Tohle je uryvek z toho konference paperu:

The second step executes the retention time alignment. This algorithm is feasible when considering two or more experiments with the same conditions (flowrate, column dimensions, etc.). In that case, it calculates the average time of the peaks maxima and shifts all the data points to reach the lowest difference between the average maxima values and the actual maxima values of all corresponding components.

Optimization problem arises where is time of peak maximum for component *i* in experiment *j*. is average time of maxima of the peaks from all experiment for component *i*.

|  |  |  |
| --- | --- | --- |
|  |  | *equation 3* |
|  | s.t. | *equation 4* |

V principu to znamena, ze nase loss function je suma (přes experimenty v clusteru) sum (přes složky) rozdilu reálných casu maxim s průměrem (pro slozku ve všech experimentech v clusteru).

Když jsem nad tim ale premejslel, je pro nas dulezity abychom nehejbali se slozkama v experimentu vzlast. Když uy chceme posunout cas v experimentu, musíme posunout všechny slozky o stejnej Time shift.

Vznika optimalizacni problém:

1. Average time pro kaydou komponentu je konstanta – spocitana v prvním kroku
2. Decision variable je Time shift, pro kazdej experiment (posouva všechny slozky v experimentu) – vektor, pro který: number of elements = number of experiments in cluster
3. Loss function je definovana nahore
4. Constrains pro Time shift jednotlivých experimentu je, ze nenulova koncentrace nemůže byt posunuta do casu T < 0!
   1. Najdu pro experiment nejnizsi cas s nenulovou koncetraci
   2. Zaporna hodnota tohoto casu je lower boundary pro Time shift
   3. Upper boundary je infinity

Algoritmus v Ret\_Time\_Cor.py by tedy měl byt postaven takhle:

1. Zjistim a ulozim si casy maxim pro kazdou slozku v kazdym experiment v clusteru
2. Vypocitam average time pro kazdou slozku
3. Nadefinuju si Time shift vector s hodnotou 0 pro kazdej experiment (initial value)
4. Loss funkce nebude pocitat jen sumu sum jak je uvedeno vyse, ale bude mit argument Time shift (vektor s poctem elemntu stejnym jako pocet experiment) na zaklade kteryho posune vsechny maxima v experiment. Az po posunu vypocte celkovou loss value pro vsechny slozky a vsechny experiment (suma sum)
5. Vypoctu si lower boundaries pro kazdy element (experiment) v Time shift vektoru
6. Zavolam scipy.optimize.minimize
   1. Fun = lossfunction
   2. X0 – initial time shift
   3. Nepotrebujeme extra arguments
   4. method = „Nelder-Mead“

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html#scipy.optimize.minimize>