UPPSALA
UNIVERSITET

UPTEC X 22025
Examensarbete 30 hp
Juni 2022

Pipeline for Next Generation Sequencing data of phage displayed libraries to support affinity ligand discovery

Ella Schleimann-Jensen

Civilingenjörsprogrammet i molekylär bioteknik

UPPSALA
UNIVERSITET

Pipeline for Next Generation Sequencing data of phage displayed libraries to support affinity ligand discovery

Ella Schleimann-Jensen

Abstract

Affinity ligands are important molecules used in affinity chromatography for purification of significant substances from complex mixtures. To find affinity ligands specific to important target molecules could be a challenging process. Cytiva uses the powerful phage display technique to find new promising affinity ligands. The phage display technique is a method run in several enrichment cycles. When developing new affinity ligands, a protein scaffold library with a diversity of up to 10^{10}-10^{11} different protein scaffold variants is run through the enrichment cycles.

The result from the phage display rounds is screened for target molecule binding followed by sequencing, usually with one of the conventional screening methods ELISA or Biacore followed by Sanger sequencing. However, the throughput of these analyses are unfortunately very low, often with only a few hundred screened clones. Therefore, Next Generation Sequencing or NGS, has become an increasingly popular screening method for phage display libraries which generates millions of sequences from each phage display round. This creates a need for a robust data analysis pipeline to be able to interpret the large amounts of data.

In this project, a pipeline for analysis of NGS data of phage displayed libraries has been developed at Cytiva. Cytiva uses NGS as one of their screening methods of phage displayed protein libraries because of the high throughput compared to the conventional screening methods. The purpose is to find new affinity ligands for purification of essential substances used in drugs.

The pipeline has been created using the object-oriented programming language R and consists of several analyses covering the most important steps to be able to find promising results from the NGS data. With the developed pipeline the user can analyze the data on both DNA and protein sequence level and per position residue breakdown, as well as filter the data based on specific amino acids and positions. This gives a robust and thorough analysis which can lead to promising results that can be used in the development of novel affinity ligands for future purification products.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala

Handledare: Gustav Myhrinder Ämnesgranskare: Adam Ameur

Examinator: Siv Andersson

Robust analys av data för att främja rening av essentiella läkemedelsämnen

Populärvetenskaplig sammanfattning

Ella Schleimann-Jensen

Läkemedel består av specifika ämnen som hjälper till att bidra till läkemedlets funktion. Utan de essentiella ämnena i ett läkemedel kommer det inte att verka som tänkt. Detta säger någonting om hur viktigt det är att kunna framställa dessa ämnen. Ofta befinner sig de ämnen som behövs i komplexa blandningar med andra ämnen som man inte alls vill ha med i sitt läkemedel. Därför behöver man kunna rena fram de signifikanta ämnena från en sådan typ av blandning. För detta används ofta en teknik kallad affinitetskromatografi. Denna teknik består bland annat av ett långt, vertikalt rör i vilket de viktiga ämnena kan fångas upp medan de övriga ämnena rinner ut. För att kunna möjliggöra att just de viktiga ämnena fångas upp, innehåller röret bland annat så kallade affinitetsligander vilka har affinitet, dvs är specifika, mot det viktiga ämnet vilket gör att de fastnar då de tillsätts i röret.

Att hitta affinitetsligandrar som binder just de ämnen man vill rena fram kan vara en svår process. På Cytiva där de bland annat jobbar med att hitta nya affinitetsligandrar, använder de sig av en specifik teknik som genererar stora mängder data. Datan består av sekvenser av deoxiribonukleinsyra, eller DNA, från olika ämnen man testar som potentiella affinitetsligandrar. DNA är den mest grundläggande byggstenen för allt som finns på jorden, även du är uppbyggd av det! De DNA-sekvenser som datan består av är rätt så korta, men väldigt många vilket gör att det är svårt att bara med ögat säga om det finns någon affinitetsligand som verkar binda till det viktiga ämnet man vill rena fram. Till detta behövs en robust dataanalys som kan leta och filtrera i de stora mängderna av data. Detta är precis vad som har utvecklats i detta projekt.

Jag har utvecklat flera olika dataanalyser som kan användas för att just från dessa stora mängder DNA-sekvenser hitta den affinitetsligand som binder till det viktiga ämnet bäst, och som kan hjälpa till att fånga upp det vid användning av affinitetskromatografi. Resultatet blev flera sammanhängande analyser som från den beskrivna datan kan ge användaren lovande resultat om vilken av de affinitetsligander som testats som binder bäst till det viktiga ämnet. Detta är i form av olika visualiseringar och statistik över de stora datamängderna som användaren sedan kan grunda sina resultat på.

Table of contents

1 Introduction

Affinity ligands are important molecules that can be used to purify substances from complex mixtures using affinity chromatography, for example pharmaceutical substances used in drugs. Finding affinity ligands which bind to the target molecule under the right conditions and have the required specifications can be a challenging process with different approaches.

Phage display is a commonly used technique using protein displaying bacteriophages to find new affinity ligands for a specific target. The phage display technique is run in several cycles where a protein scaffold library with a diversity of up to 10^{10}-10^{11} different protein variants are incubated with the target molecule. The sequences binding the target will be enriched between each round giving the possibility to find binding molecules for a specific target.

To examine the potential binding molecules further, a screening method is needed, where ELISA or Biacore are two common alternatives. The protein sequences of the generated hits against the specific target are then sequenced using Sanger sequencing. The drawback with these conventional screening methods is the low throughput. Only a few hundred potential binding molecules are often screened. During the past few years, NGS or Next Generation Sequencing has increased in popularity as a screening method for phage display libraries, which gives the ability to generate millions of sequences from each phage display round. This is of course revolutionary compared to the throughput of the conventional screening methods, but it also comes with new challenges.

NGS of phage display libraries generates huge amounts of data for which a robust bioinformatic analysis is needed to make sense of the data and find potential binding molecules for the target. That is why the aim of this project is to create an analysis pipeline for NGS data from phage displayed libraries for Cytiva. This is needed because of the lack of such analyses for the huge amounts of data produced when using NGS as a screening method for phage displayed libraries. The pipeline should be able to handle such big amounts of data and provide the possibility to analyze and interpret the data. The pipeline should be somewhat automated and give the user a simple way to analyze and interpret the data based on the purpose of a project using this method.

1.1 Prior work

Some prior analyses fulfilling the same purpose have previously been developed at Cytiva, but still requires a lot of manual work and needs to be further developed. Similar developed data analyses from other studies are quite specific to those projects and cannot be applied directly to the projects at Cytiva. Therefore, a more general method is needed to obtain robust

statistics, visualizations and enable filtering of the huge amounts of data generated when using NGS as a screening method for phage displayed libraries.

2 Background

In this part all relevant background needed to understand the importance and details of the project and the pipeline functions are described.

2.1 Affinity chromatography

Affinity chromatography is a purification method based on specific biological interactions between two molecules. It is one of the most powerful and widely used chromatography techniques for purification of target molecules from complex mixtures (Urh *et al.* 2009, Ayyar *et al.* 2012). To achieve the purification, the stationary phase consists of a matrix and an immobilized binding molecule, called affinity ligand, with the ability to interact exclusively with the target molecule in a reversible binding (Urh *et al.* 2009, Rodriguez *et al.* 2020). A mixture containing the target molecule constitutes the mobile phase (Arora *et al.* 2017). The interactions between the binding molecule and the target molecules are typically reversible to enable both capturing and release during the isolation of the target molecule (Urh *et al.* 2009, Rodriguez *et al.* 2020). When running the mixture through the affinity chromatography column, the target molecules are captured in the column due to the specific biological interactions between the binding and target molecules. Other components of the complex mixture pass through since they are not specific to the affinity ligands in the stationary phase (Arora *et al.* 2017). The capturing of the target molecules in the column enables separation from the mixture, then the target is released, often by a change in pH or salt concentration to break the binding. The target molecule is thereby isolated from the other components in the original mixture.

2.1.1 Affinity ligands

Affinity ligands are proteins that have a specific interaction towards a target molecule and can be used to develop therapeutics and purification methods of therapeutics. Affinity ligands are the binding molecules used in the stationary phase of affinity chromatography (Rodriguez *et al.* 2020). They are the primary component of a successful affinity chromatography (Arora *et al.* 2017) since they are specific to the target molecule based on interactions between the affinity ligand and the target. The ability to purify a target molecule depends on these biological interactions to the affinity ligand, which strengthens the importance of using affinity ligands exclusively binding to the target molecule.

2.2 Phage display

The phage display technology is a powerful tool to find binding molecules, such as affinity ligands, for a specific target (Pande *et al.* 2010). With the help of a highly diverse phage display library running through several enrichment cycles, potential binding molecules to the target molecule can be found. Phage display can be performed under different conditions which can affect the result.

2.2.1 Phage library preparation

The first step of a phage display run is to prepare a phage library. To generate target specific molecules, a protein scaffold is often used as a backbone structure with specific residues being mutated (Ryvkin *et al.* 2018). This generates a combinatorial protein library with a diversity of up to 10^{10}-10^{11} different protein scaffold variants (Ravn *et al.* 2013). The protein scaffolds are co-expressed on the surface of bacteriophages by inserting the protein library with their respective coding DNA sequences into the DNA of bacteriophages. This is done via phagemid vectors, giving a connection between the phenotype and genotype of each scaffold variant in the protein library (Figure 1). The bacteriophages with the co-expressed proteins on their surface constitutes the phage library which then is run through the phage display cycles (Figure 2).

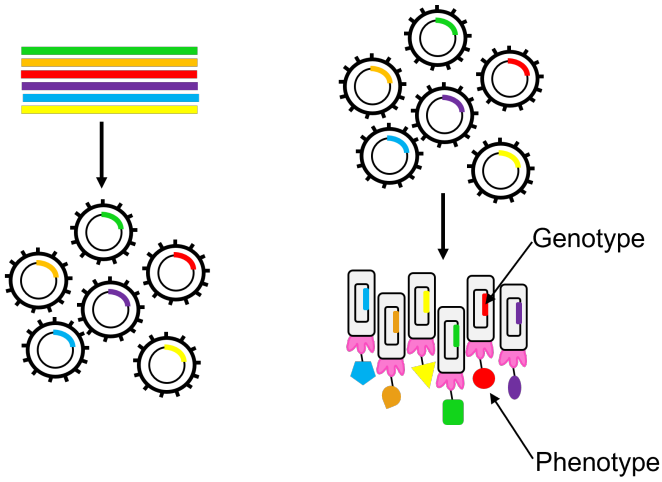

Figure 1: The figure shows how a protein scaffold library is co-expressed on the surface of bacteriophages and the connection between the phenotype and genotype of each protein scaffold variant.

2.2.2 The phage display cycle

The phage display technology is run through several cycles via an enrichment process called bio-panning (Figure 2). The first step is to incubate the prepared phage library with the target molecule. Some bacteriophages will potentially interact with and bind to the target molecule through the protein on its surface, while the vast majority will not. The bacteriophages binding the target are captured while the ones not binding are washed away. The genetic sequences of the binding ligands are extracted, amplified and expressed on the surface of phages again, creating a sub-phage library. This cycle is run several times, often three to five,

causing the best binding ligands to increase for each round. The phage display can be run under different conditions to affect the outcome of the bio-panning cycles and to discover the best possible binding molecules. Examples of different conditions could be to run the phage display with different concentrations of the target molecule, different temperature or pH elution, varying number of washes etc.

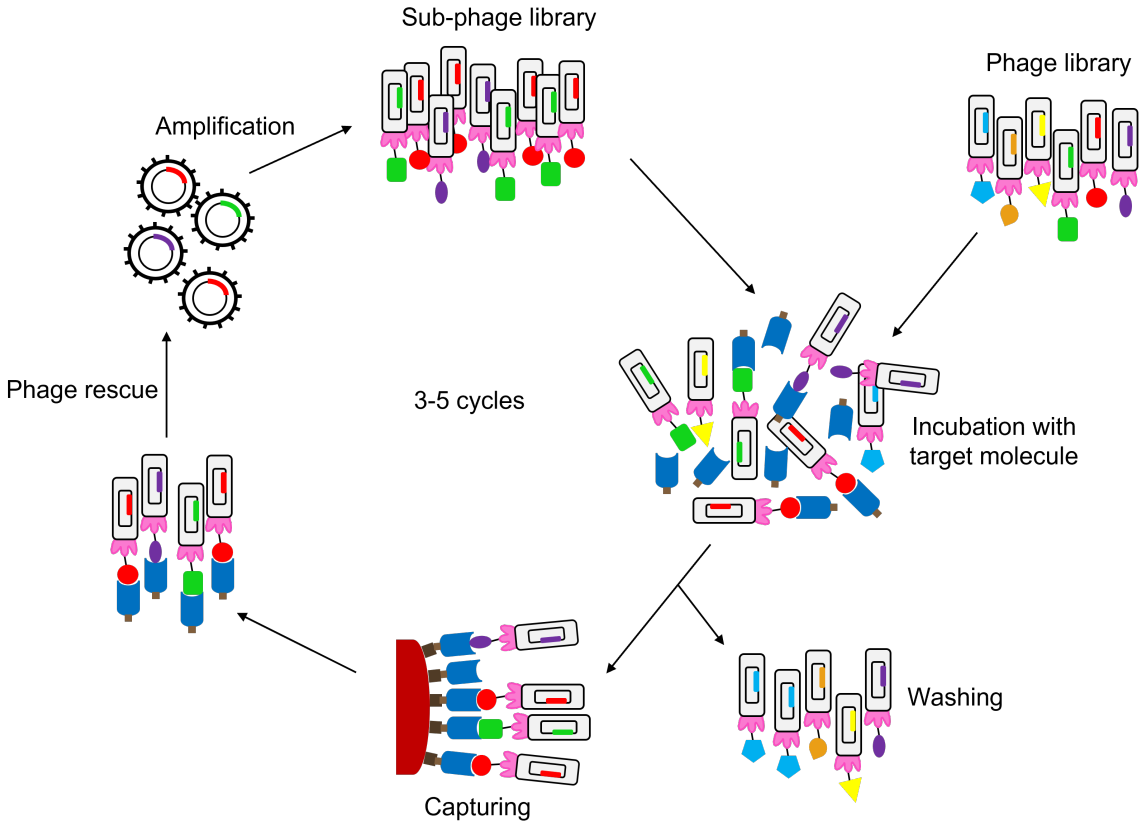

Figure 2: The figure shows the phage display cycle containing incubation, washing, capturing, phage rescue, amplification and the creation of a sub-phage library. In the incubation step the bacteriophages are mixed with the target molecules, the bacteriophages not binding the target are then washed away during the washing while the ones binding the target are captured. The DNA sequences from these clones are extracted and amplified from which a sub-phage library is created, containing the binding clones from the first round. The sub-phage library is then run through the same cycle. This is typically done 3-5 times where the best binding clones are enriched.

2.2.3 Screening

After the enrichment process in the bio-panning cycles, the protein library is normally analyzed using conventional colony screening methods such as ELISA or Biacore assays to investigate potential target binding (Pande *et al.* 2010, Ravn *et al.* 2013). The generated hits against the specific target are then sequenced using Sanger sequencing. A drawback with these conventional colony screening methods is the low throughput, with normally only 100-500 screened variants (Ravn *et al.* 2013), therefore new screening methods such as NGS have been developed. Depending on what screening method that is used the throughput, analysis time, quality and other parameters can differ. The screening occurs for the extracted sequences from each or some of the phage display rounds.

2.3 NGS

NGS or Next Generation Sequencing is a sequencing technique capable of massive parallel sequencing (McCombie *et al.* 2019). The Illumina next generation sequencing is based on the sequencing by synthesis concept and is performed on a flow cell. The samples are prepared by adding adaptors containing indices and regions complementary to oligos on the flow cell, (Illumina 2017). The adaptors are added to the ends of the sequences where all unique samples are provided with unique indices. When samples are added to the flow cell, they bind to the oligos complementary to the adaptors. This enables amplification, creating clusters of each sequence on the flow cell (Figure 3). The sequencing is performed in cycles parallelly for each cluster on the flow cell. Fluorescently tagged nucleotides are added to the flow cell in each cycle, and the complementary base to the current position in the sequence in each cluster is incorporated. This emits a characteristic color for the base and will give knowledge of which base is in each position of the sequence. The sequencing is performed both on forward and reverse strands for each sequence on the flow cell, giving the possibility to pair both strands to obtain contiguous sequences. The technique enables for hundreds of millions of clusters to be sequenced simultaneously in a massively parallel process.

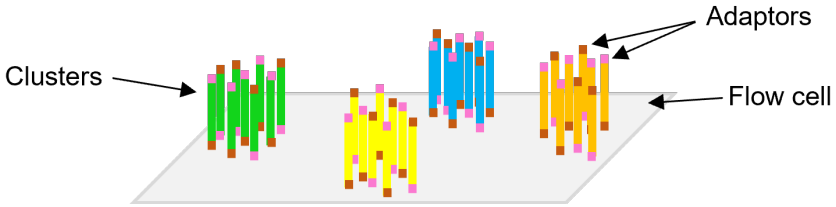

Figure 3: Clusters of sequences on a flow cell during an NGS analysis. In an NGS analysis each sequence is amplified in several steps after they are added to the flow cell, this creates clusters of each unique sequence with both forward and reverse strands. These are the green, yellow, blue and orange clusters in the figure. The red and pink endings of the sequences represent adaptors which are complementary to oligos on the flow cell, to which the sequences bind. After the creation of clusters, the sequences are sequenced in a massively parallel process.

2.4 NGS of phage display libraries

The use of NGS of phage display libraries has become increasingly popular in the recent years (Ravn *et al.* 2013). NGS of phage-displayed libraries results in millions of sequences from each step in the bio-panning process. Therefore, the enrichment of unique library clones can be analyzed throughout the bio-panning, compared to conventional screening methods (Yang *et al.* 2017). The conventional screening methods such as ELISA or Biacore combined with Sanger sequencing, can only analyze and sequence up to a few hundred clones since a lot of manual work is involved (Matochko *et al.* 2012). NGS is a powerful complement to Sanger sequencing to find new interesting findings and to confirm results. Data analysis and bioinformatic studies of the NGS data of phage-displayed libraries are crucial to evaluate and process the large amounts of data and to extract the potential binding molecule sequences.

Thereafter, these NGS-derived binding molecules are evaluated regarding their target molecule binding and protein characteristics.

2.5 R

R is an object-oriented programming language of statistical computing and graphing (R Core Team 2022). It comes with several solutions and packages for bioinformatic analysis and is commonly used in bioinformatics. It has been developed since the late 20th century by the R core Team.

2.5.1 RStudio

RStudio produces free and open-source software, developed for R and other open-source programming languages such as Python (RStudio 2022). It was launched in 2009 when they released their first free and open-source products and has kept on developing until today. RStudio is one of the most frequently used software in combination with R in the world.

2.5.2 R project

An R project is a function developed in R by RStudio (RStudio 2022). It simplifies project work in R by creating a root directory from where the R project user simply can run a project and get input and output files, scripts and raw data arranged into separate folders which simplifies a complex analysis in R.

3 Methods

In this project a bioinformatic analysis pipeline for NGS data of phage display libraries has been developed. The purpose of the pipeline is to help develop new affinity ligands for specific target molecules. The pipeline consists of several different continuous analyses where inputs are based on outputs from previous analyses. By providing several statistical analyses which produces plots and data tables visualizing the data in different levels, the pipeline can help to find new possible affinity ligands.

3.1 Raw data for pipeline

The analysis pipeline was built for handling NGS data of phage displayed libraries. Cytiva uses NGS as a screening method for phage displayed libraries in the process of finding new affinity ligands used in affinity chromatography products. The NGS method produces a huge amount of data in form of DNA sequences collected in one file for each phage display round and track analyzed. The pipeline was developed to take these files in *.csv* format as input after quality control and merging of paired sequence reads, performed separately by the user. The file names must follow the format *optionalname_RoundX_TRACKNAME.csv* to be suitable

for the analyses. “X” is the number of the round and “Trackname” is one capital letter defining the bio-panning track. The pipeline can handle different sizes of raw data files as well as different sequence lengths.

3.1.1 Raw data for report

In this report NGS data from an internal project at Cytiva using a protein scaffold library selected against two different target molecules have been used to show the functionalities of the pipeline. The two different targets are named A and B and constitutes as two different tracks in this report. Target A have been run through two bio-panning rounds and target B through three bio-panning rounds and results from each round will be shown and discussed in this report. The combinatorial protein scaffold library used in this report, for both target A and B consists of seven randomized positions with any of the 20 natural amino acids except cysteine and proline.

3.2 Tools

This bioinformatic analysis pipeline has been developed in RStudio version 4.1.3, (R Core Team 2022). Several different R packages have been used to provide the analyses with efficiency. All used packages for the pipeline can be found in Appendix A (Table A 1). The project function of RStudio have been used as a base for the pipeline to provide the user with arrangement of raw data, outputs and inputs. Each analysis takes different types of *.csv* files produced in earlier analyses as input, as well as RStudio console inputs. Outputs are plots and new *.csv* files.

3.3 Implementation

The different scripts in the pipeline constituting the different analyses have been developed continuously during the whole project. The focus has been on developing one function at a time, but as several functions were developed, previously developed functions could be adapted to provide with efficiency, user-friendliness and a better connection between the different functions.

3.4 Evaluation

Throughout the whole project the pipeline has continuously been evaluated and troubleshooted. This has been done by running scripts and functions on different datasets with different characteristics and prerequisites provided by Cytiva, such as varying sizes of datasets or differing lengths of sequences. This has enabled debugging and to find ways to generalize the functionalities as much as possible against a large amount of variety among datasets suited for the pipeline.

4 Pipeline

The developed pipeline is divided into eleven different scripts and three helping functions where different scripts and functions provide the pipeline with several functionalities to analyze the NGS data from different aspects. This part of the report will describe the pipeline divided into the different functionalities containing one or several scripts or functions. Figure 4 shows the structure of the whole pipeline with all scripts and functions. The flow of all scripts in the form of a pipeline will also be described, as well as the continuous work with user-friendliness and efficiency.

4.1 Developed scripts and functions

4.1.1 Create a project

The whole analysis pipeline is based on the project function in RStudio. To start an analysis, a project folder containing the right subfolders and scripts is required. Along with the pipeline, a script for starting an R project is therefore provided. This script is called *create_project.R* and is run in R. When running the script, the user is asked to enter a project name, location for the project and from where to fetch the pipeline scripts and functions, which is provided to the user in a separate folder.

This creates a new R project folder at the chosen location. The project folder contains subfolders for scripts, output in form of plots and datasets, and raw data. To open the project in an R environment there is an *.Rproj* file created. When opening the *.Rproj* file an R-environment is opened with a defined path to the specific project. All scripts and functions for the pipeline are created to be run inside an R project environment. The project function provides with a better arrangement of the scripts, raw data and output files.

4.1.2 Initial analysis

The initial analysis constitutes the beginning of the pipeline and is done by running the script *seq_occurrences_sort_aa.R*. Each of the other analyses is also based on the output files from this analysis. It was developed to take raw data as input which it fetches from the raw data folder in the project. The raw data files should be *.csv* files containing already quality controlled DNA sequences from Illumina NGS runs of phage displayed libraries. The initial analysis uses one of the developed helping functions, *DNA_to_aa.R* retrieved from the script subfolder in the project, to translate the DNA sequences in each raw data file into protein sequences. This translating function uses a nested command containing the DNAStringSet and Translate functions provided by the R package Biostrings (Pagès *et al.* 2022), available in the software Bioconductor. Before this function is run, the user specifies the reading frame of the protein.

The script then counts the occurrence of each unique protein sequence and orders the sequences according to the occurrence. The unique protein sequences, original DNA

sequences as well as the occurrence of each sequence is stored in a new, ordered dataset. If several DNA sequences translate into the same protein sequence, the most occurring DNA sequence is saved for that protein, but also the number of DNA sequences translating into this protein.

To be able to backtrack sequences through different pipeline output results, the initial analysis also provides each unique protein sequence of all datasets in a whole project with a unique ID number. This is done by using an additional helping function, *seq_ID.R*. This function merges all protein sequences from each file into one dataset and gives each unique sequence a unique ID number. These IDs are saved in the new sorted datasets for each sequence in each file.

At last, the initial analysis calculates the normalized value of each protein sequence occurrence by dividing the occurrence by the total number of sequences in each dataset. This is to provide comparable numbers if datasets are of different sizes. From each raw data file an output file containing the unique IDs, protein sequences, DNA sequences, number of DNA sequence per protein sequence, the occurrence as raw number and normalized number, is produced. These are saved in *.csv* format in the output folder and sorted by the occurrence of protein sequences. All statistics and analyzes in the other developed functions of the pipeline are based on these output files.

4.1.3 Sequence diversity

The next functionality of the pipeline is developed to get a visual representation of the outcome from the phage display rounds and the difference between the datasets regarding diversity. To do this, a simple analysis over sequence diversity has been implemented, *seq_diversity.R*. This script takes one or several sorted files produced in the initial analysis as input which can originate from different phage display rounds and tracks. The script orders the input files according to track name extracted from the file names. To estimate the sequence diversity, it calculates the number of unique sequences in each input file. The number of unique sequences in each file is divided by the total number of sequences to get the proportion of unique sequences. These values are plotted as line graphs between different rounds, showing rounds on the X-axis and percentage unique sequences on the Y-axis. If several tracks are present, these will be separated by different colors and specified by track name in the legend (Figure 5). The plot is created by using the R package ggplot2, (Hadley Wickham 2016), inside the Tidyverse package collection, and is saved for the user to further examine.

4.1.4 Top sequences

The pipeline provides two ways to analyze top sequences from the datasets based on two different approaches. The first approach is developed to analyze the most occurring protein sequences using the script *top_seq.R*. It takes several inputs starting with a sorted file from the initial analysis from which to extract a specified number of top occurrence sequences, which is also specified by the user. Several sorted files from the project are then chosen by the user in which the extracted top sequences are searched for. The number of occurrences of each top

sequence in each input file as well as unique sequence IDs are saved in a matrix which can be saved as a *.csv* file if preferred by the user. A plot showing the occurrence of the top occurrence sequences in the different datasets is also produced if preferred by the user (Figure 6, Figure 7). This plot is also created by using the R package ggplot2.

The second approach of finding top sequences in the pipeline is based on factor rise of sequences between two phage display rounds. This analysis is developed in the script *top_seq_factor_rise.R*. As well as the first top sequence analysis it takes one sorted file produced in the initial analysis as input and a specified, quite high number of top occurrence sequences are extracted. The reason for choosing a high number of sequences in this step is to find potential sequences with high factor rise hidden far down the top occurrences list. The next input to this script is another sorted file from the initial analysis for comparison to the first input file. The extracted sequences are searched for in the comparison file and the factor rise of these are calculated between the two chosen phage display round datasets. The sequences are then sorted by factor rise instead of occurrence, and a specified number of top factor rise sequences are extracted. These can be saved as a *.csv* file along with the factor rise for each sequence between the two phage display rounds and unique sequence IDs for further analyze if preferred by the user.

4.1.5 Per position analysis

The pipeline also provides the user with an analysis per sequence position, i.e., on individual amino acid level. Already in the initial analysis the input DNA sequences are translated into protein sequences using the separate developed translating function *DNA_to_aa.R*. These sequences are saved in the sorted files produced in the initial analysis. The per position analysis is quite time consuming for large datasets, so the pipeline provides two ways to run this analysis. Either to queue several datasets and enable the user to run the analysis in the background, or to run one dataset at a time.

To queue several datasets, the script *mother_script_aa.R* can be used. This combines two scripts and enables the user to run several datasets through these two after each other by collecting all user input at the beginning of the script. The datasets are first run through the script *aa_occurrences.R* which produces a matrix containing the occurrence of each amino acid in each position among all sequences in a dataset. The script collects the protein sequences from the input file of sorted format and splits each protein sequence by position. In a nested loop that goes through each position in each sequence, the matrix is appended with the amino acid occurrence in each position. This script also calculates the percentage of frameshift in the dataset using two different approaches. One approach estimates the fraction of stop codons in each position which is an indication of frameshifts. The other approach estimates the fraction of "incorrect" amino acids in each framework position which also could be an indication of frameshift. These numbers are given as an output in a table directly in the R console for the user to reflect on. The produced matrix is saved as a *.csv* file. If this script is run individually only one dataset can be run at a time.

Thereafter the mother script will use the produced matrix to run the script *aa_histogram.R*. The histogram script produces a histogram based on the amino acid distribution in the sequences (Figure 8A, Figure 8B, Figure 9A, Figure 9B, Figure 9C). For this script, the user chooses what positions that should be included in the histogram, preferably the varying positions in the protein scaffold library. Each input file run through the mother script will output a histogram, based on normalized occurrence values of each amino acid in each chosen position. If the histogram is run separately, the matrix saved from the script *aa_occurrences.R* is used as input file and the user is then also able to choose a title for the histogram.

The last part of the per position analysis, the script *aa_heatmap.R*, is not included in the mother script. This script takes the matrix produced in the script *aa_occurrences.R* from two different phage display rounds or tracks as input. The script then calculates the factor rise of each amino acid in each sequence position and plots it in a heatmap showing if the occurrence of each amino acid in each chosen sequence position has increased or decreased between the two phage display rounds or tracks (Figure 10, Figure 11). As for the histogram script, the occurrence values are normalized before the factor rise is calculated to get accurate results independent of the dataset size.

4.1.6 Filtering

The pipeline also provides a filtering function to enable a deeper analysis of the datasets, *filter_seq.R*. This comes in handy when using the NGS analysis in synergy with Sanger sequencing where the user may have recognized sequence motifs to remove. This could be motifs of affinity ligands with unspecific interactions to the target molecule. The script lets the user choose in what positions and what amino acids in those positions to filter and creates two new files containing the extracted sequences with the specified amino acids in the specific positions and one containing the filtered original dataset. The input is one sorted file from the initial analysis in which the script catches the column containing protein sequences and splits the amino acids one by one. A new matrix is created which connects each amino acid in each position to the ID of the actual protein sequence. From this matrix the unique IDs of each sequence containing the chosen amino acids in the specified positions are collected. By using these IDs, a subset from the original input file with the sequences containing the specified positions and amino acids can be produced as well as a subset without these sequences. The two subsets are saved as *.csv* files for the user to perform further analyses on.

4.1.7 Exporting sequences

After performing a full analysis, the user may have found some interesting results of sequences that seems promising. The last step in the analysis pipeline is therefore an exporting function, *export_seq.R*, to which the user can enter unique sequence IDs and export these sequences into a new *.csv* file. The script takes one or several sorted files from the initial analysis as input as well as either a list of IDs or IDs directly written in the RStudio console. If there are several input files these are merged into one big file. The protein and DNA sequence and the unique ID of the first occurrence of each chosen ID is extracted into a new

file which is saved as a *.csv*. This file can be used to summarize the NGS analysis with the chosen and extracted sequences of interest. These sequences should be further evaluated in the wet lab for their target molecule binding properties and protein characteristics.

4.2 Pipeline flow description

One of the purposes with this bioinformatic analysis pipeline was that it was supposed to be automated for the most part. This was somehow fulfilled by successive scripts with inputs dependent on previous script outputs and few user inputs. The pipeline can though be run based on different aspects depending on what the purpose of the analysis is. Therefore, the pipeline is quite adaptable depending on the user needs.

Figure 4 shows the pipeline flow including helping functions and each script input and output. The scripts are in bold inside green boxes with an *.R* ending, and the functions are in yellow boxes with an *.R* ending. The inputs are shown in pink boxes and the outputs in blue boxes.

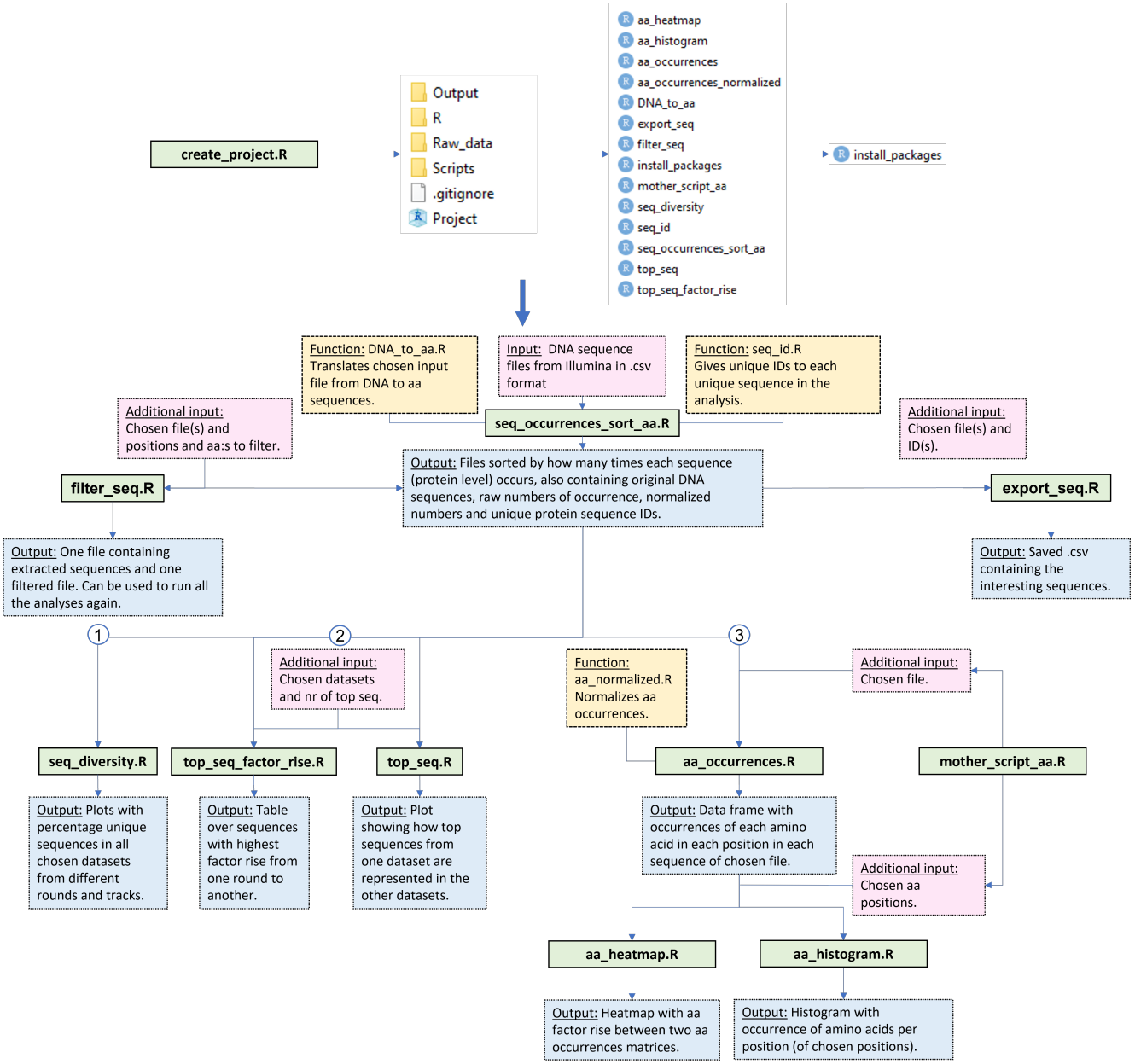

Figure 4: Each developed script and function for the pipeline shown as a script tree including all inputs and outputs. The scripts are shown in bold inside green boxes with an *.R* ending, and the functions are shown in yellow boxes with an *.R* ending. The inputs are shown in pink boxes and the outputs are shown in blue boxes.

The user first needs to create a project folder which is done by running the script *create_project.R* (Figure 4). All pipeline scripts will then be in the "Scripts" folder inside of the project folder and the raw data in the "Raw data" folder. Before starting the analysis, the user can also run the provided script *install_packages.R* to get access to all necessary packages for the whole pipeline. Then the analysis can be started inside of the R project environment.

Each of the different tracks in the analysis is dependent on the initial analysis, shown as *seq_occurrences_sort_aa.R* in Figure 4, in which each input dataset is summarized depending

on the sequence distribution. The files created in this script are used for any future analysis in the pipeline. The user can then choose to analyze this data according to three different aspects.

The sequence diversity analysis, *seq_diversity.R*, script branch 1 in Figure 4, is the simplest functionality of the pipeline which can give the user a visual interpretation of the outcome from the phage display rounds. The output from this analysis is for the user to further analyze and get new insights about datasets from the phage display rounds.

The top sequence aspect is another script branch to follow in the pipeline to get some results of promising sequences, see script branch 2 in Figure 4. By analyzing top sequences from different point of views, either by occurrences or enrichment factor increase, different sequences can be found.

The per position analysis is a way of breaking down the data and analyze the sequences on another level, see script branch 3 in Figure 4. This is the third track in the pipeline also following the initial analysis and can give the user an idea of how the distribution of amino acid changes over phage display rounds by outputting histograms and heatmaps over distribution and factor rise.

All results from the whole pipeline can be combined with results from other sequencing methods to get new insights about the data using *filter_seq.R* (Figure 4). The filtering functionality of the pipeline comes in handy here since discoveries from other results may bring the necessity to remove sequences. For example, affinity ligands that bind the target molecule at an unwanted epitope or sticky affinity ligands with unspecific interactions. The filtering function creates filtered, sorted files that can be run through the whole analysis again to make new findings. This together with the exporting functionality in which the user can export the most promising sequences at the end of an analysis, binds the whole pipeline together, and enables an iterative analyzing process of the NGS data, see *export_seq.R* in Figure 4.

4.3 Efficiency and user-friendliness

The efficiency of the pipeline has been considered throughout the whole developing process. It has been enabled by creating helping functions for calculations or functionalities that is used several times through the whole pipeline as well as reusing created variables and files in several scripts. The initial analysis and the per position analysis can though be quite time consuming for large datasets and if a project contains a big number of datasets. This has been handled by viewing the script progress in the R console when running the scripts and providing a way to run the per position analysis on several datasets without the need of any user input or actions in between. This enables the possibility to run the analysis in the background with no hands-on time, for example over night. The initial analysis is automatically run on all raw data files provided for the whole project.

All these functionalities contribute to the user-friendliness of the pipeline. There are also many other factors contributing. All user inputs given through the whole pipeline are given in the same way which reduces the learning curve of the pipeline for the user. It is always clear what kind of input file that is required when choosing files for scripts. The whole pipeline is also simply described in a user instruction document that will be provided to the user along with the needed scripts and functions. Inside all scripts there are descriptions of what each section is doing to enhance the understanding of the process for the user.

5 Results

This part will show and describe all pipeline output based on the raw data described in section 3.1.1.

5.1 Sequence diversity

As described in section 4.1.3, the sequence diversity functionality is developed for the user to get a visual representation of the dataset in regard to the number of unique sequences from the phage display cycle. Figure 5 shows a plot created with the script *seq_diversity.R*, with the data described in section 3.1.1. The graph shows that the diversity among the sequences decreases for both target A and B between phage display round 1, 2 and 3.

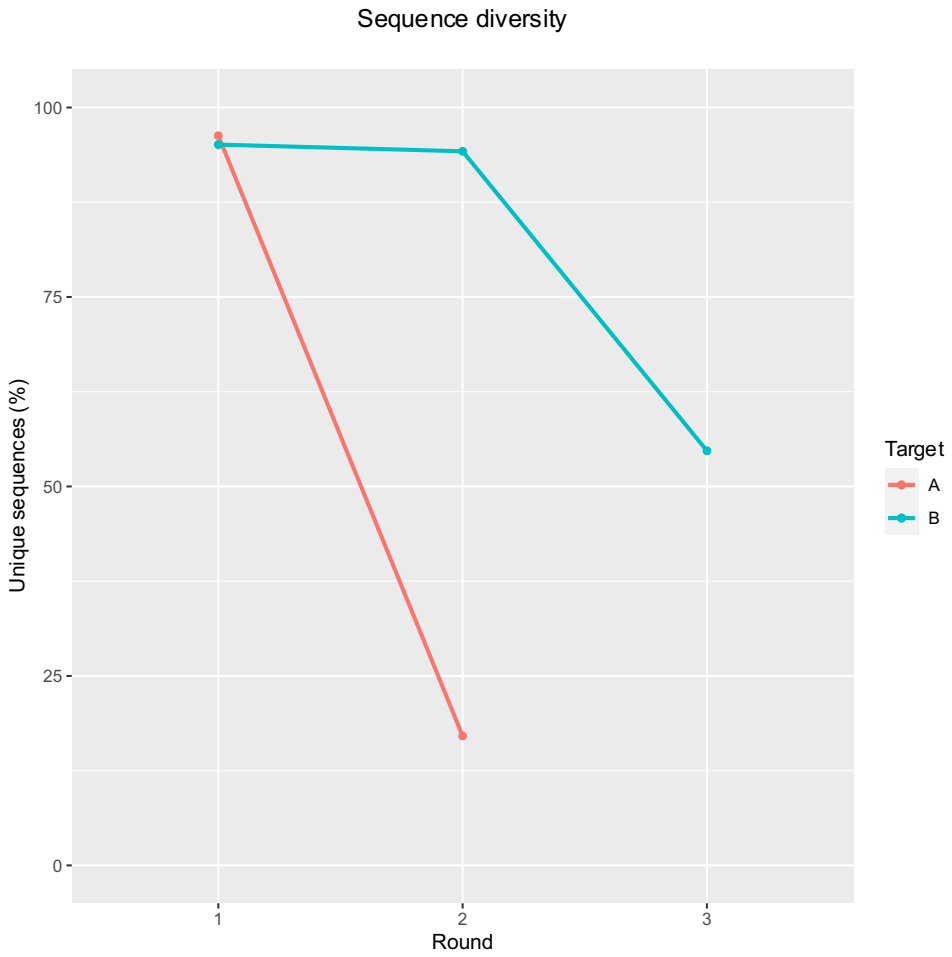

Figure 5: The sequence diversity for target molecule A and B over phage display round 1, 2 and 3. The phage display rounds are shown on the X-axis and the percentage unique sequences are shown on the Y-axis. Target A was run in two rounds and target B in three rounds.

5.2 Top sequences

The pipeline provides two different ways to analyze top sequences of the datasets. One way is to look at most occurring sequences from one phage display round and in plot and table form examine how these sequences occur from this round to other phage display rounds. Figure 6 shows how the ten most occurring sequences from phage display round 2 for target A increases from round 1 to 2. Figure 7 shows the how the ten most occurring sequences from phage display round 3 for target B increases from round 1, to 2 to 3.

The other aspect of examining top sequences with this pipeline is to look at highest factor increase between rounds. This function outputs a table containing information about the sequences with highest factor increase. The thousand most occurrent sequences from phage display round 2 for target A and phage display round 3 for target B were extracted and factor increase were calculated in regard to round 1 for target A and to round 2 for target B, to show the output of this function (Table 1, Table 2).

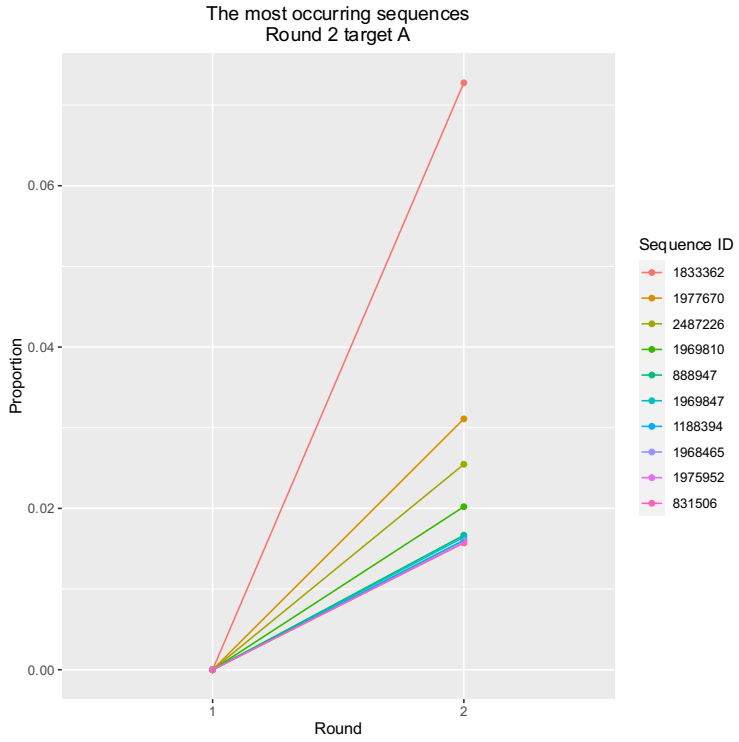

Figure 6: The increase between phage display round 1 and 2 of the 10 most occurring sequences in round 2 for target A. The X-axis shows the rounds, and the Y-axis shows the proportion of occurrence in the dataset. The unique protein IDs of the top sequences are related to the graph color in the legend.

Table 1: The top ten sequences from phage display round 2 with highest factor rise between round 1 and 2 for target A. The last number of the reference sequence ID shows the location based on occurrence in the original sorted dataset.

Unique sequence ID	Reference sequence ID	Factor rise between round 1 and 2
1833362	TargetA_Round2_A_1	18221.45
1977670	TargetA_Round2_A_2	15575.40
2487226	TargetA_Round2_A_3	12756.42
1969810	TargetA_Round2_A_4	10123.36
888947	TargetA_Round2_A_5	8352.00
1969847	TargetA_Round2_A_6	8231.04
1188394	TargetA_Round2_A_7	8041.11
1968465	TargetA_Round2_A_8	8025.12
1975952	TargetA_Round2_A_9	7893.16
831506	TargetA_Round2_A_10	7871.17

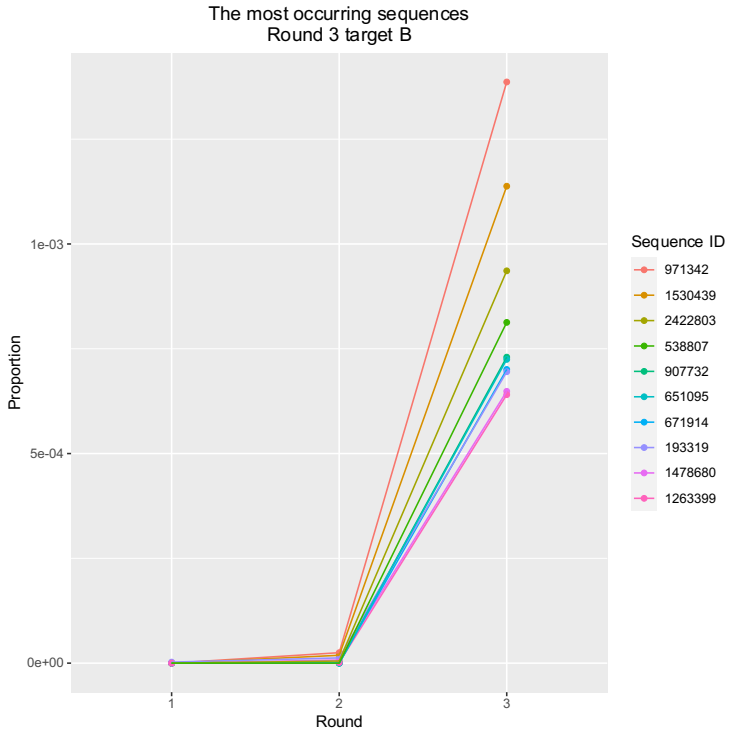

Figure 7: The increase between phage display round 1, 2 and 3 of the 10 most occurring sequences in round 3 for target B. The X-axis shows the rounds, and the Y-axis shows the proportion of occurrence in the dataset. The unique protein IDs of the top sequences are related to the graph color in the legend.

Table 2: The top ten sequences from phage display round 3 with highest factor rise between round 2 and 3 for target B. The last number of the reference sequence ID shows the location based on occurrence in the original sorted dataset.

Unique sequence ID	Reference sequence ID	Factor rise between round 2 and 3
538807	TargetB_Round3_B_4	651.54
907732	TargetB_Round3_B_5	585.10
651095	TargetB_Round3_B_6	580.81
671914	TargetB_Round3_B_7	561.53
1263399	TargetB_Round3_B_10	513.30
1856918	TargetB_Round3_B_12	468.30
1594669	TargetB_Round3_B_13	451.15
1782278	TargetB_Round3_B_14	440.43
1944575	TargetB_Round3_B_15	438.29
2176146	TargetB_Round3_B_18	431.86

5.3 Per position analysis

The per position analysis can produce both histograms of the amino acid distribution in chosen positions of one dataset, as well as heatmaps comparing the amino acid distribution in chosen positions between two datasets. This part shows the amino acid distribution in form of both histograms and heatmaps for target A and B.

5.3.1 Histograms

The histograms produced in the per position analysis of the pipeline shows the distribution of the amino acids in the seven varying positions for the protein scaffold library used in the phage display procedure. Figure 8 shows the histograms for target A, where figure 8A and 8B shows the histograms for round 1 and round 2 respectively. Figure 9 shows the histograms for target B, where figure 9A, 9B and 9C shows the histograms for round 1, round 2 and round 3 respectively.

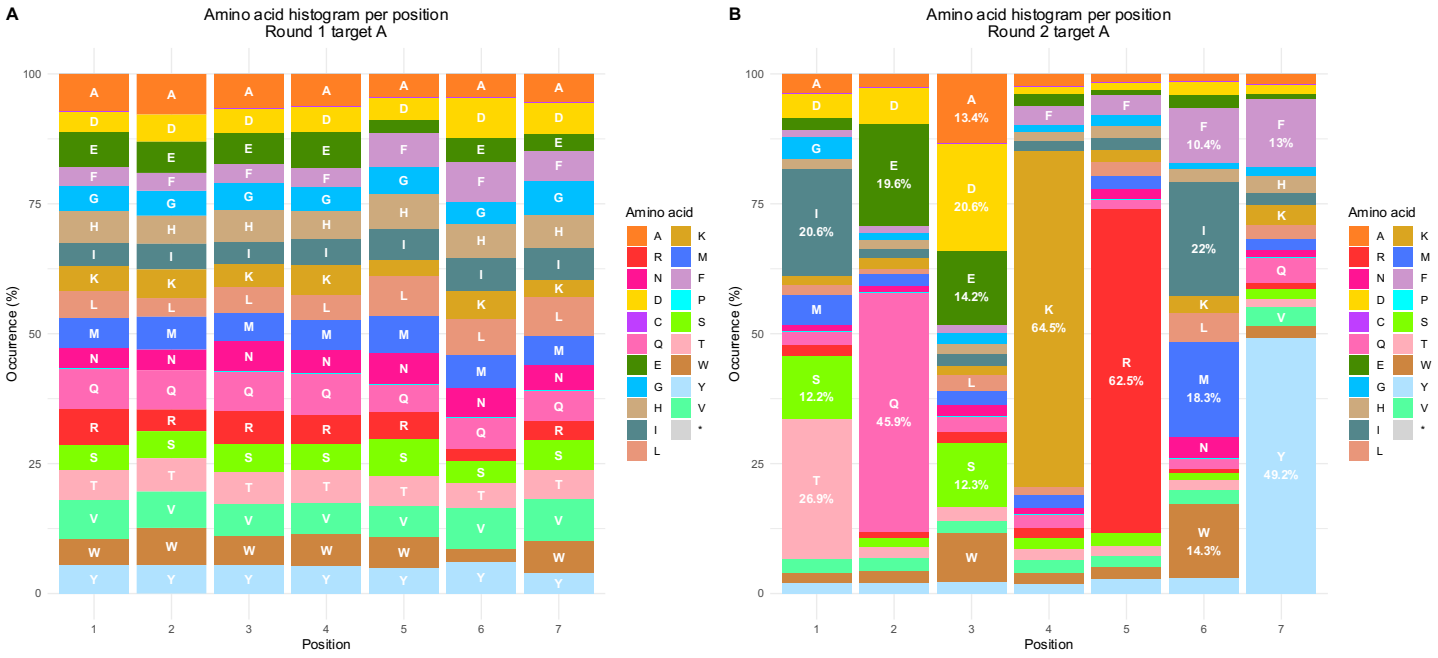

Figure 8: Histograms showing the amino acid distribution for the seven varying positions in the protein library from phage display round 1 and 2 for target A. Figure 8A shows the histogram for round 1 and figure 8B shows the histogram for round 2. The varying positions are shown on the X-axis and the occurrence in percent is shown on the Y-axis. The color for each amino acid can be seen in the legend.

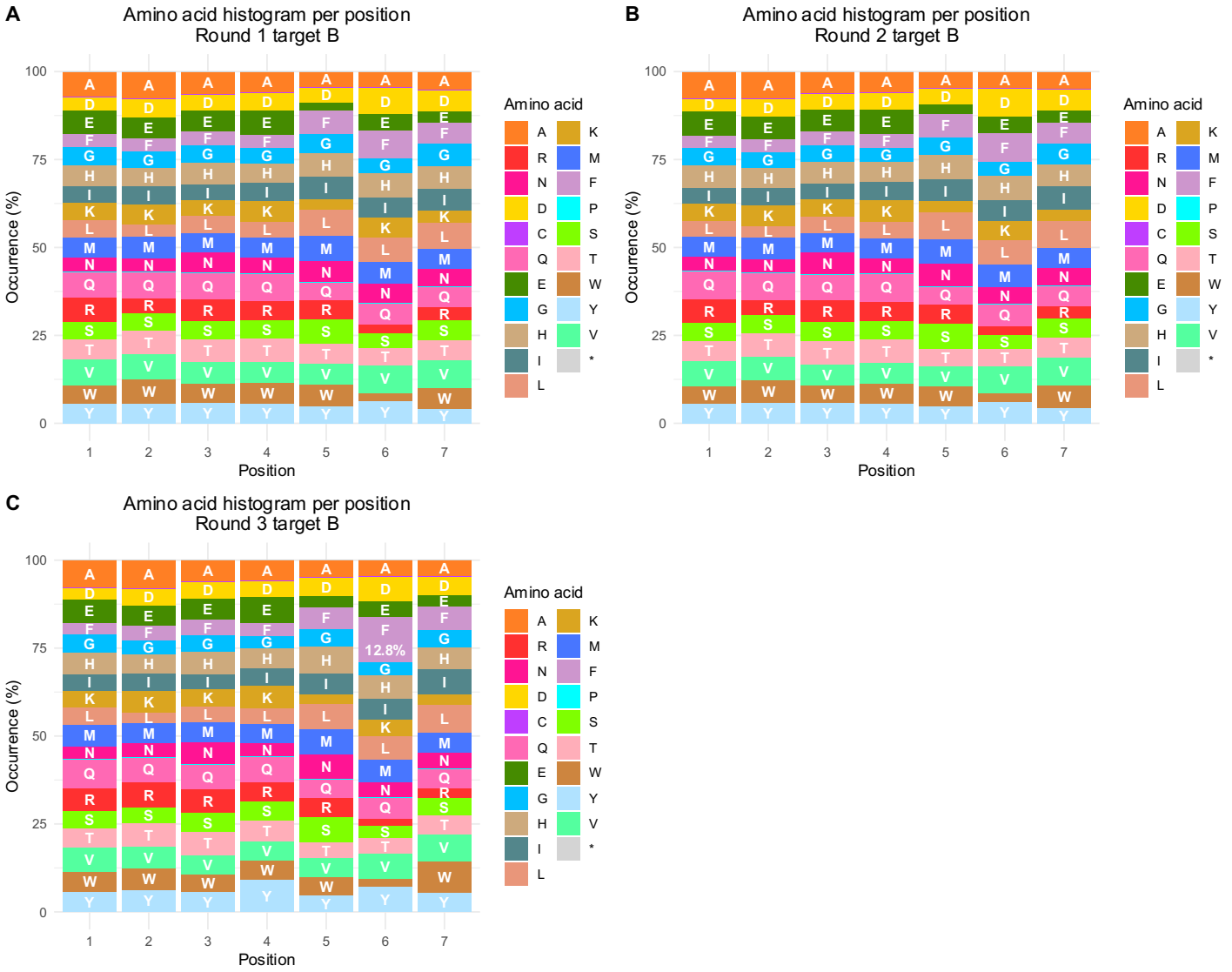

Figure 9: Histograms showing the amino acid distribution for the seven varying positions in the protein library from phage display round 1, 2 and 3 for target B. Figure 9A shows the histogram for round 1, figure 9B shows the histogram for round 2 and figure 9C shows the histogram for round 3. The varying positions are shown on the X-axis and the occurrence in percent is shown on the Y-axis. The color for each amino acid can be seen in the legend.

5.3.2 Heatmaps

The heatmaps produced in the pipeline shows the factor increase between phage display round 1 and 2 of all allowed amino acids in each of the seven varying positions in the protein library for target A and the factor increase between round 2 and 3 for target B. Figure 10 and Figure 11 shows the heatmaps for target A and target B respectively.

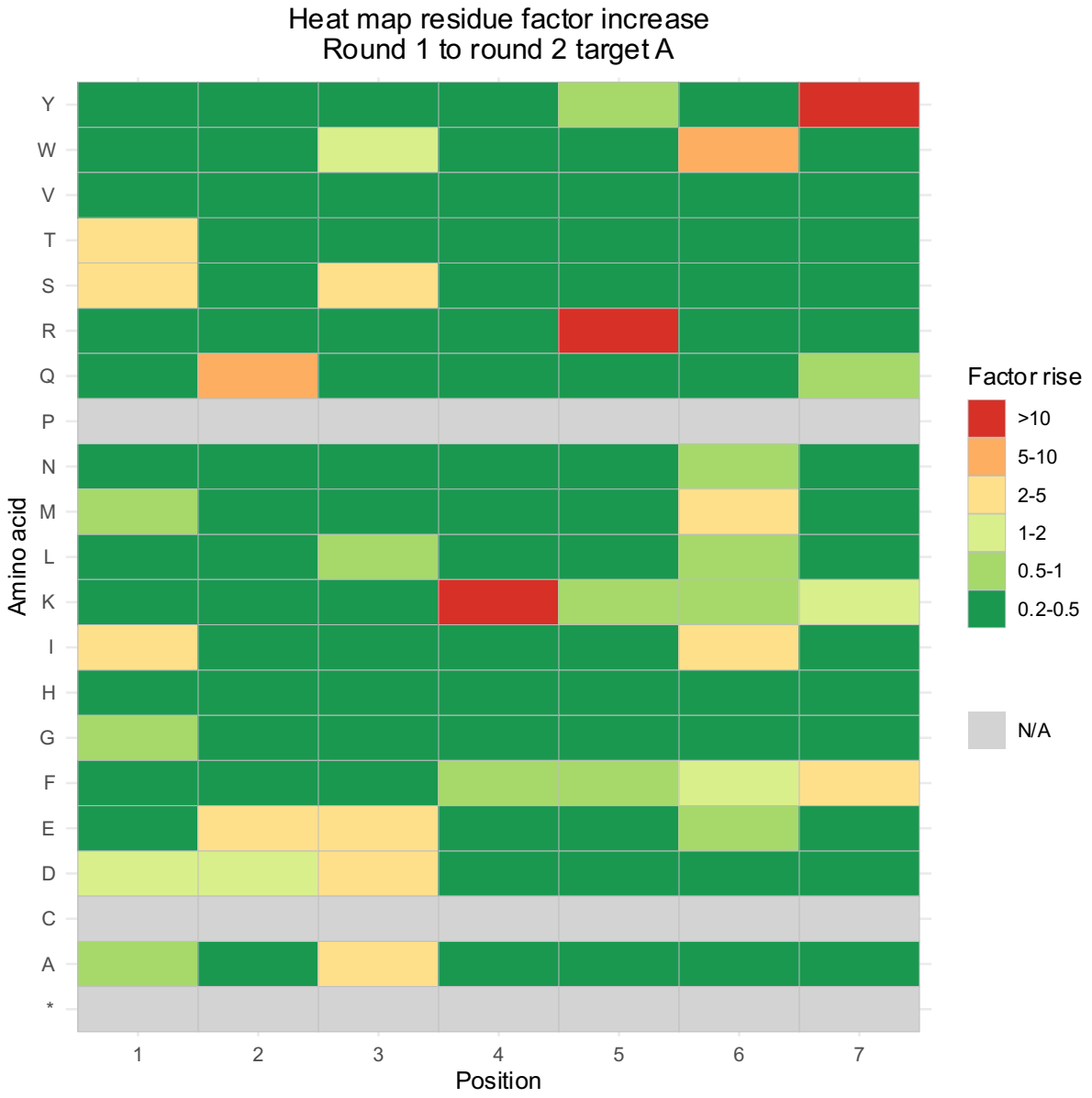

Figure 10: Heatmap showing the factor increase between phage display round 1 and 2 of the seven varying positions in the protein library for target A. The plot shows varying positions in the protein library on the X-axis and amino acids on the Y-axis. Amino acids with N/A values are not allowed in the varying positions in the protein scaffold library design.

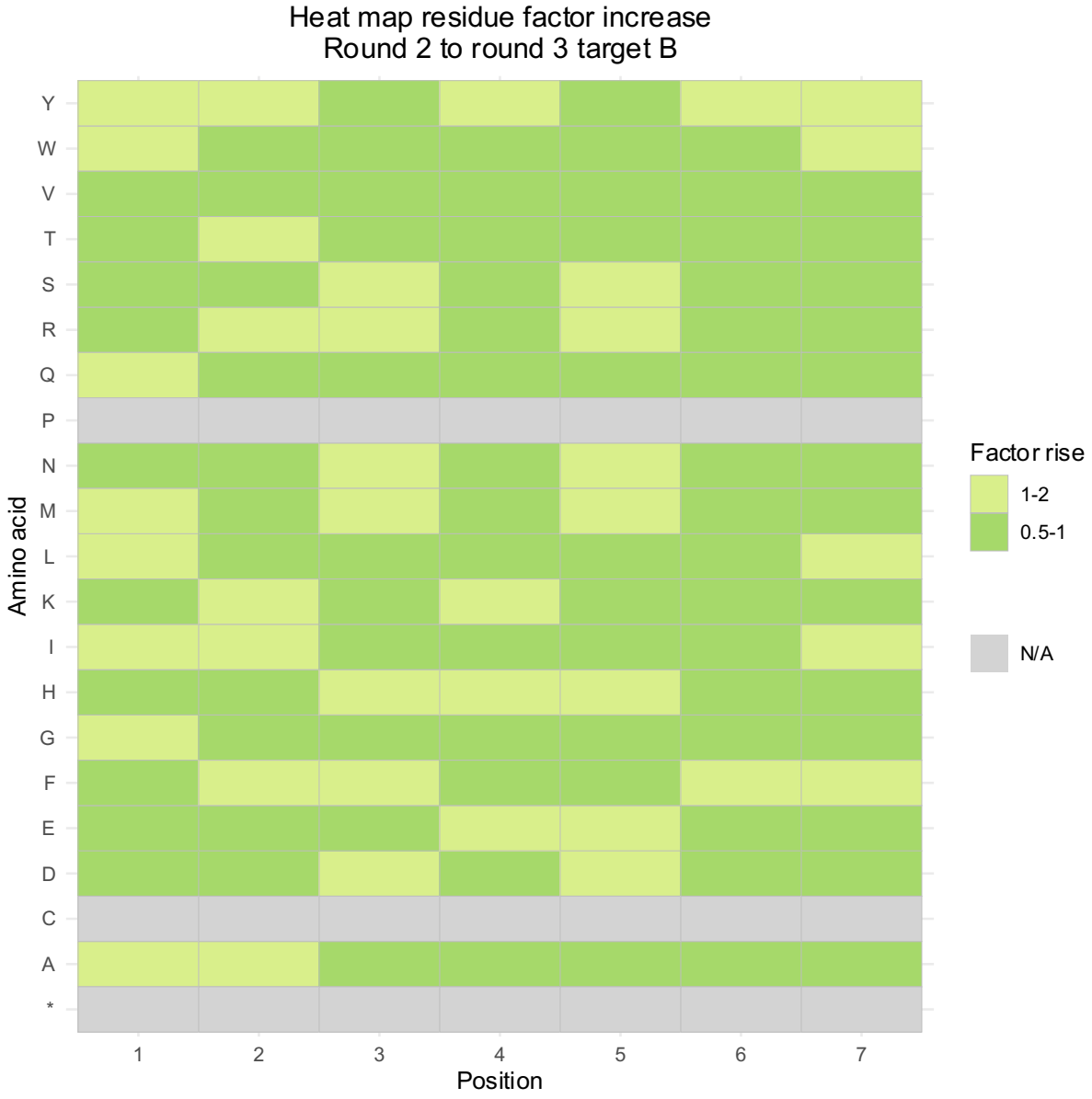

Figure 11: Heatmap showing the factor increase between phage display round 2 and 3 of the seven varying positions in the protein library for target B. The plot shows varying positions in the protein library on the X-axis and amino acids on the Y-axis. Amino acids with N/A values are not allowed in the varying positions in the protein scaffold library design.

5.4 Filtering

One filtering was made on the dataset from phage display round 2 of target A. The sequences containing the amino acids Y (tyrosine) and F (phenylalanine) in the varying position 7 in the protein library were extracted in the filtering. The filtered dataset contains 84% of the sequences from the original dataset and 16% of the sequences were filtered away. Figure 12 shows a histogram with the amino acid distribution in the filtered dataset without the extracted sequences.

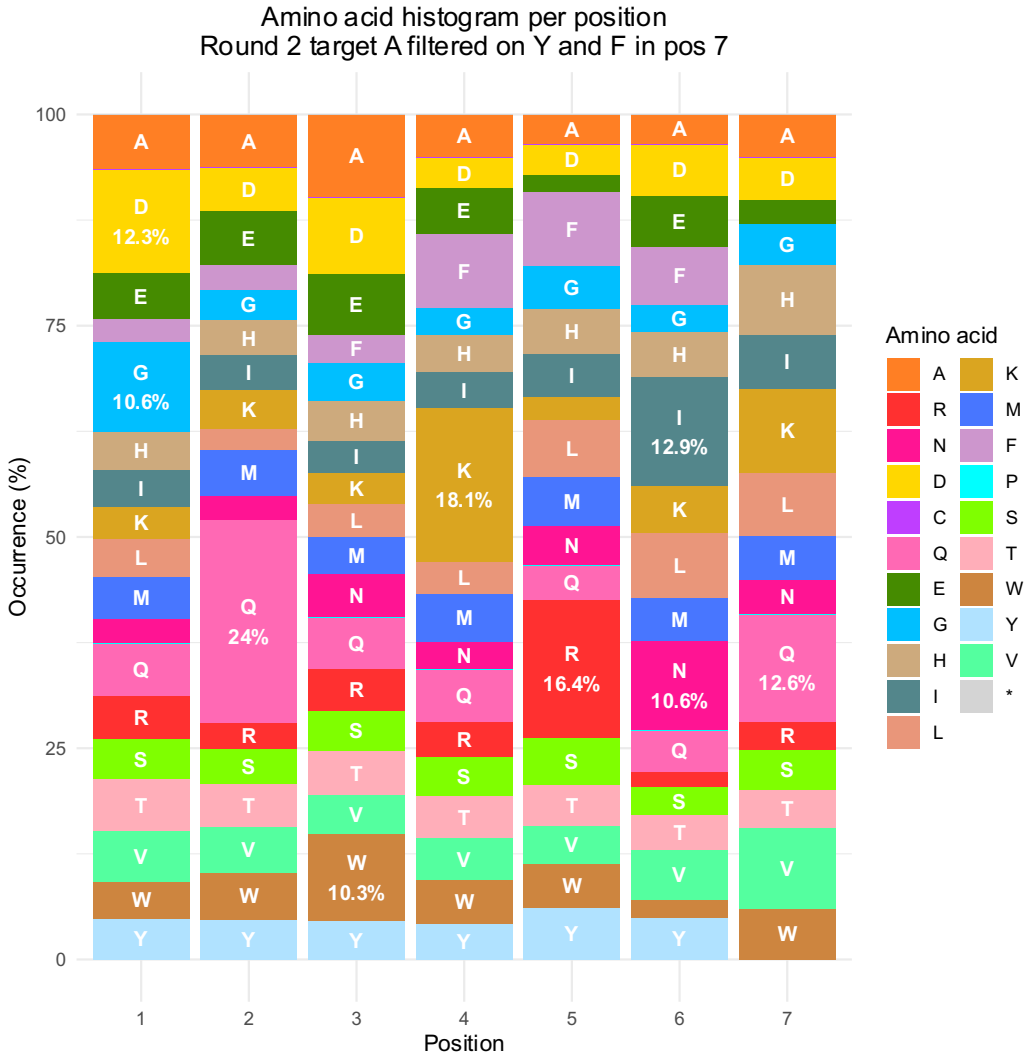

Figure 12: Histogram showing the amino acid distribution for the seven varying positions in a filtered protein library from phage display round 2 for target A. The filtering was made for the amino acids Y and F in varying position 7. The varying positions are shown on the X-axis and the occurrence in percent is shown on the Y-axis. The color for each amino acid can be seen in the legend.

6 Discussion

This pipeline contains several useful tools for analyzing NGS data from phage displayed libraries. It covers the most important steps and functionalities to be able to find interesting results from an NGS run.

The first, crucial step is to create the sorted files. This script was developed to sort the output files on protein sequence occurrences and not DNA sequence occurrences, which is advantageous since there may be several DNA sequences translating into the same protein sequence. The output from this script is also useful to get a grip of the data and to how it behaves. The sequence diversity functionality is a way to visualize the behavior of the data over several phage display rounds and tracks. You would expect the sequence diversity to decrease over rounds if the phage display were successful since the best binding affinity ligands will be enriched over the phage display rounds. If this is not the case, it gives an opportunity to reflect on what could have gone wrong, or if any of the known prerequisites led to that result. Figure 5 shows the sequence diversity plot for target A and target B over phage display round 1, 2 and 3. Both targets follow the expected downward trend, with especially a significant decrease for target A. Target B only decreases slightly between phage display round 1 and 2, but significantly more between round 2 and 3. This functionality only gives an indication that sequences have been enriched during the phage display cycles, but does not tell the user anything about what potential sequences that bind the target. For this purpose, the top sequence has been developed.

The most intuitive way to look at top sequences of an NGS dataset from a phage display library is to look at the most occurring sequences, since the sequences you would expect to bind the target the best are the most occurring in the later phage display rounds. Therefore, the first aspect of the top sequence functionality is developed, since this can say a lot about the potentially best binding sequences. But the best binding sequences does not have to be the most occurring sequences, since the sequence occurrence can vary from start. The second aspect of the top sequence functionality is therefore developed to look at the top factor rise sequences instead. A high factor rise between two rounds is often an even better sign of potential binding sequences, and the sequences with highest factor rise does not necessarily have to be the most occurring sequences. Although this seems to be the case for target A, if comparing the ID numbers in Figure 6 and Table 1 it can be concluded that the top ten most occurring sequences in round 2 are also the top ten highest factor rise sequences. This can also be concluded by examining the reference sequence IDs which represents the original location in the list sorted by top occurrences.

For target B on the other hand, Figure 7 shows that the sequence with ID 971342 is the most occurring sequence in round 3 and it also seems to be the one with highest increase between the two rounds, based on the slope of the graph. But Table 2 containing the top factor rise sequences for target B shows that this is not the case. It shows that the sequence occurring

fourth most, with sequence ID 538807, is the sequence with highest factor rise between round 2 and 3. Table 2 also shows that many other top occurrence sequences have a higher factor rise than the sequence with ID 971342 as well. This is probably because these sequences occurred in a lower amount than the sequence with ID 971342 in phage display round 2. By using the highest factor rise functionality, the sequences with a high factor rise hiding further down the most occurring list could be found. This script is therefore preferably run with a high amount of extracted top sequences to look for in the comparison dataset. When comparing the output from the two top sequence aspects, new insights about the dataset can be realized.

The per position analysis was developed to give the user a way to examine the data on another level and to be able to find amino acids in the varying positions that have been enriched over a big proportion of the sequences. For a naïve library, i.e., the protein scaffold library to be used for a phage display, an even distribution of the allowed amino acids in the varying positions is normally expected. Therefore, it can be advantageous to produce an amino acid distribution histogram over the naïve library to see if the distribution of the amino acids behaves as the theoretical design after preparing a protein scaffold library. Figure 8A and Figure 9A shows the amino acid distribution of phage display round 1 for target A and target B respectively. It can be concluded that the allowed amino acids still are quite evenly distributed in each of the seven varying positions for both targets. In round 2 for target A on the other hand, several positions have diverged against one or two amino acids, while round 2 for target B looks quite similar to round 1.

Figure 8B shows the amino acid distribution of phage display round 2 for target A, where especially position 4, 5 and 7 have diverged against the amino acids K (lysine), R (arginine) and Y (tyrosine) respectively. This could mean that this is an advantageous pattern for the specific binding between the affinity ligand and target A. Figure 9B shows the amino acid distribution of phage display round 2 for target B, which looks quite similar to round 1 of target B which indicates that not much has happened between the rounds. This result can also be confirmed in Figure 5, where it can be seen that the sequence diversity only slightly decreases between phage display round 1 and 2 for target B. Figure 9C shows the amino acid distribution of phage display round 3 for target B where the only difference from round 2 is that F (phenylalanine) in varying position 6 is the only amino acid that has increased slightly, which can be concluded if Figure 9B and Figure 9C is compared. This does not necessarily mean that the phage display was failed but rather that a specific amino acid motif has not been enriched for the target molecule binding.

Figure 10 and Figure 11 are heatmaps showing the increase or decrease of each amino acid in each of the seven varying positions between round 1 and 2 for target A and round 2 and 3 for target B. These can be used as a complement to the histograms, since they clearly show what amino acids that increased and decreased between phage display rounds. The heatmap for target A shows the same results as concluded from the histograms; the most significant

increases of amino acids are K, R and Y in positions 4, 5 and 7 respectively, although it is also seen that quite a few other amino acids also increase in several positions. For target B the heatmap is useful since the histograms are quite vague to compare. In the heatmap it is clearer which amino acids that actually increased between phage display round 2 and 3, even though it was just a slight increase.

The results from the amino acid histograms and heatmaps could profit from comparing with results from conventional screening and sequencing methods, such as ELISA or Biacore followed by Sanger sequencing. If a specific pattern has shown some interesting results using ELISA or Biacore, such as sequences containing this pattern has an unspecific interaction to the target molecule or similar, the sequences containing this pattern can be filtered away using the filtering functionality of the pipeline. Then all analyses can be run again to find new promising sequences including other specific patterns.

Figure 12 shows the histogram of a filtering that has been applied in the varying position 7 of the dataset from phage display round 2 of target A. If this is compared with Figure 8B, which is the original histogram from round 2 of target A, it can be seen that the original clear motif including K, R, Y in the varying positions 4, 5 and 7 respectively, is not as clear after the filtering. When extracting the sequences with the amino acids Y (tyrosine) and F (phenylalanine) in the varying position 7, K and R in position 4 and 5 respectively, are significantly reduced. Even though they are still the amino acids occurring the most in these positions, this proves that these amino acids together are clear motifs in the original dataset. The occurrence of the enriched amino acids in position 1, 2, 3 and 6 were also reduced with this filtering. Some new motifs can also be seen; G (glycine) and D (aspartic acid) in position 1 and N (asparagine) in position 6 increased. In conclusion, the filtering functionality can be used to provide with evidence for new motifs hidden by other enriched motifs, and to conclude if several amino acids in different positions are connected in an important motif.

6.1 Future development

One significant future development with this analysis pipeline is to create a software. This to make it more user-friendly and clearer. Due to time restrictions with this project, this was not implemented but some research in the area was made. One promising suggestion on how to build a software from the developed scripts, is to use R Shiny which is an R package developed by RStudio (RStudio 2022). This package comes with easy solutions for creating apps and software based on programming in R. Since all R scripts for each function of the pipeline already exists, it would not be a very time-consuming process to create this app. The learning curve for the package is probably quite high, but after learning the useful functions, this pipeline could easily be developed into a software using the R Shiny package.

Besides from this, several functionalities could also be developed as an addition to the already existing functionalities. An example of that could be a way to quality control the raw data

inside the pipeline or couple the pipeline to another program that would perform the quality control. Today Cytiva has a well-functioning way to quality control the raw data which is why this has not been prioritized during this project. There are of course many other functionalities that could be added to the pipeline which there is a possibility to do, but the finished pipeline outcome from this project covers quite a good range of analyses to be able to find some good results from NGS data of phage display libraries.

7 Conclusion

To conclude, this project has led to a well-functioning pipeline with analyses covering the whole analysis process of NGS data from phage displayed libraries. It can give results on several levels of the raw data and give the user a good idea of what sequences that potentially bind the target best. A future work could be to develop a software based on this pipeline.

8 Acknowledgement

I would sincerely like to thank my supervisor Gustav Myhrinder at Cytiva which has been a helping hand through the whole project and has always been open for discussions. This has been extremely helpful for me. I also thank the whole group I have been working in at Cytiva for such a warm welcome and inclusion during my whole semester at Cytiva.

Many thanks also go to my subject reader Adam Ameur for helping me with the distribution of the report as well as uncertainties about the whole project.

Thanks to my examinator Siv Andersson and to the course coordinator Lena Henriksson for answering my questions about the course and report in general as well as giving me feedback.

References

Arora S, Saxena V, Ayyar BV. 2017. Affinity chromatography: A versatile technique for antibody purification. Methods 116: 84–94.

Ayyar BV, Arora S, Murphy C, O'Kennedy R. 2012. Affinity chromatography as a tool for antibody purification. Methods 56: 116–129.

Hadley Wickham. 2016. ggplot2: Elegant Graphics for Data Analysis. online 2016: https://ggplot2.tidyverse.org/. Accessed May 6, 2022.

Illumina. 2017. An Introduction to Next-Generation Sequencing Technology.

Matochko WL, Chu K, Jin B, Lee SW, Whitesides GM, Derda R. 2012. Deep sequencing analysis of phage libraries using Illumina platform. Methods 58: 47–55.

McCombie WR, McPherson JD, Mardis ER. 2019. Next-Generation Sequencing Technologies. Cold Spring Harbor Perspectives in Medicine 9: a036798.

Pagès H, Aboyoun P, Gentleman R, DebRoy S. 2022. Biostrings: Efficient manipulation of biological strings. doi 10.18129/B9.bioc.Biostrings.

Pande J, Szewczyk MM, Grover AK. 2010. Phage display: Concept, innovations, applications and future. Biotechnology Advances 28: 849–858.

R Core Team. 2022. R: The R Project for Statistical Computing. The R Foundation. online 2022: https://www.r-project.org/. Accessed May 6, 2022.

Ravn U, Didelot G, Venet S, Ng K-T, Gueneau F, Rousseau F, Calloud S, Kosco-Vilbois M, Fischer N. 2013. Deep sequencing of phage display libraries to support antibody discovery. Methods 60: 99–110.

Rodriguez EL, Poddar S, Iftekhar S, Suh K, Woolfork AG, Ovbude S, Pekarek A, Walters M, Lott S, Hage DS. 2020. Affinity chromatography: A review of trends and developments over the past 50 years. Journal of Chromatography B 1157: 122332.

RStudio. 2022. RStudio | Open source & professional software for data science teams. online 2022: https://www.rstudio.com/. Accessed May 6, 2022.

Ryvkin A, Ashkenazy H, Weiss-Ottolenghi Y, Piller C, Pupko T, Gershoni JM. 2018. Phage display peptide libraries: deviations from randomness and correctives. Nucleic Acids Research 46: e52.

Urh M, Simpson D, Zhao K. 2009. Chapter 26 Affinity Chromatography: General Methods. In: Burgess RR, Deutscher MP (ed.). Methods in Enzymology, pp. 417–438. Academic Press,

Yang W, Yoon A, Lee S, Kim S, Han J, Chung J. 2017. Next-generation sequencing enables the discovery of more diverse positive clones from a phage-displayed antibody library. Experimental & Molecular Medicine 49: e308.

Appendix A – R packages

Table A 1 shows all R packages in R version 4.1.3 used in the analysis pipeline developed in this project. It includes the general functionality of each function, but also the purpose of each package in the pipeline.

Table A 1: All R packages used in the pipeline in R version 4.1.3.

Package	Functionality	Pipeline purpose
Biostrings	Fast manipulation of large biological sequences or sets of sequences.	To translate DNA sequences into protein sequences and to handle the huge amounts of sequences in several ways.
Tidyverse	A collection of R packages designed for data science.	Several different packages from the Tidyverse collection have been used for different purposes in this pipeline.
Ggplot2	A system for creating nicely looking graphics in R. The package is a part of the Tidyverse collection.	The package has been used to plot each graph produced in this pipeline.
Dplyr	A package for data manipulation. Is a part of the Tidyverse collection.	The package has been used to mutate and filter the data based on specific variables in several of the pipeline functions.
Tibble	A package to create effective and simple data frames in R. Is a part of the Tidyverse collection.	The package has been used to create several tables printed in the R console for some of the developed pipeline functionalities.
Stringr	A package to provide fast common string manipulations. Is a part of the Tidyverse collection.	The package has been used to manipulate sequences in string format and create subsets of strings in the pipeline.
Forcats	A package for handling factors in R which is used to handle categorical variables. Is a part of the Tidyverse collection.	The package has been used to handle categorical variables in different datasets handled in the pipeline.

Data.table	A package used for data manipulation such as creating subsets and group data.	The package has been used to handle data frames produced in the pipeline.
Rstudioapi	A package for manipulating and handling documents open in R.	The package has been used to interact with documents opened in R in this pipeline.
Gtools	A package providing several functions to assist R programming.	The package has been used to handle variable in different forms in the pipeline.
Reshape2	A package built to transform data between different formats.	The package has been used to melt datasets before plotting.
Stats	A package containing statistical functions for R.	The package has been used to calculate different statistical calculations in several functions of the pipeline.
Svglite	A package necessary for the creation of .svg files.	The package has been used to create plots in .svg format which is a vector graphic format suitable for documents and presentations.
Usethis	A package for package and project creation inside of R scripts.	The package has been used in the create project function of this pipeline.