Drug Safety

Timing Matters: A Machine Learning Method for the Prioritization of Drug-Drug Interactions through Signal Detection in the FDA Adverse Event Reporting System and Their Relationship with Time of Co-exposure

Vera Battini[1,2]*, Marianna Cocco[1], Maria Antonietta Barbieri[3], Greg Powell[4], Carla Carnovale[2], Emilio Clementi[2,5], Andrew Bate[6,7], and Maurizio Sessa[1]

[1] Department of Drug Design and Pharmacology, University of Copenhagen, Copenhagen, Denmark

[2] Pharmacovigilance & Clinical Research, International Centre for Pesticides and Health Risk Prevention, Department of Biomedical and Clinical Sciences (DIBIC), ASST Fatebenefratelli-Sacco University Hospital, Università degli Studi di Milano, Milan, Italy.

[3] Department of Clinical and Experimental Medicine, University of Messina, Messina, Italy.

[4] Safety Innovation and Analytics, GSK, Durham, NC, USA.

[5] Scientific Institute, IRCCS E. Medea, Bosisio Parini, LC, Italy.

[6] GSK, London, UK.

[7] London School of Hygiene and Tropical Medicine, University of London, London, UK.

Corresponding author:

Vera Battini, MPharm, Cand. PhD

Department of Drug Design and Pharmacology,

University of Copenhagen,

Jagtvej 160, Copenhagen 2100, Capital Region, Denmark

vera.battini@unimi.it

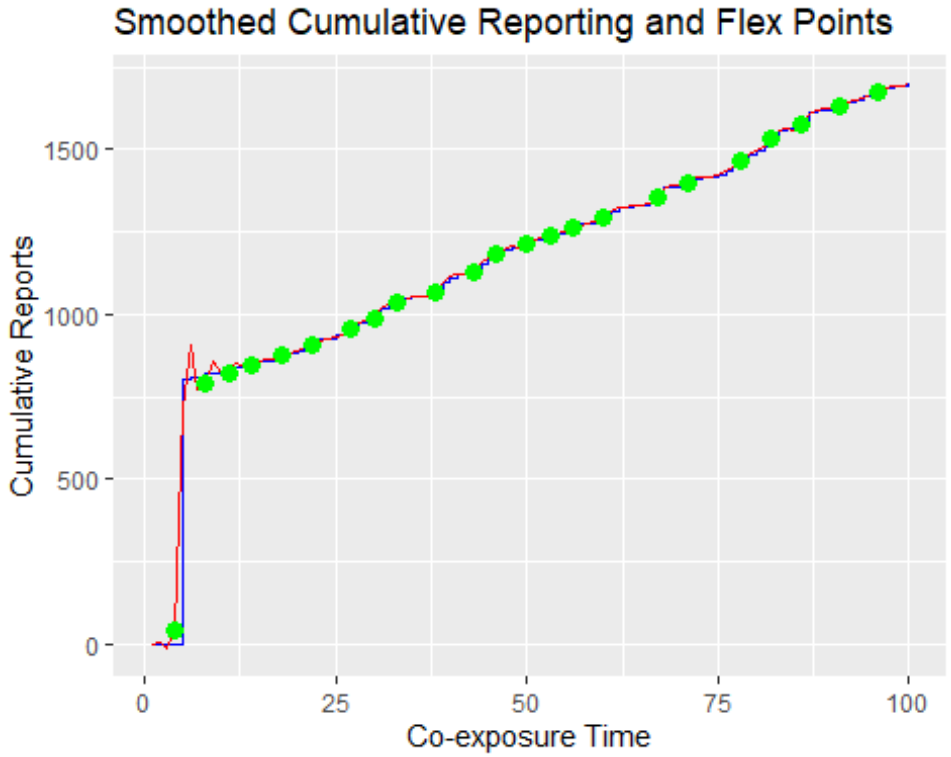
Electronic Supplementary Material 1

Developer: Sessa Maurizio

2024-03-13

```
# Load necessary libraries
library(ggplot2)

# Set seed for reproducibility
set.seed(123)

# Generate simulated data (cumulative exponential function with log2(x) rate
and a single big flex point)
time <- 1:100 # Time period

# Generate reports based on a progressively increasing rate
reports <- sapply(time, function(t) rexp(1, rate = log2(t + 1) / 1)) # Slower
increase rate based on log2(x)/1

# Introduce the single large increase at the desired flex point (time = 5)
big_increase <- 800 # Size of the big increase
reports[5] <- reports[5] + big_increase

# Generate smaller rate of increase for the remaining days
reports[6:100] <- rep(rexp(94, rate = 0.1), each = 1)

## Warning in reports[6:100] <- rep(rexp(94, rate = 0.1), each = 1): number o
f
## items to replace is not a multiple of replacement length

# Cumulative reports
cum_reports <- cumsum(reports)

# Smooth the cumulative distribution function
smoothed_cdf <- smooth.spline(time, cum_reports)

# Calculate the derivative of the smoothed cumulative distribution function
smoothed_cdf_derivative <- diff(smoothed_cdf$y) / diff(smoothed_cdf$x)

# Find local maxima in the derivative (indicating flex points)
flex_points <- which(diff(sign(diff(smoothed_cdf_derivative))) == -2) + 1

# Plotting the cumulative distribution function with smoothing and flex point
cdf_plot <- ggplot(data.frame(time, cum_reports)) +
  geom_step(aes(x = time, y = cum_reports), direction = "hv", color = "blue")
+
  geom_line(aes(x = smoothed_cdf$x, y = smoothed_cdf$y), color = "red") +
  geom_point(data = data.frame(time = smoothed_cdf$x[flex_points], cum_report
```

```
s = smoothed_cdf$y[flex_points]), aes(x = time, y = cum_reports), color = "gr
een", size = 3) +
  labs(x = "Co-exposure Time", y = "Cumulative Reports", title = "Smoothed Cu
mulative Reporting and Flex Points")

print(cdf_plot)
```




```
# Display the first flex point
if(length(flex_points) > 0) {
  print(paste("Flex point with the first increase in cumulative reports at ti
me:", smoothed_cdf$x[flex_points[1]]))
} else {
  print("No flex points found.")
}
```

```
## [1] "Flex point with the first increase in cumulative reports at time: 4"
```