Statistical analysis

Demo1:

Part 1)

* General R project organization

Part 2)

* Rmarkdown and knitting

Part 3)

* Biostrings package to manipulate biological strings
* countPattern -> counts number of “pattern” inside DNAstring
* vcountPattern -> count in multiple DNAstrings from one vector
* matchPattern -> position where “pattern” starts and ends
* oligonucelotideFrequency -> counts of oligonucleotides sequences of “width” (works for multiple sequences)

Part 4)

* stringr package to manipulate strings
* string\_replace\_all -> replace all nucleotides “x” for “y” in DNAstring
* str\_split -> separate string at given character (simplify = TRUE to return matrix/vector)

Part 5)

* for loops

Part 6)

* Apply functions

Exercise 1:

Question 2)

* Read in fasta files, plot counts, create dataframe, count number of specific AA, write to file

Question 3)

* Transform corrupted string, str\_tp\_upper, str\_replace\_all

Question 4)

* Read in multiple files, calculate cpm, take highest

Demo 2:

Part 1) Data Frames

* read\_csv to read in data into dataframe (readr)

Part 2) Pipes

* “function1” %>% “function2” to give output of function1 as input to function2

Part 3) Manipulating dataframes with dplyr

* slice -> to select arbitrary rows/columns
* filter -> filter column by specified values
* select -> select specific columns
* mutate -> create new column
* rename -> rename existing column (can be done by combining mutate and select)
* transmute -> combine mutate select and rename into one command
* recode -> combined with mutate to change specific value in column for all entries of that values
* group\_by -> groups rows by value in specific column, all calculations perfomed over multiple columns are now only done inside the groups
* summarise -> creates summary, one line per group

Part 4) Joining daraframes

* inner\_join -> only take keys present in both
* full\_join -> take all keys from either dataframe
* left\_join -> take keys in first dataframe
* right\_join -> take keys in second dataframe

Part 5) tidyr

* spread -> from long to wide
* gather -> from wide to long
* tibbles do not work with rownames

Exercise 2:

Question 1)

* make dataframe using pipe
* make wide format into long
* join dataframes
* count -> count instances of value in column

Question 2)

* read in csv file
* transmute
* left\_join
* filter
* recode
* summarise and arrange

Question 3)

* use lapply to read in multiple files from directory
* calculate cpm and count cpm > 1
* transform long to wide and into matrix

Demo 3:

Part 1) ggplot2

* Basic plotting, highly points value based, aesthetics, point size/shape, coloring, manual colors, scales, RColorBrewer (colorRampPalette)
* Axis, set lim, set scale
* Theme, Faceting
* Other geoms (density, barplot, violin + boxplot)
* Saving figures
* Multipanel (cowplot)

Part 2) ComplexHeatmap

* Colors and names
* Clustering and labels
* Annotation -> bars, boxplots
* Legend and titles

Exercise 3:

* Different type of plots

Demo 4:

* CATALYST package
* Import data, create sce (prepData), get overview of sce, plots counts

Part 1) Clustering

* Compute clusters in dimension xdim ydim for 2 till maxK
* Clustree visualization

Part 2) Type- and state-markers

* type\_markers <- Access type markers (state\_markers)
* plotExprHeatmap <- plot expression heatmap of type markers by cluster\_id

Part 3) Dimensionality reduction

* runDR(sce, dr = “type of dr”) -> perform dimensionality reduction
* plotDR(sce,color\_by= ”meta12”/”patient\_id”/”condition”,”genes expression”,facet\_by=”codition”) -> plot dimensionality reduction

Part 4) Pseudobulk-level MDS plot

* pbMDS(sce, by = “sample\_id”/”both”) -> compute and plot mulit-dimensional scaling

Part 5) Differential analysis

* plotMultiHeatmap(sce,hm1=FALSE,scale=”never”, hm2 = state\_markers(sce)[1:choose\_how\_many\_of\_the\_markers]) -> plots a heatmap with expressions of state genes for all the markers
* identify subpopulation specific chantes in expression across conditions (diffcyt)
* plotDiffHeatmap(sce,tbl)
* plot\_Abundances(sce,by=”sample\_id”/”cluster\_id”)investigate the relative abundance of genes
* plotFreqHeatmap

Exercise 4: Cytof

Part 1) Loading the data (from ExperimentHub)

Part 2) Construct sce

* download panel, use prepData

Part 3) Clustering

* cluster, look at table, plotAbundances for one metak for all clusters

Part 4) type and state markers

* test type markers assumption

Part 5) dimensionality reduction

* runDR, plotDR -> patient effect?, condition effect?

Part 6) differential discover

* exploratory data analysis (mds)

Exercise 6:

* find highly variable genes, run PCA,TSNE -> find patient/technology effects
* clustering, buildSNNGraph (using pca), plot pca tsne colored by cluster
* find markers gene for each cluster, violing plot (plotExpression)
* marker genes plotted on tsne/pca
* find reference and assign labels