

# Wprowadzenie do R i RStudio

Analizy bioinformatyczne w badaniach genomowych | Ćwiczenia 1

8.10.2024

# Plan zajęć

## **1. Część teoretyczna**

- Wprowadzenie do R
- Podstawy języka R
- Wprowadzenie do Rstudio
- Pakiety i rozszerzenia

## **2. Przegląd funkcji bazowych R**

## **3. Część praktyczna A**

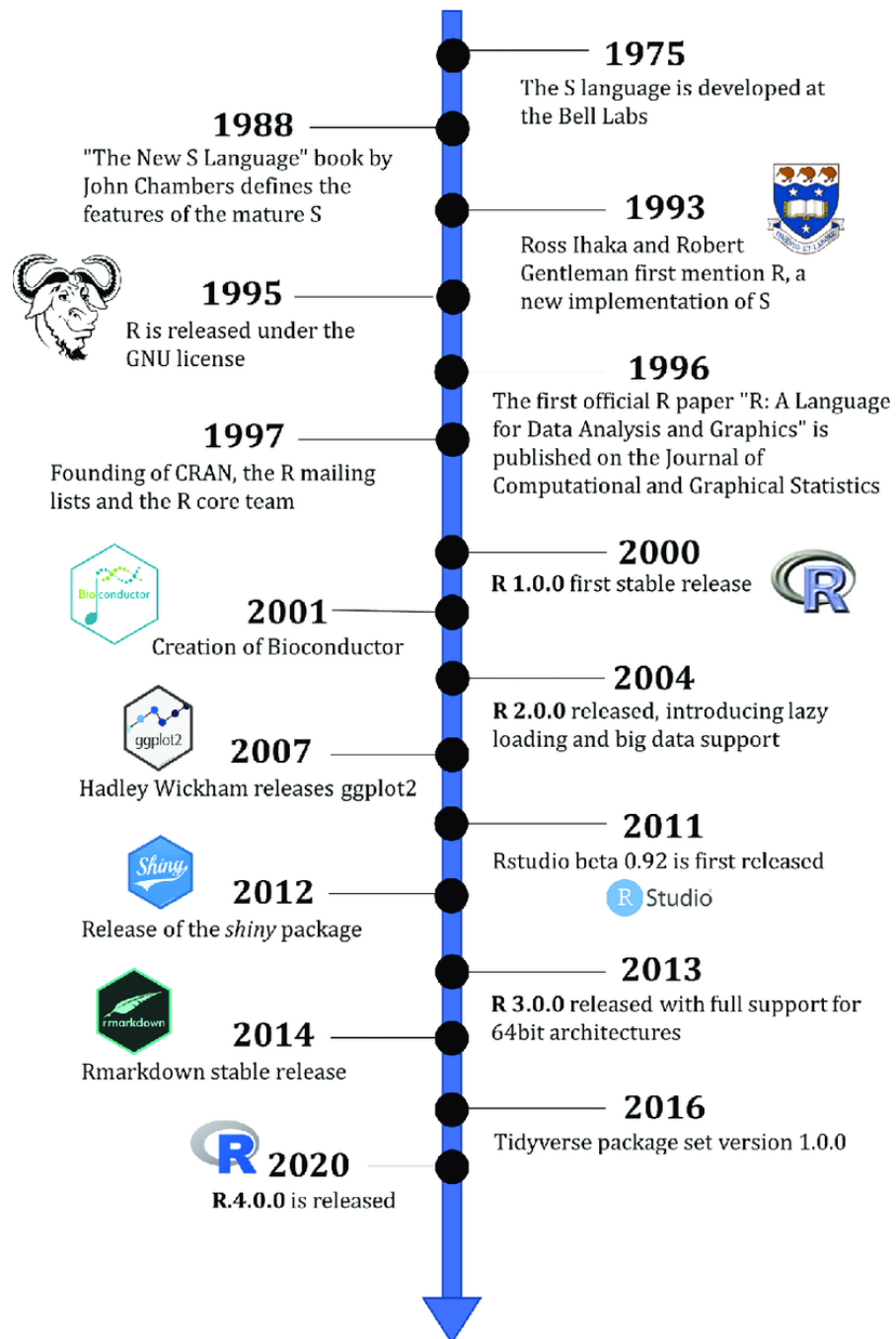
Wspólnie poznajemy możliwości R i Rstudio

## **4. Część praktyczna B**

Samodzielna praca z manipulacją danymi

# Wprowadzenie do R

# Historia i rozwój języka R



# Filozofia R

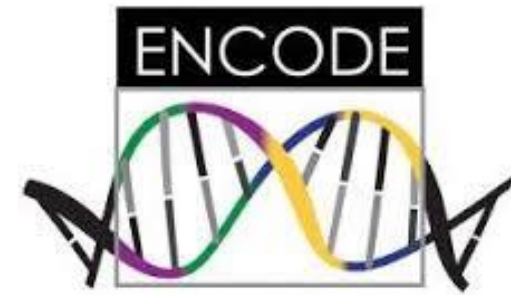
- rozwijany przez społeczność
- wszystko w otwartym dostępie
- nastawienie na wykorzystanie niekomercyjne (badania naukowe)

# Zastosowania R w bioinformatyce i badaniach genomowych

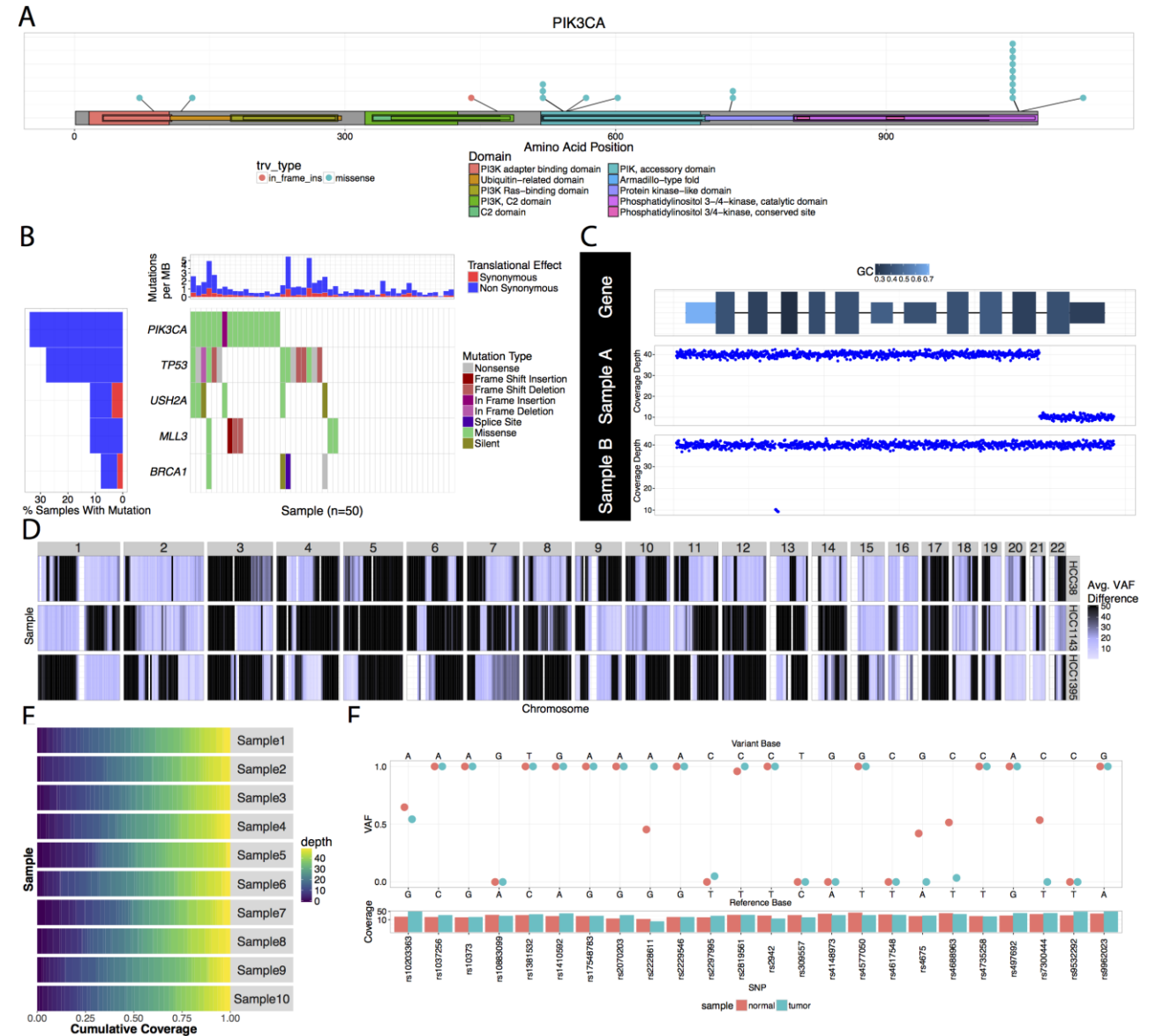
- Analiza sekwencji genomowych
- Przetwarzanie danych z mikromacierzy i NGS (Next-Generation Sequencing)
- Wizualizacja wyników analiz

# Wykorzystanie R w badaniach i projektach

- ENCODE
- 1000 Genomes
- GTEx

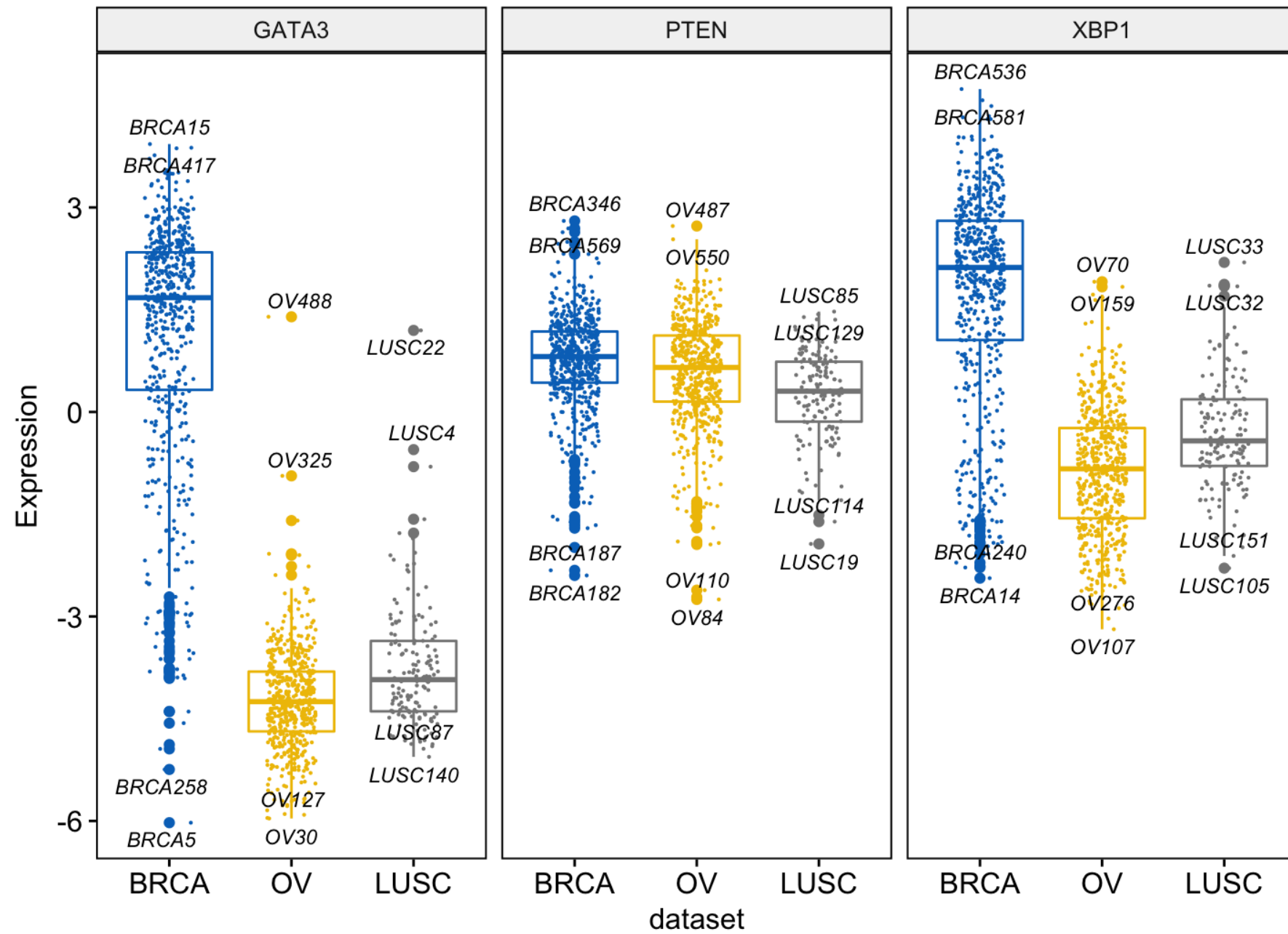


# Wizualizacja danych genomowych





dataset BRCA OV LUSC





# Podstawy języka R

# Składnia i struktura języka

- Komentarze - #

```
#to jest komentarz
```

- Przypisywanie wartości

```
#metoda 1:  
x <- 5  
#metoda 2:  
y = 10
```

Wydzielanie podzbioru ze zbioru danych - \$

```
podzbior <- zbior$podzbior
```

# Typy danych w R

Wektory – jednowymiarowe struktury danych

```
liczby <- c(1, 2, 3, 4, 5)
```

Macierze – dwuwymiarowe tablice jednego typu danych

```
macierz <- matrix(1:9, nrow = 3)
```

Ramki danych (data frames, df)

```
ramka <- data.frame(  
  imie = c("Anna", "Piotr"),  
  wiek = c(28, 34)  
)
```

# Typy danych w R

Listy: kolekcje elementów różnych typów

```
lista <- list(wektor = liczby, ramka = ramka)
```

Czynniki (factors): zmienna kategoryczna

```
czynniki <- factor(c("mały", "duży", "średni"))
```

# Operatory arytmetyczne, logiczne i relacyjne

**Arytmetyczne:** +, -, \*, /, ^

**Porównania:** ==, !=, <, >, <=, >=

**Logiczne:** & (i), | (lub), ! (nie)

**Warunkowe:** if(condition) { expr } else { expr }

ifelse(test, yes, no)

any(x)

all(x)

which(x)

# Podstawowe funkcje (base)

- matematyczne

- `sqrt(x)`
- `log10(x)`
- `sin(x)`, `cos(x)`, `tan(x)`
- `ceiling(x)`
- `floor(x)`
- `trunc(x)`

- statystyczne

- `mean(x)`
- `sd(x)`
- `min(x)`
- `max(x)`
- `range(x)`
- `summary(x)`
- `table(x)`
- `cor(x, y)`
- `cov(x, y)`

- opisowe

- `sqrt(x)`
- `log10(x)`
- `sin(x)`, `cos(x)`, `tan(x)`
- `ceiling(x)`
- `floor(x)`
- `trunc(x)`



# Wprowadzenie do RStudio

# Co to jest Rstudio?

- Zintegrowane środowisko programistyczne (Integrated Developer Environment, IDE) dla R
- Ułatwia pisanie kodu, zarządzanie projektami i wizualizację wyników

# Zalety RStudio

- Możliwość tworzenia projektów
- Każda część składni widoczna w innym kolorze
- Automatyczne uzupełnianie nawiasów i innych operatorów
- Automatyczne debugowanie kodu (sort of...)
- Szybki podgląd wizualizacji wyników
- Łatwe przemieszczanie się po skrypcie
- Brak konieczności znajomości pracy w command line
- Może służyć jako IDE również do innych języków programowania

# Interfejs użytkownika

- Konsola (Console): miejsce wykonywania poleceń w czasie rzeczywistym
- Edytor skryptów (Source): pisanie i edycja skryptów R
- Środowisko robocze (Environment): przegląd zmiennych i danych
- Panel wyjściowy (Plots, Files, Packages, Help): wyświetlanie wykresów, zarządzanie plikami i pakietami, dostęp do pomocy

Project: (None)

EnvironmentHistoryConnectionsTutorial

Import Dataset142 MiB

List

RGlobal Environment

Data

all_parameters_summ...	2716 obs. of 36 variables	
all_samples_summary	302 obs. of 5 variables	
angle_data	266 obs. of 2 variables	
angle_data_back	249 obs. of 2 variables	
angle_data_frames	List of 15	
angle_data_head	244 obs. of 2 variables	
angle_data_back	147 obs. of 2 variables	

FilesPlotsPackagesHelpViewerPresentation

InstallUpdate

Name	Description	Version	
System Library			
abind	Combine Multidimensional Arrays	1.4-5	
ade4	Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences	1.7-20	
adegenet	Exploratory Analysis of Genetic and Genomic Data	2.1.10	
ade4phylo	Exploratory Analyses for the Phylogenetic Comparative Method	1.1-13	
agricolae	Statistical Procedures for Agricultural Research	1.3-5	
alabama	Constrained Nonlinear Optimization	2022.4-1	
AlgDesign	Algorithmic Experimental Design	1.2.1	
ape	Analyses of Phylogenetics and Evolution	5.6-2	
aplot	Decorate a 'ggplot' with Associated Information	0.1.9	
apTreeshape	Analyses of Phylogenetic Treeshape	1.5-0.1	
askpass	Safe Password Entry for R, Git, and SSH	1.1	
assertive	Readable Check Functions to Ensure Code Integrity	0.3-6	
assertive.base	A Lightweight Core of the 'assertive' Package	0.0-9	
assertive.code	Assertions to Check Properties of Code	0.0-3	
assertive.data	Assertions to Check Properties of Data	0.0-3	
assertive.data.uk	Assertions to Check Properties of Strings	0.0-2	
assertive.data.us	Assertions to Check Properties of Strings	0.0-2	
assertive.dattimes	Assertions to Check Properties of Dates and Times	0.0-3	
assertive.files	Assertions to Check Properties of Files	0.0-2	

tibution-plots1.py xheritability-calculator4.R xcoordinates-cleaned2.py xeuclidean-distance-plotter2.py x

Source on SaveRunSource

```
1 library(readxl)
2
3 extract_pedigree <- function(indiv_id, data, edges = data.frame(from = character(0), to = character(0))
4   # Check if the indiv_id exists in the dataset
5   if(!indiv_id %in% data$Indiv) {
6     return(edges)
7   }
8
9   horse <- data[data$Indiv == indiv_id, ]
10
11   if(!is.na(horse$Sire)){
12     edges <- rbind(edges, data.frame(from = as.character(horse$Sire), to = as.character(indiv_id)))
13     edges <- extract_pedigree(horse$Sire, data, edges)
14   }
15
16   if(!is.na(horse$Dam)){
17     edges <- rbind(edges, data.frame(from = as.character(horse$Dam), to = as.character(indiv_id)))
18     edges <- extract_pedigree(horse$Dam, data, edges)
19   }
20
21   return(edges)
22 }
```

75:41 (Top Level) R Script

ConsoleTerminalBackground Jobs

R 4.1.2 ~/

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/RData]

# Personalizacja ustawień i korzystanie z pomocy

- możliwość zmiany tła, kolorów czcionek, wielkości poszczególnych okien
- pakiet 'learnr' -> specjalnie do nauki R
- zakładka 'Help' w panelu wyjściowym
- Skróty klawiszowe

Pakiety i rozszerzenia

# Instalacja i zarządzanie pakietami

Instalacja pakietu:

```
install.packages("nazwa_pakietu")
```

Ładowanie pakietu:

```
library(nazwa_pakietu)
```

*Można to też zrobić przez panel wyjściowy, ale lepiej się nie przyzwyczajać...*



# Przykładowe pakiety

- **ggplot2**: najbardziej zaawansowany pakiet do wizualizacji danych
- **dplyr**: manipulacja danymi, funkcja pipe (`%>%`)
- **knitr**, **rmarkdown**: pozwala na połączenie kodu, wyników i opisu w jednym dokumencie
- **readxl**: umożliwia wczytywanie danych z plików Excel
- **plotly**: interaktywne wykresy
- **Shiny**: interaktywne aplikacje internetowe
- **esquisse**: interfejs graficzny pozwalający na tworzenie wykresów bez znajomości ggplot2

# Bioconductor: pakiety bioinformatyczne

- Pierwsza wersja platformy BioConductor powstała w 2001 r., zaraz po pierwszej stabilnej wersji języka R
- Oprócz R korzysta także z innych języków, m. in. Python, C, Swift
- Setki pakietów do analizy sekwencji, mikromacierzy etc.



Koniec teorii! Pytania?