

Peak Alignment

MC 2.1.1 - Analytische Chemie im Master

Christoph Steinbeck, et al.

17/05/2018

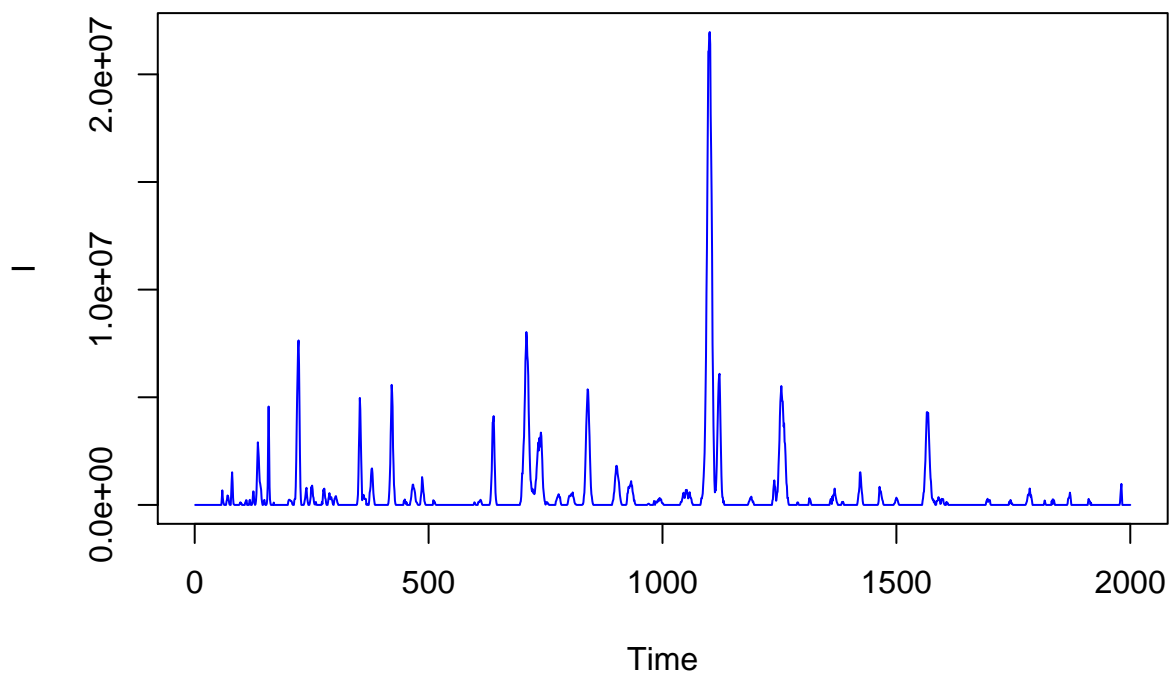
Präambel

Dies ist ein begleitendes Übungsblatt zur Vorlesung Analytische Chemie I (MC 2.1.1). Sie finden hier überwiegend den R-Code für die Beispiele, die in der Vorlesung besprochen wurden. Zu jedem Code-Beispiel, dass Sie in der Umgebung R-Studio ausführen können, finden Sie in der Regel eine oder mehrere Aufgaben. Bitte konzentrieren Sie sich nicht nur auf die Aufgaben, sondern studieren Sie die Code-Beispiele und versuchen Sie sie im Detail zu verstehen. Merken Sie sich wiederverwendbare Muster in den Lösungen und nutzen Sie die Zeit mit diesen Beispielen zum Experimentieren und Spielen.

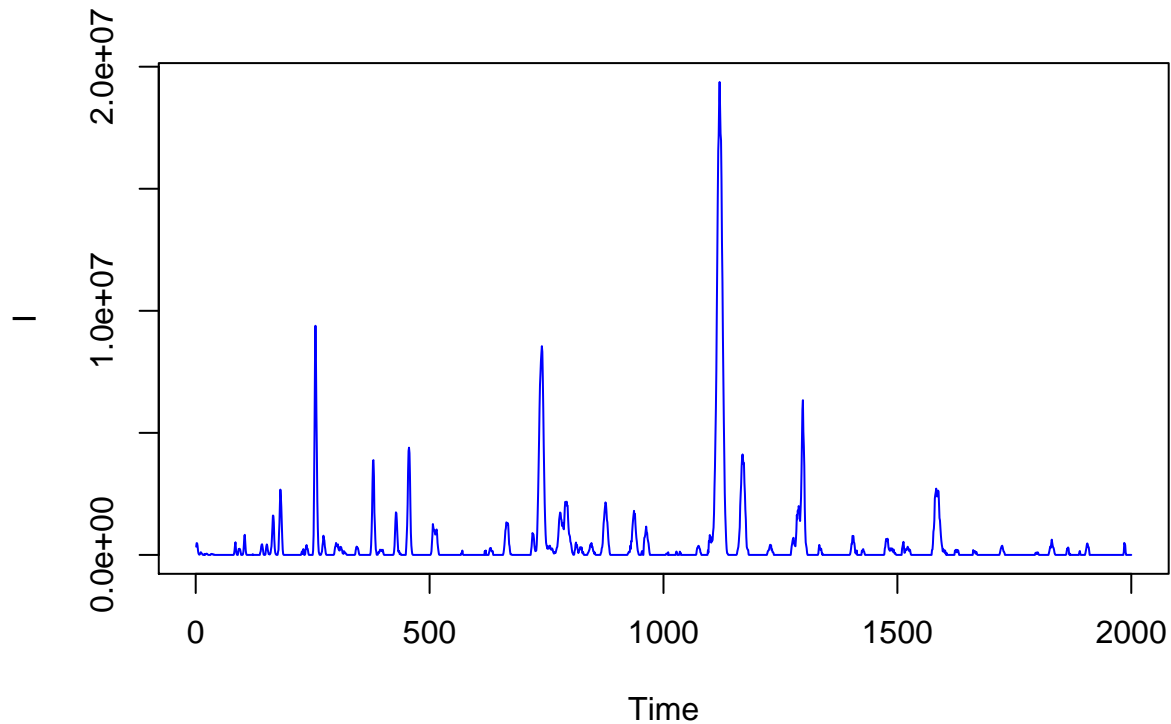
Peak Alignment

Wir laden und visualisieren zunächst unsere Beispieldaten - zwei LC/MS Chromatogramme aus dem lcms Paket

```
library(ptw)
data("lcms")
plot(lcms[1,,2], type = "l", xlab = "Time", ylab = "I", col="blue")
```



```
plot(lcms[1,,3], type = "l", xlab = "Time", ylab = "I", col="blue")
```



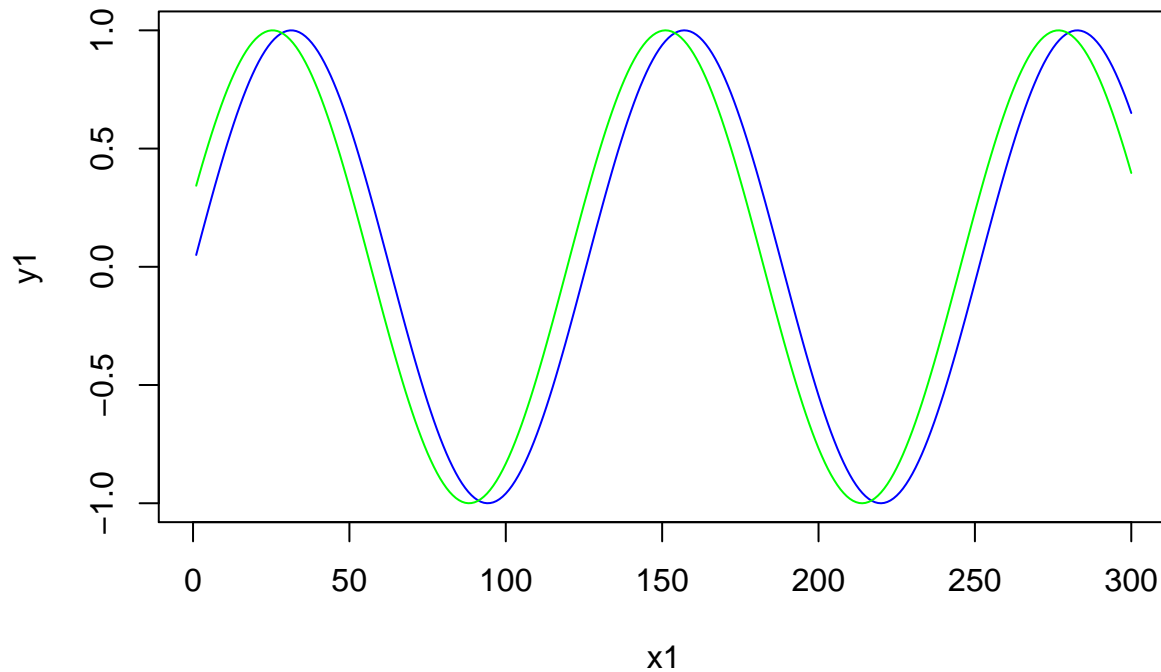
AUFGABE: Der obige Code erzeugt zwei getrennte Plots. Erzeugen Sie einen Plot mit beiden Chromatogrammen im gleichen Plot, jeden in einer eigenen Farbe Ihrer Wahl, und mit einem Y-Offset für eines der beiden Chromatogramme zur besseren Sichtbarkeit.

Spektrenähnlichkeit mit der Weighted Cross Correlation (WCC)

In diesem Übungsteil erzeugen wir zwei gegeneinander versetzte Sinus-Funktionen und berechnen jeweils den Wert der WCC

```
library(ptw)
x1 <- 1:300
y1 <- sin(x1/20)
x2 <- 1:300
y2 <- sin(x2/20 + 2)
resultRed = wcc(y1, y2, 1)
plot(x1, y1, type="l", col="blue")
y2 <- -sin(x2/20 + .3)
lines(x2, y2, type="l", col="green")
resultGreen = wcc(y1, y2, 1)
mytitle <- paste("Weighted Cross Correlation: ", resultGreen)
title(mytitle)
```

Weighted Cross Correlation: 0.958760478587113



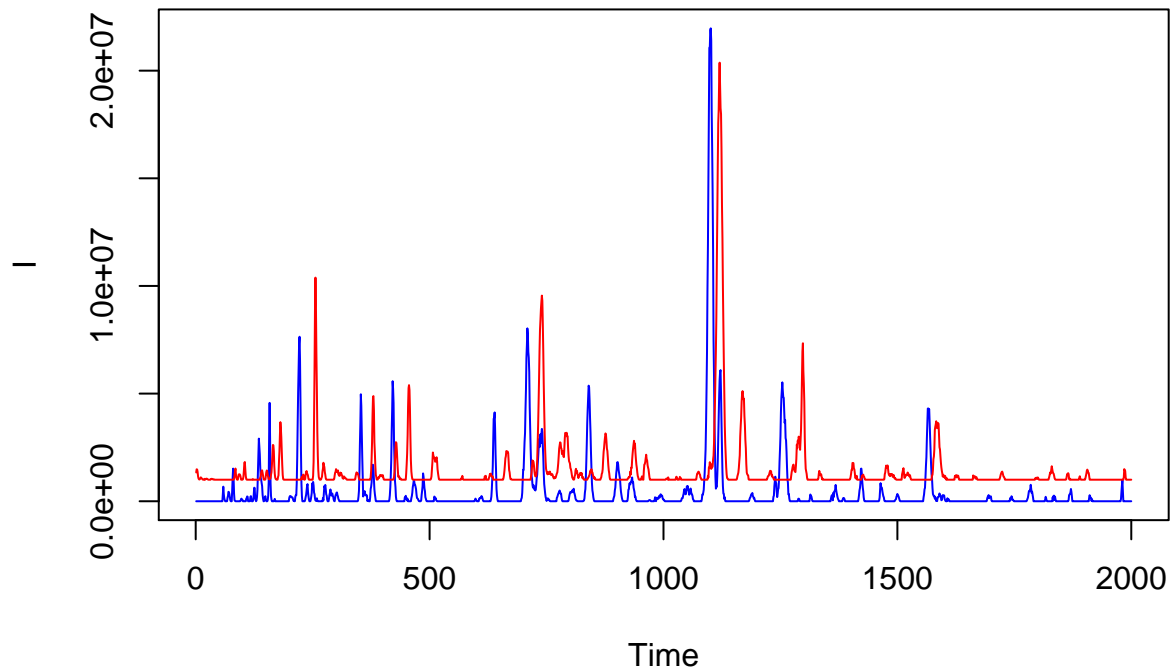
AUFGABE: a) Lesen Sie die Hilfe zur WCC und verstehen Sie die Parameter. b) Variieren Sie den Grad der Verschiebung der Sinusfunktionen zueinander und beobachten Sie den Effekt auf den Wert der WCC. c) Beachten Sie wie der Wert der WCC seinen Weg in die Überschrift des Graphen findet. d) finden Sie heraus, wie man den WCC-Wert als Legende in den Plot übernimmt

Peaks alinieren

Als erstes Beispiel für ein realistisches Peak Alignment schauen wir uns die ersten beiden LCMS chromatogramme aus dem Prostata-Datensatz an:

```
plot(lcms[1,,2], type = "l", main="Erste beiden Spektren aus LCMS-Datensatz", xlab = "Time", ylab = "I")
lines(lcms[1,,3] + 1e6, type = "l", col = "red")
```

Erste beiden Spektren aus LCMS-Datensatz



Wir verwenden die Weighted Cross Correlation als Maß für deren Übereinstimmung:

```
resultGreen = wcc(lcms[1,,2], lcms[1,,3], 1)
resultGreen
```

```
## [1] 0.2205827
```

AUFGABE: Merken Sie sich diesen Wert. Wir brauchen ihn später.

Peak Alignment mit Parametric Time Warping

Mit Hilfe der Funktion `ptw()` - Parametric Time Warping - führen wir nun als erste eine Alignierung der Chromatogramme aus.

AUFGABE: Lesen Sie die Hilfe zu `ptw` und den hier verwendeten Parametern.

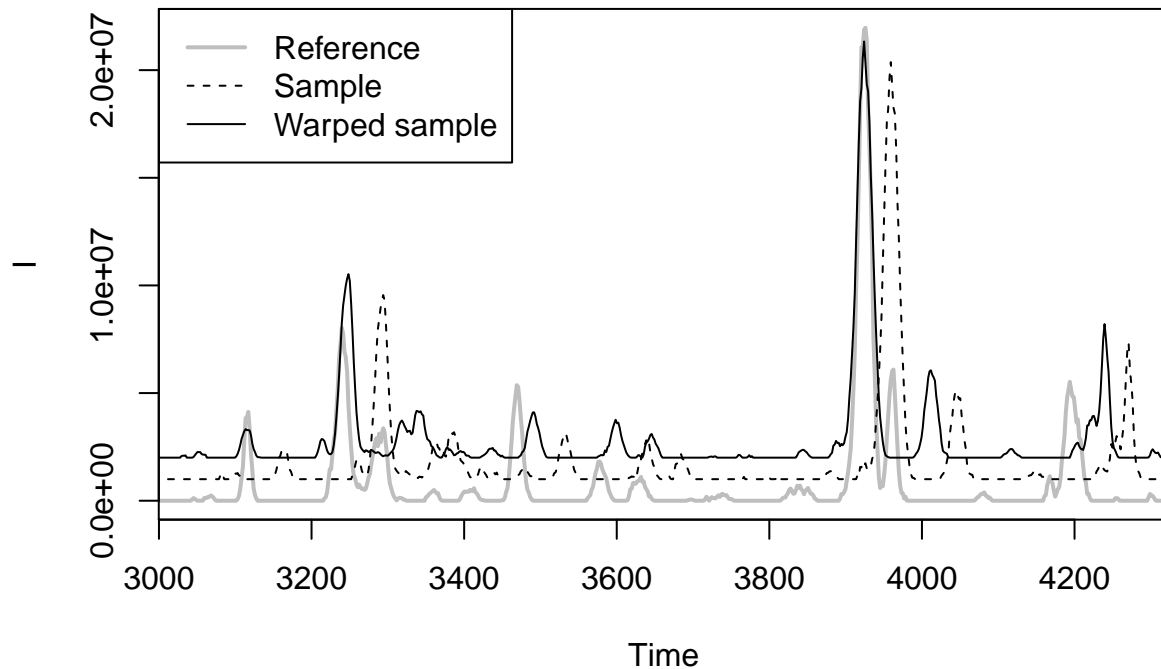
```
sref <- lcms[1,,2] # Zuweisung des ersten LC-MS-Spektrum an eine Variable (Spectrum Reference) zur leichteren Referenzierung
ssamp <- lcms[1,,3] # Dito, analog.
lcms.warp <- ptw(sref, ssamp, init.coef = c(0, 1, 0))
summary(lcms.warp)
```

```
## PTW object: individual alignment of 1 sample on 1 reference.
##
## Warping coefficients:
##      [,1]      [,2]      [,3]
## [1,] -43.04656 1.027225 -5.890607e-06
##
## Warping criterion: WCC
## Warping mode: forward
## Value: 0.08779711
```

AUFGABE: Studieren Sie die Zusammensetzung des PTW Objektes. Hinweise dazu finden sich ebenfalls in der Hilfe.

Nun plotten wir zum besseren Verständnis alles drei Spektren, d.h. die beiden Eingabespektren sref, ssamp sowie das gewarppte Spektrum in einem Graphen.

```
sref <- lcms[1,,2]
ssamp <- lcms[1,,3]
lcms.warp <- ptw(sref, ssamp, init.coef = c(0, 1, 0))
plot(time, sref, type = "l", lwd = 2, col = "gray", xlim = c(time[600], time[1300]), xlab = "Time", ylab = "I")
lines(time, ssamp + 1e6, lty = 2)
lines(time, lcms.warp$warped.sample + 2e6)
legend("topleft", lty = c(1,2,1), col = c("gray", 1, 1),
legend = c("Reference", "Sample", "Warped sample"), lwd = c(2, 1, 1))
```



AUFGABE: Beachten Sie die Parameter lty und lwd im Plot-Commando (Siehe Hilfe) und merken Sie sich diese Optionen für eventuelle spätere eigene Plotaufgaben.

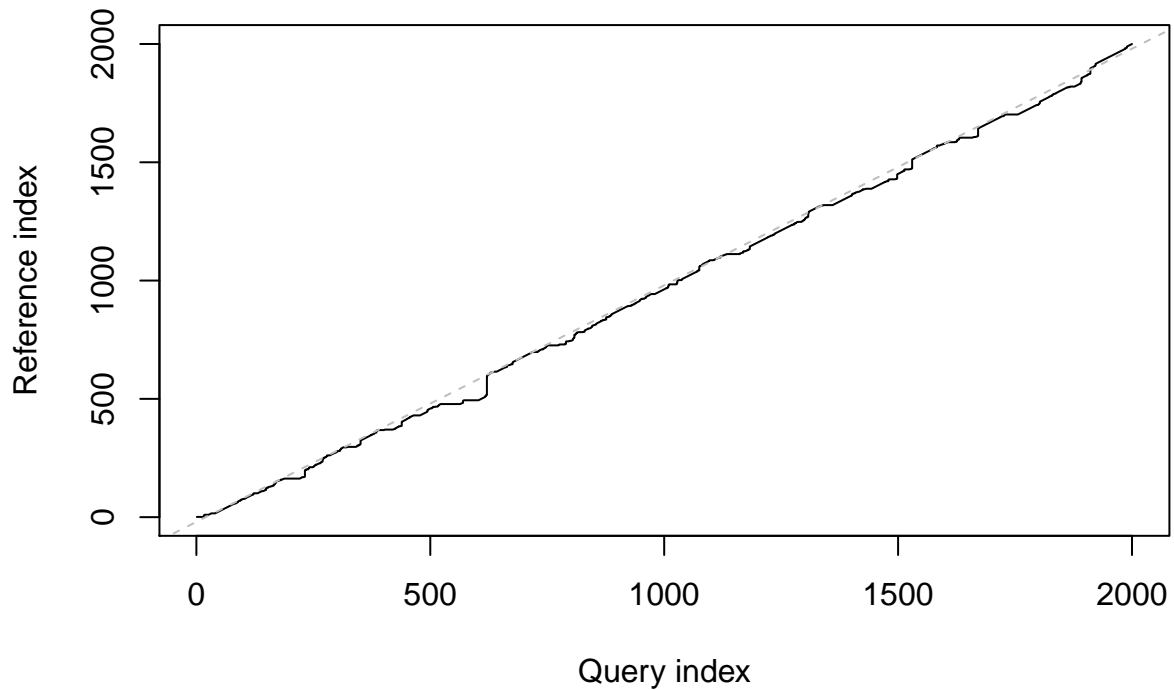
Peak Alignment mit Dynamic Time Warping

Zu guter letzt probieren wir noch die Methode des Dynamic Time Warping aus.

```
library(dtw)

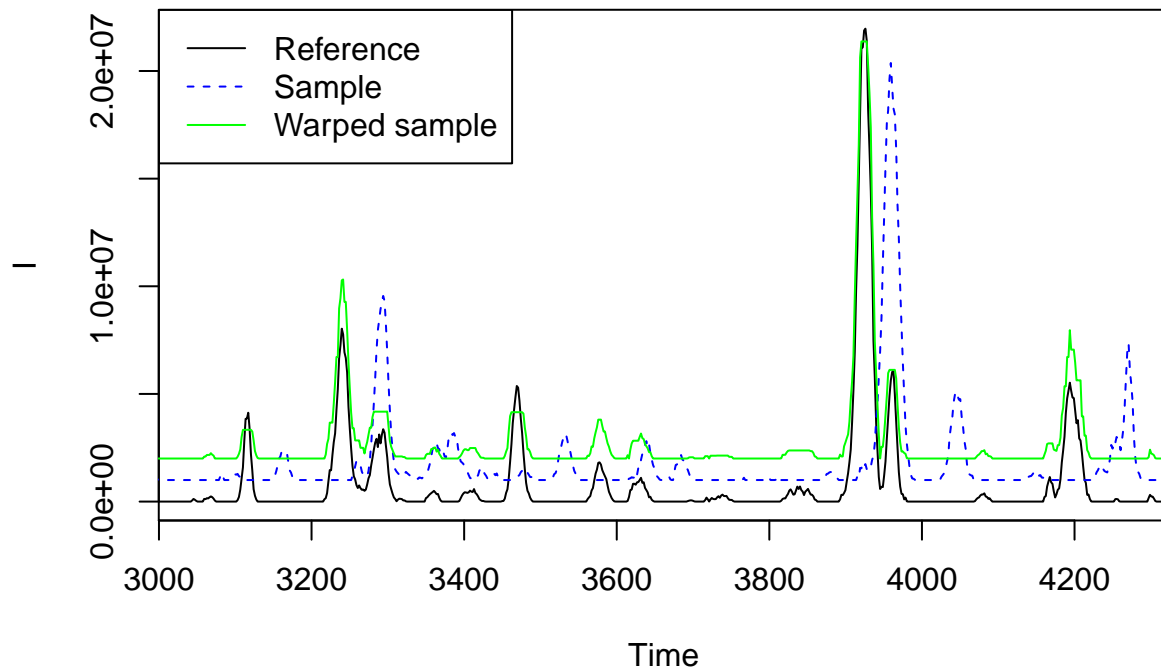
## Loading required package: proxy
##
## Attaching package: 'proxy'
##
## The following objects are masked from 'package:stats':
##
##   as.dist, dist
##
## The following object is masked from 'package:base':
##
```

```
##      as.matrix
## Loaded dtw v1.20-1. See ?dtw for help, citation("dtw") for use in publication.
warpfun <- dtw(ssamp, sref)
plot(warpfun)
abline(-20, 1, col = "gray", lty = 2)
```



AUFGABE: Vergegenwärtigen Sie sich die Bedeutung des obigen Graphen.

```
library(dtw)
wx2 <- warp(warpfun)
plot(time, lcms[1,,2], type = "l", xlab = "Time", ylab = "I", xlim = c(time[600], time[1300]))
lines(time, lcms[1,,3] + 1e6, lty = 2, col = "blue")
lines(time, lcms[1, wx2, 3] + 2e6, col = "green")
legend("topleft", col = c("black", "blue", "green"), lty = c(1,2,1),
legend = c("Reference", "Sample", "Warped sample"))
```



AUFGABE: Berechnen Sie den Wert der Weighted Crosscorrelation nach der Alinierung. Vergleichen Sie den Erfolg mit der ursprünglichen Situation.