User Guide

Application:

Robust K-PD model for activated clotting time prediction

and UFH dosing individualization during cardiopulmonary bypass

The proposed UFH Bayesian dosing individualization strategy was implemented in a R function. Based on available ACT measurement, patient's bodyweight and dosing history, this code produces a graphic with an estimation of the individualized dosing regimen and the corresponding ACT prediction with 95% confidence and prediction intervals, either with a continuous or intermittent UFH dosing strategy.

How to use the code in 6 steps:

1) Load the .R file provided in the supplementary material of the article in R or RStudio.

Notice that the code is divided in 3 sections; A, B and C:

Robust_ACT_Script.R

Source on Save   Run   Source

```r
#### A - Individual Patient Data ####

# 1 - Recommended ACT target during surgery (seconds) :
ACT_Target <- 400
# 2 - Values of measured ACTs of the patient (seconds) :
ACT_Values_Measured <- c(110, 445, 402, 502, 307, 447)
# 3 - Times of ACT measurements (minutes) :
ACT_Measurement_Times <- c(0.0, 5.4, 27.0, 56.4, 85.2, 113.4)
# 4 - Patient weight (kg) :
Patient_Weight <- 85
# 5 - Heparin boluses given (total UI per shot) :
Bolus_Given <- c(25000, 10000, 5000)
# 6 - Times where the heparin boluses were given (minutes) :
Bolus_Times <- c(6.0, 31.8, 96.0)
# 7 - Infusion rates of heparin infusions given to the patients (UI per hour) :
Infusion_Rates_Given <- c(0)
# 8 - Start times of heparin infusions (minutes) :
Infusion_Times <- c(0)
# 9 - Duration of given heparin infusions (minutes) :
Infusion_Durations <- c(0)

#### B - Launch Function ####

# Launch continuous UFH strategy function:
Robust_ACT_Continuous(ACT_Target, ACT_Values_Measured, ACT_Measurement_Times, Patient_Weight,
                      Bolus_Given, Bolus_Times, Infusion_Rates_Given, Infusion_Times,
                      Infusion_Durations)

# Launch intermittent UFH strategy function:
Robust_ACT_Intermittent(ACT_Target, ACT_Values_Measured, ACT_Measurement_Times, Patient_Weight,
                        Bolus_Given, Bolus_Times, Infusion_Rates_Given, Infusion_Times,
                        Infusion_Durations)

#### C - Function creation ####

# Continuous UFH strategy function creation:
Robust_ACT_Continuous <- function(ACT_Target, ACT_Values_Measured, ACT_Measurement_Times,
                                  Patient_Weight, Bolus_Given, Bolus_Times,
                                  Infusion_Rates_Given, Infusion_Times,
                                  Infusion_Durations) {...}

# Intermittent UFH strategy function creation:
Robust_ACT_Intermittent <- function(ACT_Target, ACT_Values_Measured, ACT_Measurement_Times,
                                    Patient_Weight, Bolus_Given, Bolus_Times,
                                    Infusion_Rates_Given, Infusion_Times,
```

2) Run the entire C section of the code ("**C – Function creation**"). This will create and store the functions that we will use later.

3) Fill the 9 parts of the A section of the code ("**A - Individual Patient Data**") with the patient's individual data; patient body weight, ACT measurements, unfractionated heparin (UFH) boluses and infusion already administered, as follows:

```
#### A - Individual Patient Data ####
```

1 - Enter the target ACT used during the surgery, in seconds, as in the example below:

```
# 1 - Recommended ACT target during surgery (seconds):
ACT_Target <- 400
```

2 - Enter the ACT values (in seconds) that were measured since the beginning of the surgery, including the ACT at time 0, separated by comas, as in the example below:

```
# 2 - Values of measured ACTs of the patient (seconds):
ACT_Values_Measured <- c(110, 445, 402, 502, 307, 447)
```

3 - Enter the times (in minutes) when the ACTs above were measured, starting with t= 0 for the first ACT measurement.

There must be exactly as many "ACT_Measurment_Times" as there are "ACT_Values_Measured".

In this example, the second ACT was measured 5.4 minutes after the first one, the third ACT was measured 27 minutes after the first ACT (after t= 0) etc...

```
# 3 - Times of ACT measurements (minutes):
ACT_Measurment_Times <- c(0.0, 5.4, 27.0, 56.4, 85.2, 113.4)
```

4 - Enter the patient body weight in kg, as in the example below:

```
# 4 - Patient weight (kg):
Patient_Weight <- 85
```

5 - Enter the different amounts of UFH already administered by intravenous boluses (in total UI per shot), separated by comas, as in the example below:

```
# 5 - Heparin boluses given (total UI per shot):
Bolus_Given <- c(25000, 10000, 5000)
```

6 - Enter the times when UFH intravenous boluses mentioned above were administered, in minutes since t= 0. There must be exactly as many "`Bolus_Times`" as there are "`Bolus_Given`".

In the example below, the first bolus was administered 6 minutes after t=0, in other words 6 minutes after the measurement of the first ACT:

```
# 6 - Times where the heparin boluses were given (minutes):
Bolus_Times <- c(6.0, 31.8, 96.0)
```

7 - Enter the different infusion rates of UFH already administered by infusion (in UI per hour):

Enter 0 if no infusions were administered.

```
# 7 - Infusion rates of heparin infusions given to the patients (UI per hour):
Infusion_Rates_Given <- c(0)
```

8 - Enter the starting times of the infusions above, in minutes since t= 0:

There must be as many "`Infusion_Times`" as there are "`Infusion_Rates_Given`":

```
# 8 - Start times of heparin infusions (minutes):
Infusion_Times <- c(0)
```

9 - Enter the duration (in minutes) of the infusions above:

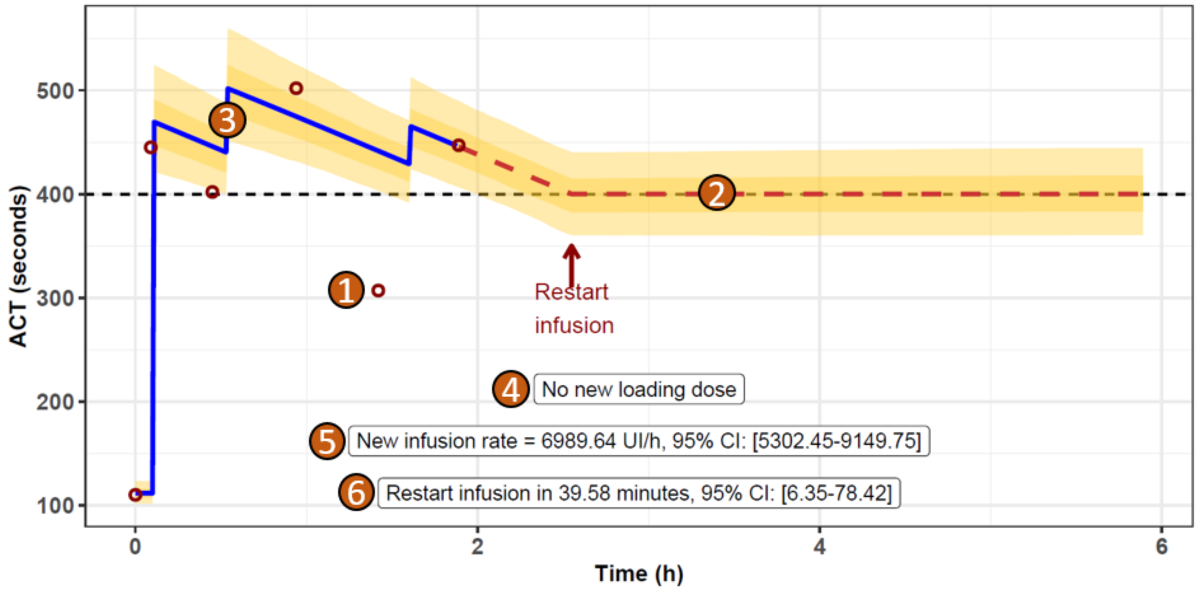
There must be as many "`Infusion_Durations`" as there are "`Infusion_Rates_Given`".

```
# 9 - Duration of given heparin infusions (minutes):
Infusion_Durations <- c(0)
```

4) After filling those 9 parts, run the entire A section of the code. The patient's data are now saved in the memory.

5) Finally, in the B part of the code ("**B - Launch Function**"), using the patient's data we filled and stored in 3) and 4), run the desired function we stored in 2): the function "`Robust_ACT_Continuous()`" to use the continuous HNF dosing strategy, or the function "`Robust_ACT_Intermittent()`" to use the intermittent HNF dosing strategy.

6.1) With the "`Robust_ACT_Continuous()`" function, the output will be as follow:



This graphic output displays:

- 1: The ACTs already measured punctually during the surgery (red circles).

- 2: The ACTs values predicted by the robust K-PD model (dashed red line) with the 95% prediction interval (yellow ribbon).

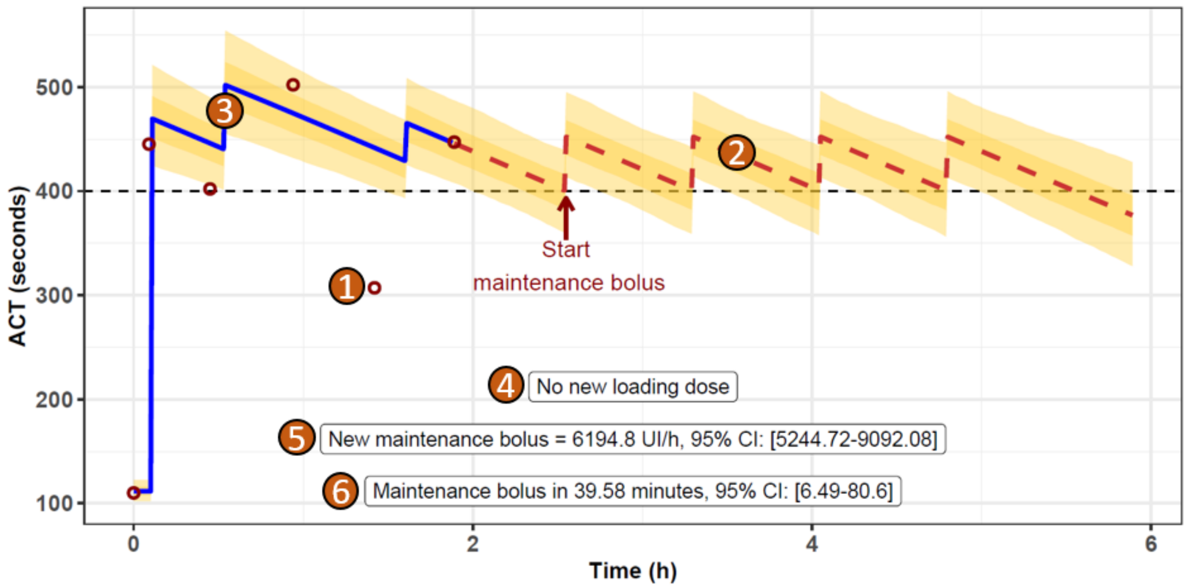
- 3: The fitted ACTs values (plain blue line) with the 95% prediction interval (yellow ribbon).

- 4: The value of the next UFH loading dose to be administered (in UI) after the last ACT measurement (in the example, the last ACT value is above the ACT target, therefore no new loading dose is required).

- 5: The UFH infusion rate of the next infusion to administer, in UI per hour.

- 6: The time to wait since the last ACT measurement to the administration of the next UFH infusion (in the example, the last ACT value is above the ACT target, therefore the new infusion should be administered in 40 minutes)

6.2) With the "`Robust_ACT_Intermittent()`" function, the output will be as follow:



This graphic output displays:

- 1: The ACTs already measured punctually during the surgery (red circles).

- 2: The ACTs values predicted by the robust K-PD model (dashed red line) with the 95% prediction interval (yellow ribbon).

- 3: The fitted ACTs values (plain blue line) with the 95% prediction interval (yellow ribbon).

- 4: The value of the next UFH loading dose to be administered (in UI) after the last ACT measurement (in the example, the last ACT value is above the ACT target, therefore no new loading dose is required).

- 5: The new UFH maintenance bolus to administer, in UI per hour.

- 6: The time to wait since the last ACT measurement to the administration of the next UFH maintenance bolus (in the example, the last ACT value is above the ACT target, therefore the new UFH maintenance bolus should be administered in 40 minutes)