1$ and $1 \leq q \leq P-1$, there are two corresponding nodes $V'_{ij,pq+}$ and $V'_{ij,pq-}$; for each $V_{ij,pq}$ where $p = 1$ and $1 \leq q \leq P-1$, there is one corresponding node $V'_{ij,pq+}$; for each $V_{ij,pq}$ where $p = P$ and $1 \leq q \leq P - 1$, there is one corresponding node $V'_{ij,pq-}$. For each $e \in E_{ij}$, we have a new edge $e' \in E'_{ij}$ connecting the faces from the right side of e to the left side. This edge is directed (with positive direction by convention). The edge weights are the same as for the primal graph G_{ij}. An example is shown in **Fig. S1b**.

Definition 8 – GTW graph

The GTW graph $G_{gtw} \coloneqq \{V_{gtw}, E_{gtw}\}$ is defined as the *integrated graph* of all DTW graphs $\{G'_{ij} | 1 \leq i < j \leq N\}$ with the integration guided by the neighborhood of warping functions, such that $V_{gtw} = \{V'_{ij} | 1 \leq i < j \leq N\}$ and

$$E_{gtw} = E'_{ij} \big| 1 \leq i < j \leq N \cup \{\left(V'_{ij,pq+}, V'_{kl,pq+}\right), \left(V'_{ij,pq-}, V'_{kl,pq-}\right)\} \big| ((\mathrm{i},\mathrm{j}),(\mathrm{k},\mathrm{l})) \in Neib.$$

All newly introduced edges $\left(V'_{ij,pq+}, V'_{kl,pq+}\right)$ and $\left(V'_{ij,pq-}, V'_{kl,pq-}\right)$ are bi-directional with capacity λ_1 as shown in **Fig. S1d**. An example of a GTW graph with two pairs of curves is shown in **Fig. S1e**.

Definition 9 – Labeling of the graph

L is a *labeling* of graph G if it assigns each node in G a binary label. L can induce a cut set $C = \{(s,t) | L(s) \neq L(t), (s,t) \in E_G\}$. The corresponding cut (or flow) is $cut(L) = cut(C) = \sum_{(s,t) \in C} weight(s,t)$, where $weight(s,t)$ is the weight on the edge between nodes s and t.

Based on its construction, a labeling L for the graph G_{gtw} can be written as $L = \{L_{ij} | 1 \leq i < j \leq N\}$, where L_{ij} is a labeling for the DTW graph G'_{ij}. Thus, we can express the minimum cut problem for the graph G_{gtw} as:

$$\min_L g(L) = \min_{L:=\{L_{ij}|1\le i<j\le N\}} \sum_{1\le i<j\le N} cut(L_{ij}) + \lambda_1 \sum_{((i,j),(k,l))\in Neib} cut(L_{ij}, L_{kl}), \qquad (S10)$$

where $cut(L_{ij})$ is the cut of all edges for G'_{ij} and $cut(L_{ij}, L_{kl})$ is the number of the cut edges between two neighboring DTW graphs G'_{ij} and G'_{kl}.

As proved in (Wang, et al., 2016), the GTW problem as stated in (S9) is equivalent to the minimum cut problem on the GTW graph G_{gtw} if we set $\lambda_1 = 2\kappa_1$. Once we apply any existing network flow algorithm to solve this minimum cut problem, the GTW problem would be solved, and all pairwise warping functions are available.

Stage 2: Finding multiple alignment based on the constraint of pairwise alignments

In the stage 2 of ncGTW, the result from the stage 1, $\{\Phi_{i,j}\}$, is used to estimate $\{\Phi_{i,c}\}$, which are the final goal of multiple alignment problem. Here, N warping functions need to be estimated. On contrary to stage 1, where neighboring information as constraints, in stage 2 the warping function $\{\Phi_{i,j}\}$ are set as constraints to estimate $\{\Phi_{i,c}\}$. For example, according to the warping function $\Phi_{i,j}$, point x_{ip} in x_i should be aligned to point x_{jq} in x_j. Then, we expect that x_{ip} and x_{jq} should be aligned to the same position on the reference x_c. That is, we want the warping of x_{ip} and x_{jq} on the reference is *consistent*.

Definition 10 – inconsistency between two sample points

For any given integer pair $(p_i, p_j) \in \Phi_{i,j}$, where p_i is the point index of x_i and p_j is the point index of x_j. The inconsistency between x_{ip_i} and x_{jp_j} is defined as follows:

$$\text{incons}\left(x_{ip_i}, x_{jp_j}; \Phi_{i,j}\right) = \left|\max_{(p_i,q_i)\in\Phi_{i,c}} q_i - \max_{(p_j,q_j)\in\Phi_{j,c}} q_j\right| + \left|\min_{(p_i,q_i)\in\Phi_{i,c}} q_i - \min_{(p_j,q_j)\in\Phi_{j,c}} q_j\right|,$$

which could be considered as the distance between two continuous integer sets, as shown in **Fig. S2a**. The inconsistency quantifies how much the alignment deviates from our expectation. If the inconsistency is zero, we call these two points are consistent.

Definition 11 – inconsistency between two warping functions

For any given valid warping function $\Phi_{i,j}$, the inconsistency between two warping functions $\Phi_{i,c}$ and $\Phi_{j,c}$ is defined as follows:

$$\mathrm{incons}\left(\Phi_{i,c}, \Phi_{j,c}; \Phi_{i,j}\right) = \sum_{(p_i,p_j)\in\Phi_{i,j}} \left| \max_{(p_i,q_i)\in\Phi_{i,c}} q_i - \max_{(p_j,q_j)\in\Phi_{j,c}} q_j \right| + \left| \min_{(p_i,q_i)\in\Phi_{i,c}} q_i - \min_{(p_j,q_j)\in\Phi_{j,c}} q_j \right|, (S11)$$

which is equivalent to the sum of the inconsistency between two sample points x_{ip_i} and x_{jp_j}, where $\left(p_i, p_j\right) \in \Phi_{i,j}$. Likewise, if the inconsistency between these two warping functions is zero, then the two warping functions are consistent.

Intuitively speaking, stage 2 tries to identify the final warping functions from all pairwise ones, with the constraint that the sum of inconsistency between all warping function pairs as low as possible. However, due to the noise and other factors, in the real data, there are always some contradictions among the pairwise warping functions. For example, from $\Phi_{i,j}$, we know that x_{ip} and x_{jq} are aligned together, and from $\Phi_{j,k}$, we know that x_{jq} and x_{kr} are aligned together, but from $\Phi_{k,i}$, we know that x_{kr} and x_{io} (not x_{ip}) are aligned together. This kind of contradictions will make the alignment tend to be the trivial "all to one" mapping, if we want to minimize the inconsistency. To solve this problem, here we introduce another constraint that makes the warping functions tend to choose "one to one" mapping, to avoid the trivial mapping. In the real implement, we redesign the weight of the edges in the DTW graphs. Thus, the exact values of neither x_i nor

x_c do no matter in this stage. In the other words, the weights of the edges on the DTW graph of x_i and x_c is not based on the points of x_i and x_c. This property is the reason why our approach does not rely on the selection or estimate of the reference curve. Therefore, x_c is called a "virtual reference".

Definition 12 – non-diagonality of a warping function

For any given valid warping function $\Phi_{i,c}$, the associated non-diagonality is defined as follows:

$$\mathrm{nondiag}(\Phi_{i,c}) = \sum_{(p,q)\in\Phi_{i,c}} \mathbb{1}\left((p-1,q) \in \Phi_{i,c} \vee (p,q-1) \in \Phi_{i,c}\right), \qquad (S12)$$

where $\mathbb{1}(\cdot)$ is the indicator function. This definition is the same as the total number of vertical and horizontal edges in the corresponding path. The smallest value of non-diagonality is zero (one-to-one mapping, the diagonal line). An example of non-diagonality is shown in **Fig. S2a**.

In order to obtain the consensus final alignment from pairwise warping functions, we design ncGTW problem which tries to balance the non-diagonality (S12) and inconsistency (S11) among final warping functions:

$$\min_{\Phi} f(\Phi) = \min_{\Phi=\{\Phi_{i,c}|1\le i\le N\}} \sum_{1\le i\le N} \mathrm{nondiag}(\Phi_{i,c}) + \kappa_2 \sum_{1\le i<j\le N} \mathrm{incons}(\Phi_{i,c}, \Phi_{j,c}; \Phi_{i,j}), \qquad (S13)$$

where the first term relates to the warping path for each final warping function, and the second term is the constraints between each warping function pair. The relation between these two terms is similar to the two terms in the GTW problem (S9). Thus, as same as GTW, an ncGTW problem could also be transformed to an ncGTW graph, and solved by maximum flow algorithms.

Definition 13 – ncGTW graph

The ncGTW graph $G_{ncgtw} \coloneqq \{V_{ncgtw}, E_{ncgtw}\}$ is defined as the *integrated graph* of all DTW graphs $\{G'_{ic} | 1 \leq i \leq N\}$ with the integration guided by the pairwise warping functions, such that $V_{ncgtw} = \{V'_{ic} | 1 \leq i \leq N\}$ and $E_{ncgtw} = \{E'_{ic} | 1 \leq i \leq N \cup \left(V'_{ic,p_i q+}, V'_{jc,p_j q+}\right) | (p_i, p_j) \in \Phi_{i,j} \cup \left(V'_{ic,p_i q-}, V'_{jc,p_j q-}\right) | (p_i, p_j) \in \Phi_{i,j}\}$. That is, all newly introduced edges are guided by $\Phi_{i,j}$ as shown in **Fig. S2b**. Also, all these new edges are bi-directional with capacity λ_2. For the edges in V'_{ic}, the capacity of edges corresponding to vertical or horizontal path is one, and zero for edges corresponding to diagonal path, so that the non-diagonality of the warping function $\Phi_{i,c}$ is equivalent to the cost of the warping path on the corresponding DTW grid G_{ic}.

Like GTW problem, the ncGTW problem as stated in equation (S13) is equivalent to the minimum cut problem on the ncGTW graph G_{ncgtw} if we set $\lambda_2 = 2\kappa_2$. Moreover, a labeling L for the graph G_{ncgtw} can be written as $L = \{L_{ic} | 1 \leq i \leq N\}$, where L_{ic} is a labeling for the DTW graph G'_{ic}. Therefore, we can express the minimum cut problem for the graph G_{ncgtw} as:

$$\min_{L} g(L) = \min_{L \coloneqq \{L_{ic} | 1 \leq i \leq N\}} \sum_{1 \leq i \leq N} cut(L_{ic}) + \lambda_2 \sum_{1 \leq i < j \leq N} cut\left(L_{ic}, L_{jc}\right), \qquad (S14)$$

where $cut(L_{ic})$ is the cut of all edges for G'_{ic} and $cut\left(L_{ic}, L_{jc}\right)$ is the number of the cut edges between two connecting DTW graphs G'_{ic} and G'_{jc}.

Again, after applying any generic maximum flow algorithm, the ncGTW problem would be solved, and all final warping functions $\{\Phi_{i,c}\}$ are available.

Relationship between hyper-parameter and the solutions of Stage 1 and Stage 2

With a specific value of hyper-parameter $\lambda_1(\kappa_1)$ in Stage 1, we can obtain from Equation S10 the corresponding label set L and the minimum cut. Similarly, in Stage 2, with $\lambda_2(\kappa_2)$ given, we can obtain the corresponding label set and the minimum cut from Equations S12. We also obtain the final warping functions corresponding to that minimum cut. If we change the value of λ_1 or λ_2, the minimum cut solution may or may not change. Since the solution space is discrete, a very minor change of λ_1 or λ_2 may not lead to the change of solution. If the change of the hyper-parameter is large enough, we may get a different minimum-cut solution.

The edges that are cut in any minimum cut solution can be grouped into two categories: those inside each DTW graph and those between DTW graphs. Therefore, the minimum cut obtained in Stage 1 or 2 can be viewed as a function of hyper-parameter λ (λ_1 in Stage 1 or λ_2 in Stage 2):

$$cut_{total}(\lambda) = cut_D(\lambda) + \lambda \times cut_B(\lambda), \qquad (S14)$$

where $cut_D(\lambda)$ is the cut for all DTW graphs and $cut_B(\lambda)$ is the total number of the cut edges between any two connecting DTW graphs. Note that λ (λ_1 or λ_2) can be replaced with κ (κ_1 or κ_2) since they are equivalent (Wang, et al., 2016). If we can test all possible values of the hyper-parameter, we can get all possible solutions of warping functions and we can choose the best one from them. However, practically this is intractable and consumes too much time. Instead, we hope to find some special properties of Equation S14 and utilize those properties in the search for the optimal value of hyper-parameter.

Interestingly, we found that the total cut $cut_{total}(\lambda)$ in Equation S14 is a concave non-decreasing piecewise linear function of λ, and each line segment is corresponding to a specific result (a set of warping functions) in Stage 1 or Stage 2. We will utilize those properties to reduce the search space of hyper-parameters and to obtain efficient approximate strategies. In this section, we will prove the property. We first introduce some lemmas.

Lemma 1 $cut_{total}(\lambda)$ is a non-decreasing function.

Proof: Assuming $\lambda'' > \lambda'$, if $cut_{total}(\lambda)$ is not a non-decreasing function, then there should be at least one pair of λ' and λ'' satisfies:

$$cut_D(\lambda') + \lambda' \times cut_B(\lambda') > cut_D(\lambda'') + \lambda'' \times cut_B(\lambda'').$$

Since $\lambda'' > \lambda'$, we can obtain:

$$cut_D(\lambda') + \lambda' \times cut_B(\lambda') > cut_D(\lambda'') + \lambda'' \times cut_B(\lambda'') > cut_D(\lambda'') + \lambda' \times cut_B(\lambda''),$$

and

$$cut_D(\lambda') + \lambda' \times cut_B(\lambda') > cut_D(\lambda'') + \lambda' \times cut_B(\lambda'').$$

By definition, $cut_D(\lambda')$ and $cut_B(\lambda')$ should be the cuts which give the minimum cut when λ is λ'. However, the above equation shows that when λ equals λ', $cut_D(\lambda'')$ and $cut_B(\lambda'')$ can give even lower total cut. Therefore, there is a contradiction. Thus, $cut_{total}(\lambda)$ is a non-decreasing function. □

Lemma 2 $cut_D(\lambda)$ is a non-decreasing step function, and $cut_B(\lambda)$ is a non-increasing step function. The interval of each step of $cut_D(\lambda)$ is the same as the one of $cut_B(\lambda)$.

Proof: Assuming $\lambda'' > \lambda' \geq 0$, then

$$cut_D(\lambda') + \lambda' \times cut_B(\lambda') \leq cut_D(\lambda'') + \lambda' \times cut_B(\lambda''),$$

and

$$cut_D(\lambda'') + \lambda'' \times cut_B(\lambda'') \leq cut_D(\lambda') + \lambda'' \times cut_B(\lambda'),$$

since $(cut_D(\lambda'), cut_B(\lambda'))$ and $(cut_D(\lambda''), cut_B(\lambda''))$ should give the minimum cut when λ is λ' and λ'' respectively. Thus, from the above two inequalities, we can obtain:

$$\lambda'\left(cut_B(\lambda') - cut_B(\lambda'')\right) \leq cut_D(\lambda'') - cut_D(\lambda') \leq \lambda''\left(cut_B(\lambda') - cut_B(\lambda'')\right),$$

and thus

$$cut_B(\lambda') \geq cut_B(\lambda''),$$

$$cut_D(\lambda'') \geq cut_D(\lambda').$$

Therefore, $cut_D(\lambda)$ is a non-decreasing function, and $cut_B(\lambda)$ is a non-increasing function. Also, the possible values of $cut_D(\lambda)$ and $cut_B(\lambda)$ are discrete and countable, since the edges in DTW graphs and the edges between DTW graphs are countable. Thus, $cut_D(\lambda)$ is a non-decreasing step function, and $cut_B(\lambda)$ is a non-increasing step function.

Moreover, $cut_D(\lambda)$ and $cut_B(\lambda)$ will change together. When $cut_B(\lambda)$ increases / decreases, $cut_D(\lambda)$ should decrease / increase. If $cut_D(\lambda)$ does not decrease / increase, which means the previous $cut_B(\lambda)$ does not give the minimum cut and it is a contradiction. Therefore, the interval of each step of $cut_D(\lambda)$ is the same as the one of $cut_B(\lambda)$. □

Lemma 3 In each step of $cut_D(\lambda)$ and $cut_B(\lambda)$, $cut_{total}(\lambda)$ is a linear function with a positive slope.

Proof: From Lemma 2, we know that in each step of $cut_D(\lambda)$ and $cut_B(\lambda)$, $cut_D(\lambda)$ and $cut_B(\lambda)$ are both constants. Thus, Equation (S14) becomes

$$cut_{total}(\lambda) = C_D + \lambda \times C_B,$$

where C_D and C_B are constants. Thus, in each step of $cut_D(\lambda)$ and $cut_B(\lambda)$, $cut_{total}(\lambda)$ is a linear function with a positive slope C_B. □

Theorem 1 The solved minimum cut from Stage 1 or Stage 2 is a concave non-decreasing piecewise linear function of λ_1 of λ_2, and each line segment corresponding to a specific alignment result in Stage 1 or Stage 2.

Proof: From **Lemma 2**, we know that $cut_D(\lambda)$ and $cut_B(\lambda)$ are not continuous since they are step functions. To prove Equation (S12) is continuous, we need to prove that λ between each step of $cut_D(\lambda)$ (and also the same point of $cut_B(\lambda)$) is continuous for Equation (S12). That is, assuming C_D and C_B is the value of $cut_D(\lambda)$ and $cut_B(\lambda)$ for step i, and C'_D and C'_B is the value of $cut_D(\lambda)$ and $cut_B(\lambda)$ for step $i+1$, from **Lemma 3**, we need to prove:

$$C_D + \lambda \times C_B = C'_D + \lambda \times C'_B,$$

for λ between each step of $cut_D(\lambda)$. If $C_D + \lambda \times C_B > C'_D + \lambda \times C'_B$, then $cut_{total}(\lambda)$ is not a non-decreasing function, which contradicts **Lemma 1**. If $C_D + \lambda \times C_B < C'_D + \lambda \times C'_B$, it means C_D and C_B give lower total cut than C'_D and C'_B for step $i+1$, which contradicts the fact that C'_D and C'_B always give the minimum cut for step $i+1$. Thus, $C_D + \lambda \times C_B = C'_D + \lambda \times C'_B$, and $cut_{total}(\lambda)$ is a continuous function. In addition, the slope of each line segment of $cut_{total}(\lambda)$ is non-increasing by **Lemma 2** and **Lemma 3**, so $cut_{total}(\lambda)$ is a concave non-decreasing piecewise linear function of λ. From **Definition 9** and **Definition 13**, we know that after obtaining $cut_D(\lambda)$ (the cut of DTW graphs), the warping function set is available for Stage 1 (if λ is λ_1) or Stage 2 (if λ is λ_2). Also, in each line segment of $cut_{total}(\lambda)$, $cut_D(\lambda)$ is the same. Thus, each line segment of $cut_{total}(\lambda)$ corresponds to a specific alignment result in Stage 1 or Stage 2. □

We can immediately obtain two corollaries:

Corollary 1 The first line segment of $cut_{total}(\lambda)$ is corresponding to solving each DTW graph separately.

The first line segment of $cut_{total}(\lambda)$ starts when λ is zero. Under such circumstance, there is no additional edge connecting DTW graphs, so we can solve each DTW graph separately to obtain $cut_D(\lambda)$.

Corollary 2 The last line segment of $cut_{total}(\lambda)$ is corresponding to solving a *single* DTW graph.

When λ is large enough (for example, larger than the maximum value of $cut_D(\lambda)$), no additional edge will be cut to avoid large value of cut. As a result, $cut_B(\lambda)$ is zero. Hence, the warping functions (paths) of all the DTW graphs are the same to avoid cutting any additional edge. Therefore, we can create a new DTW graph whose topology is the same as any existing DTW graph. The capacity of each edge in the new graph is the summation of the capacities of corresponding edges from all existing graphs. The warping function obtained from the new DTW graph is the same as the warping functions of all original DTW graphs of the last segment of $cut_{total}(\lambda)$.

The strategy of finding the line segments of $cut_{total}(\lambda)$

To obtain the best alignment result of Stage 1 and Stage 2, we need to tune the hyperparameters λ_1 and λ_2 (κ_1 and κ_2). However, since the alignment results are countable (line segments), we can tune the hyperparameters much more efficiently by finding different line segments, instead of trying different values of λ_1 and λ_2 blindly. Assuming there are total k line segments, from **Corollary 1**, we can obtain the first line segment (segment 1) by solving each DTW graph separately, and from **Corollary 2**, we can obtain the last line segment (segment k) by solving a new DTW graph. As shown in **Fig. S3**, we can find a crossing point by extending segment 1 and segment k respectively. The λ corresponding to the crossing point is the new hyperparameter we can try to obtain a new segment i. Again, we can find another two new segments by extending segment i to find the crossing points with segment 1 and segment k. Repeating the steps for newly obtained line segments, we can obtain different values of λ roughly uniformly along line segments. When this step is repeated with enough times, all line segments are identified if needed.

Local or global optimum

In Stage 1 of ncGTW, we claimed that we could obtain the global optimum of a GTW problem. That is, we can solve the maximum flow problem from the GTW graph with global optimum. One should notice that this global optimum is not the global optimum of the multiple alignment problem. In fact, the multiple alignment problem is an NP-hard problem. It is possible that the global optimum of GTW is just the local optimum of multiple alignment. In spite of that, ncGTW still has a superior advantage. In different fields, the ways of evaluation of the result of multiple alignment are very different. However, ncGTW can adjust the weights of the additional edges according to the evaluation method. Thus, with the structure information, ncGTW can approach to the global optimum of the multiple alignment better than other methods, with great flexibility to various evaluation criteria.

Experiments

We evaluate the performance of ncGTW on both simulated and real datasets. Any neighborhood structure among samples can be incorporated into ncGTW as long as the structure can be represented as a graph. Although weights on edges can also be naturally integrated into ncGTW, prior knowledge on weights is very application-dependent and thus we assume the neighborhood structure has no weight. In this set of experiments, we test three different structures: line, block, and uniform (non-informative). Three peer methods, DBA, CPM, and GTW were selected for comparison due to their representativeness. DBA is a DTW based method that iteratively computes barycenter and aligns all samples to the barycenter (Petitjean, et al., 2011). CPM uses the hidden Markov model to learn the prototype function (Listgarten, et al., 2005). Since GTW does not provide the reference, we need to manually supply one if we want to apply GTW to the multiple alignment problem. In the experiments, each time we used one sample as a reference. We went through all samples one by one and take the average of all scores. For a visual demonstration, we choose the most informative one. In the following experiments, one can see that a bad reference may ruin the alignment of GTW. Also, other flaws of directly applying GTW on multiple alignment are demonstrated.

Both quantitative measurements and visual assessment were used to evaluate the performance. We adopt two quantitative criteria that are frequently used in the literature. They are mean correlation coefficient (MCC) and simplicity (SP) (Jiang, et al., 2013). After alignment, the correlation among samples is expected to increase. Thus, the mean of the correlation of all sample pairs can be considered as a quantification of the alignment quality. The definition of simplicity here is the sum of the fourth power of all singular values, where the sum of all singular values is normalized to be one. The singular values are computed based on the data matrix, where each row represents a sample. The idea is that if it is a good alignment, the first singular value should dominate others. Hence, larger simplicity means better alignment. Note that the largest possible value for simplicity is one.

For the simulation data, since we have the ground truth, we test each peak separately so that we can know the alignment quality of each peak. For the real data, the numerical evaluations for each peak are not applicable since we do not have the ground truth.

Simulation data generation

To understand the performance of each method on different structures, we first conduct three case studies on synthetic datasets with line, block, and non-informative structures respectively. To prevent the impact of the other factors, the shapes of all peaks are sinusoidal. Also, the distance between the neighboring peaks is set to be the same for all simulated samples, and the shifts between the neighboring simulated samples are all one (for block structure, the shift between the two batch is seven). The apex intensity of each peak is extracted from the samples with similar peak intensity in the MESA dataset. Moreover, to fully observe the impact of the missing peak problem, we randomly picked a simulated sample and remove one peak from it.

To compare all the methods on a small-scale real dataset, we picked samples from the MESA dataset. For the line structure, we picked samples with a clear linear structure. For the block structure, we picked two groups of samples. Within each group, the RT shifts are similar, but the RT shift between groups is obvious. For non-informative structure, the samples for line structure are re-used but without the structure information.

Case study on line structure

When samples change gradually according to a certain variable (such as time, **Fig. S4a**), we call it to have a line structure, which can be converted to triangles (**Fig. S4b**) as an input structure between pairwise alignments. To evaluate the performance of different methods, we first designed a simulation dataset containing five samples. The first sample is shown in **Fig. S5a** and all five samples are plotted in **Fig. S5b**. Note that all the samples contain three peaks, except the fourth sample (purple dash line, the third peak of which is missing). This phenomenon of missing peaks occurs frequently in real applications.

Fig. S5c-f shows the alignment result of DBA, CPM, GTW, and ncGTW, and **Table S1** shows the evaluation scores of each peak. As the ground truth, the two peaks pointed by black arrows belong to the first and second group, respectively. The MCC and SP of all peaks of all methods are improved comparing with the original curves. However, DBA wrongly aligned the peaks in sample 4, which leads to bad scores of MCC and SP. CPM aligned all the peaks correctly, so it got good scores for all three peak groups in three evaluation. In **Fig. S5e**, the reference of GTW is the fourth sample. Since the fourth sample lacks the third peak, we can see from the figure that the third peaks of all samples are not aligned well. This is an example showing that the reference may have a huge effect on alignment. Moreover, the variance of the fourth sample is the largest one. In fact, many existing methods posit that the sample with the largest variance should be selected as the reference. As we shown here, reference selection is indeed a hard problem. Even with averaging, the scores of peak 3 of GTW are still much worse than other methods. ncGTW produced accurate alignment as evidenced by both visual assessment and quantitative scores. One may notice that the MCC of peak 3 are all smaller than 0.6, which is due to the missing peak in the fourth sample.

Case study on block structure

Block structure means there are several blocks formed by samples in the dataset. Within each block, the shifts are small. Between blocks, the shift is larger. Suppose we have ten samples in a dataset where every five samples form a block. **Fig. S6** shows how to connect these pairs. We can separate these pairs into three types. The first type is the alignment within the first block. The second type is between the two blocks. The third type is within the second block. Only the same type of neighbors will be connected. To test this structure, we generated a ten-sample dataset, and every five samples form a block. **Fig. S7a** shows the first sample, which contains three peaks. **Fig. S7b** shows the eighth sample, of which the third peak is missing. **Fig. S7c-d** show the two blocks in the dataset. **Fig. S7e** shows all the samples.

Fig. S7f- i show the results of DBA, CPM, GTW, and ncGTW, and **Table S2** shows the peak scores for each method. DBA wrongly aligned the two peaks in the eighth sample, due to the same reason as in the previous section. CPM has the worst performance since CPM separated all the peaks into four groups, not three. The reason may be that in **Fig. S7e**, it somehow shows four peak groups, and in each group, there are at least five peaks. Thus, CPM incorrectly treated them as four peak groups, not three. The reference of GTW is the tenth sample, which contains three peaks. With a good reference, in **Fig. S7h**, GTW have the performance as good as ncGTW. However, since there is a missing peak in the eighth sample, after averaging, the MCC of peak 3 is a little bit lower than ncGTW. Again, the MCC of peak 3 is not close to one for all methods, and this is also due to the missing peak in the eighth sample.

Case study on non-informative structure

Sometimes we may know nothing about the structure of the dataset. In this section, we demonstrate the experiments on the dataset without any structural information. For simulation, here we consider a ten-sample dataset. Without structural information, in the first stage of ncGTW, we add edges between the pairs that have the same aligning sample or reference sample. For example, for $G'_{1,2}$, we will connect it to $G'_{1,i}$, where i is from 3 to 10. Likewise, for $G'_{2,1}$, we will connect it to $G'_{i,1}$, where i is from 3 to 10. In this dataset with 10 simulated samples, there are two samples without peak 1, another two without peak 2, and still another two without peak 3. Thus, there are only four samples with all three peaks. **Fig. S8a** shows a sample without peak 2. **Fig. S8b** shows all ten samples. **Fig. S8c-f** show the results of DBA, CPM, GTW, and ncGTW, and **Table S3** shows the peak scores for each method. DBA again wrongly aligned the sample shown in **Fig. S8a** (pointed by arrows). CPM aligned all the peaks well but with some small drifts in some samples and distortions, so the scores are not as good as ncGTW but still better then DBA. With the sample shown in **Fig. S8a** as reference, GTW again misaligned the peak 2 group as shown in **Fig. S8e**. Since there are six samples with a peak missing, the average scores of GTW are all relatively low. Even without any structure information, ncGTW has the best performance.

Case study on small scale real LC-MS dataset

In this section, we conducted tests on a real liquid chromatography-mass spectrometry (LC-MS) experiments. First, ten samples were selected and ordered by the time as they were assayed. Assuming the properties of the equipment were gradually changing, we impose a line structure among these samples. **Fig. S9a** shows one sample from the ten samples. Clearly, there are nine peaks in the sample. **Fig. S9b** shows all ten samples together. Note that the intensity of the corresponding peaks between samples are very different, and some peaks are missing.

Fig. S9c-f show the results of DBA, CPM, GTW, and ncGTW. DBA wrongly aligned the peak pointed by the arrow to the third group of peaks (that peak should be aligned to the second group). The reason is that DBA is based on DTW. At some steps, DTW aligned a peak to a peak with similar intensity, but these two peaks are not in the same group. DTW based methods have the tendency to align the peaks with similar intensities together. CPM also aligned the same peak wrongly to the third group. Similar to DBA, the intensity of the peak is so strong that CPM decided to align this peak to the group with a higher intensity. The reference sample of GTW is just the one shown in **Fig. S9a** and GTW aligned the arrow-pointed peak correctly, while the fourth and fifth peak groups were misaligned. Again, ncGTW aligned all the peaks well, so that the nine peak groups are clearly separated and none of them is clumped together.

Next, we chose 20 samples from two batches where each batch contains 10 samples. Among the samples in each batch, the retention time drift is small. However, between the first and the second batches, the drift is larger. **Fig**. **S10a-b** show an example of each batch, and there are three peaks for each two sample. However, the third peak of most samples is missing. **Fig. S10c-d** show the first and second batches in the dataset. There are three peak groups in both batches. However, for the third peak group, only two samples in the first batch and only three samples in the second batch have the peak. To see other peaks clearer, in **Fig. S10e**, we show all the samples by changing the scale of the y-axis.

Fig. S10f-i show the results of DBA, CPM, GTW, and ncGTW. DBA aligned more than ten peaks to the third peak group with serious distortions. There should be five peaks in that peak group. CPM also aligned wrongly the third peak group, since there are too many missing peaks in the third peak group. The reference of GTW is the first sample of the first batch, as shown in **Fig. S10a**. Though there is no missing peak in the reference, GTW still tends to align the peaks with similar intensity together. Thus, the third peak group has more than five peaks and the shapes of the peaks are distorted. Only ncGTW aligned correctly the five peaks to the third peak group. This example demonstrated the significant improvement of ncGTW compared with GTW. We can see the advantage that ncGTW does not need a certain reference, and there is no distortion after alignment.

To test ncGTW on data with no structure among samples (or in the scenario that we are not sure about its structure), we use the same ten LC-MS samples in **Fig. S9b** but we do not incorporate the a priori structural information. Instead, we apply the non-informative structure by adding edges between the pairs with the same aligning sample or reference sample in the first stage of ncGTW.

Fig. S9a-d are the same as **Fig. S11a-d**, since DBA and CPM do not consider structural information already. As shown in **Fig. S11e-f**, unlike DBA and CPM, GTW and ncGTW still well aligned the pointed peak. However, without the structural information, GTW misaligned peak 3 and peak 4. Even without structural information, ncGTW accurately aligned all the peaks.

Discussion

Readers may be aware of the significant progress in aligning multiple DNA sequences. One may wonder why similar progress has not been seen in the general category of multiple alignment and why good ideas for DNA sequence alignment cannot be directly transferred to other applications. In our view, this is not surprising because many effective approaches for DNA sequences explicitly or implicitly rely on the assumption of the existence of the evolution tree. Yet, the tree structure is very special and many applications cannot be described by a tree structure. Our ncGTW can be understood as an extension of tree structure to any graph structure. In addition, unlike methods designed for DNA sequences, our method models all samples simultaneously. Though we did not test our algorithm on DNA sequencing data, we expect to see favorable performance due to the integrative modeling nature.

The experiments clearly demonstrated the power of ncGTW. When structural information is available, it is anticipated to see increased accuracy of alignment due to the use of extra information. It is a little bit counterintuitive to observe that ncGTW still performed better when there was no informative structure. From hindsight, this is also expected because one can always borrow information from other samples if we use a Bayesian perspective and consider all samples forming a prior distribution. This phenomenon is also related to Stein’s paradox in the estimation theory (Efron and Morris, 1977).

The Implementation of ncGTW on Large Dataset

For a large dataset, it is very time-consuming to simultaneously align all the samples for a profile-based alignment method. When the sample number is large, the ncGTW graph becomes extremely huge, and it may take hours or even days to solve the maximal flow problem. Thus, in the practical implementation, ncGTW splits the whole dataset into several sub-datasets, and performs alignment on each small dataset. In this way, the numbers of nodes and edges in the graph for each small dataset will decrease significantly comparing to the original graph. Also, since these small datasets are aligned independently, the computation time can be further reduced with parallel computing. (number of threads limited by the number of CPU cores). After the alignment of each small dataset, we build a "super-sample" for each small dataset. Then, align these super-samples to obtain the warping functions of super-samples. With the warping functions within each small dataset and of the super-samples, we can obtain the final warping functions for each sample. We called this hierarchical alignment process as "two-layer ncGTW". **Fig. S12** shows the diagram of two-layer ncGTW. In the following sections, we will explain each layer in details.

The first layer of two-layer ncGTW

In the first layer, to begin with, we need to decide how many sub-datasets we should split the original dataset into. For the computational efficiency, we recommend in each sub-dataset there should be at least 10 samples. For example, if there are 500 samples in the original dataset, it is recommended to split the dataset into 10 sub-datasets (50 samples in each). Then, we apply ncGTW on these sub-datasets independently. If the CPU cores the user have are more than 10, the computation time of this layer is equivalent to apply ncGTW on a 50-sample dataset once. After the alignment of each sub-dataset is done, we obtain the warping functions of all samples. With the warping functions, within each sub-dataset, the samples can be aligned to the same RT axis, and a “super-sample” will be generated by taking the average on the aligned samples. After the super sample of each sub-dataset is generated, we build a super-dataset with these super samples and send the super-dataset to the second layer with the warping functions of all samples.

The second layer of two-layer ncGTW

In the second layer, ncGTW is applied to the super-dataset to align the super-samples. Since the super-samples can be viewed as samples, we can directly apply ncGTW on them. Therefore, we can obtain a set of warping functions for super-samples. With the warping functions from the first layer (sample to super-sample) and the ones for super samples (super-sample to the final axis), all the samples can be aligned to the final axis, so that the alignment of all samples is done.

Different p-value thresholds of misalignment detection

In the main text, we set the threshold as 0.05 to detect the misaligned features. If we adjust the threshold, as shown in **Table S4**, more truly misaligned features are indeed identified when the threshold is less stringent, while expectedly the number of false positives also increases. In our ncGTW package, users have the freedom to loosen the threshold for recovering more misaligned features. Note that our experimental results also show that, when sample size is relatively large, neither the true positives nor false positives increases with further loosened p-value threshold. This is expected and actually beneficially because p-value threshold is only applied to the first of the two cooperated criterions in the misalignment detection. More precisely, when sample size is large with less stringent p-value threshold, the detection power (true positives) increases while at the potential cost of more false positives; thus, the second criteria (disjoint sets of sample indices - highly associated with true positives) is specifically designed to reduce such unwanted side-effect. For more interested user reference, the effectiveness of the second criterion against various degrees of overlapped sets of sample indices is experimentally demonstrated via **Table S5-8.**

Supplementary Tables

Peak1		
Scores / Methods	MCC	SP
Before alignment	0.3071	0.4240
DBA	0.5683	0.5095
CPM	0.9697	0.9390
GTW	0.9988	0.9986
ncGTW	0.9999	0.9999
Peak2		
Scores / Methods	MCC	SP
Before alignment	0.3072	0.4407
DBA	0.5671	0.5118
CPM	0.8943	0.8169
GTW	0.9999	0.9999
ncGTW	0.9999	0.9999
Peak3		
Scores / Methods	MCC	SP
Before alignment	0.0848	0.3959
DBA	0.4912	0.9999
CPM	0.5042	0.9384
GTW	0.3986	0.8791
ncGTW	0.5099	0.9999

Table S1. Peak scores for four methods of line structure. ‘Before alignment’ serves as the baseline. MCC and SP represent mean correlation coefficient and simplicity respectively. The range of either score is between 0 and 1 with higher score indicating better performance.

Peak1		
Scores / Methods	MCC	SP
Before alignment	0.1163	0.1937
DBA	0.7894	0.8326
CPM	0.3923	0.4603
GTW	0.9835	0.9778
ncGTW	0.9998	0.9998
Peak2		
Scores / Methods	MCC	SP
Before alignment	0.1165	0.1937
DBA	0.7885	0.7957
CPM	0.3988	0.4801
GTW	0.9301	0.9379
ncGTW	0.9999	0.9998
Peak3		
Scores / Methods	MCC	SP
Before alignment	0.0859	0.2002
DBA	0.7521	0.9999
CPM	0.3408	0.5831
GTW	0.6926	0.9407
ncGTW	0.7481	0.9998

Table S2. Peak scores for four methods of block structure.

Peak1		
Scores / Methods	MCC	SP
Before alignment	0.0686	0.2293
DBA	0.5475	0.9996
CPM	0.5931	0.9090
GTW	0.5312	0.8433
ncGTW	0.6750	0.9991
Peak2		
Scores / Methods	MCC	SP
Before alignment	0.0695	0.2545
DBA	0.2809	0.7712
CPM	0.5041	0.8091
GTW	0.4759	0.8482
ncGTW	0.6673	0.9989
Peak3		
Scores / Methods	MCC	SP
Before alignment	0.0542	0.2184
DBA	0.3108	0.5238
CPM	0.5980	0.9419
GTW	0.4707	0.8427
ncGTW	0.7435	0.9990

Table S3. Peak scores for four methods of non-informative structure.

Cut-off p-value \ Dataset	Rotterdam iQC (44)	Rotterdam study (1000)	MESA iQC (335)	MESA study (1977)
0.01	48 (35 + 13)	44 (32 + 12)	61 (58 + 3)	49 (48 + 1)
0.05	57 (41 + 16)	45 (32 + 13)	61 (58 + 3)	49 (48 + 1)
0.1	60 (42 + 18)	46 (33 + 13)	62 (58 + 4)	49 (48 + 1)
0.2	68 (44 + 24)	47 (34 + 13)	62 (58 + 4)	49 (48 + 1)

Table S4. The table of detected misaligned features at different cut-off p-values (the first criterion) with zero index overlapping rate (the second criterion). The numbers after the name of the datasets are the sample number. The first number after the detected misaligned features is the number of the true positives, and the second number is the number of the false positives. For example, there are 44 samples in the Rotterdam iQC dataset. The number of the detected misaligned feature is 48 with 0.01 as cut-off, and the number of true positives and false positives are 35 and 13 respectively.

Cut-off p-value \ Dataset	Rotterdam iQC (44)	Rotterdam study (1000)	MESA iQC (335)	MESA study (1977)
0.01	48 (35 + 13)	57 (33 + 24)	70 (58 + 12)	53 (49 + 4)
0.05	61 (42 + 19)	60 (34 + 26)	75 (58 + 17)	54 (49 + 5)
0.1	65 (43 + 22)	64 (35 + 29)	76 (58 + 18)	57 (49 + 8)
0.2	71 (45 + 26)	65 (36 + 29)	79 (58 + 21)	57 (49 + 8)

Table S5. The table of detected misaligned features at different cut-off p-value with index overlapping rate threshold as 0.1.

Dataset / Cut-off p-value	Rotterdam iQC (44)	Rotterdam study (1000)	MESA iQC (335)	MESA study (1977)
0.01	53 (35 + 18)	80 (36 + 44)	92 (58 + 34)	56 (49 + 7)
0.05	75 (42 + 33)	84 (37 + 47)	101 (58 + 43)	56 (49 + 7)
0.1	85 (43 + 42)	93 (39 + 54)	103 (58 + 45)	59 (49 + 10)
0.2	99 (46 + 53)	97 (41 + 56)	110 (58 + 52)	60 (50 + 10)

Table S6. The table of detected misaligned features at different cut-off p-value with index overlapping rate threshold as 0.5.

Dataset / Cut-off p-value	Rotterdam iQC (44)	Rotterdam study (1000)	MESA iQC (335)	MESA study (1977)
0.01	59 (35 + 24)	81 (36 + 45)	99 (58 + 41)	56 (49 + 7)
0.05	82 (43 + 39)	85 (37 + 48)	110 (58 +52)	56 (49 + 7)
0.1	91 (44 + 47)	94 (39 + 55)	114 (58 + 56)	59 (49 + 10)
0.2	104 (46 + 58)	98 (41 + 57)	123 (58 + 65)	60 (50 + 10)

Table S7. The table of detected misaligned features at different cut-off p-value with index overlapping rate threshold as 0.75.

Dataset / Cut-off p-value	Rotterdam iQC (44)	Rotterdam study (1000)	MESA iQC (335)	MESA study (1977)
0.01	62 (35 + 27)	81 (36 + 45)	101 (58 + 43)	56 (49 + 7)
0.05	85 (43 + 42)	85 (37 + 48)	112 (58 + 54)	56 (49 + 7)
0.1	95 (44 + 51)	94 (39 + 55)	116 (58 + 58)	59 (49 + 10)
0.2	109 (46 + 63)	98 (41 + 57)	125 (58 + 67)	60 (50 + 10)

Table S8. The table of detected misaligned features at different cut-off p-value with index overlapping rate threshold as 1.

Dataset	m/z	RT	Metabolite annotation
MESA	431.4	184.9	alpha-tocopherol-glucuronide
MESA	629.4	184.9	alpha-tocopherol-glucuronide
MESA	430.4	187.1	alpha-tocopherol-glucuronide
MESA	1176.7	275.1	Ganglioside GM3
MESA	1175.7	275.2	Ganglioside GM3
MESA	520.5	275.6	Ganglioside GM3
MESA	844.6	275.6	Ganglioside GM3
MESA	599.5	293.3	Phosphatidylinositol
MESA	879.5	293.8	Phosphatidylinositol
MESA	575.5	293.9	Phosphatidylinositol
MESA	857.5	293.9	Phosphatidylinositol
MESA	576.5	293.9	Phosphatidylinositol
MESA	881.5	294.4	Phosphatidylinositol
MESA	903.5	294.6	Phosphatidylinositol
MESA	882.5	294.7	Phosphatidylinositol
MESA	600.5	294.9	Phosphatidylinositol
MESA	907.5	307.5	Phosphatidylinositol
MESA	625.5	307.8	Phosphatidylinositol
MESA	908.5	332.6	Phosphatidylinositol
MESA	907.5	333.6	Phosphatidylinositol
MESA	886.5	334.0	Phosphatidylinositol
MESA	885.5	334.0	Phosphatidylinositol
MESA	603.5	334.3	Phosphatidylinositol
MESA	604.5	334.3	Phosphatidylinositol
MESA	605.5	334.4	Phosphatidylinositol
MESA	910.5	334.4	Phosphatidylinositol
MESA	909.5	334.4	Phosphatidylinositol
MESA	931.5	334.5	Phosphatidylinositol
MESA	932.5	334.5	Phosphatidylinositol
MESA	628.5	334.6	Phosphatidylinositol
MESA	341.3	334.7	Phosphatidylinositol
MESA	342.3	334.7	Phosphatidylinositol
MESA	627.5	334.7	Phosphatidylinositol
MESA	269.2	335.2	Phosphatidylinositol
MESA	925.5	335.3	Phosphatidylinositol
MESA	911.6	338.7	Phosphatidylinositol
MESA	629.5	344.0	Phosphatidylinositol
MESA	630.6	346.8	Phosphatidylinositol
Rotterdam	104.1	24.2	Choline
Rotterdam	342.3	111.5	Lysophosphatidylinositol
Rotterdam	623.3	112.4	Lysophosphatidylinositol
Rotterdam	341.3	112.5	Lysophosphatidylinositol
Rotterdam	583.3	115.4	Lysophosphatidylinositol
Rotterdam	909.5	284.1	Phosphatidylinositol
Rotterdam	603.5	285.0	Phosphatidylinositol
Rotterdam	604.5	285.0	Phosphatidylinositol
Rotterdam	627.5	285.0	Phosphatidylinositol
Rotterdam	628.5	285.0	Phosphatidylinositol
Rotterdam	885.5	284.8	Phosphatidylinositol

Table S9. The annotations of features in MESA and Rotterdam datasets with their m/z and RT positions.

Supplementary Figures

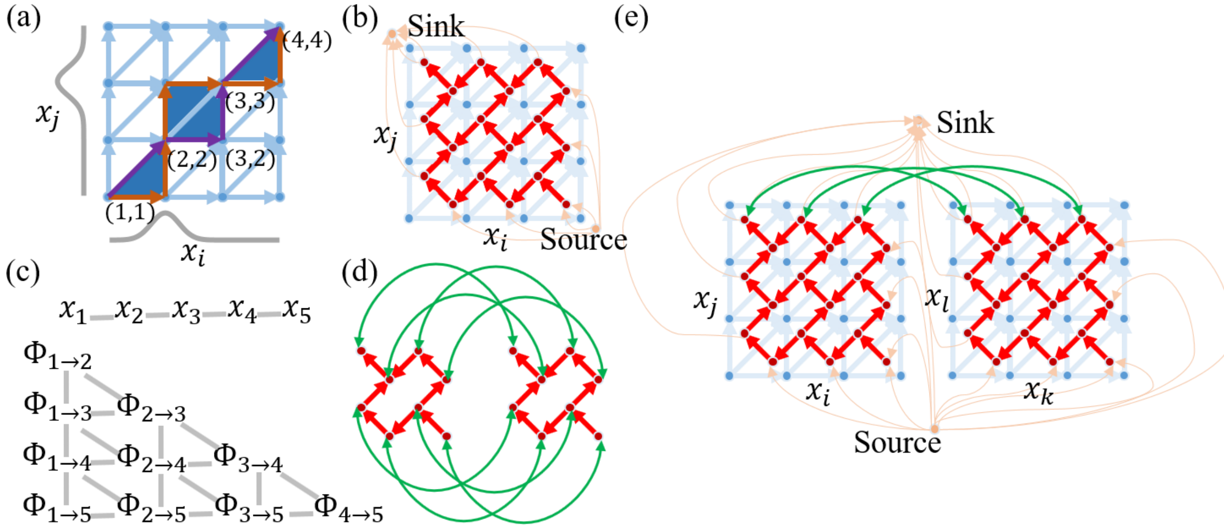

Figure. S1. Figures of DTW grids, DTW graphs, structural information diagram for GTW, and GTW graph. (a) A DTW grid for aligning x_i to x_j. The purple path and the orange path correspond to two different warping functions. Each node in the grid corresponds to a pair of points, one from x_i and the other from x_j. For example, node (3, 2) corresponds to the third point on x_i (x_{i3}) and the second point on x_j (x_{j2}). The weight of an edge is determined by its starting node. For example, the weight of the edge ((3, 2), (3, 3)) is given by the distance between (x_{i3}, x_{j2}). The distance between the purple path and the orange path is defined as the area in dark blue (four triangles in this case). The corresponding warping function of the purple path is {(1, 1), (2, 2), (3, 2), (3, 3), (4, 4)}. (b) A DTW grid and the corresponding DTW graph. The blue lines and dots form the original DTW grid, and the red and orange lines and dots form the corresponding DTW graph. Note here orange lines link only the vertices (those red dots enclosed by blue lined exterior triangle) to a single source or sink. (c) An example of structural information between samples and the induced structural information between pairwise warping functions. Suppose there are five LC-MS samples and the run orders of which are exactly 1, 2, 3, 4, and 5. These samples are expected to have continuous changing profiles from smaller indices to larger indices. More importantly, the continuous changing gives rise to the similarities between warping function. For example, the warping function for curve pair (x_1, x_2) is similar to the warping function for (x_2, x_3). In general, curve pairs (x_i, x_{i+1}) and (x_{i+1}, x_{i+2}) are considered as neighbors. Similarly, curve pairs (x_i, x_j)

and (x_i, x_{j+1}), along with curve pairs (x_i, x_j) and (x_{i+1}, x_j) are also neighbors. The diagram shows all warping function neighboring information. (d) Two small DTW graphs connected together. Green lines are the additional edges linking the corresponding vertices of the DTW graphs. (e) A GTW graph formed by two linked DTW graphs. Two warping functions ($\Phi_{i,j}$ and $\Phi_{k,l}$) are neighbors. Orange edges connect vertices to source and sink. Green edges link the corresponding vertices in two graphs. For clarity, we only show links between top three vertices. Other vertices are linked in the same way.

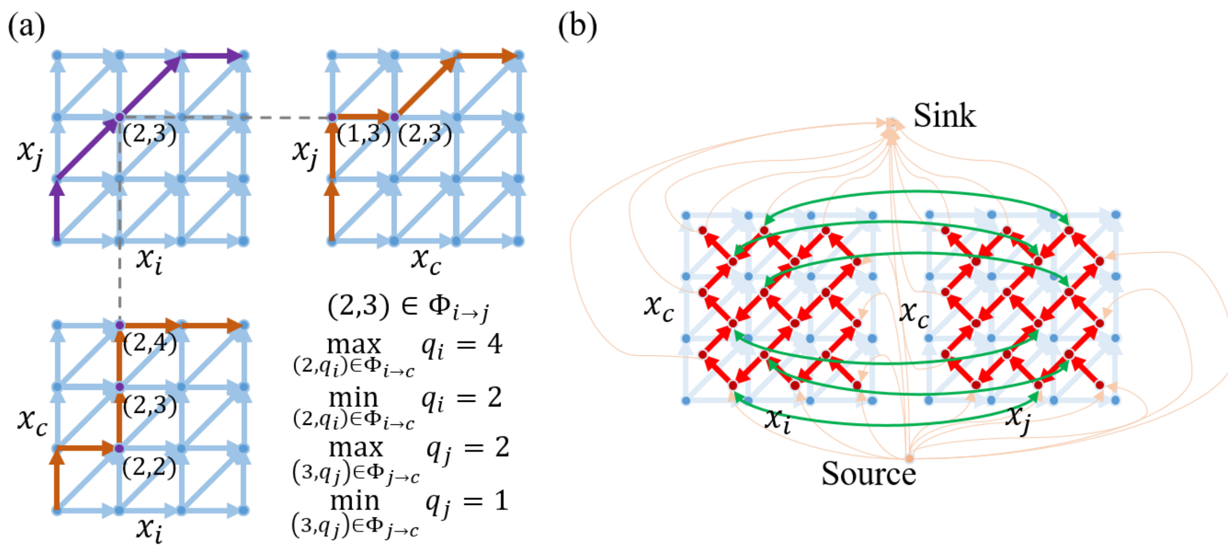

Figure S2. The illustration of Stage 2 of ncGTW. (a) Example of inconsistency and non-diagonality calculation. From the upper-left DTW grid, to achieve consistency, the 2nd point on x_i and the 3rd point on x_j should be aligned to the same position on the virtual reference, because $(2,3) \in \Phi_{i,j}$. It is clearly not the case by looking at the other two DTW grids. From $\Phi_{i,c}$ (lower-left DTW grid), we know that the 2nd point on x_i is aligned to points 2, 3, and 4 on the reference. From $\Phi_{c,j}$ (upper-right DTW grid, the inverse of $\Phi_{j,c}$), we know that the 3rd point on x_j is aligned to points 1 and 2 on the reference. Thus, by definition the inconsistency from (2, 3) is $|4-2| + |2-1| = 3$. The total inconsistency is calculated along all nodes in $\Phi_{i,j}$. The non-diagonality of $\Phi_{i,c}$ is 6, since there are totally 6 corresponding vertical and horizontal paths in the DTW grid. The non-diagonality of $\Phi_{i,j}$ is 2. (b) The graph of stage 2 of ncGTW. From stage 1, we have all pairwise warping functions. If from the warping function $\Phi_{i,j}$, we know that the 2nd point in x_i is aligned to the 3rd point in x_j, this graph shows how to link the related vertices, where x_c is the virtual reference.

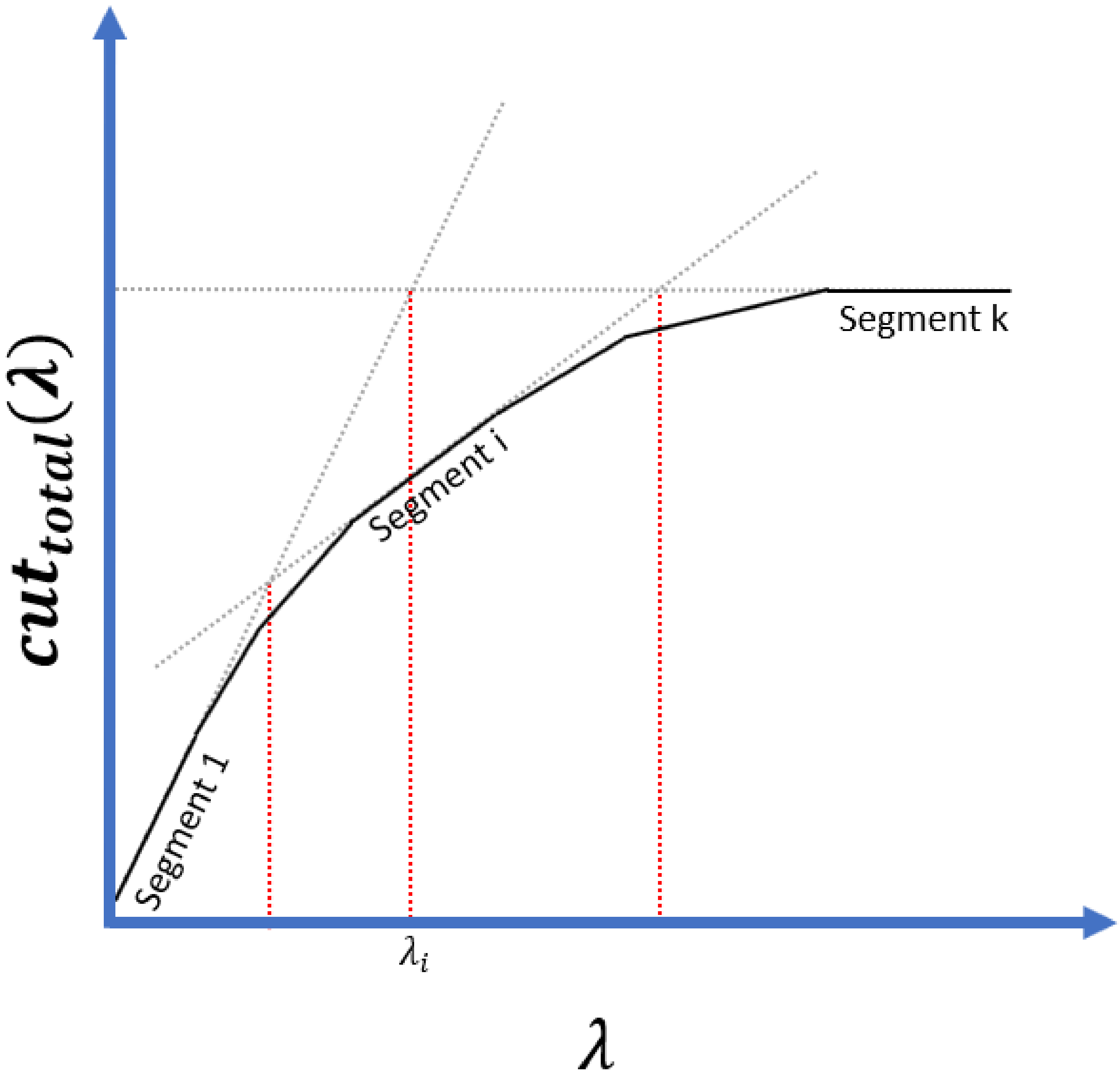

Figure S3. The illustration of the strategy of finding the line segments of $cut_{total}(\lambda)$. The first line segment (segment 1) can be found by solving each DTW graph separately (**Corollary 1**). The last line segment (segment k) can be found by solving the new DTW graph (**Corollary 2**). After extending segment 1 and segment k, we can obtain a crossing point, whose corresponding position on the λ axis is λ_i. With λ_i, we can solve the minimum cut problem to identify segment i. Similarly, if we extend segment i, we can find the crossing points with segment 1 and segment k and we can further identify new segments. If we repeat this step, more line segments can be identified.

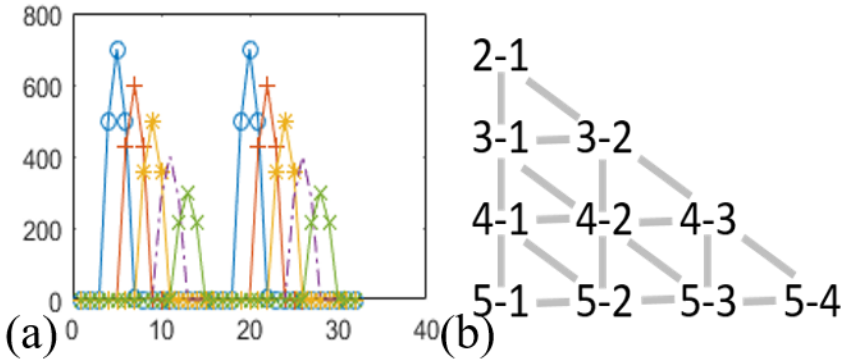

Figure S4. Example of five-curve with line structure. (a) Five continuously changing samples: the first one is drawn in “o”, second in “+”, third in “*”, fourth in “-”, and fifth in “x”. The shifts between the directly neighboring samples are all two points. (b) The induced neighborhood structure between warping functions.

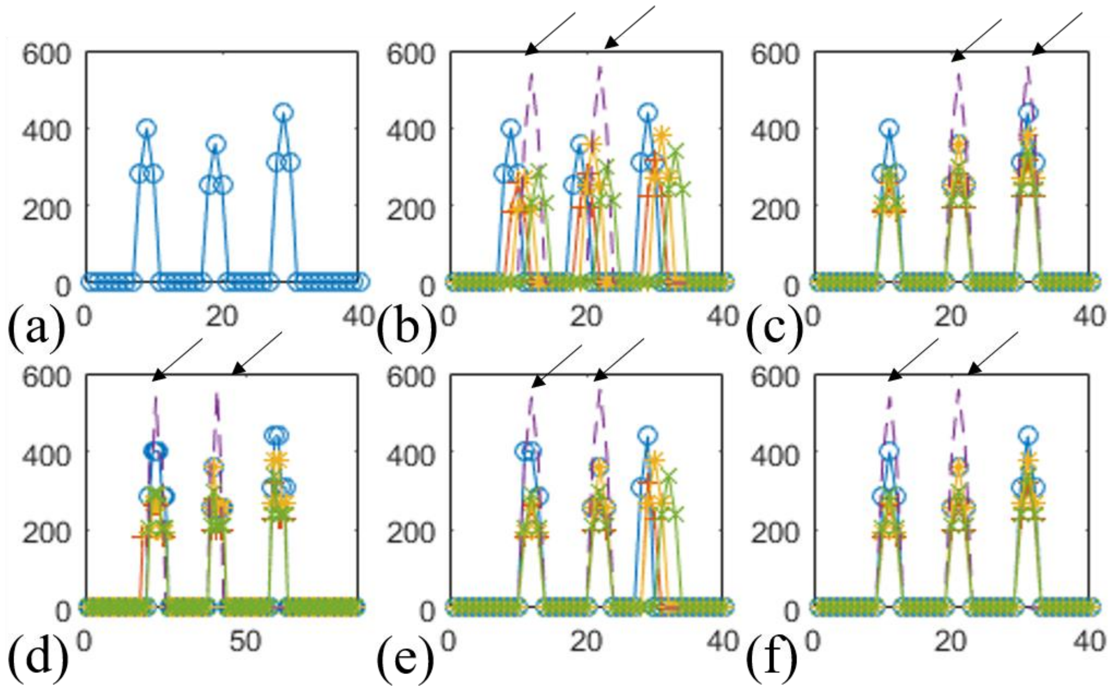

Figure S5. Case study on line structure. (a) The first sample. (b) The five synthetic samples are drawn in the order of “o”, “+”, “*”, “-”, and “x”. The shifts between neighboring samples are all one. All samples contain three peaks, except for the fourth sample, which misses the third peak (the other two are pointed by the arrows). (c) Alignment result of DBA. (d) CPM. (e) GTW. (f) ncGTW.

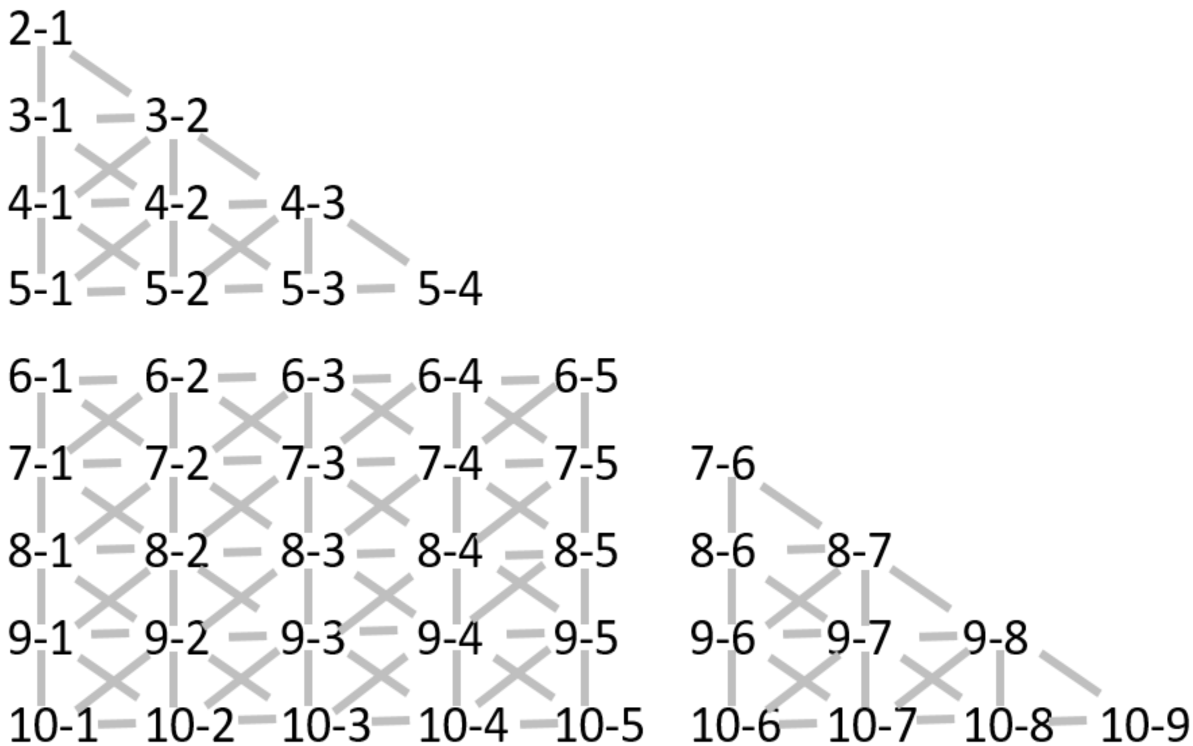

Figure S6. The way to connect samples in a dataset with two blocks (5 samples in each block). One can see there are three types of pairs: within the first group, between 2 groups, and within the second group.

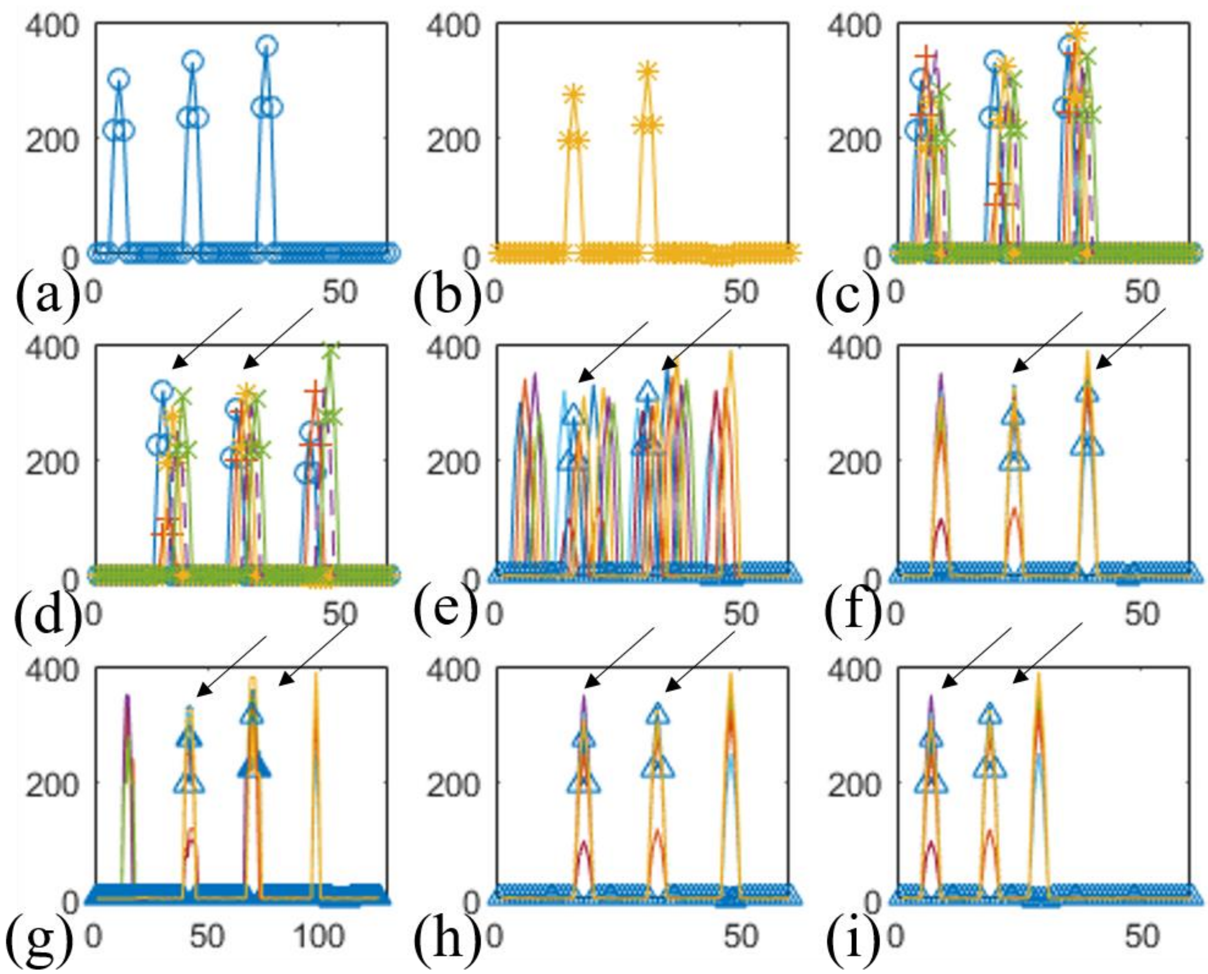

Figure S7. Case study on block structure. (a) The first sample. (b) The eighth sample. The third peak is missing. (c) The first block of the dataset. The samples are drawn in the order of "o", "+", "*", "-", and "x". The shifts between neighboring samples are all one. (d) The second block of the dataset. The five samples are drawn the same way as the previous subfigure. In the following subfigures, only the eighth sample is drawn with marks (triangle). Its peaks are pointed by arrows. (e) All ten samples. The shift between the two blocks is seven points. (f) Alignment result of DBA. (g) CPM. (h) GTW. (i) ncGTW.

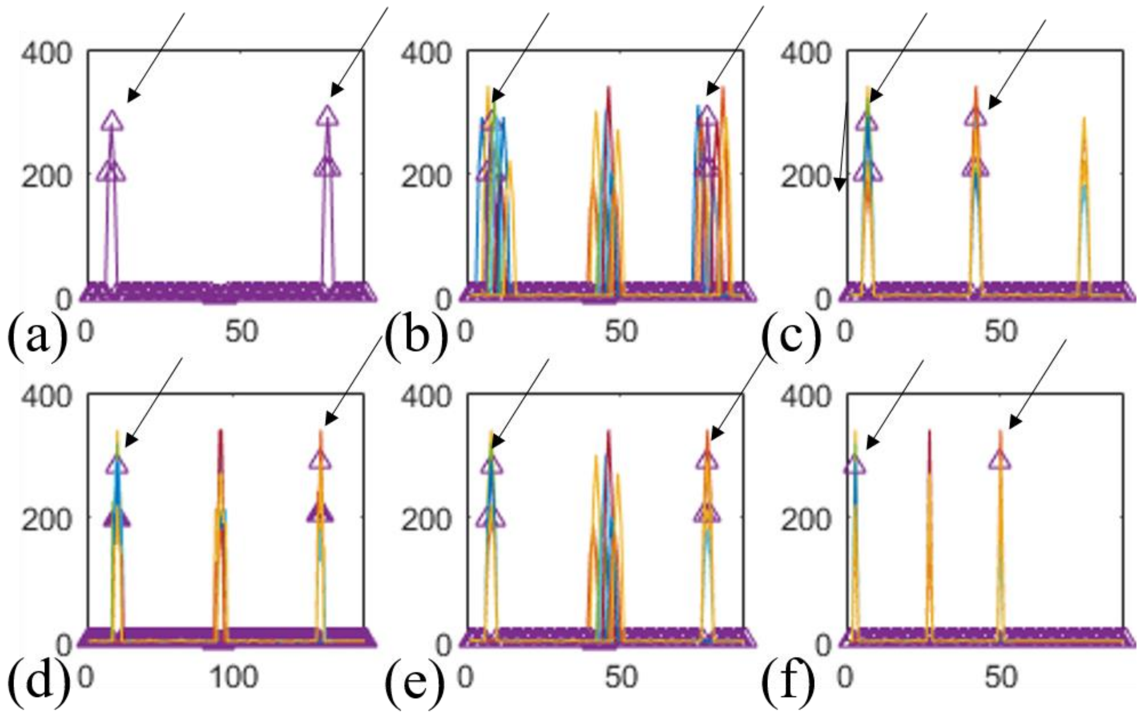

Figure S8. Case study on non-informative structure. (a) A sample without peak 2 (marked as triangles in the following subfigures). (b) All ten simulated samples. (c) Alignment result of DBA. (d) CPM. (e) GTW. (f) ncGTW.

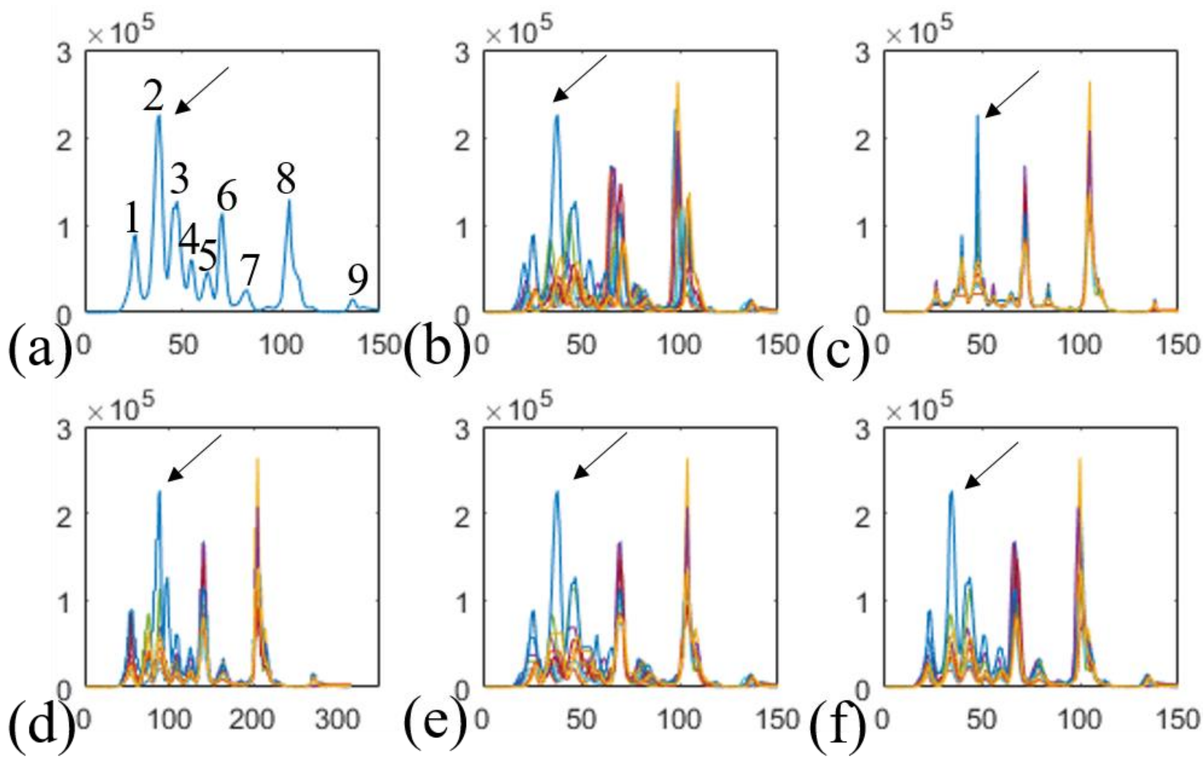

Figure S9. Case study on real LC-MS dataset of ten samples. (a) One exemplary sample with nine clear peaks marked with index. The second peak (pointed by an arrow) was wrongly aligned by most methods except GTW and ncGTW. (b) All ten samples. (c) Result of DBA. (d) CPM (e) GTW (f) ncGTW.

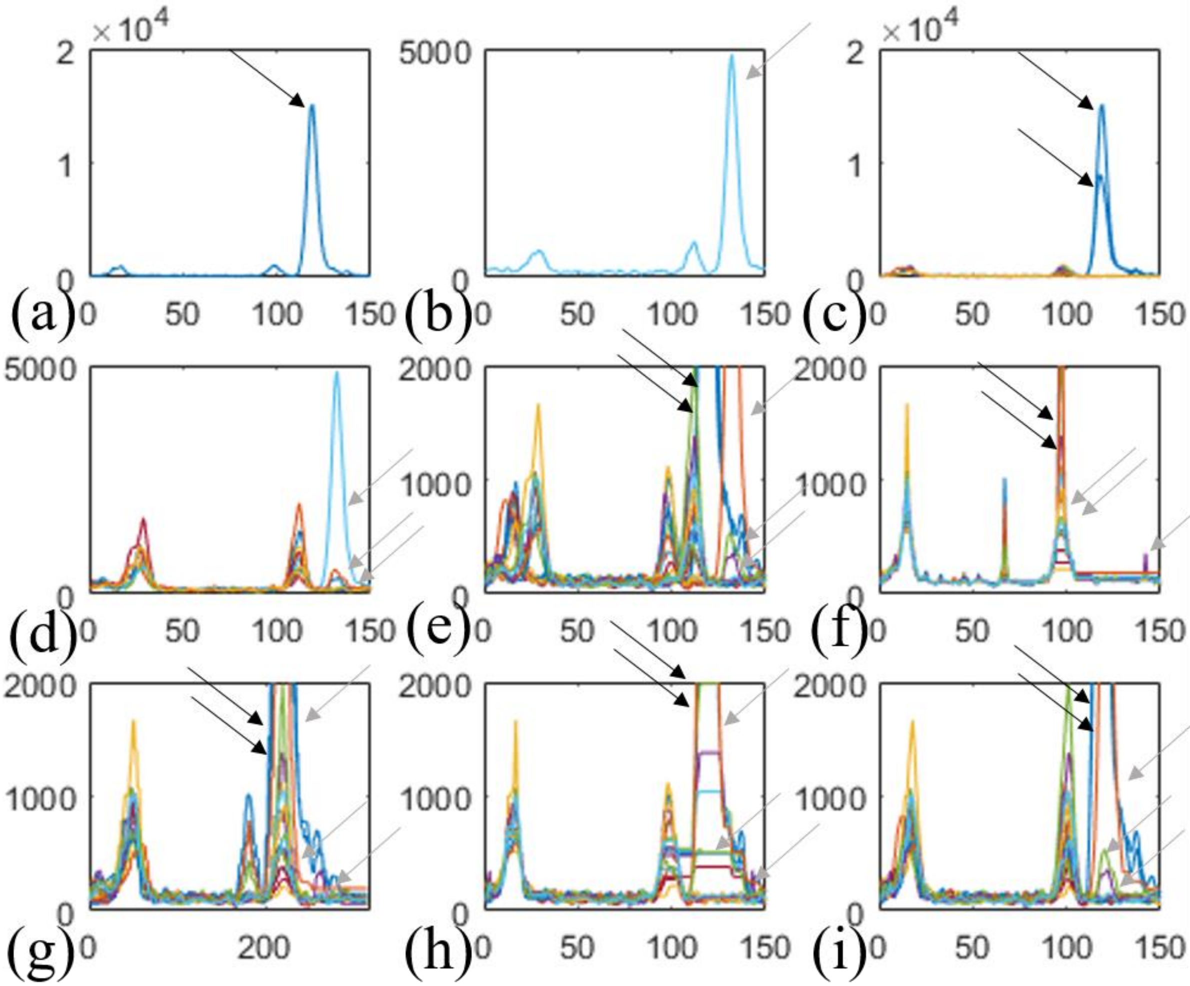

Figure S10. Case study on real LC-MS dataset of twenty samples from two batches. (a) The first sample from the first batch, in which there are three peaks. (b) The sixth sample from the second batch, in which there are also three peaks. (c) The first batch. For the third peak group, only two samples have the peak (pointed by the black arrows). (d) The second batch. Only three samples have the peak in the third peak group (pointed by the gray arrows). (e) All twenty samples. (f) Result of DBA. (g) CPM. (h) GTW. (i) ncGTW.

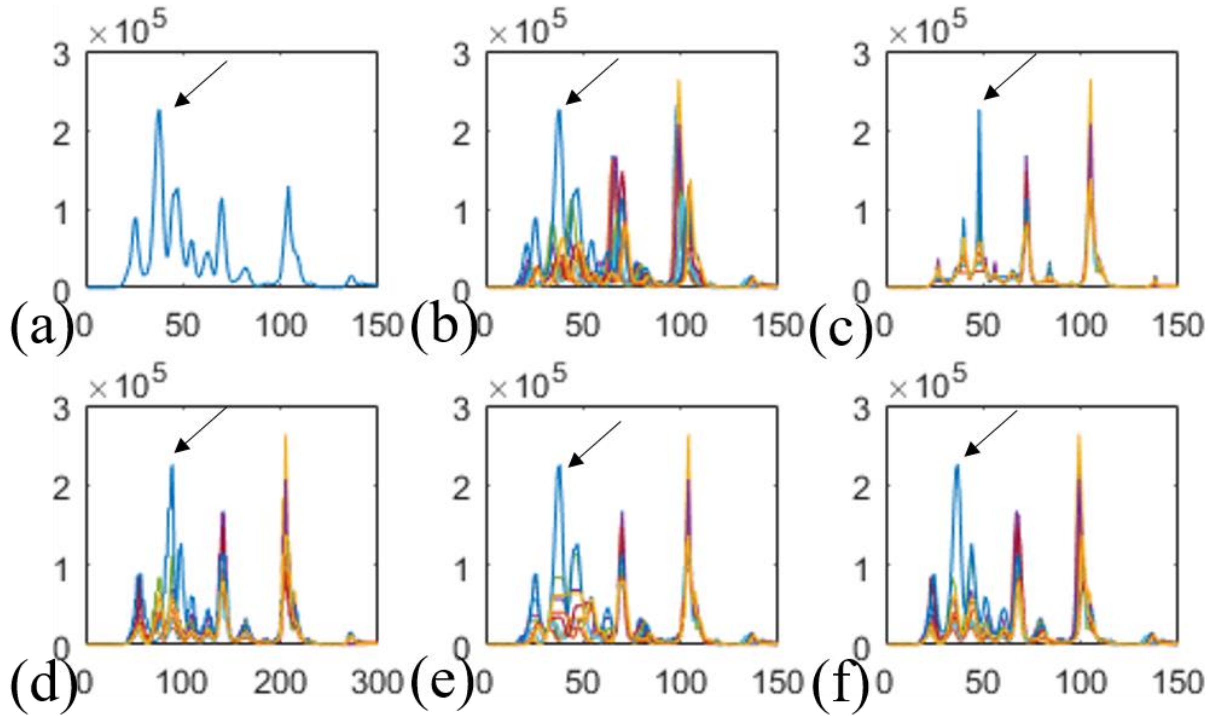

Figure S11. Case study on real LC-MS dataset of ten samples without prior knowledge of the structure information. (a) One exemplary sample with nine clear peaks marked with index. The second peak (pointed by an arrow) was wrongly aligned by most methods except GTW and ncGTW. (b) All ten samples. (c) Result of DBA. (d) CPM. (e) GTW. (f) ncGTW.

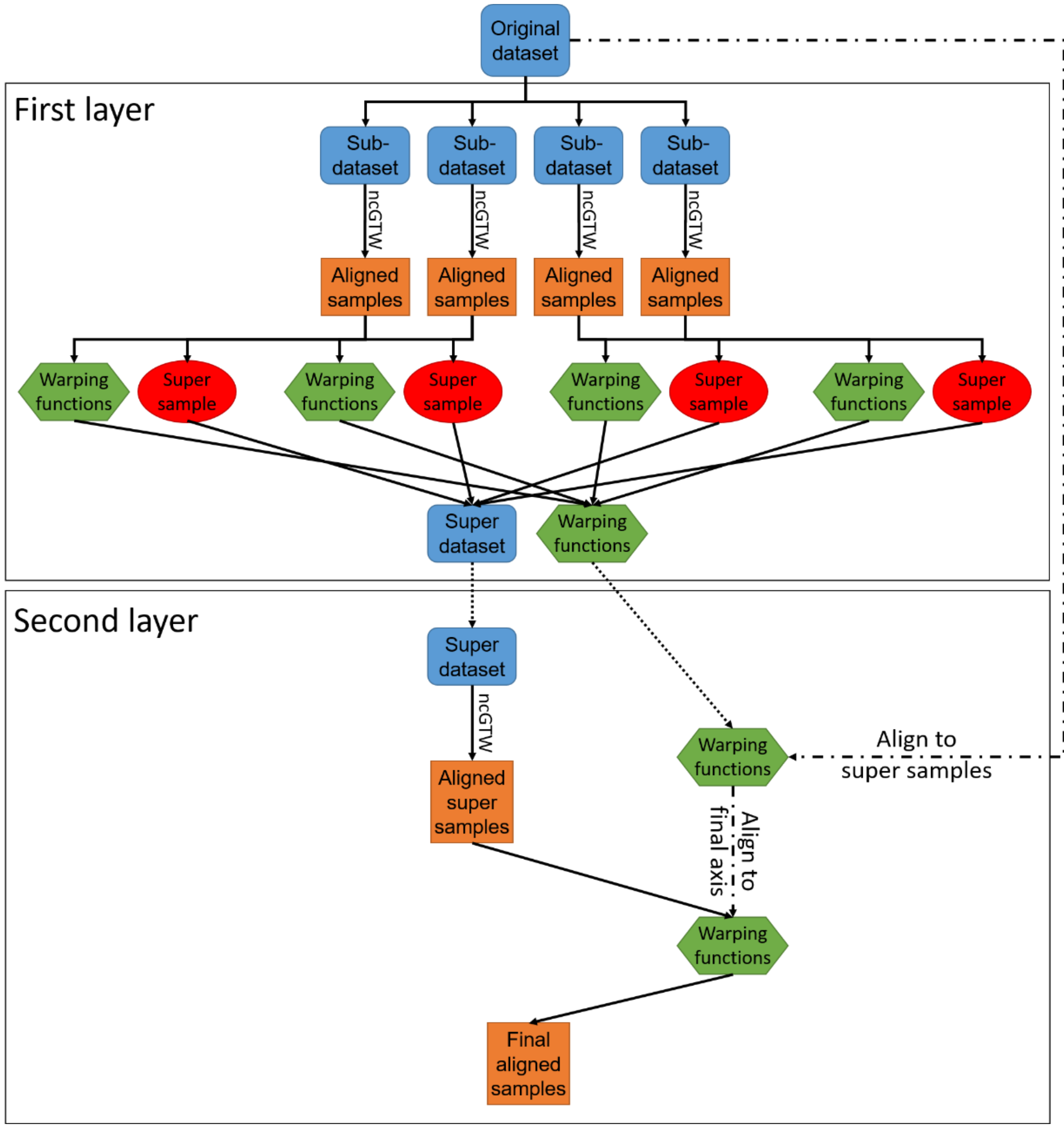

Figure S12. Diagram of two-layer ncGTW

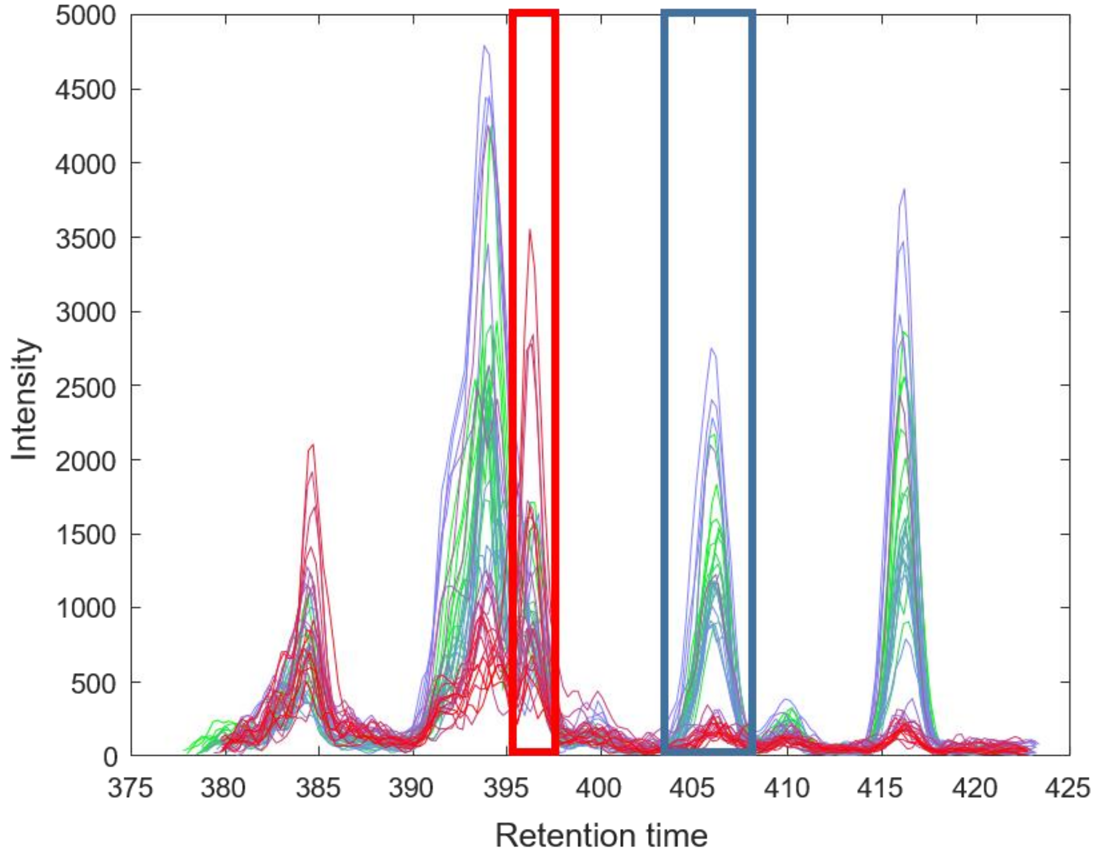

Figure S13. An XCMS aligned feature from the Rotterdam dataset as an example of false positives of the misalignment detection algorithm. One can see that all the peaks are aligned well. That is, unlike **Fig. 6b** in the main article, no peaks are spread consecutively across RT, Thus, this detected feature is considered as a false positive. The sample indexes in the red box are 34, 36, and 40, and the p-value is 0.034. Apparently, there are more than three peaks in the red box, but only three of them are detected. Moreover, the sample indexes in the blue box are 1, 4, 9, 21, 23, and 24, and the p-value is 0.047. Again, there are many peaks that are not detected in the blue box. Both of the p-values are lower than the threshold and the index sets are disjoint, so this feature is reported as misaligned.

References

Arnold, B.C., Balakrishnan, N. and Nagaraja, H.N. A first course in order statistics. Siam; 1992.

Connor, R.J. The Sampling Distribution of the Range from Discrete Uniform Finite Populations and a Range Test for Homogeneity. *Journal of the American Statistical Association* 1969;64(328):1443-1458.

Efron, B. and Morris, C. Stein's paradox in statistics. *Scientific American* 1977;236(5):119-127.

Jiang, W., *et al.* Comparisons of five algorithms for chromatogram alignment. *Chromatographia* 2013;76(17-18):1067-1078.

Korte, B., *et al.* Combinatorial optimization. Springer; 2012.

Listgarten, J., *et al.* Multiple alignment of continuous time series. In, *Advances in neural information processing systems*. 2005. p. 817-824.

Petitjean, F., Ketterlin, A. and Gançarski, P. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition* 2011;44(3):678-693.

Wang, Y., *et al.* Graphical time warping for joint alignment of multiple curves. In, *Advances in Neural Information Processing Systems*. 2016. p. 3648-3656.