**Lesson 4. Computing Biological Models in RStudio**

**LESSON 4 GOALS**

1. Learn what the provided data sheets mean in a computational and a biological context and be able to effectively filter mass transcriptomic data to identify significant genes.

1. Learn what an SIR model is, how the algorithm coerces the population dynamics to evolve over the course of time, and how SIR models can be used in real life.
2. Learn how to build a computational model that finds the optimal harvest rate for a fish species dependent on biological parameters specific to the species and understand the relevance as to why this modeling is important.

**LESSON 4.1 – Filtering transcriptomic data to identify key genes involved in the pathogenesis of Systemic Lupus Erythematosus**

**BACKGROUND**

Systemic Lupus Erythematosus (SLE) is a complex autoimmune disease that can affect various organs and systems in the body.

The exact cause of SLE is not fully understood, but it involves the immune system mistakenly attacking healthy tissues, leading to inflammation and damage.

Common physiological symptoms of SLE include joint pain, skin rashes, fatigue, fever, and organ involvement like kidneys, heart, and lungs.

The prevalence of SLE varies worldwide, but it is estimated that approximately 0.1% to 0.2% of the global population is affected, with a higher incidence in women, particularly during their childbearing years.

Treatment for SLE aims to control symptoms, reduce inflammation, and prevent organ damage. Medications such as anti-inflammatory drugs, corticosteroids, and immunosuppressants are commonly prescribed. Additionally, lifestyle modifications and regular medical monitoring play crucial roles in managing the condition.

As research progresses, targeted therapies and personalized treatment approaches hold promise for improving the management of SLE and enhancing the quality of life for those affected by this chronic autoimmune disorder.

Systemic Lupus Erythematosus (SLE) remains a challenging autoimmune disease whose exact etiology is yet to be fully elucidated. While it is proposed to have a genetic component, the precise cause of SLE is still unknown.

To shed light on this enigmatic condition, **our assignment focuses on investigating potential hereditary factors by analyzing mRNA-seq data.** *Our goal is to identify key genes that exhibit significant upregulation or downregulation in SLE patients compared to healthy individuals.*

Through this comprehensive analysis, we hope to uncover crucial molecular pathways and genetic signatures associated with SLE, providing valuable insights into the disease's underlying mechanisms and potential therapeutic targets. By advancing our understanding of the genetic basis of SLE, this research endeavor aims to pave the way for more personalized and effective treatment strategies, ultimately improving the lives of individuals affected by this complex autoimmune disorder.

**DATA EXPLANATION**

The provided data spreadsheets are crucial for our analysis on Systemic Lupus Erythematosus (SLE) and offer valuable insights into gene expression patterns. *The data has already been pre-analyzed, and it includes four essential columns:*

**1. Gene name:** This column identifies the specific genes that were analyzed in the study. Each row represents a different gene transcript.

**2. Log2 fold change:** The Log2 value represents the fold change in gene expression between SLE patients and healthy individuals. A positive Log2 value indicates higher gene expression in SLE patients compared to healthy individuals, while a negative Log2 value indicates lower gene expression in SLE patients.

**3. p-value:** The p-value is a statistical measure that assesses the significance of the differences in gene expression between SLE patients and healthy individuals. A p-value below 0.05 is considered statistically significant, indicating consistent and reliable data.

**4. Study name:** This column provides a unique identifier for each study from which the data was obtained. It allows us to differentiate between datasets and maintain clarity during the analysis.

The first data sheet serves as an example problem sheet, containing Log2 values, p-values, and identifiers from two different studies—one being a perfect dataset, and the other containing many NA (Not Available) values that need to be transformed into a new data frame. This sheet serves as a practice exercise to familiarize ourselves with the dataset and hone our filtering skills.

The second data sheet, the composite dataset, is the primary focus of our analysis for the homework problem. It combines data from multiple studies, providing a comprehensive overview of gene expression patterns in SLE patients versus healthy individuals. By scrutinizing this composite dataset and applying appropriate filtering and statistical analysis, we aim to identify crucially modulated genes associated with SLE, contributing to a deeper understanding of the disease and potential therapeutic targets.

Overall, this script we will be utilizing is a comprehensive analysis that filters and visualizes transcriptomic data related to SLE, allowing researchers to identify and prioritize genes that are significantly upregulated or downregulated in SLE patients compared to healthy individuals. The volcano plot provides an intuitive way to visualize the data and facilitates the identification of biologically relevant genes for further investigation.

**EXPLANATION OF SCRIPT**

*This RStudio script conducts transcriptomic analysis and generates a volcano plot to identify significantly modulated genes in Systemic Lupus Erythematosus (SLE) compared to healthy individuals. Here's an explanation of each step:*

**Step One: Load in data**

-The script reads the data from the CSV file "RLESSON4\_PART1\_DATAEXAMPLE.csv" into a variable called "ExampleData".

**Step Two: Remove NAs and coerce into a dataframe**

- The script removes any rows containing missing values (NAs) from the "ExampleData" and stores the filtered data in a new dataframe called "example".

**Step Three: Filter significant data**

- The script filters the "example" dataframe to include only rows with a corrected p-value less than 0.05 and log2 fold change values greater than or equal to +/-2. The resulting dataframe is stored in "sigfilt.example".

**Step Four: Filter insignificant data**

- The script filters the "example" dataframe to include only rows with log2 fold change values between -2 and 2 (exclusive) or corrected p-values greater than 0.05. The filtered data is stored in "insigfilt.example".

**Step Five: Transform p-values by taking the -log10 for the volcano plot**

- The script calculates the negative log10 values of the corrected p-values for both "sigfilt.example" and "insigfilt.example" dataframes. This transformation is commonly done in volcano plots to better visualize significant p-values.

**Step Six: Find duplicates**

- The script calculates the occurrences of duplicate gene names in the "sigfilt.example" dataframe and stores the counts in the variable "sigduplicates.example." Duplicates indicate that certain results occurred multiple times across different studies, making them more significant.

**Step Seven: Create a volcano plot**

- The script generates a scatter plot with log2 fold change values (SLE/Healthy) on the x-axis and negative log10 transformed corrected p-values on the y-axis. This plot is known as a "volcano plot" and is commonly used in transcriptomic analysis to visualize the distribution of significant gene expression changes. The data points representing significantly modulated genes are colored red, while insignificantly modulated genes are colored gray. Vertical lines are drawn at +/-2 to indicate the boundaries for significant fold changes.

**LESSON 4.2 – Creating a Susceptible-Infected-Recovery (SIR) infectious disease model of COVID-19**

In the face of infectious disease outbreaks, the scientific community relies on powerful tools like SIR models to unravel the complex dynamics and make informed predictions. The Susceptible-Infectious-Recovered (SIR) model is a mathematical framework used to study the progression of infectious diseases within a population. This model has proven instrumental in guiding public health responses and formulating strategies to mitigate the impact of epidemics.

At its core, the SIR model divides the population into three main categories: Susceptible (S), Infectious (I), and Recovered (R) individuals. These categories are dynamic and continuously change as the epidemic unfolds over time. Initially, a large portion of the population falls into the Susceptible category, as they have not yet encountered the infectious agent.

As the disease spreads, some susceptible individuals come into contact with infected individuals, transitioning into the Infectious category. The rate at which this transmission occurs depends on factors like the infectiousness of the disease and the frequency of interactions among people. As the infection progresses, a fraction of the Infectious population recovers from the disease and becomes immune, moving them into the Recovered category.

The flow of the population dynamics from one category to the next is crucial in understanding the trajectory of the epidemic. As the number of susceptible individuals decreases over time due to infections and recoveries, the spread of the disease may start to slow down, leading to a decline in new cases. Eventually, if enough individuals become immune through either recovery or vaccination, the disease may die out, or at least, its impact will be significantly reduced. See figure below for a visual aid demonstrating the flow of population dynamics over time as more people are exposed to the agent and recover.



The SIR model's predictive power lies in its ability to forecast the course of an epidemic based on initial conditions, infectiousness, and other relevant parameters. By analyzing historical data and fitting the model to observed trends, scientists and health experts can estimate the potential size of an outbreak, determine when it may peak, and evaluate the effectiveness of different intervention measures.

It's important to note that the SIR model is a simplified representation of the complex realities of disease transmission. More sophisticated models, like SEIR (which includes an Exposed category) or agent-based models, may be used to account for additional factors. Nevertheless, the SIR model remains an invaluable tool for gaining insights into the spread of infectious diseases, enabling authorities to make informed decisions to protect public health and save lives.

Overall, this RStudio script allows researchers and health experts to understand and visualize the progression of an infectious disease over time and assess the potential impact of different interventions, such as vaccination, on the disease dynamics.

**EXPLANATION OF SCRIPT**

*The provided RStudio script is an implementation of the Susceptible-Infectious-Recovered (SIR) model to simulate the progression of an infectious disease, representing COVID-19 in this context. The script sets certain parameters, such as population size, initial numbers of susceptible, infectious, recovered, and deceased individuals, infection rate, death rate, recovery rate, birth rate, and other variables related to vaccination and immunity loss. The main goal of the script is to calculate and visualize the frequency of susceptible, infectious, recovered, and deceased individuals over time using a for loop.*

Let's break down the script into sections:

**Step One: Set Parameters**

-In this section, various parameters are defined, each representing a specific aspect of the disease and the population. These parameters include the initial population size (`initial.pop`), the number of susceptible individuals initially (`s.initial`), the number of infectious individuals initially (`i.initial`), the number of recovered individuals initially (`r.initial`), and the number of deceased individuals initially (`d.initial`).

-Other parameters include the infection rate (`infection.rate`), death rate (`death.rate`), recovery rate (`recovery.rate`), normalized death rate (`norm.death.rate`), birth rate (`birth.rate`), and immunity loss rate (`immunity.loss.rate`). Additionally, there are parameters related to vaccination, including `vaccinations`, which is initialized to 0.

**Step Two: Calculate Variables Using SIR Model**

-This section defines a function `SIR\_Model`, which implements the SIR model using the defined parameters and iteratively calculates the number of susceptible, infectious, recovered, and deceased individuals over a specified time period (`t`) using a for loop. The model calculates the changes in each category based on the flow of the population dynamics and updates the counts accordingly.

**Step Three: Plot Loop Output**

-Finally, the script uses the `SIR\_Model` function to generate data on the frequencies of susceptible, infectious, recovered, and deceased individuals over time (`outputList`). It then creates a plot using this data to visualize the course of the infectious disease.

-In the plot, time (years) is represented on the x-axis, and the number of individuals in each category (susceptible, infectious, recovered, deceased) is represented on the y-axis. The number of susceptible individuals is shown in blue, infectious individuals in red, recovered individuals in green, and deceased individuals in black. A legend is provided to identify each curve on the graph.

**LESSON 4.3 – Determining optimal harvest rate of a fish species based on parameters of health span of the species**

Fisheries play a crucial role in providing food and livelihoods for millions of people worldwide. However, improper management of these valuable aquatic resources can lead to overexploitation and ecological imbalances, threatening the very populations they seek to sustain. To address this challenge, researchers and policymakers are increasingly turning to computational models to determine optimal harvest rates.

At the heart of these models are biological parameters specific to each fish species, such as growth rates, reproductive patterns, mortality rates, and carrying capacities. By integrating this data into sophisticated algorithms, scientists can simulate the dynamics of fish populations under various harvesting scenarios.

The goal is to find the "sweet spot" - the optimal harvest rate that strikes a delicate balance between sustainable yield and population stability. When the harvest rate matches the natural replenishment rate of the fish, the population remains homeostatic, allowing for continuous productivity without depletion.

Computational models can assess the impact of different fishing intensities over time, offering invaluable insights into the long-term consequences of specific harvesting strategies. By accounting for biological uncertainties and environmental fluctuations, these models enable policymakers to make informed decisions that promote responsible and sustainable fishing practices.

Ultimately, harnessing the power of computational models empowers fisheries management to avoid catastrophic collapses of fish populations while maximizing the yield potential. These efforts not only safeguard the health of marine ecosystems but also secure the livelihoods of fishing communities and contribute to global food security in an ever-changing world.

Overall, this script allows for the investigation of how different harvest rates impact the fish population's expected yield and variability. It demonstrates the use of computational modeling to explore ecological dynamics and informs decision-making for sustainable fishery management.

**EXPLANATION OF SCRIPT**

*This RStudio script is a simulation and analysis of a simple ecological model that represents the dynamics of a fish population subject to harvesting. The main purpose of the script is to explore how different harvesting rates (h) affect the expected yield and variability in fish population sizes over time.*

Let's break down the script step by step:

**Step 0: Set up model and parameters**

- The script starts by defining various model parameters:

 - 'r': The per capita growth rate of the fish population.

 - 'rSpread': The spread in the birth rate, indicating random variation in the birth rate around 'r'.

 - 'm': The per capita mortality rate of the fish population.

 - 'h': The target harvest rate, which represents the proportion of the fish population harvested each year.

 - 'N\_initial': The initial population size of the fish population.

 - 'runs': The number of times the simulation will be run.

**Step 1: Set up treatments and datasheet**

- The script creates an empty matrix called 'dataSheet' to store the results of each simulation run.

- It defines a vector 'h\_vec' that contains a sequence of harvest rates, ranging from 0.05 to 1 with a step size of 0.1.

**Step 2: Run experiments and collect data**

- The script runs a loop over different values of 'h' from the 'h\_vec' vector.

- For each 'h' value, it runs another loop 'runs' number of times to simulate the fish population dynamics over 100 years.

- Within the inner loop, the fish population size ('N') is updated yearly based on the growth, mortality, and harvest rates.

- The population size at year 50 to 100 is used to calculate the mean yield for that particular run with the current 'h' value. The results are then appended to the 'dataSheet'.

**Step 3: Plot and analyze data**

- After running the simulations, the script aggregates the results by calculating the mean and variability (standard deviation) of the yields for each 'h' value from 'dataSheet'.

- The script then proceeds to create a plot to visualize the relationship between 'h' (harvest rate) and both the expected yield and yield variability.

- The plot shows two lines: one representing the expected yield ('Expected Yield') and another representing the variability ('Variability') for different 'h' values.

- The points on the plot show the data points collected from the simulations, and the legend explains the meaning of the lines and points.

**HOMEWORK PROBLEMS**

**\*Highlighted regions indicate where your answers to the questions should be written. You will NOT submit your RMD file or screenshots from your code. *If your answers are correct, then it is clear your code was correct.\****

1. **Complete the same transcriptomic data analysis now with the composite dataset.**

* 1. Submit a list of gene names with their frequency that appear as duplicates in the dataset at least 3 times. This list will be around ~30 genes. List of ~30 duplicated genes (with their frequency value) between datasets:
		1. xxx (freq = )
		2. xxx (freq = )
		3. xxx (freq = )
		4. xxx (freq = )
		5. xxx (freq = )
		6. xxx (freq = )
		7. xxx (freq = )
		8. xxx (freq = )
		9. xxx (freq = )
		10. xxx (freq = )
		11. xxx (freq = )
		12. xxx (freq = )
		13. xxx (freq = )
		14. xxx (freq = )
		15. xxx (freq = )
		16. xxx (freq = )
		17. xxx (freq = )
		18. xxx (freq = )
		19. xxx (freq = )
		20. xxx (freq = )
		21. xxx (freq = )
		22. xxx (freq = )
		23. xxx (freq = )
		24. xxx (freq = )
		25. xxx (freq = )
		26. xxx (freq = )
		27. xxx (freq = )
		28. xxx (freq = )
		29. xxx (freq = )
		30. xxx (freq = )
	2. Select any 3 genes of interest and conduct a brief literature analysis to propose a hypothesis as to why the transcription of these genes may be modulated in SLE/LN patients comparative to healthy individuals. List the three genes of interest you selected:
		1. xxx
			1. Proposed hypothesis:
		2. xxx
			1. Proposed hypothesis:
		3. xxx
			1. Proposed hypothesis:
1. **Complete the same SIR model with the following modifications:**
	1. Change the vaccination efficacy rate to **0.96.**
	2. Only observe the change in population dynamics over the course of **15 years** not 50.
	3. Provide a description of how the two graphs(example and homework problem) are different and create an argument of why vaccination intervention is important using the graphs as your evidence.
		1. Graph comparison description:
		2. Argument:
2. **Complete the same fishery harvest model now with Bluefin Tuna’s per capita growth rate and spread.**

* 1. Change the per capita growth rate parameter to **0.286**. This is the per capita growth rate of Bluefin Tuna (the most commonly fished tuna) as of 2022.
	2. Change the spread in birth rate to **0.2**. Tuna is very well studied, and due to this their spread in birth rate is smaller as there are mass datasets calculating the birth rate to a relatively precise spread.
	3. Change the starting population to **100,000**.
	4. Change the **harvest rate vector** to **sample from 0.05 to 1 by increments 0.01.**
		1. Paste a screenshot of your example graph:
		2. Paste a screenshot of your homework problem graph:
		3. What is the optimal harvest rate for the theoretical made-up fish in the example problem? *(Hint: view the ‘expectedYield’ datasheet and find the harvest rate with the highest yield.):*
		4. What is the optimal harvest rate for Bluefin Tuna?:
		5. Is the optimal harvest rate for Bluefin Tuna lower or higher than the theoretical fish? Why? What parameter(s) did we change that affected the harvest rate?: