R for experimentalists:

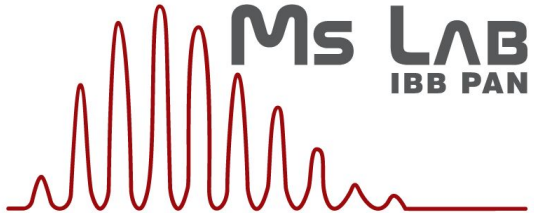
HDX-MS example

Weronika Puchała

WhyR? Warsaw, September 2019

About me:

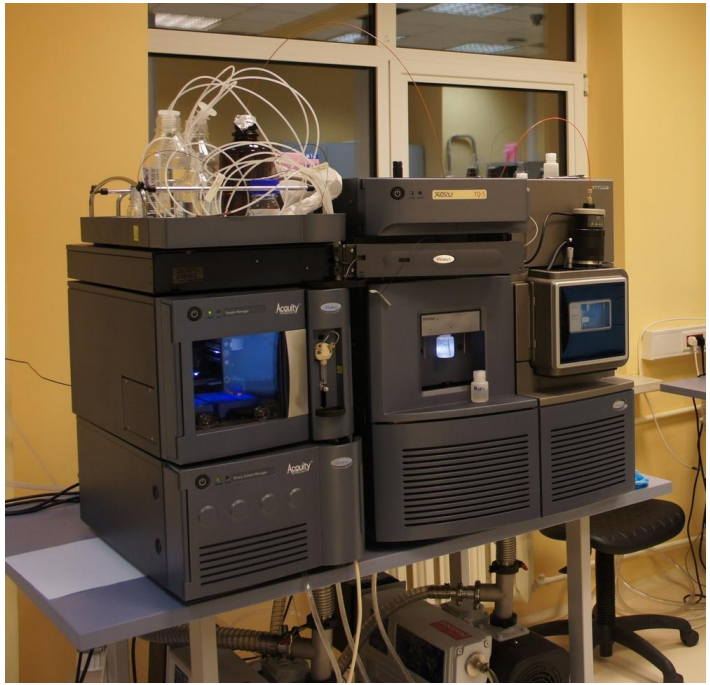
- MSc in Physics
- commercial background
- currently: PhD in Biophysics in Mass Spectrometry Lab
- Warsaw based



contact: puchala.weronika@gmail.com

Presentation overview:

- Hello
- Biology 101
- The purpose
- The app
- The obstacles
- A little bit of code
- The end



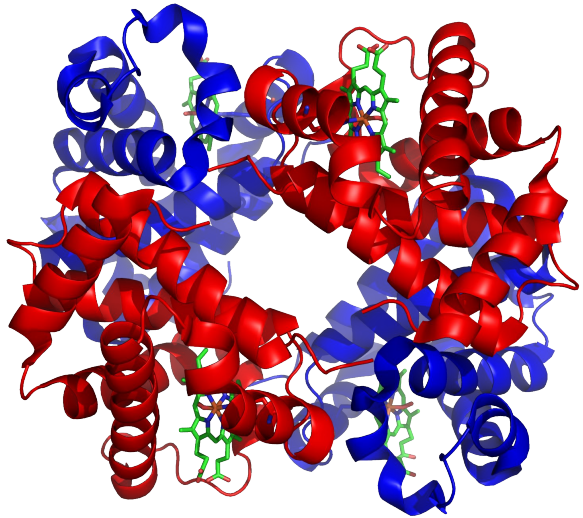
The motivation

- the natural sciences are mostly not yet automated
- experimentalists have no interest in learning computer skills - they’re interested in making the experiments [sic!]
- minimize time spent on data processing for commercial analysis
- some methods too new to be well-described

World of proteins

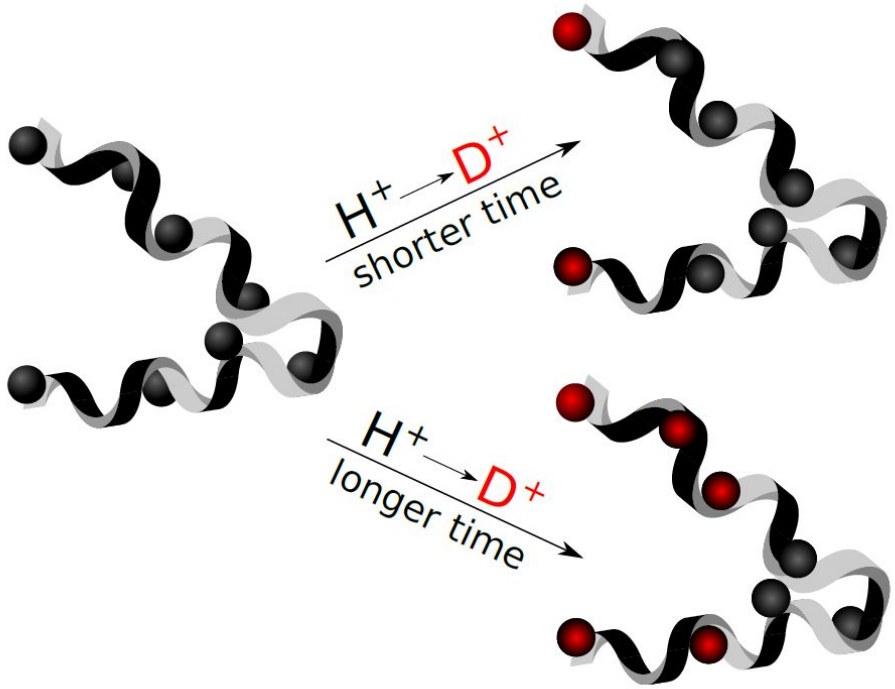
Structure:

- primary: chain
- secondary: α elix / β pleated sheet
- tertiary: 3D
- quaternary: relation between subunits



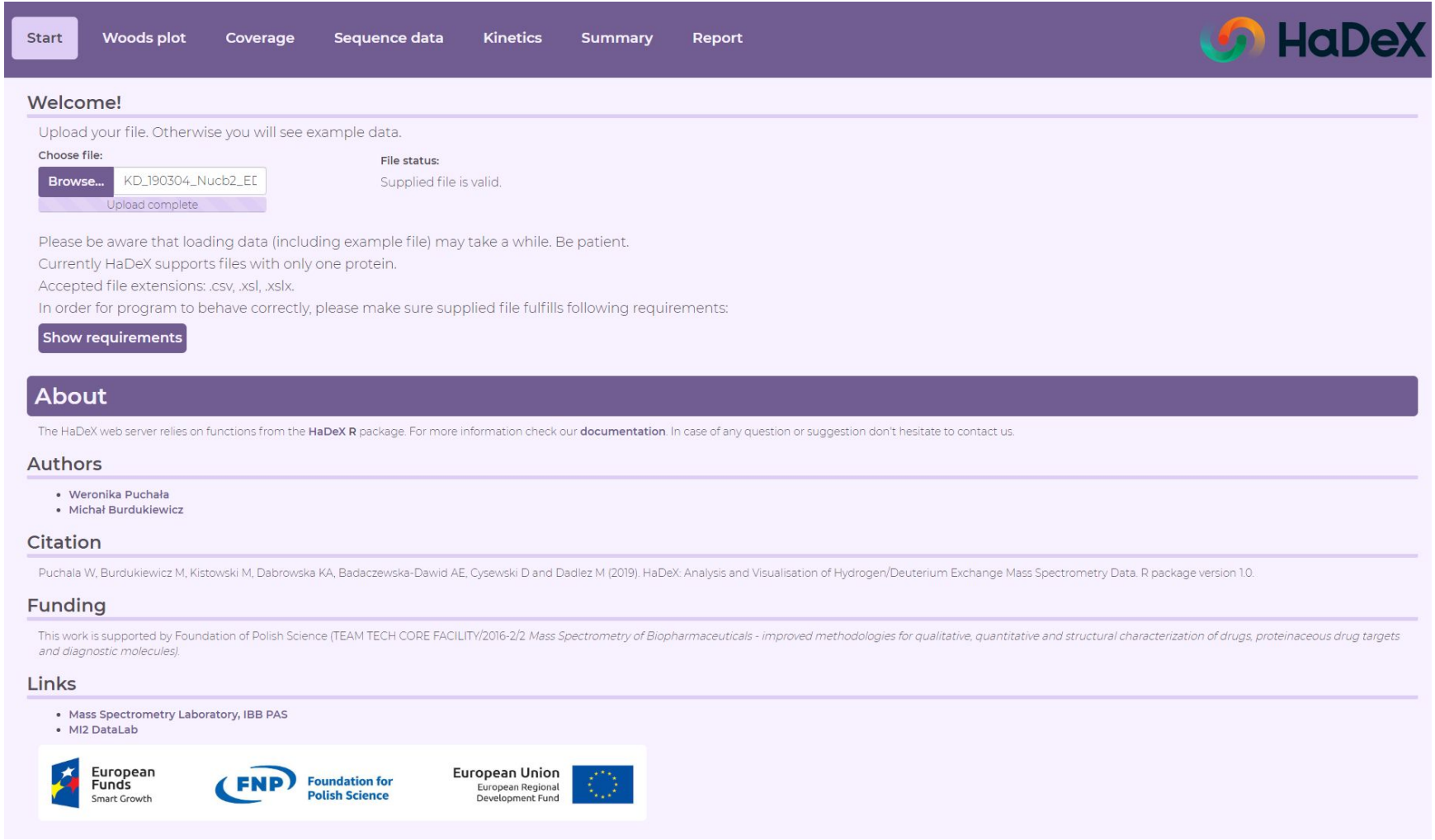
source:https://en.wikipedia.org/wiki/Hemoglobin

HDX-MS



How HaDeX answers users' needs?

- three forms: package, web-server, standalone software for sensitive data
- point-and-click application
- results available immediately
- every step of the calculations is downloadable
- publication quality plots
- easy creation of the report of the results
- all the methodology described in vignette


Start
Woods plot
Coverage
Sequence data
Kinetics
Summary
Report
HaDeX
Welcome!
Upload your file. Otherwise you will see example data.
Choose file:
Browse...
KD_190304_Nucb2_EE
Upload complete
File status:
Supplied file is valid.
Please be aware that loading data (including example file) may take a while. Be patient.
Currently HaDeX supports files with only one protein.
Accepted file extensions: .csv, .xsl, .xslx.
In order for program to behave correctly, please make sure supplied file fulfills following requirements:
Show requirements
About
The HaDeX web server relies on functions from the HaDeX R package. For more information check our documentation. In case of any question or suggestion don't hesitate to contact us.
Authors
Weronika Puchała
Michał Burdukiewicz
Citation
Puchala W, Burdukiewicz M, Kistowski M, Dabrowska KA, Badaczewska-Dawid AE, Cysewski D and Dadlez M (2019). HaDeX: Analysis and Visualisation of Hydrogen/Deuterium Exchange Mass Spectrometry Data. R package version 1.0.
Funding
This work is supported by Foundation of Polish Science (TEAM TECH CORE FACILITY/2016-2/2 Mass Spectrometry of Biopharmaceuticals - improved methodologies for qualitative, quantitative and structural characterization of drugs, proteinaceous drug targets and diagnostic molecules).
Links
Mass Spectrometry Laboratory, IBB PAS
MI2 DataLab
European Funds
Smart Growth
FNP
Foundation for Polish Science
European Union
European Regional Development Fund

Data structure

multidimensional & complex

```
> dat <- read_hdx(system.file(package = "HaDeX", "HaDeX/data/KD_180110_CD160_HVEM.csv"))
> str(dat)
'data.frame':   4069 obs. of  15 variables:
 $ Protein     : chr  "db_CD160" "db_CD160" "db_CD160" "db_CD160" ...
 $ Start       : int  1 1 1 1 1 1 1 1 1 1 ...
 $ End         : int  15 15 15 15 15 15 15 15 15 15 ...
 $ Sequence    : chr  "INITSSASQEGTRLN" "INITSSASQEGTRLN" "INITSSASQEGTRLN" "INITSSASQEGTRLN" ...
 $ Modification: logi  NA NA NA NA NA NA ...
 $ Fragment    : logi  NA NA NA NA NA NA ...
 $ MaxUptake   : num  14 14 14 14 14 14 14 14 14 14 ...
 $ MHP         : num  1591 1591 1591 1591 1591 ...
 $ State       : chr  "CD160" "CD160" "CD160" "CD160" ...
 $ Exposure    : num  0 0 0 0.001 0.001 0.167 0.167 0.167 0.167 1 ...
 $ File        : chr  "KD_160527_CD160_sekw_05" "KD_160527_CD160_sekw_05" "KD_160527_CD160_sekw_05" "KD_160527_CD160_IN_01" ...
 $ z           : int  1 2 3 2 3 2 2 2 2 2 ...
 $ RT          : num  3.23 3.24 3.24 3.26 3.26 ...
 $ Inten       : num  6592 394066 173526 232221 110675 ...
 $ Center      : num  1591 796 531 796 531 ...
>
```

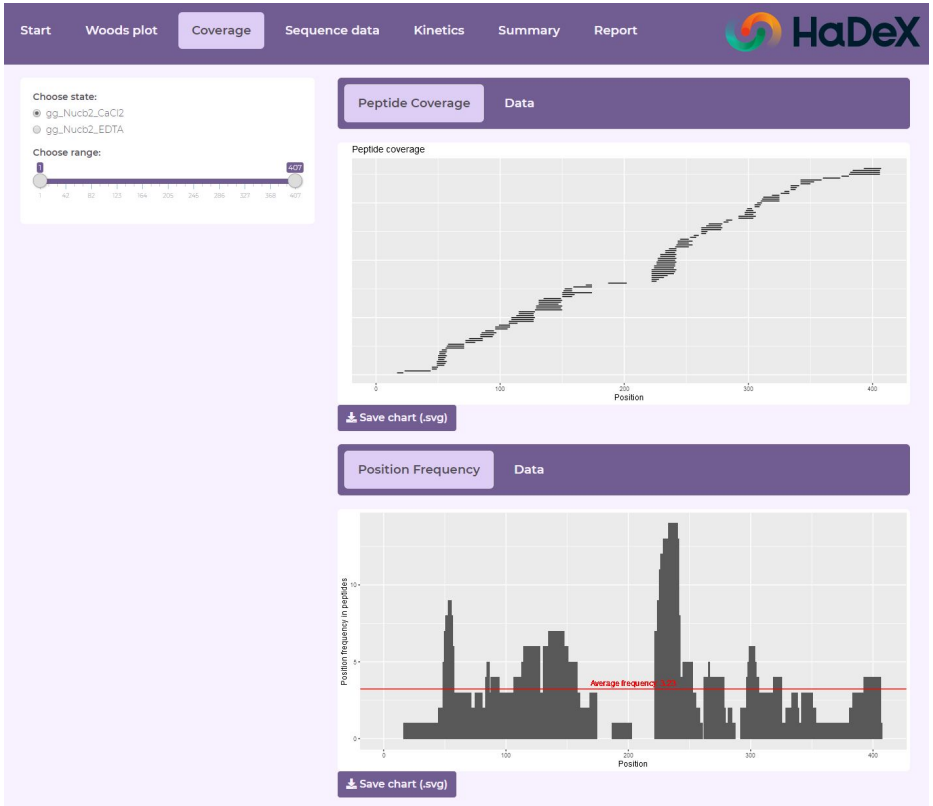
experimental data!

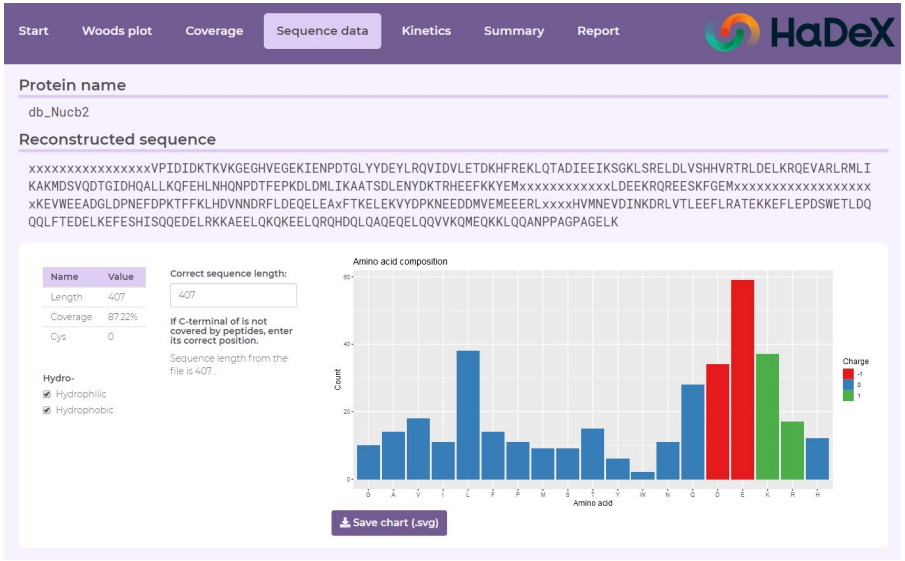
File validation

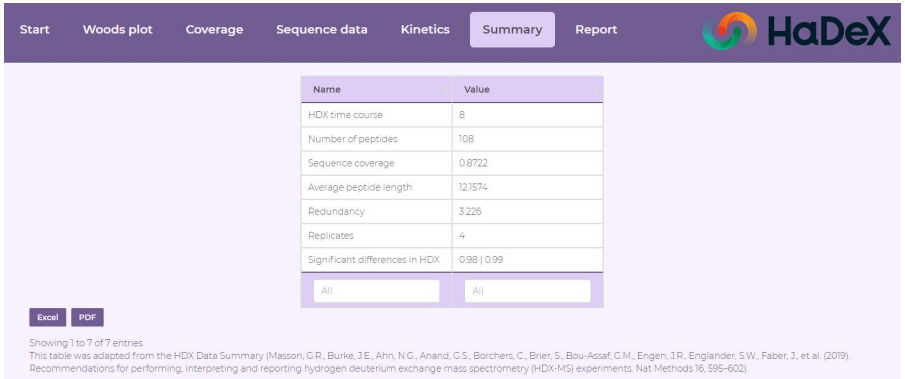
```r
#check for dynamx2 file
colnames_v_2 <- c("Protein", "Start", "End", "Sequence",
                  "Modification", "Max Exchangers",
                  "MHP", "State", "Exposure", "File",
                  "z", "RT", "Inten", "Center")

if(all(colnames_v_2 %in% colnames(dat))){
  dat <- upgrade_2_to_3(dat)
}

#check for dynamx3 file
colnames_v_3 <- c("Protein", "Start", "End", "Sequence",
                  "Modification", "Fragment", "MaxUptake",
                  "MHP", "State", "Exposure", "File", "z",
                  "RT", "Inten", "Center")

colnames_presence <- colnames_v_3 %in% colnames(dat)

if(!all(colnames_presence)) {
  err_message <- paste0(ifelse(sum(!colnames_presence) > 0,
                               "A supplied file does not have required columns: ",
                               "A supplied file does not have the required column "),
                        paste0(colnames_v_3[!colnames_presence], collapse = ", "), ".")
  stop(err_message)
}
```

Mysterious columns & different versions

Additional information







Start | Woods plot | Coverage | Sequence data | Kinetics | Summary | Report

HaDeX

Select parameters for the plot.

Theoretical calculations

Choose values type:

relative

absolute

Comparison plot parameters:

Choose time parameters:

IN	CHOSEN	OUT
0.001	1	1440

Choose states for comparison:

gg_Nucb2_CaCl2

gg_Nucb2_EDTA

Adjust colors

Woods plot parameters:

State 1	State 2
gg_Nucb2_CaCl2	gg_Nucb2_EDTA

Confidence limit 1:	Confidence limit 2:
98%	99%

Adjust plot:

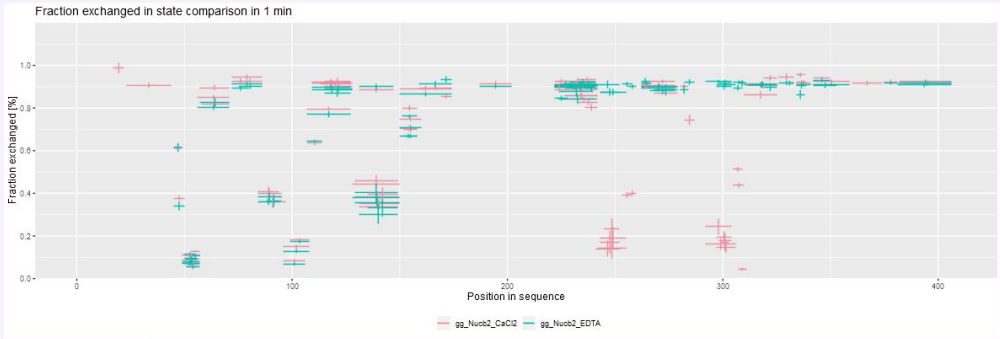
Choose y range for comparison plot: -2 to 2 (0, 1.2)

Choose y range for Woods plot: -2 to 2 (-0.5, 0.5)

Choose x range for both plots: 1 to 407

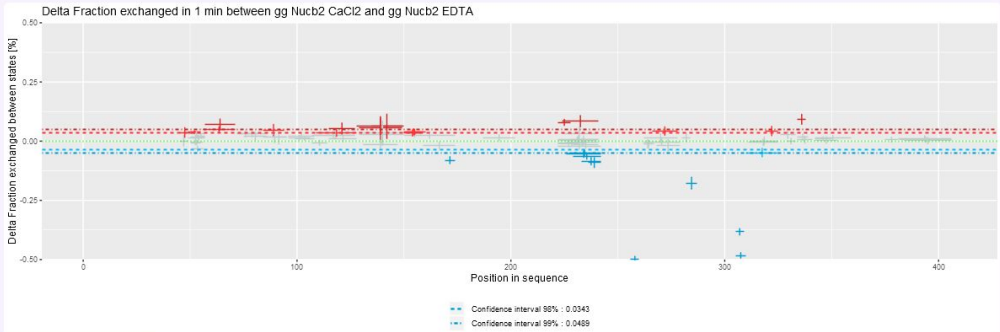
Adjust labels

Comparison plot | Data



Save chart (.svg)

Woods plot | Data



Save chart (.svg)

How is it calculated ?

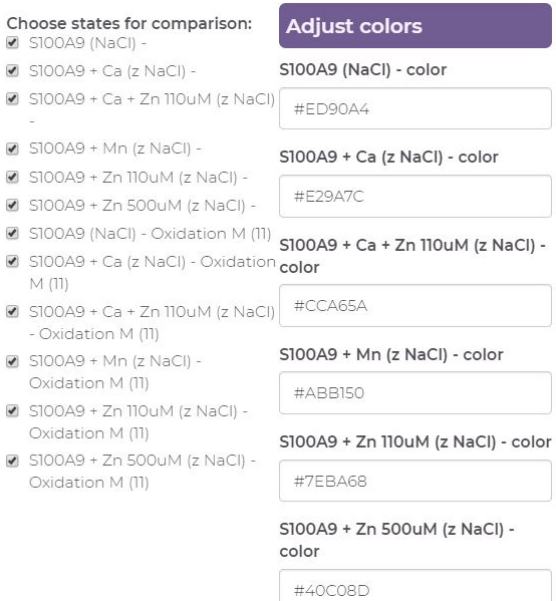
Changing colors on the plot

ui.R

```
tags$button("Adjust colors",
            class = "collapse-button",
            `data-toggle`="collapse",
            `data-target`="#colorss"),
tags$div(
  class = "hideable",
  id = "colorss",
  uiOutput("states_colors")
),
class = "states-colors-column"
```

server.R

```
##

comparison_plot_colors <- reactive({

  hcl.colors(length(states_from_file()), palette = "Set 2", alpha = NULL, rev = FALSE, fixup = TRUE)

})

##

output[["states_colors"]] <- renderUI({

  lapply(1:length(states_from_file()), function(i) {
    textInput(inputId = paste0(states_from_file()[i], "_color"),
              label = paste(states_from_file()[i], " color"),
              value = comparison_plot_colors()[i])
  })
})

##

comparison_plot_colors_chosen <- reactive({

  lapply(paste0(states_from_file(), "_color"), function(i) input[[i]])

  tmp <- t(sapply(paste0(input[["compare_states"]],"_color"), function(i) input[[i]][1], simplify = TRUE))

  tmp[tmp == "NULL"] <- NA

  if (all(is.na(tmp))) {
    comparison_plot_colors()[1:length(states_from_file())]
  } else {
    coalesce(as.vector(tmp), comparison_plot_colors()[1:length(input[["compare_states"]])])
  }

})
```



Start | Woods plot | Coverage | Sequence data | Kinetics | Summary | Report

HaDeX

Select parameters for the plot.

Theoretical calculations

Choose values type:

- relative
- absolute

Choose time parameters:

IN: 0.001 | OUT: 1440

Choose peptide:

Sequence	State	Start	End
VPIDID	gg_Nucb2_CaCl2	17	22
VPIDID	gg_Nucb2_EDTA	17	22
KTKVKGEGHVEGEKIENPDTGL	gg_Nucb2_CaCl2	23	44
KTKVKGEGHVEGEKIENPDTGL	gg_Nucb2_EDTA	23	44
YYDEY	gg_Nucb2_CaCl2	45	49
YYDEY	gg_Nucb2_EDTA	45	49
YYDEYL	gg_Nucb2_CaCl2	45	50
YYDEYL	gg_Nucb2_EDTA	45	50
YLRQVID	gg_Nucb2_CaCl2	49	55
YLRQVID	gg_Nucb2_EDTA	49	55
All	All		

Showing 1 to 10 of 216 entries

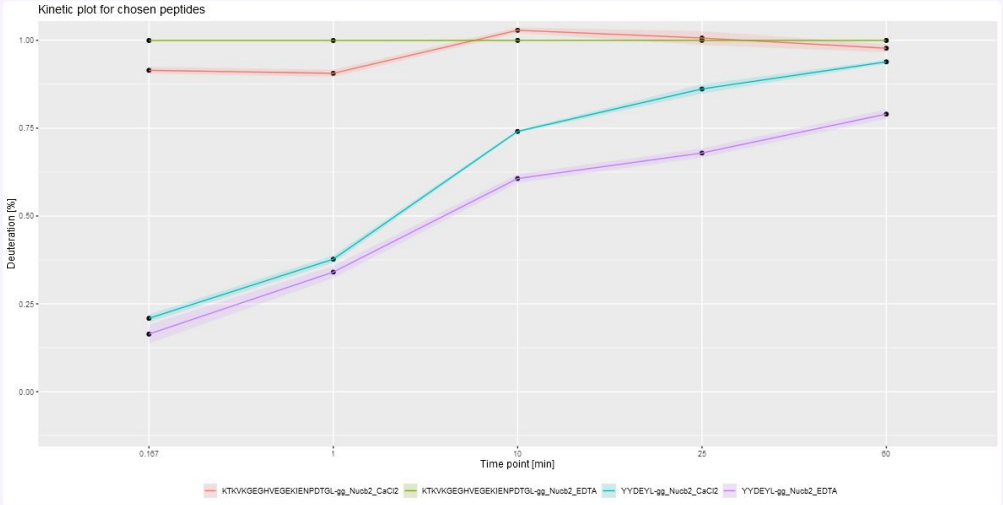
Previous 1 2 3 4 5 ... 22 Next

Choose y range for kinetic plot:

-0.5 -0.1 1 2

Adjust labels

Kinetic plot | Data



Save chart (.svg)

Can we change the colors?

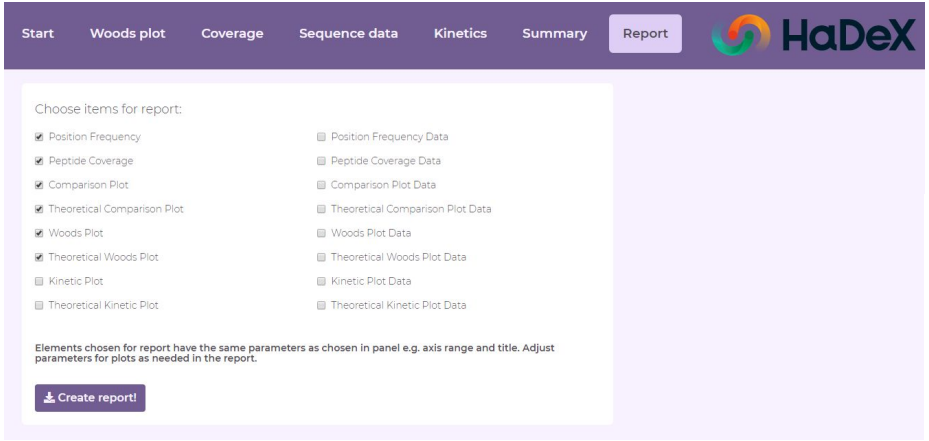
Code tips

ui.R

```
choices = c( 0 ,  1 ,  5 ,
h5("Choose peptide:"),
DT::dataTableOutput("peptide_list_data"),
br()
```

server.R

```
##

output[["peptide_list_data"]] <- DT::renderDataTable({

  datatable(data = peptide_list(),
            class = "table-bordered table-condensed",
            extensions = "Buttons",
            options = list(pageLength = 10, dom = "tip", autoWidth = TRUE, target = 'cell'),
            filter = "bottom",
            rownames = FALSE)

})

##

kin_dat <- reactive({

  validate(need(input[["peptide_list_data_rows_selected"]], "Please select at least one peptide from the table on the left."))

  bind_rows(apply(peptide_list()[input[["peptide_list_data_rows_selected"]], ], 1, function(peptide){
    calculate_kinetics(dat = dat(),
                       protein = protein_name(),
                       sequence = peptide[1],
                       state = peptide[2],
                       start = as.numeric(peptide[3]),
                       end = as.numeric(peptide[4]),
                       time_in = as.numeric(input[["kin_in_time"]]),
                       time_out = as.numeric(input[["kin_out_time"]]))
  }))

})

##
```



Knit ▾ Insert ▾

```
---
title: "hadexReport"
author: ""
date: ""
output:
  html_document:
    theme: united
    highlight: textmate
    toc: true
    toc_float: true
    number_sections: false
    df_print: paged
---

<style type="text/css">
span {
      display: block;
      max-width: 100%;
      word-wrap: break-word;
    }
</style>



Report generated on `r Sys.time()` using the [HaDeX](http://github.com/michbur/HaDeX) R package.

Detected input file: `r ifelse(is.null(input[["data_file"]][["name"]]), "none", input[["data_file"]][["name"]])`.

MD5 hash of the input file: `r ifelse(is.null(input[["data_file"]][["name"]]), "none (example file)",
tools::md5sum(as.character(input[["data_file"]][["datapath"]])))`.

# Summary

Protein name : `r as.character(unique(dat()[["Protein"]]))`

Protein sequence : <span>`r protein_sequence_colored()`</span>

Sequence length : `r input[["sequence_length"]]`
```

```
### TAB: REPORT ###

##

output[["export_action"]] <- downloadHandler(

  filename <- "HaDeX_Report.html",

  content <- function(file) {

    rmarkdown::render(input = "report_template.Rmd",
                      output_file = file, quiet = TRUE)

  })

##
```

How to deal with the “hidden” parameters?

Availability

R package
CRAN

1.0

Github

Web-server

Standalone
software

Plans for HaDeX 2.0

- multiple input file types
- object-based
- multiprotein analysis
- different approach on statistics
- high-resolution algorithm
- new requested features

take-home message

- so much to do in the natural sciences
- simply things work very good, no need to use fancy/complicated stuff
- sometimes coding is not the hardest part
- close and thorough contact with user is for your benefit

Thanks!

Acknowledgements:

- Michał Dadlez - IBB PAN
- Michał Burdukiewicz - MINI PW
- Dominik Cysewski - IBB PAN
- Dominik Rafacz - MINI PW - frontend developer