It is undisputable that translational research fields in general and cancer research in particular have greatly benefited from the advent and availability of Big Data as a great resource to identify trends, patterns and correlations that often help guide and validate research questions. Nevertheless, even basic analysis of publicly available data can be often outsourced by research groups due to a lack of knowledge in practical statistical or bioinformatic techniques. With this in mind, the main objective of this project was to develop a pipeline that provides access into current The Cancer Genome Atlas (TCGA) datasets with the ultimate objective of further exploiting these resources through the course of our career.

TCGA is a conjoined effort by the National Cancer Institute and the National Human Genome Research Institute that has resulted in the molecular characterization of more than 20,000 primary cancer samples, matching these to normal samples across 33 cancer types. This has been achieved by collecting and analyzing samples from more than 11,000 patients over a 12-year period. This huge achievement has provided the scientific community with genomic, epigenomic, transcriptomic, and proteomic data for which new and sophisticated technology was employed while also overcoming the nuances of different cancer types and other factors. Unsurprisingly, this has generated over 2.5 petabytes of information that is currently growing and publicly available.

The RTCGA package we are using allows us to download and integrate the variety and volume of TCGA data using patient barcode key that enables for easier data possession. This potentially has a beneficial impact on the development of science and improvement of patients' treatment. Furthermore, RTCGA package transforms TCGA data to tidy form which is convenient to use. The first lines of the code (4-10) install the latest RTCGA packages from Bioconductor needed to do our analysis. Of note, we are using version 3.8 of RTCGA.clinical. Next (lines 12-18), we needed to then install other packages in order to run the survival analysis and plot a more editable Kaplan-Meier plot than what the RTCGA package provides. This includes devtools, dplyr, survival, and survminer.

The next lines of code are used to check and download TCGA datasets. Specifically, the infoTCGA (lines 21-22) function provides a list of available cohorts for statistical analyses along with the relevant information. Additionally, the checkTCGA function (lines 24-26) lets one check the TCGA dataset names for current release date and cohort. COAD is an abbreviation for colon adenocarcinoma, a common cancer type with large datasets which allowed us to run our analysis smoothly. After we checked our data set, we then used the downloadTCGA function (lines 29-30) in order to download the TCGA data from a specified date and from a particular cohort.

We then used the readTCGA (33-39) function that allowed us to read the unzipped downloaded data files that we obtained using downloadTCGA and organized them into a data frame. Other devtools functions (sapply, grep, etc.) are then used to save the data in a more manageable format. After this has been done, a separate data frame can then be created with column names in order to have a searchable list of available information (lines 43-45). We once again used COAD clinical data. After this has been accomplished, we used the survival TCGA function in order to extract survival information from the clinical datasets that were obtained from the TCGA project (lines 49-51). We used gender as an example variable for COAD clinical survival analysis since it is a good binary variable to use and is readily available information in nearly all clinical datasets. We then used our survival packages to fit the data and ggsurvplot in order to create the Kaplan-Meier curves on the data we had obtained.

Although multiple tools and programs already exist to analyze publicly available datasets, developing our own code to bridge this gap tackles two concerns that can arise with data analysis: First, it allows us to supervise and overview every aspect of the analysis to understand and filter the essential parts that best fit the question. Second, this allows us to have a highly customizable tool that can be adapted to different aspects or future questions in a project and thus has the potential of being a very practical tool in the long run [not convinced by the phrasing here]. Implementing the techniques discussed during *Computing Skills for Biologists,* we developed a concise and replicable stream of code that takes advantage of an open source R package focused on TCGA data retrieval. Although this comprehensive package also includes statistical analysis and plotting tools, we determined that it best fits our objectives of data retrieval and ultimately recurred to different plotting and analysis packages available for R. Overall, this conceptually simple question proved to be a challenging task that highlighted the advantage of developing automated analysis with coherent pipelines and will undeniably prove useful in the future.

Work on this project was distributed rather evenly. Jacob created the first draft of the code and then Juan cleaned up the code as well as added better survival analysis packages in order to plot the data. Furthermore, writing this paper was also split in half, with Juan writing the introduction and conclusion and Jacob writing part of the rationale and code explanation. Over the entire project Jacob and Juan talked consistently and worked together to finish the project.