mNIRS Kinetics Data Processing & Curve Fitting

What do I need to know from the data?

* Data channel (name)
  + Must be numeric
* Sample rate
  + Sample to time conversion
* Missing data (NAs)
  + Interpolate? Repeat last?
* Outlier detection?
  + Remove zeros?
  + Hampel filter
    - Window length?
    - Threshold criteria (±2-4 SD?)
* What kind of data filtering?
  + Butterworth low pass
    - Order & cutoff frequency?
  + Moving average
    - Window length?
  + Smoothing spline?
    - Parameter (spar) ?
  + GAM?
* Ensemble average multiple events?

Curve Fitting

* Where to start and stop looking for relevant kinetics?
  + Manual
  + Including periods preceding & following phase-change
    - Baseline & end values
  + Multiple windows return multiple fits
* Range of data to display
  + Manual
  + Display window wider than fit window
* Where does the external phase-change occur
  + Manual
  + Step function; start/end of workload; contraction stimulus; etc.
  + As precise as possible
* How is the preceding phase modelled?
  + Baseline mean?
  + Linear regression?
  + Independent or within nonlinear fit?
* Where to stop modelling?
  + Local peak/nadir end values
    - Manual
    - No subsequent peaks within what time range (10/15/30 sec?)
  + Pseudo-plateau in monotonic function
    - e.g. levelling off of primary deoxy kinetics
* Models to fit?
  + Monoexponential / asymptotic / negative exponential
  + Sigmoidal / Logistic / Gompertz, Weibull
  + Power curve?
  + Logarithmic?
  + Non-parametric
    - Half-response time (HRT)
    - 63.2% response time
    - Slope: mean; peak
    - Time to peak slope
* Fit acceptance criteria?
  + R^2 >0.85
  + AIC, BIC
  + Plausible time values (0 to time of peak value)
* Confidence limits (precision)
  + Bootstrapping

Functional layout

* Global manual inputs:
  + NIRS label
* read\_data()
  + file\_path
    - Local folder location
    - How to manually input on Shiny?
    - Detect data type (xlsx, csv, etc.)
  + Device = c(“moxy”, “train.red”, “artinis”, “vo2\_master”, “custom”),
  + nirs\_column
    - column name
    - must exist; be numeric
  + time\_column = NULL
    - optional; be numeric or hms; be sequential
    - if doesn’t exist, create sample (idx) column
  + event\_column = NULL
    - optional
  + Output: raw\_data
* process\_data()
  + raw\_data

…

Steps

1. Read raw file
2. Find all channels
3. Extract table
4. Create index column
5. Rename target columns
6. .keep\_all other data columns
7. Removed fixed values column-wise
8. Removed outliers globally
9. Handle missing data column-wise
   * Linear interpolation
   * Omission? (for partial data?)
10. Specify sample\_rate (globally)
11. Re-sample data (globally)
12. Filter column-wise (column-wise parameters)
    * none
    * Low-pass
    * Moving-average
    * Smooth-spline
13. Shift range positive globally
14. Normalise range
    * None
    * Column-wise
    * Globally
    * Grouped
15. OUTPUT -> processed\_data
    * Step-by-step interactive visualisation? (plotly?)
    * To see effects from each processing step
16. Identify kinetics events
17. Prepare list of kinetics dataframes
    * Distinct for each kinetics event
    * Globally
    * Grouped
18. Specify kinetics fitting windows
    * Optionally re-scale baseline means to zero?
    * Event-wise
19. Specify kinetics models to fit
    * Event-wise
    * Globally
20. Fit kinetics models column-wise per data list
    * Monoexp (4 param)
    * Biexp? (how to interpret params?)
    * Logistic (4/5 param)
    * Gompertz
    * HRT/HDT (recovery/reoxygenation & deoxygenation)
    * Peak slope
      + Specify samples
21. Add to each data list
    * Fitted values (column-wise)
22. Add to metadata event-wise
    * Model
23. OUTPUT -> kinetics\_table
    * Dataframe of event-wise column-wise kinetics parameters
    * Fit comparisons
      + AIC, BIC, R^2, RMSE, MAE, MAPE, …
24. OUTPUT -> display kinetics plots
    * Interactive plotly?