Quantitative Life Sciences: from Infectious Diseases to Ecosystems

Jessy Duran Ramirez, Chloé Pasin
jessyjomary.duranramirez@uzh.ch

20th December 2022

Molecular Dynamics
Exercise sheet

GOALS:

1. Exploring how R can handle complex structures such as molecules
2. Familiarization with the process of rational drug design and lead optimization
3. Experiment with machine learning algorithms, and understand related concepts such as variable selection and overfitting
4. Understand the limitation of in-silico drug design

The current exercise will be heavily based on packages from the Bioconductor project, one of the most important open source software frameworks for bioinformatics analysis. We therefore have to install the following packages from Bioconductor:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install(c("ChemmineR"))
library(ChemmineR)
```

In addition, we will use machine-learning approaches to better understand drug-candidate selection. For this we will install the following standard package:

```
library("caret")
```

Problem 1: Molecules in R

a) We will explore and visualize molecules in R. Let us start with Tannin! Load the information of tannin into R (available on the OLAT folder).

```
tannin <- read.SDFset("Tannin.sdf")

Now plot the molecule using


plot(tannin[[1]])
```

b) Molecules in SDF (Spatial Data File) format can be downloaded from PubChem:
`https://pubchem.ncbi.nlm.nih.gov`. Just type the compound name in the search field. Then download the 2D SDF file for the molecule. Visit the webpage and download your favourite molecule! Explore it by reading it into R and plotting it.

Problem 2: Exploring potential candidates for the next drug

The year is 1994, HIV has been isolated for almost 10 years and the knowledge about its life cycle is progressing every day.

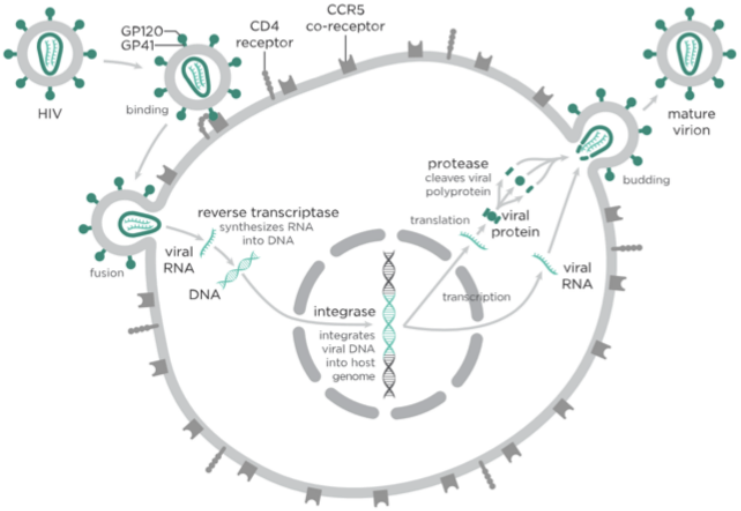

Figure 1: The life cycle of HIV. Image supplied courtesy of Thomas Splettstoesser.

In this exercise, we will consider HIV protease as a potential drug target. The following video `https://www.youtube.com/watch?v=u5GLZfCFgQM` gives you an idea about the mechanism of HIV protease.

a) Quantitative structure activity relationship (QSAR) is a quantitative approach based on empirical data to take into account the impact of a large range of molecular features on the binding of a drug to its target. Look up this method on Wikipedia and try to understand QSAR well enough so you can summarize it in your own words.
`https://en.wikipedia.org/wiki/Quantitative_structure_activity_relationship`

We will now conduct our own QSAR analysis. You are provided with a dataset of 859 compounds (desc.csv) along with their enzyme inhibition constant, Ki, in relation to the HIV-1 protease enzyme. For the purpose of this exercise we consider Ki as a measure of how strongly the compound binds the protease and hence as a measure of whether a compound is a promising drug candidate. In addition, there are 1444 features for each compound. The goal of this exercise is to use machine-learning approaches to find the features that minimize Ki; i.e., the features that minimize the concentration of the compound required to inhibit the protease. The 1444 features are derived using the PaDEL descriptor software. Explore this file for what they describe and represent. Description of the variables in dataset are provided in *Descriptors.xls* (available on the OLAT folder).

b) Note that the identification numbers in the first column are not unique. This means that for the same substance, we have information about repeated measurements of the Ki (second column). In these repeated measurements, all features are the same, but the Ki's are different. Create two subsets *descUnique1* and *descUnique2* of *desc* containing only one row per Monomer ID: 1) randomly select one experiment for each Monomer ID and 2) averaging the Ki values of repeated measurements.
Hint: check R functions *duplicated* and *aggregate*.
Continue with the subset that is more appropriate in your opinion (why?). Also, note that some features are the same for all monomers. Remove all these columns.

c) In this exercise, we have the binding activities of 613 compounds, and each compound possesses 1182 features. How do you know which features correlate best with binding activity? Is it plausible that some features are very similar to each other? E.g., molecular weight and number of carbons. Compute and visualize the correlation between some of the features using the function *cor* and *heatmap*. To move on, use *findCorrelation* to select one representative of highly correlated features.

d) Explore the correlation between some of the features with Ki using linear regression. What would happen if you were to include all the features in the linear regression model. Do you get reasonable estimates?

In order to fit models that generalize well, the concept of cross validation was introduced. `https://en.wikipedia.org/wiki/Cross-validation_(statistics)`.

e) Divide your data into two sets, a training set (70% of the data), and a validation/test set (the remaining 30%). Use the function *sample* to partition the row indices into training and test set. Examine the quantiles of Ki in the two sets, are they similar?

f) It is important that the number of features is appropriate to the number of observations and that the features do not correlate. Now, select a subset of features that best describes a potential strong binder to the HIV protease enzyme. Using the function *train* from the package *caret* and fit a model of Ki against all the available features and store it in a variable (make sure to use only the training set). You may use the following lines of code:

```
fitControl <- trainControl(method = "repeatedcv", number = ..., repeats = ...)
fit1 <- train(Ki_nM ~ ., data = ..., method = "rpart", trControl = fitControl)
```

Caret offers a plethora of models (`http://topepo.github.io/caret/available-models.html`) that you can fit, from simple linear models, all the way to deep learning and neural networks. The choice of the initial model is yours, but beware that the more complicated models might take long with high number of features. The 'rpart' method is a good starting point. Use the help function to find out what the parameters of 'trainControl' and 'train' are. You can, e.g., conduct a 10-fold cross validation.

g) Once the model is fitted you can use the function *varImp*, which gives you the top variables and their relative importance. Now retrain your model on the training set, using only the subset of features, which you consider important (note that you can also include features if you have a scientific rationale for their inclusion even if *varImp* does not support them).

h) How well does your model perform? Report RMSE and R-squared, two frequently used measures to quantify the performance of a model. Try a different method/model and assess the performance difference. For example, how does a linear model fitted using 'lm' compare to a neural network fitted using 'nnet'?

i) Now compare the performance your trained models on the test set (30% of the data which we have not used) Use the function *postResample*: `http://topepo.github.io/caret/measuring-performance.html#measures-for-regression`). How well does it perform?

j) Try to interpret your results: Do you believe that if you were given millions of molecules that your model would be able to advise you about their potential binding, and thus you would be able to create a short list of molecules to take further down the pipeline for drug development?