Peak-level vs Group-level Matching

Joe Wandy

February 17, 2015

1 Introduction

We pose the following question:

Q1. Is aligning peaks at feature-level better/worse than aligning at group-level?

We can try the following experiment to answer the question:

a. Take any two files. For each peak a in file 1, find out how many peaks in file 2 can potentially be matched within some mass & RT windows from peak a (count the # of candidate matches). Gradually increase the RT window and see what happens.

b. Repeat for group-level.

In short, approach (a) is to do alignment on feature-level, while approach (b) is operating on the group-level. The hypothesis is that on the group-level, we might see potentially fewer matching but are more precise. You can see the results for this experiment in the `check_potential_matching.ipynb` notebook.

1.1 Group-level matching

We need to come up with some scheme to measure how similar groups of peaks (clusters) are together in order to match them. Let's say from file 1 and 2, we take as input two matrices Z_1 and Z_2 produced as the results from clustering on the precursor mass/adduct relationships, so Z_1 and Z_2 are the two matrices of the probabilities of peaks to be in clusters. Set a threshold t on these matrices, then consider everything $>t$ to be in the same cluster. We use the probability product kernel [1] to measure the similarity between two clusters of mass spectra.

Consider two mass spectra clusters as mixture model, $p(x) = \frac{1}{k} \sum_k p(x|M_k, \sigma_1^2)$ and $p'(x) = \frac{1}{j} \sum_j p(x|M_j, \sigma_2^2)$, where $\sigma_1^2 = \sigma_2^2$ is some constant dependant on the MS instrument accuracy. Then, compute the probability product kernel for the two clusters as:

$$K(p, p') = \int p(x)p'(x)\, dx$$

$$\frac{1}{k}\frac{1}{j}\sum_k \sum_j \int p(x|M_k, \sigma^2)\, p(x|M_j, \sigma^2)\, dx$$

From matrix cookbook, the above product of two Normal distributions become:

$$\frac{1}{k}\frac{1}{j}\sum_k \sum_j c_c \int N_x(m_c, \Sigma_c)\, dx$$

where $\int N_x(m_c, \Sigma_c)\, dx$ integrates to 1, so we're left with:

$$\frac{1}{k}\frac{1}{j}\sum_k \sum_j c_c$$

where $c_c = \frac{1}{\sqrt{det(2\pi(\sigma_1^2 + \sigma_2^2)}} exp\left[-\frac{1}{2}(M_k - M_j)^T(\sigma_1^2 + \sigma_2^2)^{-1}(M_k - M_j)\right]$.

References

[1] Markus Heinonen, Huibin Shen, Nicola Zamboni, and Juho Rousu. Metabolite identification and molecular fingerprint prediction through machine learning. *Bioinformatics (Oxford, England)*, 28(18):2333–41, September 2012.