Alignment of protein mass spectrometry data by integrated Markov chain shifting method: supplementary notes

I. IMS semiparametric model for multiple samples

We have illustrated the integrated Markov chain shifting (IMS) mass spectrometry (MS) alignment method for the one-sample case, and we have claimed that the extension to two or multiple samples (groups) is straightforward. We present in this section a multi-sample IMS model, a corresponding algorithm for solving the nonparametric profile maximum likelihood estimation (NPMLE) and some simulation results.

The one-sample semiparametric model

$$y_i(t) = \alpha_i + \beta_i m(x_i(t) - s_i(t)) + \epsilon_i(t)$$

can be extended to incorporate multiple samples (groups) by introducing different shape functions for different samples. Suppose there are G different samples, and there are n_g individual MS curves in sample g. Let the shape function for sample g be $m_g(\cdot)$ and assume the following semiparametric model for observed curves

$$y_{gi}(t) = \alpha_{gi} + \beta_{gi} m_g(x_{gi}(t) - s_{gi}(t)) + \epsilon_{gi}$$

where $\epsilon_{gi} \sim N(0, \sigma^2)$ are independent normal errors, $x_{gi}(t)$ the observed m/z values and $s_{gi}(t)$ the unobserved random shift effects at time t, where $g = 1, ..., G$ and $i = 1, 2, ..., n_g$. The random shift is the double integral of a second-order Markov chain as described in Section 2. In order for the model to be identifiable, assume $\alpha_{g1} = 0$ and $\beta_{g1} = 1$. The NPMLE algorithm for this model is basically the same as that for the one-sample model. The only differences are that the updating formula for m_g in the original algorithm (see the algorithm in Methods section) should take summation over observations $i = 1, 2, .., n_g$, and in each iteration α_{g1}, β_{g1}, $g = 1, 2, ..., G$, should be normalized.

Table 1: Regression parameter estimates ($h^2 = 5$)

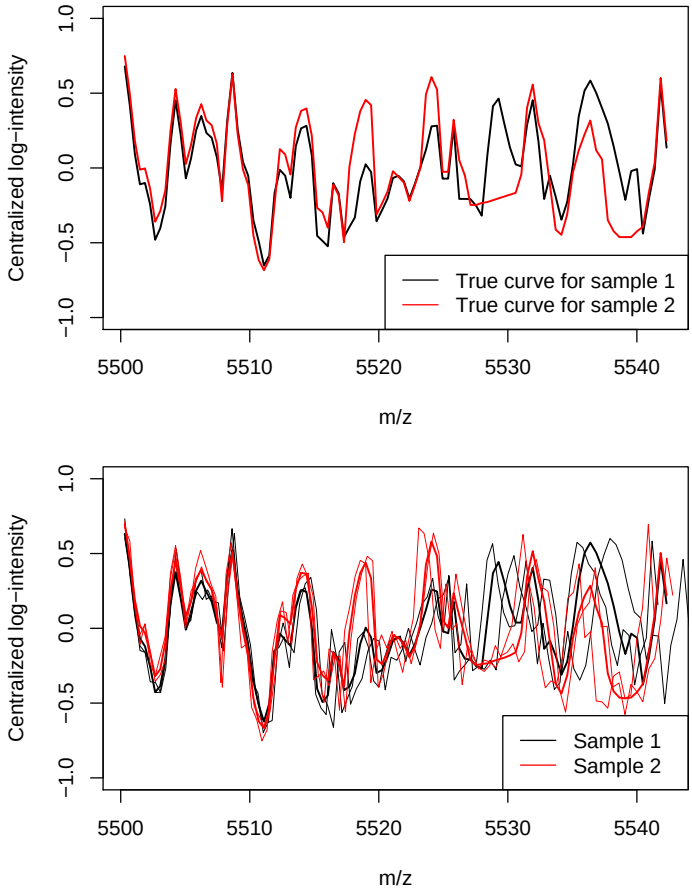
			Sample 1		Sample 2	
ID	α	β	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\alpha}$	$\hat{\beta}$
1	0	1	0.000(0.000)*	1.000(0.000)	0.000(0.000)	1.000(0.000)
2	0	1	0.000(0.014)	1.004(0.049)	0.002(0.015)	0.996(0.041)
3	-0.25	1.25	-0.255(0.017)	1.243(0.058)	-0.244(0.016)	1.255(0.048)
4	0.25	1.25	0.245(0.019)	1.251(0.058)	0.255(0.016)	1.258(0.050)
5	-0.5	1.5	-0.502(0.019)	1.505(0.054)	-0.497(0.015)	1.487(0.056)
6	0.5	1.5	0.498(0.019)	1.509(0.060)	0.502(0.017)	1.487(0.055)
7	-0.75	1.75	-0.756(0.021)	1.742(0.072)	-0.742(0.021)	1.750(0.065)
8	0.75	1.75	0.746(0.022)	1.747(0.065)	0.758(0.018)	1.746(0.068)
9	-1	2	-1.001(0.022)	2.004(0.076)	-0.996(0.023)	1.986(0.073)
10	1	2	0.996(0.023)	2.010(0.077)	1.002(0.022)	1.990(0.075)

*Standard deviations are in parentheses

Following are some simulation results for two-sample case. The true regression parameters (α, β) are shown in Table 1, the true standard deviation for the error terms is taken to be $\sigma = 0.1$. The two true curves $m_1(\cdot)$ and $m_2(t)$ for the two groups are displayed in the top panel of Figure 1. The peak patterns

are similar at most of the m/z values for these two samples, though the intensity values differ for some of the m/z values. The major difference lies at m/z value around 5530 where group 1 has a peak and group 2 is flat. We repeatedly generate 10 curves from each group, and implement 500 replicates of simulation. The bottom panel of Figure 1 displays two observed curves for each group (thin lines), the fitted curves using $h^2 = 5$ for the two groups are the two thick lines, which are seen to be very accurate estimates of the true curves as shown in the top panel. Variabilities in both m/z and intensities can be clearly seen for the observed curves in both groups. Especially at the right end, the two observed curves in each group are shifted prominently.

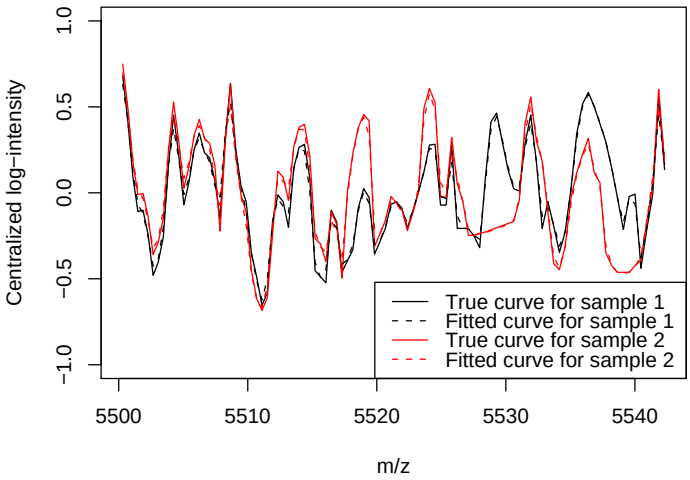
Figure 1: Two sample/group curve estimation with IMS alignment (Top panel: true curves; bottom panel: observed (thin lines) and fitted (thick lines, $h^2 = 5$) curves for the two groups).



To keep the signal information we have used a small bandwidth $h^2 = 5$, which corresponds to a 0.11% sliding window in the common peak alignment algorithm. This is because we have used Gaussian kernel, the 99% confidence interval has width $2 \times 2.576h = 11.52$ which is about $\pm 0.11\%$ of the m/z values ranging between 5500 to 5540 we had simulated. Figure 2 compares the fitted curves and the true curves for the two samples. It can be seen that the estimated curve for each sample fits the true curve almost perfectly, while the sample/group differences are kept. These fitted curves are obtained from randomly shifted curves. Table 1 lists the estimated regression parameters for $h^2 = 5$. All the estimates are close to the true values.

The standard deviation of the noise is estimated to be $\hat{\sigma} = 0.123$ with standard error 0.0026.

Figure 2: Observed and fitted curves for two-sample (thick lines: fitted curves; thin lines: observed curves with random shifts).



It is worth noting that the proposed IMS model is in fact a marginal model for log-intensities. The dependence structure of neighboring intensities is hard, if not impossible, to capture, and the normality assumption may not be true in a strict sense in practice. In some sense, the independence and normality assumptions in our model are only used to derive an effective algorithm. Validity of this approach is analogous to the validity of least squares method when the latter is applied to data that maybe neither independent nor normally distributed.

II. Derivations of the NPMLE estimates in the one-sample model

The likelihood function for the spectra can be written as

$$L(m,\theta) = \prod_{i=1}^{n} f_{\mathbf{y_i}}(\mathbf{y}_i) = \prod_{i=1}^{n} \int f_{\mathbf{y_i}|\mathbf{s}_i}(\mathbf{y}_i) f_{\mathbf{s}}(\mathbf{s}_i) d\mathbf{s}_i$$

$$= \prod_{i=1}^{n} \int \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{||\mathbf{y}_i - \alpha_i - \beta_i m(\mathbf{x}_i - \mathbf{s}_i)||^2}{2\sigma^2}\right) f_{\mathbf{s}_i}(\mathbf{s}_i) d\mathbf{s}_i, \tag{1}$$

where $\mathbf{y}_i = (y_i(u), u = 1, ..., N), \mathbf{x}_i = (x_i(u), u = 1, ..., N)$, $m(\mathbf{x}_i - \mathbf{s}_i) = (m(x_i(u) - s_i(u)), u = 1, ..., N)$. Let $f_{\mathbf{s}_i|\mathbf{y}_i}(\mathbf{s_i})$ be the conditional probability function of $\mathbf{s}_i$ given $\mathbf{y}_i$:

$$f_{\mathbf{s}_i|\mathbf{y}_i}(\mathbf{s}_i) = \frac{f_{\mathbf{y}_i|\mathbf{s}_i}(\mathbf{y}_i) f_{\mathbf{s}_i}(\mathbf{s}_i)}{\int f_{\mathbf{y}_i|\mathbf{s}_i}(\mathbf{y}_i) f_{\mathbf{s}_i}(\mathbf{s}_i) d\mathbf{s}_i},$$

where

$$f_{\mathbf{y}_i|\mathbf{s}_i}(\mathbf{y}_i) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}||\mathbf{y}_i - \alpha_i - \beta_i m(\mathbf{x}_i - \mathbf{s}_i)||^2\right).$$

We then proceed with estimating the unknown quantities as follows.

Given m, we maximize the profile likelihood over the remaining parameters to get

$$\hat{\beta}_i = \frac{\sum_{u=1}^N E_{s_i(u)|\mathbf{y}_i}[m(x_i(u) - s_i(u)) - \bar{m}_i]y_i(u)}{\sum_{u=1}^N E_{s_i(u)|\mathbf{y}_i}[m(x_i(u) - s_i(u)) - \bar{m}_i]^2}, \tag{2}$$

$$\hat{\alpha}_i = \bar{y}_i - \bar{m}_i\hat{\beta}_i, \tag{3}$$

and

$$\hat{\sigma}^2 = \frac{1}{nN}\sum_{i=1}^n\sum_{u=1}^N E_{s_i(u)|\mathbf{y}_i}(y_i(u) - \alpha_i - \beta_i m(x_i(u) - s_i(u)))^2, \tag{4}$$

where $\bar{y}_i = \sum_{u=1}^N y_i(u)/N$, $\bar{m}_i = E_{\mathbf{s}_i|\mathbf{y}_i}[\sum_{u=1}^N m(x_i(u) - s_i(u))/N]$. As we mentioned before, for all parameters to be identifiable, we need to set

$$\hat{\beta}_i = \hat{\beta}_i/\hat{\beta}_1, \quad \hat{\alpha}_i = \hat{\alpha}_i - \hat{\alpha}_1, i = 1, ..., n.$$

To derive the score function for m, we use the Hadmard derivatives for functionals. Given an arbitrary direction $h(\cdot)$, the score for $m(\cdot)$

$$\begin{aligned}
&\frac{\partial \log L}{\partial m}(h) = \sum_{i=1}^n \frac{1}{L_i(m,\theta)}\frac{\partial L_i(m + uh, \theta)}{\partial u}|_{u=0} \\
&= -\sigma^{-2}\sum_{i=1}^n \frac{\int \beta_i\langle h(\mathbf{x}_i - \mathbf{s}_i), \mathbf{y}_i - \alpha_i - \beta_i m(\mathbf{x}_i - \mathbf{s}_i)\rangle f_{\mathbf{y}_i|\mathbf{s}_i}(\mathbf{y}_i)f_{\mathbf{s}_i}(\mathbf{s}_i)d\mathbf{s}_i}{\int f_{\mathbf{y}_i|\mathbf{s}_i}(\mathbf{y}_i)f_{\mathbf{s}_i}(\mathbf{s}_i)d\mathbf{s}_i} \\
&= -\sigma^{-2}\sum_{i=1}^n E_{\mathbf{s}_i|\mathbf{y}_i}\beta_i\langle h(\mathbf{x}_i - \mathbf{s}_i), \mathbf{y}_i - \alpha_i - \beta_i m(\mathbf{x}_i - \mathbf{s}_i)\rangle \\
&= -\sigma^{-2}\sum_{i=1}^n\sum_{u=1}^N \beta_i E_{s_i(u)|\mathbf{y}_i}\left[y_i(u) - \alpha_i - \beta_i m(x_i(u) - s_i(u))\right] h\left(x_i(u) - s_i(u)\right).
\end{aligned}$$

Since this is linear in h, we may choose various h in the score to obtain a series of equations. In particular, letting $h(r) = \delta_t(r)$, the Dirac function, in the score equation $\frac{\partial \log L}{\partial m}(h) = 0$, we get

$$\hat{m}(t) = \frac{\sum_{i=1}^n\sum_{u=1}^N \beta_i(y_i(u) - \alpha_i)f_{s_i(u)|\mathbf{y}_i}(x_i(u) - t)}{\sum_{i=1}^n\sum_{u=1}^N \beta_i^2 f_{s_i(u)|\mathbf{y}_i}(x_i(u) - t)}, \tag{5}$$

which can be shown to be the solution to $\frac{\partial \log L}{\partial m}(h) = 0$ for any h.

III. Kernel approximation to the NPMLE in the one-sample model

Calculation of the conditional density $f_{s_i(u)|\mathbf{y}_i}(x_i(u) - t)$ is rather involved and inaccurate. One may approximate it by a kernel method. Let $K(\cdot)$ be a kernel density function such that $K \geq 0$, $\int K(u)du = 1$, symmetric and attains the maximum at 0. For example, the kernel function can be taken as the Gaussian density function $K(x) = e^{-\frac{x^2}{2}}$. The NPMLE for $m(\cdot)$ in (5) can be approximated through taking $h_t(u) = K(|u - t|/h)$ in the score equation for an arbitrarily small $h > 0$. Intuitively as $h \to 0$, $h_t(u)$ converges to the Dirac function $\delta_t(u)$, i.e., as $h \to 0$,

$$\int g(x)K(|x - t|/h)dx \to g(t) = \int g(x)\delta_t(x)dx,$$

if g is continuous. This may be best understood when the kernel is taken to be the Gaussian density function, for which h is the standard deviation, and as the standard deviation approaches 0, mass concentrate around the center/mean t, and the expectation $Eg(x)$ approaches $g(t)$. Thus

$$\begin{aligned} & \sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i E_{\mathbf{s}_i|\mathbf{y}_i}\left\{(y_i(u)-\alpha_i-\beta_i m(x_i(u)-s_i(u)))h_t(x_i(u)-s_i(u))\right\} \\ \approx & \sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i(y_i(u)-\alpha_i-\beta_i m(t))E_{\mathbf{s}_i|\mathbf{y}_i}K(|x_i(u)-s_i(u)-t|/h). \end{aligned} \tag{6}$$

From (5) and (6), we obtain the following approximate estimate of m

$$\hat{m}_h(t)=\frac{\sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i(y_i(u)-\alpha_i)E_{\mathbf{s}_i|\mathbf{y}_i}K(|x_i(u)-s_i(u)-t|/h)}{\sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i^2 E_{\mathbf{s}_i|\mathbf{y}_i}K(|x_i(u)-s_i(u)-t|/h)}.$$

Remark: Approximation (6) can be improved by noting

$$\begin{aligned} & \sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i E_{\mathbf{s}_i|\mathbf{y}_i}\left\{(y_i(u)-\alpha_i-\beta_i m^{(k)}(x_i(u)-s_i(u)))h_t(x_i(u)-s_i(u))\right\} \\ \approx & \sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i E_{\mathbf{s}_i|\mathbf{y}_i}\left\{(y_i(u)-\alpha_i-\beta_i m^{(k)}(t))K(|x_i(u)-s_i(u)-t|/h)\right\} \\ + & \sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i E_{\mathbf{s}_i|\mathbf{y}_i}\left\{\left[\beta_i m^{(k-1)}(t)-\beta_i m^{(k-1)}(x_i(u)-s_i(u))\right]K(|x_i(u)-s_i(u)-t|/h)\right\}. \end{aligned}$$

This results in the following iterative process for calculating the NPMLE:

$$\begin{aligned} \hat{m}^{(k)}(t) = & \frac{\sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i(y_i(u)-\alpha_i)E_{\mathbf{s}_i|\mathbf{y}_i}K\left(\frac{s_i(u)-(x_i(u)-t)}{h}\right)}{\sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i^2 E_{\mathbf{s}_i|\mathbf{y}_i}K\left(\frac{s_i(u)-(x_i(u)-t)}{h}\right)} \\ + & \frac{\sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i^2 E_{\mathbf{s}_i|\mathbf{y}_i}\left\{\left[\hat{m}^{(k-1)}(t)-\hat{m}^{(k-1)}(x_i(u)-s_i(u))\right]K\left(\frac{s_i(u)-(x_i(u)-t)}{h}\right)\right\}}{\sum_{i=1}^{n}\sum_{u=1}^{N}\beta_i^2 E_{\mathbf{s}_i|\mathbf{y}_i}K\left(\frac{s_i(u)-(x_i(u)-t)}{h}\right)}, \end{aligned}$$

which is what we have used in the algorithm. We found that with this iterative modification for estimating $m(\cdot)$, the fitting process tends to converge more quickly and the resulting estimates tend to be more accurate.