FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Oana Stoicescu

ClinFlow: An Interactive Application for Processing and Exploring Clinical Data

Master's Thesis
Degree Programme in Biomedical Engineering
April 2020

Stoicescu O. (2020) ClinFlow: An Interactive Application for Processing and Exploring Clinical Data. University of Oulu, Degree Programme in Biomedical Engineering, 73 p.

ABSTRACT

Clinical data is the most valuable resource in healthcare development, but it also comes with many challenges. When clinical researchers are required to combine medical expertise with statistical and programming knowledge, the need for data analysis tools arises. The aim of this thesis was to design ClinFlow, an application for clinical data processing and visualization based on user needs. R language and Shiny framework were selected for creating this tool. The goal was to give the means for the clinical researcher to conduct data analysis in an interactive environment, with no need for statistical programming knowledge. A case study using data from The Finnish Type 1 Diabetes Prediction and Prevention (DIPP) study was conducted to demonstrate the feasibility of this application. The initial results achieved in this case study support the previous research of the DIPP study. ClinFlow shows potential for becoming a useful data analysis tool for clinical research.

Keywords: ClinFlow, clinical data, analysis, visualization, preprocessing, type 1 diabetes.

TABLE OF CONTENTS

FOREWORD

The work in this thesis was conducted in the Biomimetics and Intelligent Systems Group (BISG) at the Department of Computer Science and Engineering of the University of Oulu, Finland, as part of the HTx Next Generation Health Technology Assesment project.

First I would like to thank my supervisors Dr. Eija Ferreira and Dr. Satu Tamminen for their endless patience, guidance and valuable advice. Thank you to Prof. Riitta Veijola and Dr. Pekka Siirtola for their valuable contribution, making this work possible. Thank you to all my colleagues at BISG, especially to my friend Gunjan Chandra for her positive and encouraging words. Finally, thank you to my fiancé Niilo for always being there for me.

Oulu, 9th April, 2020

Oana Stoicescu

LIST OF ABBREVIATIONS AND SYMBOLS

DIPP	The Finnish Type 1 Diabetes Prediction and Prevention Study
EHR	Electronic Health Records
GADA	Autoantibodies to the 65 kDa isoform of GAD
IAA	Insulin autoantibodies
IA2A	Autoantibodies to islet antigen 2
ICA	Islet cell antibodies
LOWESS	Locally Weighted Scatterplot Smoothing
MCAR	Missing Completely at Random
MDS	Multidimensional Scaling
PCA	Principal Component Analysis
SOM	Self-organizing Map
t-SNE	t-distributed Stochastic Neighbor Embedding
T1D	Type 1 Diabetes
UI	User Interface
ZnT8A	Autoantibodies to zinc transporter 8

1. INTRODUCTION

Medical data is a crucial resource in the process of healthcare progress, fundamental to the development of good care practices for patients and the success of clinical trials [1].Generally, raw medical data provides challenges with the cleanliness, reliability and completeness. Medical data is usually stored in a way that can't be easily analyzed. Population health data can be large and diverse. The variability in data types can impose a challenge, as well as merging data from different sources or database types, and there is a need for a common terminology consensus across various data sources. There are several sources for uncertainty and errors in the medical data collection phase. From sources such as lab results, medical visits or patient questionnaires, medical records are usually manually introduced in a database which leaves them prone to human errors [2]. Data from these medical records is in a wrong format (free text format), irregular and unstructured [3].

Data mining is the process of discovering patterns and previously unknown relationships within data, and it involves statistical methods and machine learning methods used for extracting novel information from a data set, in a more understandable form for further analysis [4]. For the data mining techniques to be successful in medicine, high quality medical data is needed. However, different data sets have different specific issues that can't be solved with a general preprocessing algorithm. In order to obtain the data quality necessary for efficiently applying data mining techniques, the data has to go through a preprocessing step. The data preprocessing process includes data cleaning, data integration, data transformation and feature extraction [5]. Data cleaning removes inconsistencies and errors such as extreme outlier removal, deleting duplicate entries, etc. Data integration refers to merging data from multiple sources. Data transformation transforms the variables into appropriate forms and includes scaling, normalization or conversion. Finally, feature extraction refers to creating new, more interpretable and non-redundant features from the existing ones.

A survey [6] conducted in 2018 among clinical trial researchers mainly in North America and Europe found that the biggest challenge with clinical trial data is data quality, followed by inconsistent data, which requires manual effort in aggregating, cleaning and transforming this clinical data. Most of the respondents in this survey experience issues with this manual process, which result in trial delays.

While some preprocessing methods require extensive statistical programming knowledge, others need the expertise of the medical professional, for example, in feature extraction or outlier treatment. The work in this thesis is dedicated to solving clinical data preprocessing and preparation challenges, prior to clinical data analysis and offering a platform for the researcher to easily prepare and analyze clinical data. This is achieved using the R programming language and free software environment for statistical computing and graphics[7].

1.1. Objectives and scope

The main objective is to deliver an interactive application dedicated to analyzing clinical data, that allows for some user-defined data preparation operations like filtering, feature construction and outlier treatment, and a platform for visualizing different analyses. The end goal is to facilitate the researchers to navigate data, collect subsets of data based on their research hypothesis, extract new information and perform analyses where statistical software knowledge is not necessary.

This tool is designed using the Shiny framework [8] - an R package for building interactive applications straight from R. It uses open-source technologies to offer an intuitive user interface that requires no coding or statistical software knowledge. R was chosen due to its built-in statistical analysis capabilities, and the Shiny framework makes it easy to embed R's capabilities in an interactive user interface.

We will conduct a case study to explore the need, requirements, and feasibility of such a tool, to discuss it's implementation, and to demonstrate it's operation and usefulness in processing clinical data. The case study will be conducted using data from the Finnish Type 1 Diabetes Prediction and Prevention (DIPP) study [9], containing globally unique cohort data that has been studied widely by several national and international research groups for the past two decades, with the purpose of developing strategies for prevention of type 1 diabetes. A secondary objective is to design a preprocessing framework for delivering clean and interpretable information, focused on correcting errors found in the DIPP data and extracting useful features that are well hidden in the raw data.

Following is an overview of the thesis. Chapter 2 presents a comprehensive list of data visualization tools and discusses their capabilities. Chapter 3 describes in detail the theoretical background of the statistical methods to be included in the tool. In Chapter 4, the architecture of the tool is presented along with each functionality explained. Chapter 5 contains the methodology of the case study, the preprocessing of the DIPP data and the case study results. Chapter 6 includes the discussion of the case study results and the applicability of the tool, as well as limitations and future work. Finally, the thesis will be concluded in Chapter 7.

2. BACKGROUND

Data collected during clinical trials comes with many challenges such as high volume of data, unclean data, irregularities, as well as other problems related to specific medical domains. Visualization is critical in the analysis of clinical data, not only for discovering patterns in the patient's medical visits, but also for evaluating the data quality and validity, as well as evaluating whether there is a need for improvement in the data collection practices [10].

Various tools for visualization, management and analysis of clinical trial data have been created throughout all medical specialties. These tools have been proven useful in understanding patients' medical history and improving clinicians' recognition of health trends [11].

Statistical techniques for observational analysis can be used for data derived from the routine medical visits of entire populations. However, clinical trial data can be very diverse depending on the medical domain and complex relationships between the variables cannot be easily extracted without personalized tools specific to the studied domain. Specific applications have to be adapted to domain expertise [12].

2.1. Data Visualization Tools

Multiple visualisation tools are currently available for analyzing information in the biological and clinical research. There's a wide variety of data visualisation tools including clinical, biological and general tools. EventFlow [13], LifeLines [14], LifeLines2 [15] are tools designed for clinical data visualisation and dashboard development for electronic health records (EHR). Deng and Denecke [16] used a tag cloud from radiology reports, pathology reports and surgical reports for summarising unstructured patient records. These tools are very useful for researchers to summarize, aggregate and simplify a large volume of electronic health records, for visually identifying patterns, and they offer also some data wrangling and manipulation functionalities. However, they do not include unsupervised learning methods, such as clustering, that can be useful in clinical trial data analysis. VISualization of Time-Oriented RecordS (VISITORS) [17] is used for visualizing time-oriended health records and Dynamics Icon (DICON) [18] is used for clustering and finding similarities between clusters of patients. These tools don't offer options for data cleaning or wrangling operations. Data visualisation tools, such as HARVEST [19], offer a web based infrastructure for integrating, discovering and reporting data but are restricted to the data captured in a data warehouse.

EHDviz [20] is a tool for realtime health data in a hospital setting, as well as emulating EHR for population assesment. HTPMod [21] visualizes and models biological data such as genomics and phenomics plant growth data. ExPanD [22] is an R package and tool for generalized panel data which can include visit based medical data. The last three applications include visualizations that can be used for clinical data, but offer little to no preprocessing options. Table 1 summarizes the advantages and limitations of these visualization tools.

Table 1. Advatages and limitations of data visualization tools

Tool	**Used for**	**Advantages**	**Limitations**
EventFlow [13] LifeLines [14] LifeLines2 [15]	Visualizing and summarizing patient records	Enables the visual analysis of large scale patient records.	No statistical unsupervised learning methods.
VISITORS [17]	Exploration of time-series data	Enables visualization of similarities between variables and groups of patients.	Uses raw data, no preprocessing.
DICON [18]	Cluster analysis	Highly interactive and suggestive cluster visualizations and similarity detection between different groups.	No preprocessing options other than grouping.
HARVEST [19]	Data visualizations	Supports incremental visualization updates. Tracks the user's analysis activities for better recommendations of visualizations.	Dedicated to business data. No preprocessing options.
EHDviz [20]	Time series visualization of health records	Has options for visualizing individual patients, patients from different hospital locations as well as population and cohort data visualizations.	Only time-series numeric visualizations, no preprocessing or statistical methods.
HTPMod [21]	Visualization and modelling of large-scale biological data	Has a wide variety of visualization and modelling methods.	Dedicated mostly to plant growth data and gene expression data. Limited preprocessing options and handling of missing values with automated imputation.
ExPanD [22]	Panel data visualization	Options for subsetting data based on categorical variables. A variety of panel data visualization methods including time trends. Builds fixed effects regression models.	Not dedicated specifically to clinical data. Limited preprocessing options.

Overall, the tools presented above offer a large pool of functionalities for various data types, most of all for EHRs. However, one tool cannot be generalized to all medical data, especially clinical trial data which can be highly diverse depending on the domain. Knowledge about the data collection process and how the clinical trial protocols are designed is also indispensable when processing and analyzing clinical trial data. Most of these tools either do not have preprocessing or they have automated preprocessing that can introduce bias if applied on the whole dataset. The tool we are building integrates open source visualization technologies included in some of the tools listed above, designed for the researcher to be able to verify the availability and correctness of the data and utilize domain knowledge when testing or creating hypotheses, combined with user defined data preparation functionalities for filtering, feature construction and missing data and outlier removal. As a case study, an analysis of the DIPP clinical data was conducted, a preprocessing and feature construction segment customized for the DIPP data was designed and integrated in the application. The application also offers options of exporting the data into formats that are ready to be forwarded into several other visualization tools without further preprocessing.

3. METHODS

Raw medical data is commonly stored in formats that cannot be easily analyzed by computational methods. Medical data may be collected from various images, interviews with the patient, laboratory data, and the physician's observations and interpretations, and is afterwards stored to a common database [23]. Clinical trial data contains information that require several visits to the clinic, and follows the development of the medical status of each participant. Medical records are usually manually introduced in databases which leaves them prone to human errors. Moreover, data from these medical records is in varying formats (e.g. free text format), irregular and unstructured [2]. As a result, a data preprocessing step is required. For successful preprocessing of the data, knowledge about the dataset itself and the domain studied is crucial. The data visualization process has the role of assisting the researcher to get insight into the data quality and outliers. In clinical data, visualization allows the direct involvement of the medical expert, which is an advantage over the automatic data mining techniques. Moreover, unsupervised learning techniques such as clustering can offer valuable insight into cohorts in clinical studies, and assist the researcher to generate new hypotheses, that can be verified afterwards using machine learning or statistic techniques [24].

3.1. Clinical data preprocessing

Peterkova and Michaľčonok [3] presented some general guidelines for preprocessing steps to follow to deliver clean and interpretable clinical datasets to be used in medical applications. The aim of these guidelines is to reduce the time spent on manual data preprocessing. However, this thesis underlines the importance of customizing the preprocessing methods used for each individual dataset.

Figure 1 presents a general framework that can be used for data preprocessing. It starts processing from raw-text files and the end result is a structured database.

To get a good general idea of the data and the studied problem, as well as to find irregularities from the data, the dataset needs to be visualized. For identifying the parameters, a medical hypothesis is required, defined by relevant literature or a medical expert.

Parameters extraction depends on the data source. Data can come from one or multiple sources like SQL database and/or text tables in different formats. A programming language like R or Python is required for importing and merging data from multiple sources, converting file formats and dropping irrelevant variables.

Data cleaning deals with specific problems in the data and is different from dataset to dataset. This step requires first the identification and definition of errors and error types. For example, clinical data errors could be duplicated entries, unreal entries such as birth date in the wrong century due to human errors, missing values, extreme outliers, etc.

Data normalization in clinical research refers to the process of rationalizing data to a terminology intended for algorithms to understand [25]. For example,

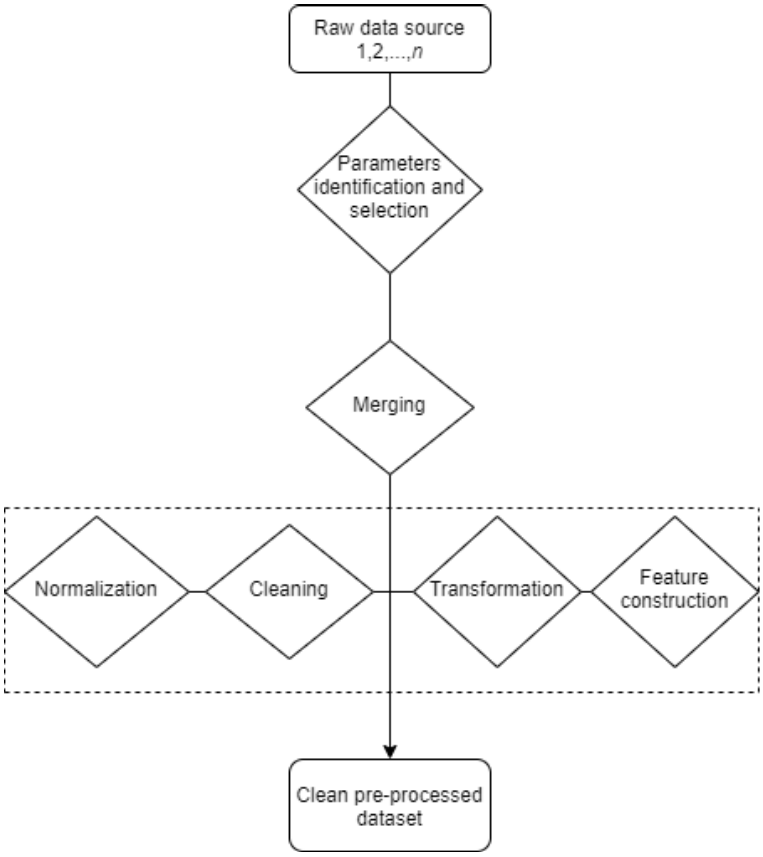


Figure 1. Medical data preprocessing diagram.

the variable formats need to be converted from "character" to "numeric" or "date", converting different measurement units, text parsing, etc.

Finally, data transformation refers to converting the data from one structure to another. This step can be intertwined with the data normalization. For example, this can contain categorisation of continuous variables using a threshold value, categorisation of text variables using certain string patterns. Another important procedure that can be included in this step is feature construction. Feature construction is a process that creates more efficient variables, derived from the existing information through some functional mapping, and adds them to the data in order to improve learning effectiveness [26].

3.2. Data visualization techniques

In order to characterize different visualization techniques, Daniel A. Keim [24] introduced a classification according to three criteria, presented in Table 2: the data type to be visualized (1), the visualization technique (2), and the type of interaction and distortion technique (3).

Clinical data is usually multidimensional data, consisting of a large number of records with multiple variables. It can be visualized using techniques for one-dimensional data by visualizing individual variables, for two-dimensional data by

Table 2. Visualization techniques according to three criteria [24].

Criteria	Specific techniques
Data type to be visualized	One-dimensional data Two-dimensional data Multidimensional data Text and hypertext Hierarchies and graphs Algorithms and software
Visualization technique	Standard 2D/3D displays Geometrically transformed displays Icon-based displays Dense-pixel displays Stacked displays
Interaction and distortion technique	Interactive projection Interactive filtering Interactive zooming Interactive distortion Interactive linking and brushing

selecting two variables, and for multidimesional data by exploring more complex properties of multiple or all variables.

The visualization techniques in this work are presented and briefly explained below.

1. Charts

   Charts are graphical representations of the data. They can be used to explore the frequency of a single variable, such as histograms or bar plots, which can also be colored according to a categorical variable.

   Density plots visualize the density distribution of a numeric variable or compare the densities of the variable according to another categorical variable.

   The relationship between a categorical and a numeric variable can be explored through Bar plots, box plots, violin plots and stripcharts. They all provide a representation of the numerical distribution for each category.

   The relationship between two numeric variables can be visualized through scatterplots, to which a third categorical variable can be added and points coloured according to the category. The scatter plot visualization can also be enhanced by fitting a regression line or a LOWESS(Locally Weighted Scatterplot Smoothing) curve, for better understanding of the correlation of two numeric variables.

   Finally, all the plots listed above can be visualized into a plot matrix such as a pairwise scatterplot, which allows easy comparison between multiple or all the relationships in the data.

2. Principal Component Analysis

One definition states that Principal Component Analysis (PCA) is a "statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components"[27]. The first principal component is the normalized linear combination of the features that has the highest variance in a scalar projection of the data, the second principal component is orthogonal to the first one and accounts for the highest possible variance, and so on, as shown in Figure 2.

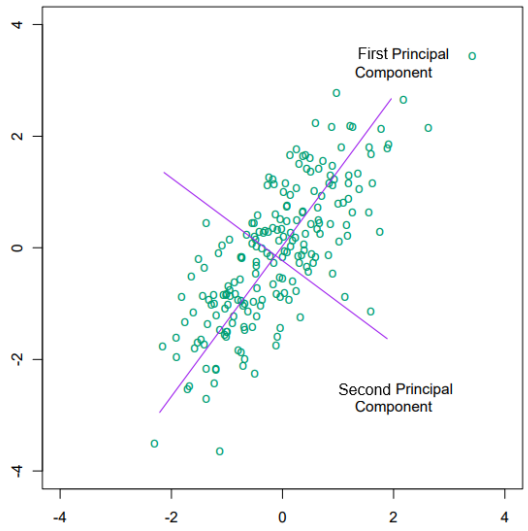


Figure 2. PCA of a multivariate distribution [28 p.67].

PCA is used for data reduction, but it can also be applied to cluster a high dimesional dataset [29]. For a dataset of p variables and n entries $x_{11}, x_{12},..., x_{np}$, centered to have the mean 0 and standard deviation 1, the first principal component is $Z_1 = [z_{11}, \ldots, z_{n1}]$ of the form

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \ldots + \phi_{p1}x_{ip}, \tag{1}$$

where $z_{11}, z_{21}, \ldots, z_{n1}$ are called the scores and $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ are the loadings of the first principal component. The loadings can be interpreted as the correlation of each element with the principal component and are calculated by $eigenvector \cdot \sqrt{eigenvalue}$. Eigenvalues indicate the amount of variance that can be explained by the component and the eigenvector indicates the direction of the variance. The second principal component is the linear combination with the highest variance out of all possible combinations that are uncorrelated with Z_1, and so on. Most of the variance in the data is usually explained by the first two or three principal components, which can be plotted in a 2D or 3D scatterplot. The scatterplot can then be colored according to a categorical variable to highlight the differences between the categories.

3. t-Distributed Stochastic Neighbor Embedding

   T-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique for dimensionality reduction, like PCA, well-suited for high-dimensional data [30]. It converts the Euclidean distances between datapoints into conditional probabilities $p_{j|i}$ that represent similarities. For the low dimesional data representation, similar conditional probabilities are calculated for the correspondent data points in the low-dimensional space, $q_{i|j}$. The aim is to minimize the mismatch between $p_{j|i}$ and $q_{i|j}$ by minimizing the Kullback-Leiber divergence (KL), also called the relative entropy C, using a gradient descent method of the form

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}, \tag{2}$$

   where $C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{p_{i|j}}$, P_i is the conditional probability distribution of data points in the high-dimensional space and Q_i is for the corresponding low-dimensional points in the embedding. This way, t-SNE maps high-dimensional data to a lower dimensional space of points with multiple features. It can be used for clustering the data based on the similarity of data points, by plotting a 2D or 3D scatterplot.

4. Multi-dimensional Scaling

   Multi-dimensional Scaling (MDS), like t-SNE is a technique designed to give a representation of high-dimensional data in a low-dimensional space, by preserving the distances between datapoints [31]. The distance measure can be Euclidean or non-Euclidean. The aim is to find an optimal configuration of points in 2-dimensional or 3-dimensional space. However, this optimal configuration might be a very poor representation of the data. If so, this will be reflected in a high stress value

$$stress = \sqrt{\frac{\sum (d_{ij} - \hat{d}_{ij})^2}{\sum d_{ij}^2}}, \tag{3}$$

   where d_{ij} is the actual distance and $\hat{d}_{ij}$ is the predicted distance on the lower-dimensional space.

   Like t-SNE, the MDS map offers a 2D or 3D scatterplot with a representation of the similarities or dissimilarities between the points in the data, where clusters can be coloured according to categories or groups.

   MDS is different from PCA in the way that PCA looks for similarities between features by computing the covariance matrix to explain variance in the data, while MDS is looking for similarities between datapoints and plots the similar datapoints closer together on the map.

5. Self-Organizing Map

   Self-Organizing Map (SOM) is a dimension reduction and clustering technique, similar to MDS, that maps the high-dimensional data to a

lower-dimension space, usually a 2D map [32 p. 105-176]. But unlike the MDS or t-SNE, SOM is a competitive learning neural network, based on unsupervised learning, meaning that it automatically assigns data points to a class or, in this case, a cluster on the map.

The map is a collection of neurons in a 2D array. Each neuron has an associated weight vector w_{ij} that corresponds to a point in the original high-dimensional feature space $x(t)$. Each observation in the dataset is assigned to one of the neurons, according to which weight vector the observation is closest to, starting with random weight vectors, and updating them as follows

$$w_{ij}(t+1) = w_{ij}(t) + \alpha_i(t)\beta_{ij}(t)[x(t) - w_{ij}(t)], \tag{4}$$

where $t = (1, 2,...n)$ is the current iteration, $\alpha_i(t)$ is the learning rate that decreases monotonically with each iteration, and $\beta_{ij}(t)$ is the neighbourhood function's influence calculated by

$$\beta_{ij}(t) = \exp(\frac{-d^2}{2\sigma^2(t)}), \tag{5}$$

where d is the minimum Euclidean distance out of all the node's calculated distances, and $\sigma(t)$ is the neighborhood function's radius that also decreases over time.

The end result can be interpreted as a combination of dimensionality reduction and clustering, where each cluster corresponds to a neuron in the 2D map. As seen in Figure 3, the SOM allows us to see structure in the clustering, by coloring according to a categorical variable.

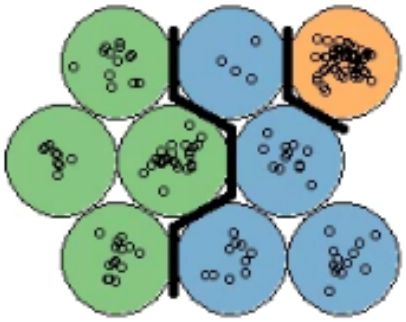


Figure 3. SOM with nine neurons. Each color corresponds to a category in the data.

6. k-means Clustering

k-means Clustering is the simplest and most popular clustering method. It works by initially randomly selecting k points in the data as the centroids of the clusters, and measuring the Euclidean distance between each data point

and each centroid, assigning that data point to the cluster with the nearest centroid. Then, the centroid is updated as the mean of all the points in the cluster. k-means Clustering is an iterative process and it stops when the centroid's values don't change anymore, or when the predefined maximum number of iterations is reached.

The optimal number of k can be decided using plots from various methods such as the Elbow Method [33], The Silhouette Method [34] or the Gap Statistic method [35] shown in Figure 4.

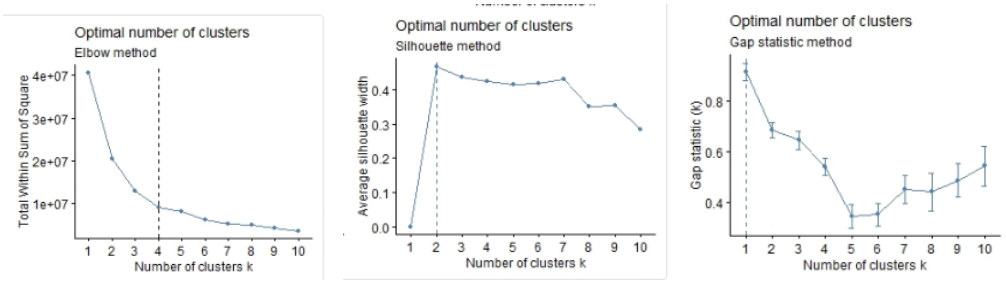


Figure 4. Plots generated by Elbow Method, Silhouette Method and Gap Statistic Method, for choosing the optimal number of clusters in k-means Clustering.

The Elbow Method plots the sum of the squared error (SSE) which is the squared distance between each member of the cluster and the centroid, for k = 1,...,10. The "elbow", where the line bends clearly to the right in the plot is the optimal k. If the plot line does not bend clearly enough to be able to identify the elbow, it may be better to use another method.

The Silhouette Method plots the coefficient $S_i = (b_i - a_i)/max(a_i, b_i)$ that tells how close each point in one cluster is to points in the neighboring clusters, where $b_i = \min_C d(i, C)$ is the smallest average dissimilarity $d(i, C)$ of each observation i to the members of the neighboring cluster C.

The Gap Statistic Method compares the total within intra-cluster variation $W_k = \sum_{r=1}^{k} \frac{1}{2i_r} D_r$, where D_r is the pairwise distance of all the members of cluster r to the centroid, for different values of k with a null reference distribution of the data, i.e. a distribution with no obvious clustering. The algorithm works as presented in Algorithm 1.

Algorithm 1 Steps of the Gap Statistic Method.

1: Compute within intra-cluster variation W_k varying the number of clusters k from 1 to 10.
2: Generate B reference data sets with a random uniform distribution and with the varying number of clusters k compute within cluster variation W_{kb}.
3: Compute the estimated gap statistic $Gap(k) = \frac{1}{B} \sum_{b=1}^{B} log(W^*_{kb}) - log(W_k)$.
4: Compute standard deviation s_k of $Gap(k)$.
5: Choose k such as $Gap(k) \geq Gap(k+1) - s_{k+1}$.

4. SHINY APP

In order for this tool to meet the objective set in this thesis, a set of requirements was defined. This chapter presents the requirements and the general architecture, as well as the functionalities of the tool.

4.1. Requirements

The requirements for the application are divided into two categories: preprocessing and visualization requirements. preprocessing requirements relate to tasks that involve dataset cleaning, transforming, building and subsetting, and visualization requirements relate to data analysis tasks that are available in this tool.

Preprocessing requirements

Assuming that the data uploaded in the application is clinical data, the application should offer two datasets. Patient data, containing constant variables that do not change over time, and visit data, containing variables that change over time for each patient or subject in the dataset.

The application should then provide a user interactive preprocessing interface with options for visualizing and filtering missing data, interactive charts for outlier selection, inspection and removal, a filter for subsetting the data according to user-selected variables, creating new constant variables by aggregating the visit data from a time interval chosen by the user or by categorizing numerical variables by user-defined parameters, and converting the variables, both constant and time-varying, into a panel data structure that can be uploaded directly into other visualization apps for time series data, for example, the R-based Shiny app "ExPanD"[22]. The tool should also offer the option of downloading the preprocessed datasets in the form of ".csv" tables.

For the DIPP data case study, the application should take raw DIPP data as input and deliver a preprocessed clean DIPP dataset, with new variables defined and constructed according to literature review and user needs. This step should offer clean data that can be used for analysis, without further cleaning and preprocessing.

Visualization requirements

The tool should provide an interface containing a comprehensive set of interactive methods for visualizing and analyzing the data. Visual inspection of data is a powerful way to find relationships between different variables or different groups in the data. The application should provide several unsupervised learning methods for clustering, with the purpose of identifying similarities between data points and exploring hidden patterns in the data. The application should also provide the user with a wide range of univariate, bivariate and multivariate charts for visualizing in detail the relationships between user selected variables and groups

in the data. The visualization has a role of assisting the researcher in finding patterns in the data, checking the validity of the findings as well as testing and creating hypotheses.

4.2. Application Architecture

For building this app, we chose the Shiny framework, an R package developed by the RStudio team enabling R programmers to create interactive visualizations for the web [36, 8, 7]. R language is a powerful tool for solving analytical challenges, but it requires programming knowledge. A Shiny app is an interface that allows all of the advanced analytics of R to be made available to the non-R users who will be making the decisions in the data analysis process. A Shiny app contains two parts. A user interface (UI) which dictates the appearance of the user input elements, and a server, which contains the backend processes. These are R scripts developed in parallel, with variable names that match the UI elements with the server calculations. In the server file, the programmer can define reactive expressions, which means that the system responds to user input, such as for example clicking a button to update the elements displayed in the UI. The shiny UI elements can be buttons, sliders, text inputs and drop down menus, which can be combined with other interactive elements from certain R packages, for example, interactive plots built with the "ggplot2" R package [37].

A standard shiny UI page has a sidebar panel, usually for user input elements, and a main panel where results are displayed, and optionally, some more user input elements. It also contains navigation bars, for changing between different Shiny pages, and tabs, to change between different main panels. A simplified scheme of the Shiny UI layout can be seen in Figure 5.

The aim of this work was to build an application that integrates some of the open-source visualization technologies found in other tools with a preprocessing interface that allows the researcher to use the knowledge about the studied domain and the data collection practices when filtering, dealing with missing data and outliers, as well as checking the quality and availability of the data. Clustering visualization methods were implemented in the application for allowing the user to identify important relationships between variables and groups in the data, as well as to discover groups of similar entries and how they are distributed in the clustering space. Various types of exploratory charts were implemented for summarizing the data. Combined with the clustering visualizations, the charts are used to explore the patterns found in the data. The visualizations are intertwined with the preprocessing as well, providing insight into any bias or need for further preprocessing that might be present in the dataset. The tool, named ClinFlow, has a modular architecture, which means that each module is a separate R script. Removing, modifying or replacing a module in the code will not affect the rest of the functionalities. Each functionality of the application has its own server module, matched with the correspondent UI module, as illustrated in Figure 6. The UI contains only three modules for each navigation bar. These are "Upload and preprocess", "Visualizations" and "Panel data".

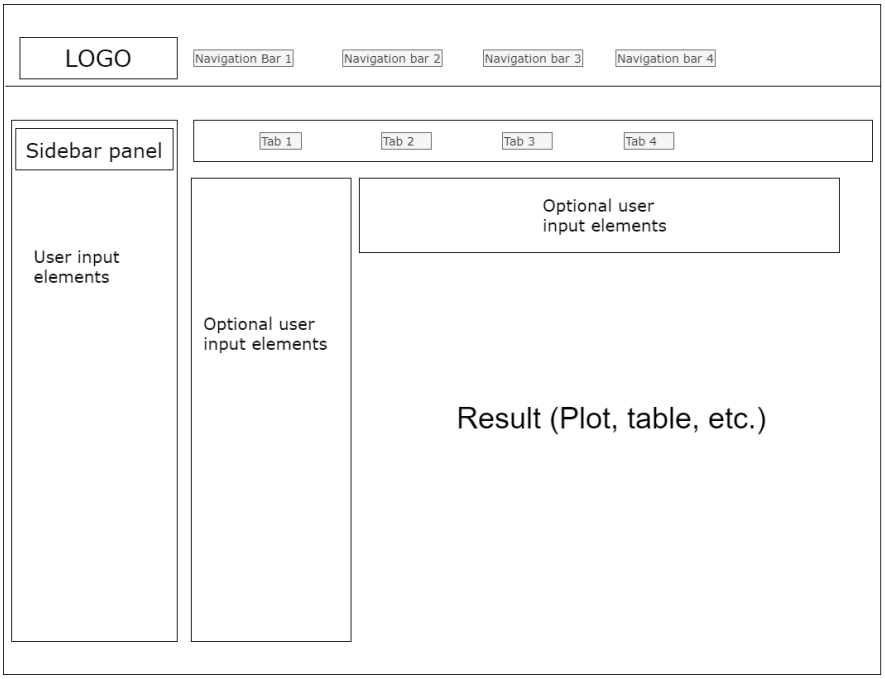


Figure 5. Basic Shiny UI layout.

The Upload and Preprocess UI module corresponds to the uploading, preprocessing, filter, aggregation, categorization, missing data map and outlier removal server modules, each with its own tab. This module allows the user to upload a dataset and perform actions of their choice to preprocess the data. In this UI module, each UI tab has a server module with the following functionalities:

- The uploading server module inputs a ".csv" table into the app.
- The preprocessing runs the data through a preprocessing pipeline[1] and divides it into patient data and visit data.
- The filter subsets the data according to user defined criteria.
- The aggregation module creates a new patient variable from aggregated visit data from a time interval defined by the user from the visit age variable.
- The categorization turns a numeric variable into a categorical variable according to user selected cut-off points.
- The missing data map displays plots of the missing data.
- The outliers server module displays interactive plots, where the user can check and delete outliers.

To match the requirements of this tool an existing Shiny module[2] by Dijun Chen et al. [21] was integrated into the app. This open-source application module was

[1]Depending on the column names in the data, some general preprocessing operations described in Chapter 5: The DIPP Case Study are applied on the data.

[2]Code for this module is from the HTPdVis module of the HTPmod app, with the same UI layout, by Dijun Chen et al. [21]. Source: `https://github.com/htpmod/HTPmod-shinyApp`

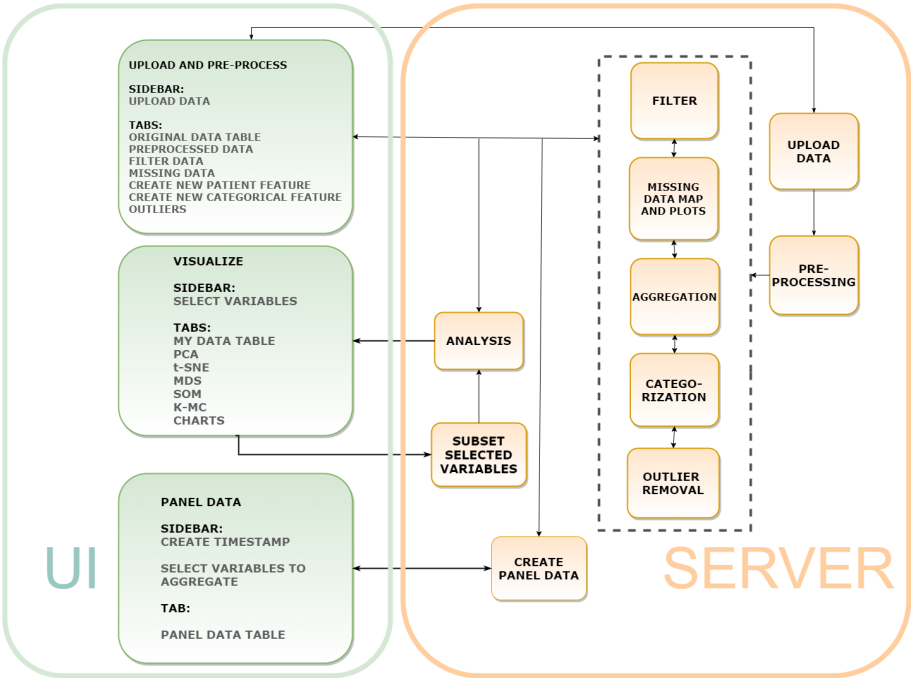


Figure 6. Architecture of the Shiny app.

distributed under the GNU General Public License. The reasoning for selecting this module is that it contains a wide range of visualization methods included in the requirements for ClinFlow, useful for exploring multivariate diverse datasets, including clinical data. The graphical representations in this module are visually pleasing and intuitive, and the clustering computations are done with built-in R functions. The integrated module required minor changes to better fit the purpose of our tool. More user options were added for choosing the variables to be used in the analysis and missing data treatment. The R package used for 3D plots was changed because the 3D representations in the original application were showing incorrect information when tested with the Iris dataset[38]. The functionalities of the module are presented later in section 4.2. The Visualization UI module has tabs for PCA, t-SNE, MDS, SOM and k-means Clustering methods and a tab for charts. Only one server module is matched with this UI module, containing a reactive function that performs analysis according to the selected tab.

The Panel data UI module is matched with the corresponding server module for turning visit data into panel data. Panel data is a multi-dimensional data that includes multiple subjects measured over a time period. In a long-term clinical study of patients monitored from birth, age can be used as a time vector, but it is impossible to have regular measurements from all the subjects at the exact same age. This server module simulates a regular time period by creating time points using age intervals, with user defined cut-off points, then it aggregates the visit data grouped by patient code for each time point to create a structured panel.

Our tool ClinFlow uses several open source R gadgets and code from other Shiny visualization apps listed in Table 3, along with R common and base packages, to integrate interactive preprocessing and analysis functionalities.

Table 3. Open source gadgets and applications used for developing ClinFlow

Name	**Type**	**Used for**
ggplot2 [37], ggpubr [39], scatterplot3d [40], corrplot [41], naniar [42]	R Packages	interactive plots
esquisse [43]	R Package and Shiny module	data filter
shinyWidgets [44], shinyBS [45], shinyjs [46], htmlwidgets [47], shinydashboard [48]	R Packages	ui elements
pcaMethods[49]	R Package	PCA
kohonen [32]	R Package	SOM
HTPmod(HTPdVis)[21]	Shiny app	Clustering and visualizations

ClinFlow is designed as a web application to be used in a web browser. The application can be hosted on a cloud hosting service such as shinyapps.io [50], or it can be locally hosted on any server that runs R.

The next section of this chapter is split into three subsections, one for each navigation bar of the app, to demonstrate how it works. The only two conditions that the data need to meet in order to use all the functionalities of the application are:

- The data has a column named "code" that contains a unique identification for each patient or group.
- The data has a column named "Age" that is a numeric vector representing time.

4.3. Application Functionalities

For demonstration purposes, we simulate a clinical dataset using the famous Iris flower dataset introduced by the British statistician and biologist Ronald Fisher in 1936 [38]. The Iris dataset consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. The variable "Species" is a factor with three levels. We have added a variable named "code" that has the same information as the "Species", so that each species name represents a unique "patient code", and another variable named "Age" that is a numeric vector, ranging from 1 to 50 for each species. Therefore, our simulated clinical data contains three subjects: "versicolor", "setosa" and "verginica", with four measurements taken regularly from age 1 to age 50. For

a more realistic data, we have also randomly replaced entries in the table with "NA" or missing values.

4.3.1. Upload and preprocessing

When opening the app, the user is first presented with a prompt in the sidebar for uploading the ."csv" dataset. Then, in the Original Data Summary tab a sample of the first rows of the uploaded raw table is displayed, along with buttons that display a summary and structure of the data. The following tab, Preprocessed Data displays the two tables, Visit and Patient data, along with buttons to download or display the summary and structure of each table. The preprocessing applied on the Iris data has added a new patient variable called "Age_follow_up" for the maximum follow up age of each patient in the data, in our case, for each species of Iris. The visit data still contains the patient variables, but the patient data only contains the constant variables. Figure 7 shows these two tabs, and the original and preprocessed table previews.

Filter Data

The next tab is the Filter Data tab, a functionality that allows the user to choose either visit data or patient data, and to select variables to filter. For the numeric variables, the filtering is done via a slider, and for the categorical ones, a multi-choice selector. If the variables contain missing entries, there is also a switch for filtering out the rows with missing values in the chosen variables. The filter is shown in Figure 8.

The filtered table is updated dynamically as the user filters the data, and the bar above the table shows in percents the amount of data preserved after filtering. The button "Update Table" updates both the visit and patient data displayed below, to include only the filtered information and it also updates the filter options accordingly. There is an option for resetting the table, which brings back the unfiltered preprocessed table from the start, and resets the filter options. This page also contains a button for displaying a summary of the variables and the variable types of the filtered data, and buttons for downloading the updated patient and visit tables.

Missing Data

In this tab, the user can choose to display a missing data map of either the visit or the patient table, and a scatterplot of missing vs. observed values from two chosen variables, in order to study the missing data mechanism. This plot can show whether the data is missing completely at random (MCAR), or not. We have introduced missing values randomly in the data, so the scatterplot displayed in Figure 9 does not show any reason for the missing values. In data analysis, usually, entries that are MCAR can be treated either by deleting them, or by imputation [51]. If the data is not MCAR, this table helps the user to study the missing mechanism and make a decision on how to deal with the missing values without introducing bias in the analysis results.

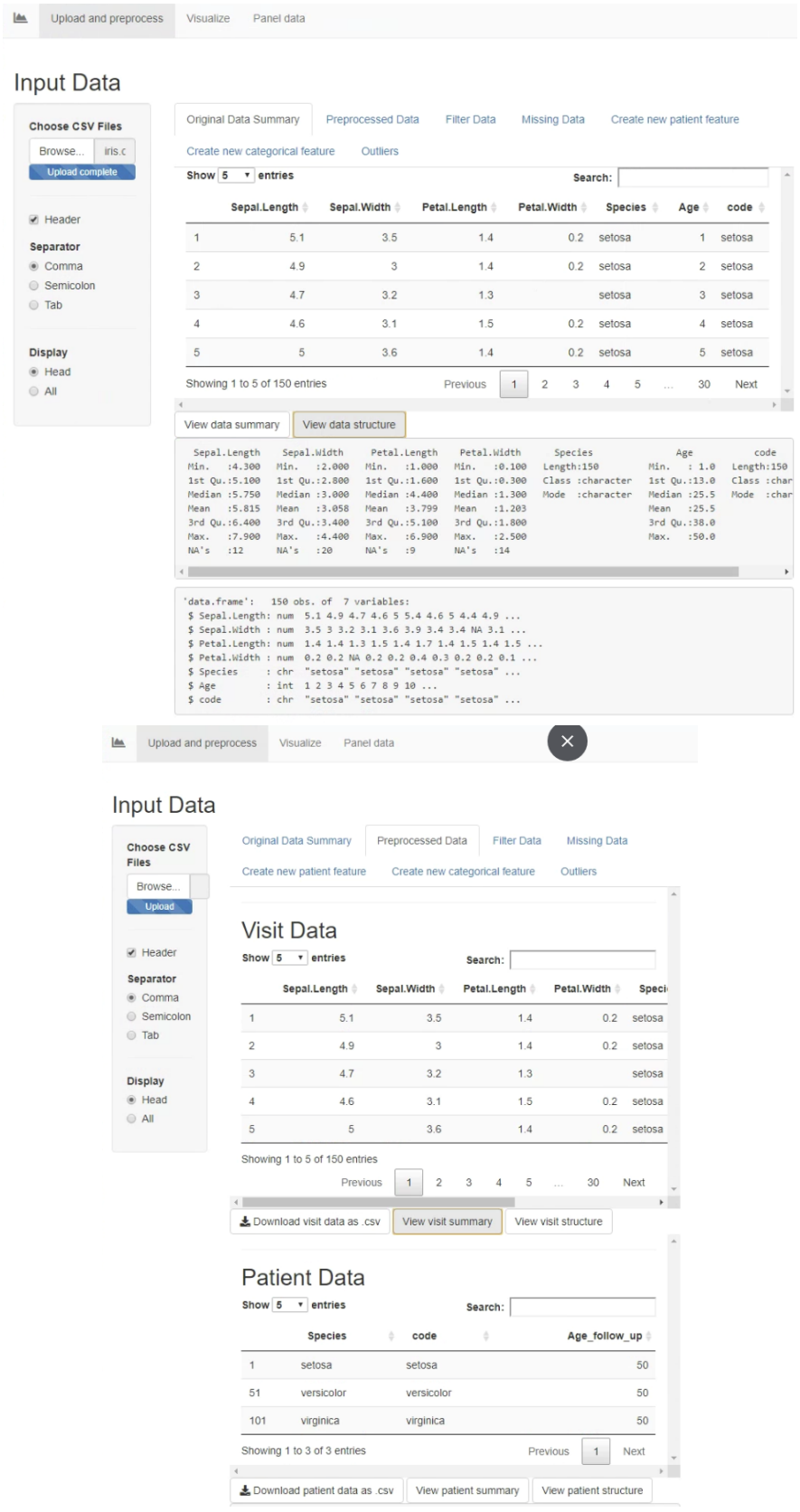


Figure 7. Original and preprocessed data in the Shiny app.

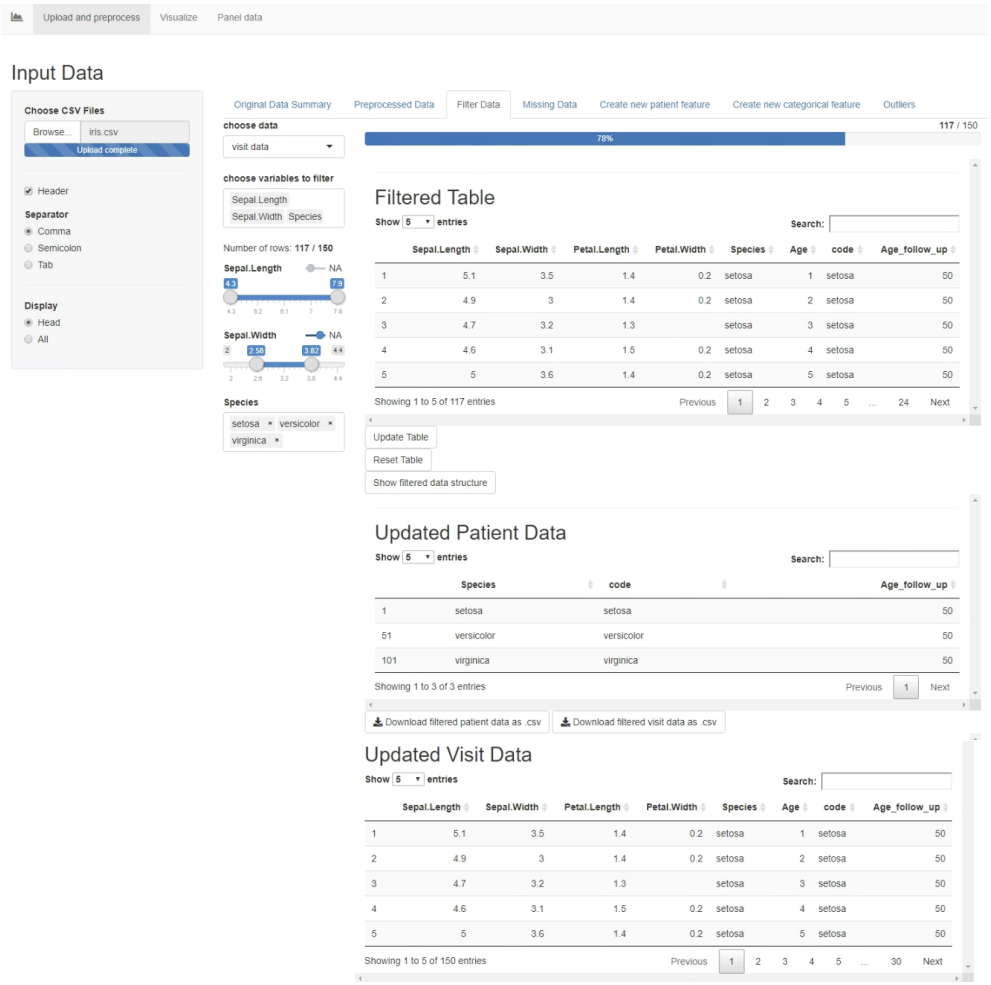


Figure 8. The Filter Data tab in the Shiny app.

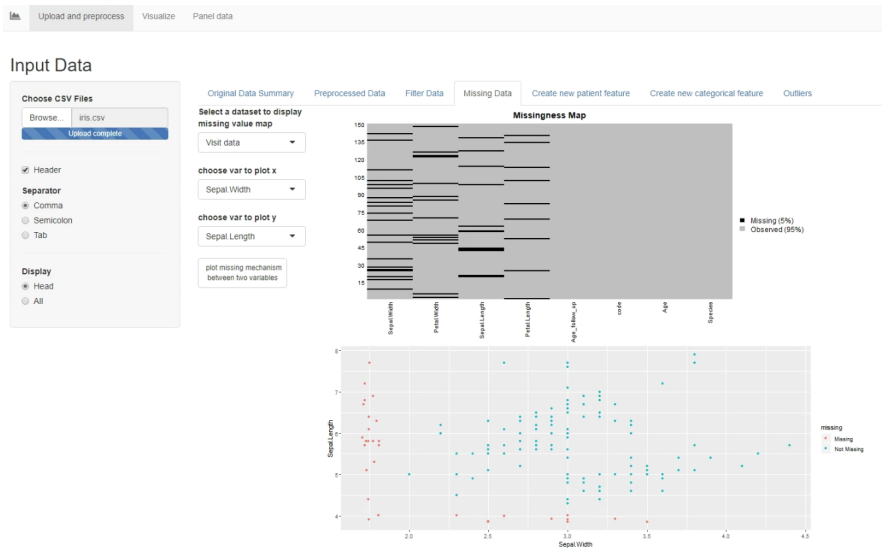


Figure 9. The "Missing Data" tab in the Shiny app.

Create new patient feature

This tab allows the user to add new user-calculated variables to the data. First, it prompts the user to type a new variable name. Then, a slider input lets the user select a range of visit ages from which the data will be aggregated. A drop-down menu allows choosing a visit variable to be aggregated and another drop-down menu allows choosing a function (sum, mean or maximum) for aggregating. For example, in Figure 10, we created a new patient variable called "sepal_max" that marks the maximum value of "Sepal.Length" for each "patient code" from the visits with age between 1 and 15. The page displays an "Updated Table" which is a preview of the patient table containing the new variable. Once the button "Save Table" is pressed, the patient data is saved and the new variable can be used in all the other tabs. This functionality works with categorical variables as well, but instead of a mathematical function, the code looks for a certain factor level. For example, if the variable has the level "TRUE" in one or more visits in the chosen age range, the new patient variable will mark "TRUE", otherwise "FALSE". This is very useful for including a certain period in the life of a patient together with the other non-constant features in the analysis. For example, in a clinical dataset, the user can calculate the maximum weight that a patient reached in the first year of life.

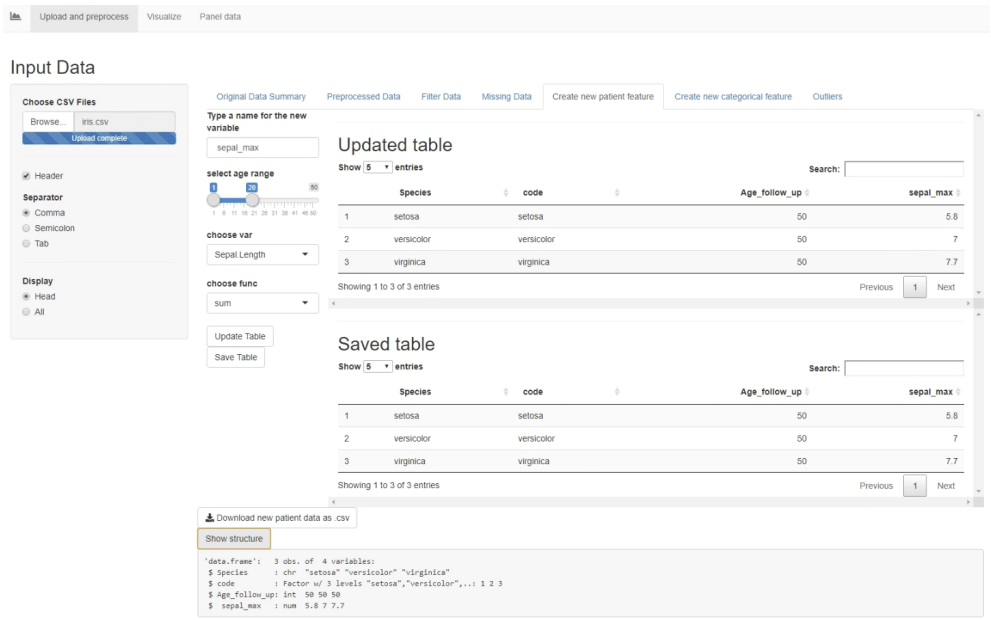


Figure 10. The "Create a new patient feature" tab in the Shiny app.

Create new categorical feature

This tab allows the user to create a new feature, either in the patient table or the visit table, based on another numerical feature in that table. The user must choose a numerical variable in the drop down menu, then type the cut-off points, separated by comma, for splitting the variable into intervals closed on the left and open on the right. The last interval is closed on both sides. A new categorical

variable is created with as many levels as there are intervals, labeled as shown in Figure 11. The "Updated Table" displays a preview of the table containing the new categorical variable, and the "Save Table" button adds the new variable to the dataset. The new feature can then be used for conducting group-based analysis. For example, in Figure 11, we have created the variable "sepal_category" that splits the data into two groups based on the sepal length.

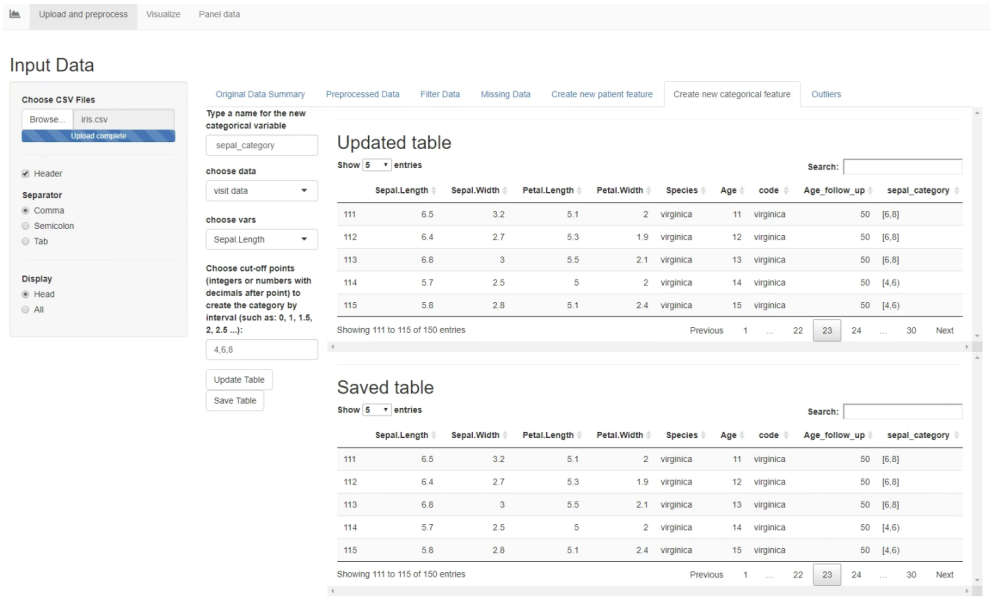


Figure 11. The "Create a new categorical feature" tab in the Shiny app.

Outliers

This tab allows the user to plot a scatterplot and a boxplot (Figure 12) of two chosen variables, and colour the points by a categorical variable, in order to identify outliers. The scatterplot shows the points in the data and a regression line. The points that are further from the regression line should be investigated as possible outliers. The boxplot shows a representation of the distribution of values on the Y axis, as a box with the edges as the first and third quartile and a median line in between. If the X axis is a categorical variable, the boxplot shows distributions of the data for each category. The values that are far away from the median and the quartiles should be investigated as outliers. The data can be either patient or visit data, and the plots are interactive. Clicking on a point in the scatterplot, or selecting multiple points by dragging and then pressing "Toggle points" will move those entries from the original table into the outlier table, where the user can study them and decide whether they should be removed from the data or not. Once the user presses the "Save new data" button, the new data table without outliers is saved, and the toggled entries are deleted. The "Reset" button, brings the table and the plots back to the original state. This functionality is useful because it allows the user to use the domain knowledge to decide whether an entry is an outlier or not. Sometimes in clinical data, a value can fall far from the mean and still be considered

normal, therefore, a generic outlier detection mechanism is not recommended, without first applying expert knowledge. As stated in the U.S. Food and Drug Administration's guidelines "Statistical Principles for Clinical Trials" [52], "Clear identification of a particular value as an outlier is most convincing when justified medically as well as statistically, and the medical context will then often define the appropriate action."

Once the table is saved in any of these tabs, the dataset is updated dynamically and all the user input options are updated to the new dataset as well, which means that any new created variables will appear in the drop-down menus and will be available in the filter options. The dataset can be reverted back to the original state in the Filter Data tab, by pressing the button "Reset Table". This will delete any new created variables and will bring back any deleted or filtered out entries.

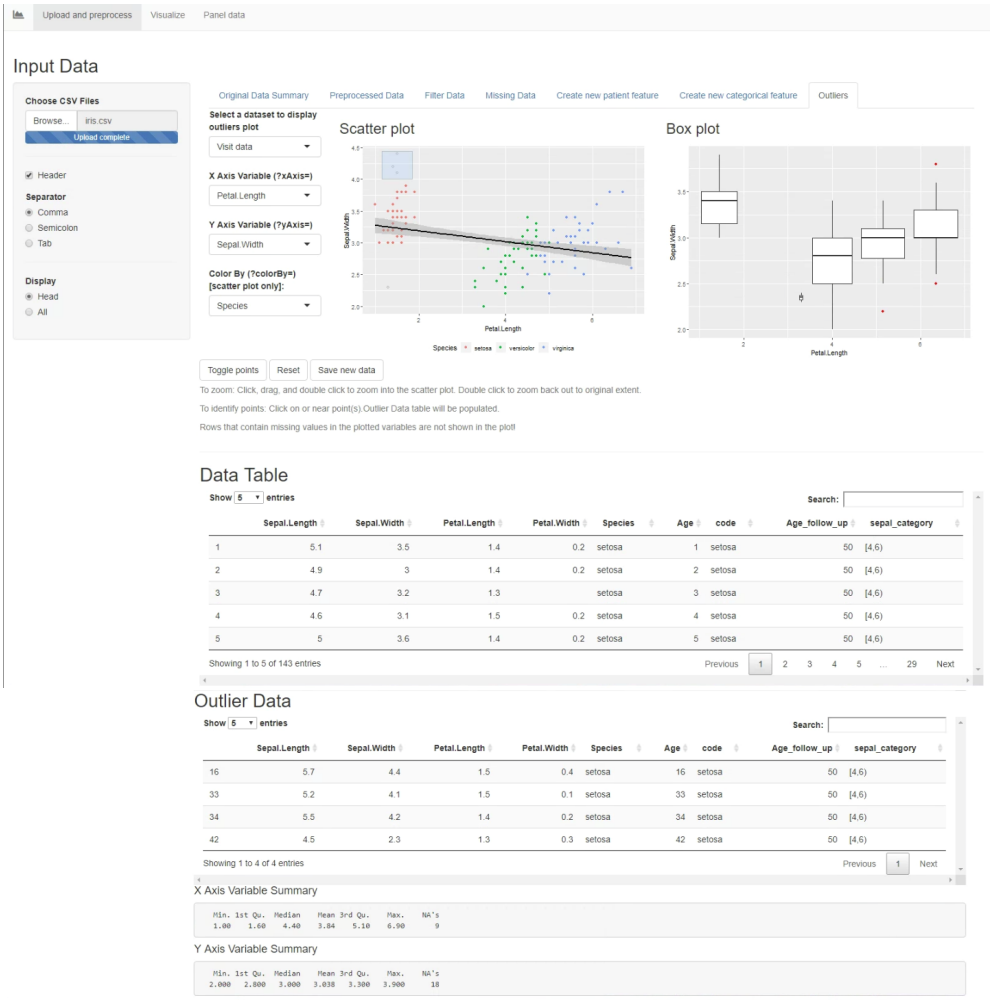


Figure 12. The "Outliers" tab in the Shiny app.

This dynamic data preparation interface is useful because it allows the researcher to easily create subsets and groups in the data according to the research question, or quickly analyze and compare different groups in order to create a hypothesis.

4.3.2. Visualizations

The second navigation bar includes a tab for viewing the table to be analyzed (Figure 13), tabs for different clustering methods (Figures 14, 15), and a tab for various charts (Figure 16). In the sidebar, the user can select either the patient or the visit table, then choose the variables and missing data treatment to use in the clustering methods. The numeric variables are used for the clustering, and the categorical variables can be used to customize the colours and the shapes of the points in the cluster plots.

The user can choose how to treat missing values in the numeric variables used for clustering, either by deleting the rows containing missing values, estimating the missing values using bayesian PCA [53] or not treating them at all. The clustering methods give an error if missing values are present in the numeric data, so the user must make an informed decision on how to proceed. It is recommended to study the missing mechanisms in the data and make a subset, using the data filter, that doesn't include missing values that are not MCAR. Imputation or deletion of the missing entries that are not MCAR can introduce bias in the analysis. If the categorical variables contain missing values, the NA entries are automatically assigned the label "Missing" and they appear as a category in the data, in order to preserve as much information as possible. The clustering visualizations allow for user customization of some parameters, rotation of the 3D plot, and saving the plots in various formats.

Clustering is useful in identifying groups of similar entries in the table. Combined with the coloring and categorization options, the user can explore the reasons behind the similarities found in the data points and identify important relationships between variables.

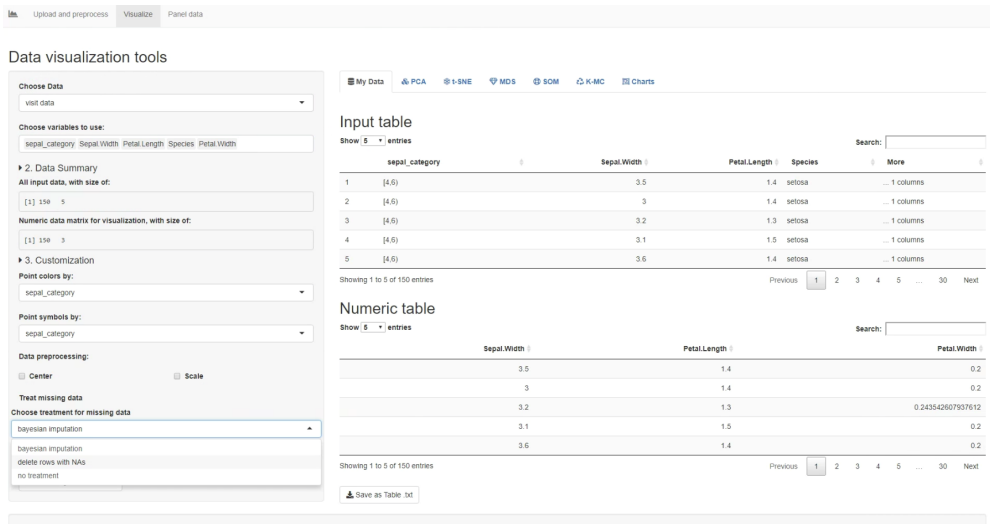


Figure 13. The Visualization Sidebar and tables in the Shiny app.

The Charts tab (Figure 16) includes a drop-down menu for choosing different types of charts for univariate, bivariate or multivariate plotting. According to the chart type to be visualized, more user input fields are activated for selecting which features to plot. The colors for the charts are set using the customization

on the sidebar. This functionality can use the table with missing values, because the plots delete the missing entries from the chosen features automatically.

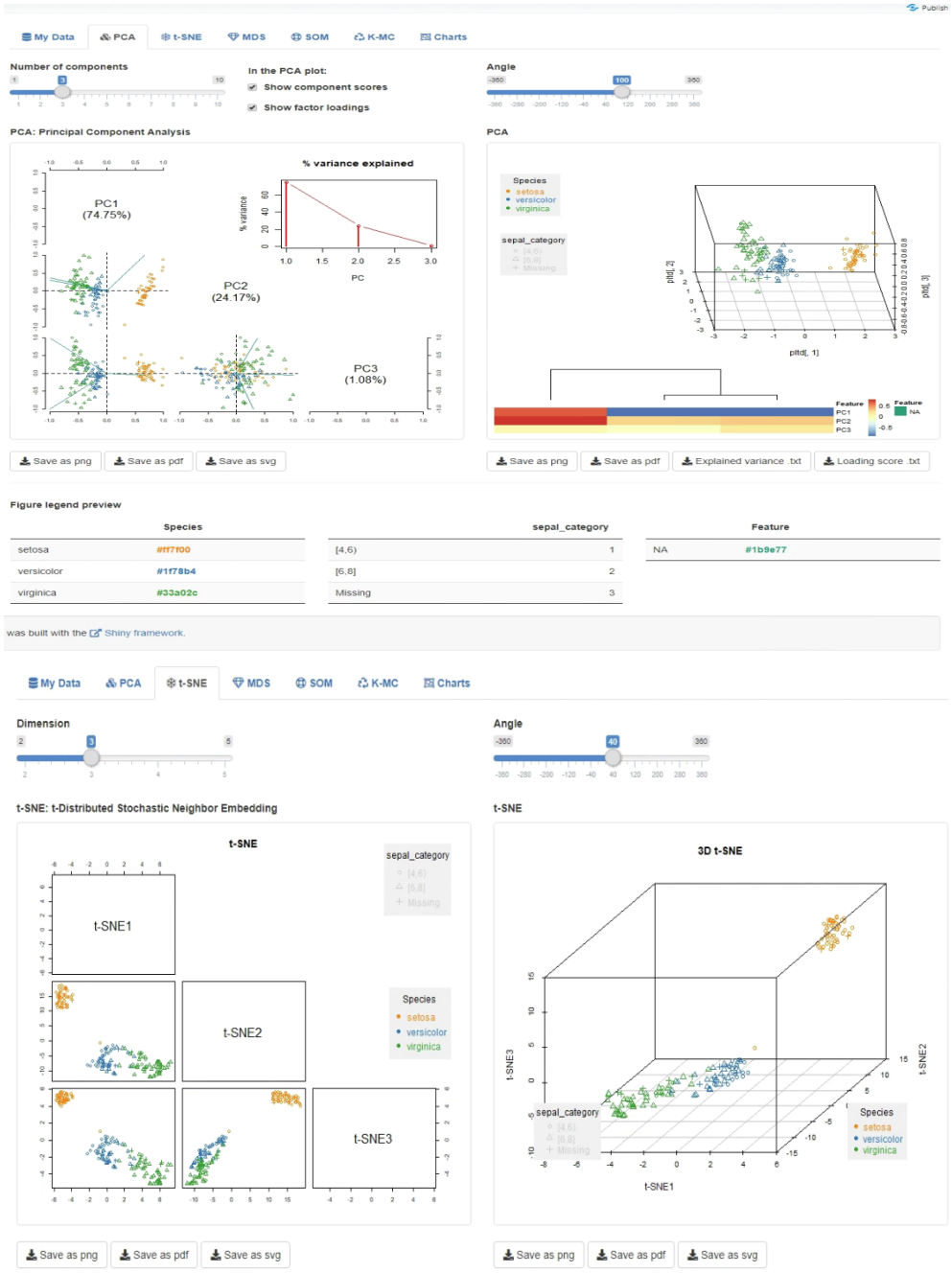


Figure 14. The clustering visualization methods tabs in the Shiny app for PCA and t-SNE.

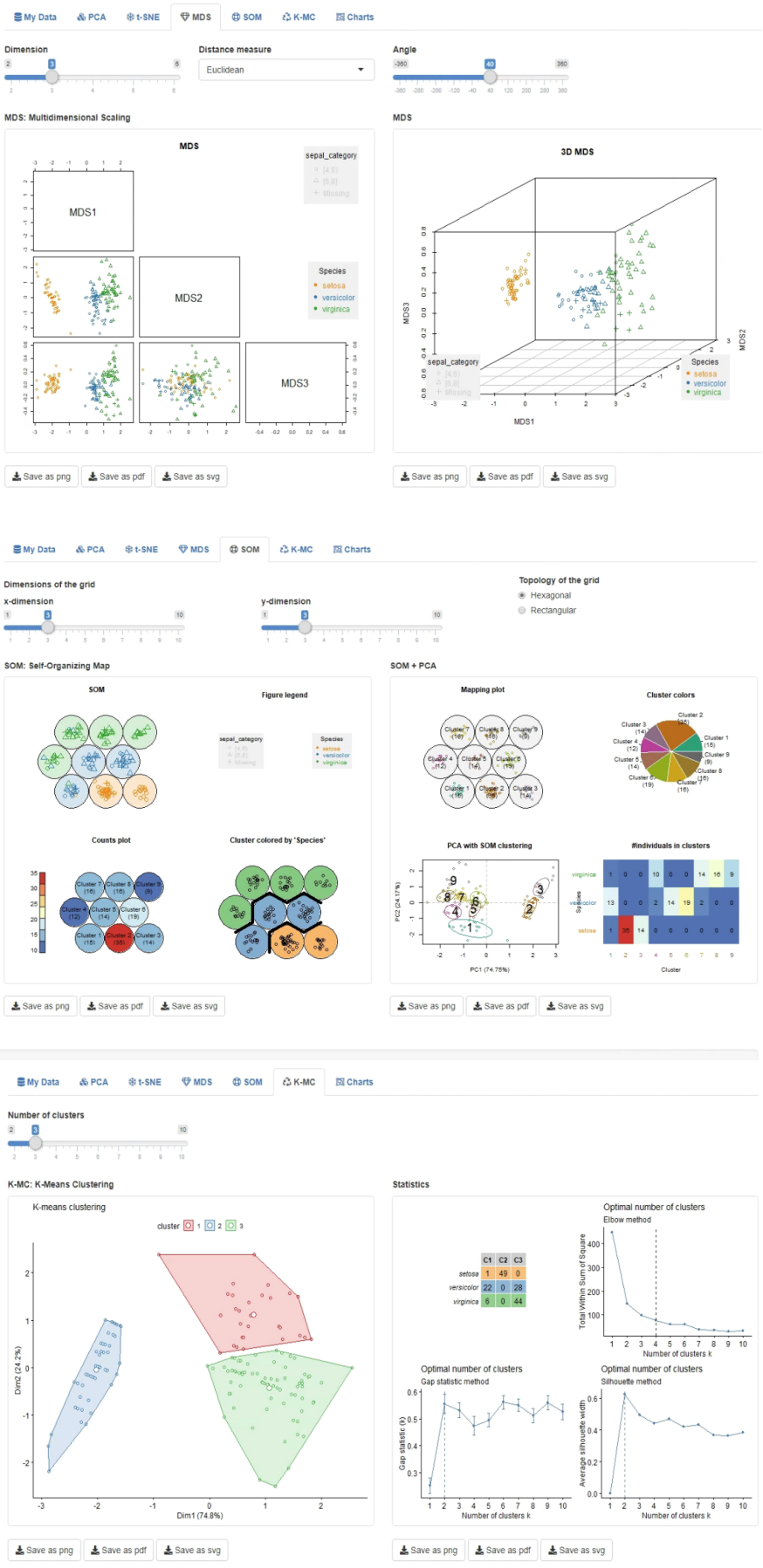


Figure 15. The clustering visualization methods tabs in the Shiny app for MDS, SOM, k-Means.

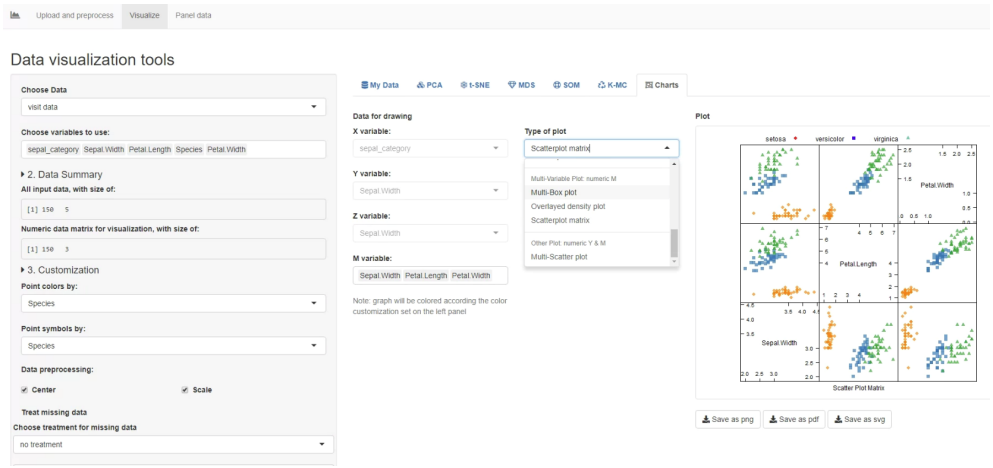


Figure 16. The Charts tab in the Shiny app.

4.3.3. Panel data

The last functionality in this application is a panel data creation tool that turns the visit age into a timestamp and allows the user to choose the variables and the functions to aggregate for each time point of each patient. The time points are introduced manually, separated by comma, and they do not have to be regular. The timestamp created is an ordinal factor. Other fields let the user choose which variables to aggregate by maximum, sum, mean and also add patient variables to the panel data. The constant patient values will repeat for each time point. The result is a structured panel with multiple measurements over time for each patient code (Figure 17). The panel data can be downloaded as a ".csv" table, and it can be uploaded directly into other tools, such as the panel data tool "ExPanD" [22] or used for time trend analysis, without any other preprocessing. Figure 18 shows an example of a time trend plot generated using panel data created with ClinFlow.

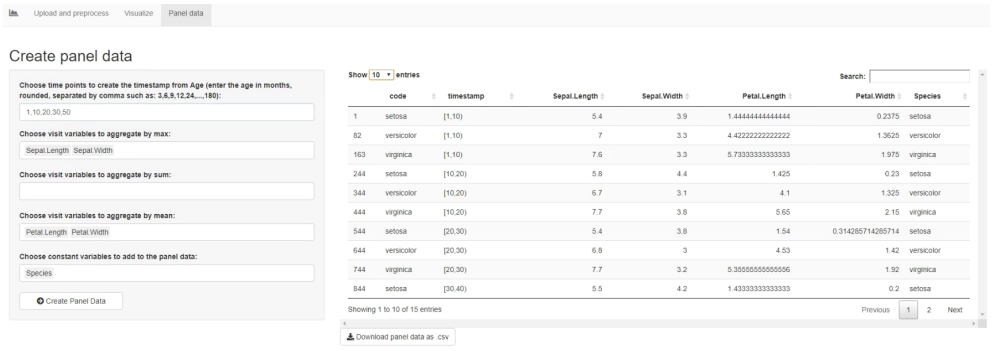


Figure 17. The Panel Data tab in the Shiny app.

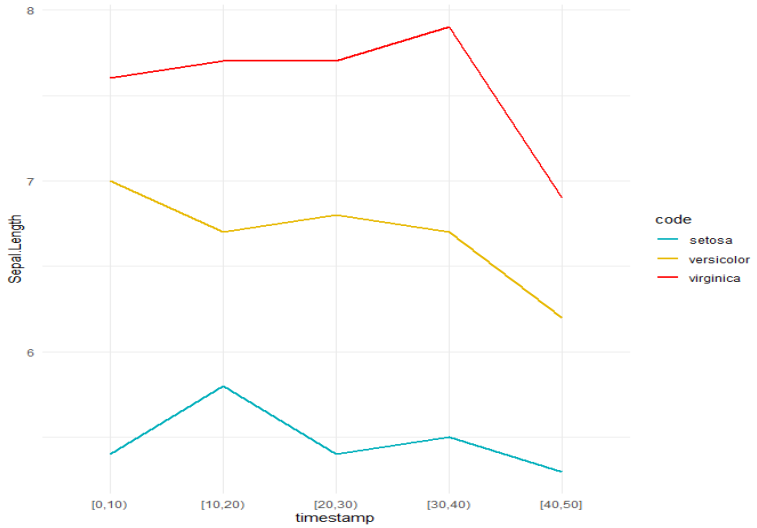


Figure 18. A time trend plot example that used the Iris panel data generated with ClinFlow.

5. THE DIPP CASE STUDY

Finland has the highest incidence of type 1 diabetes (T1D) in the world among children, the annual rate is currently 64/100,000 children under the age of 15 years [54]. At the Pediatric Diabetes Clinic, Oulu University Hospital, approximately 60 children with newly onset type 1 diabetes are diagnosed annually.

The Finnish Type 1 Diabetes Prediction and Prevention (DIPP) study [9] was launched in Finland in 1994 as a large scale observational follow-up population study, established in the university hospitals of Tampere, Turku and Oulu. The study was designed for improvement of strategies for T1D prediction and for the development of new techniques to prevent this disease. The data collected for this study consists of an intensive longitudinal follow up of children with genetic risk of T1D and their close relatives. There have been more than 220,000 newborns screened for genetic susceptibility of T1D (data as of June 2017). Children with an increased risk have had follow-up visits every 3 months for one year and then every 6-12 months until approximately 15 years old, or until diagnosed with T1D.

The DIPP study is different from the usual clinical trial because it monitors healthy children, and is focused on early prediction and prevention of a disease, rather than treatment outcomes.

In this chapter, we will identify the most prominent topics in T1D prevention research and present the DIPP case study. We will design a preprocessing framework for the DIPP data and we will analyze the DIPP data using the Shiny app, to replicate some of the results found in the reviewed literature.

5.1. Type 1 Diabetes

Type 1 diabetes is a chronic auto-immune disease characterized by the loss of insulin producing beta cells in the pancreatic islets. It has a higher incidence among those that are genetically susceptible. This disease can be held under control only with regular insulin injections. T1D may become symptomatic in the first years of life for some patients, while it might take more than 20 years with no symptoms for others [55].

Although the process of autoimmune destruction takes place in genetically susceptible individuals, the rapidly rising incidence strongly suggests that a combination of genetic [56], environmental and immunologic factors are involved in the pathogenesis of T1D. Some environmental factors included are certain viruses and early life diet and gut microbiota. For example, rubella [57] and enteroviruses [58] have been associated with an increased risk of T1D. Some infections have been shown to cause diabetes in animals [59], however, common vaccinations like MMR (vaccine for measles, mumps and rubella) have not caused a decrease in the T1D incidence. Enterovirus infection has been associated with an increased risk of T1D because in several studies enteroviruses have been found in the pancreatic tissue obtained from organ donors with T1D, and more recently, enteroviral structures have also been found from pancreatic biopsies of newly-diagnosed T1D patients. These viruses are known to damage cell functions through various mechanisms [60].

The diet and gut microbiota also play a role although it is not yet clear. Some studies [61] [62] have found that early introduction of cow's milk and cereals in the infant's diet together with a short breastfeeding period may lead to increased risk of T1D. However, prospective cohort studies have not confirmed these findings, and results from Trial to Reduce IDDM in the Genetically at Risk (TRIGR) [63], a large intervention trial, suggested that avoidance of cow's milk during the first eight months of life does not prevent from islet autoimmunity or development of T1D.

Maternal consumption of sour milk and red meat was related to increased disease risk [61] [64], but maternal consumption of root vegetables, potatoes, berries, fresh milk and cheese have been associated with a decreased risk [65, 66].

Early microbial exposures from pets have also been studied and an association between indoor dogs and a decreased risk of T1D has been found [67]. However, this finding needs to be confirmed in other populations. Some studies found evidence of an association between mother's age at birth and T1D [68]. According to the studies, a very small percentage of the increase in the incidence of childhood type 1 diabetes in recent years could be explained by increases in maternal age.

However, the onset of T1D is preceded by the appearance of islet autoantibodies detectable in the peripheral circulation. These autoantibodies may appear at birth in the cord blood, if they are transmitted from the mother, but children can start developing their own autoantibodies even as young as six months old [69], while the seroconversion rate in the general population peaks at around two years old [70].

There are five autoantibodies that are currently known to predict T1D. These include islet cell antibodies (ICA), insulin autoantibodies (IAA), autoantibodies to the 65 kDa isoform of GAD (GADA), the insulinoma-associated antigen (IA2A), and zinc transporter 8 (ZnT8A). Individuals who have positivity in at least two of the autoantibodies listed above have a 70% risk of developing T1D in the following few months to fifteen years [71, 72].

Some factors like a young seroconversion age, higher titres of ICA, IAA and IA2A at seroconversion and autoantibody multipositivity (positivity in more than one autoantibody at the same time) have been associated with rapid disease progression (1.5 years between seroconversion and diagnosis) [73]. Also, the season of birth has been found to have an association with disease progression. Slow progressors (>7.25 years between seroconversion and diagnosis) were born more frequently in the fall, whereas other progressors were born more often in the spring [74].

Recently, several large scale randomized controlled trials have been designed to prevent T1D. The European Nicotinamide Diabetes Intervention Trial (ENDIT) [75], The Diabetes Prevention Trial of Type 1 Diabetes (DPT-1) [76] and The Finnish Type 1 Diabetes Prediction and Prevention (DIPP) study [77] have used nicotinamide, parenteral insulin, oral insulin [78] and nasal insulin. Unfortunately, they did not prevent or delay the onset of T1D.

The factors that influence the risk of T1D, especially the mechanism of autoimmune destruction, have been the subject of many DIPP articles. The autoantibodies have an important role in the T1D disease progression, especially the positivity of multiple autoantibodies [72]. The associations between

seroconversion, autoantibody levels and disease progression have been widely explored as well [79, 73, 74].

In this case study, we used the Shiny app to inspect the DIPP data from Oulu University Hospital and the relationships between islet autoantibodies and progression to T1D. We tried to confirm the findings in the study of Knip et al, "Role of humoral beta-cell autoimmunity in type 1 diabetes" [79] that demonstrated which type of autoantibodies, autoantibody combinations and age at seroconversion have the highest risk of disease progression, and the article of Pöllänen et al, "Characterisation of rapid progressors to type 1 diabetes among children with HLA-conferred disease susceptibility", that demonstrated which factors are the most prominent at seroconversion in those children with a rapid disease progression.

5.2. Preprocessing of DIPP data

Since 1994 over 17 000 children with a genetic risk have had regular visits every three months for one year, then every 6-12 months until approximately fifteen years old. At each visit, standard clinical data such as weight and height measurements, questionnaires about diet and breastfeeding, and blood samples for measuring autoantibodies were collected. Before 2003, only ICA was measured, and if the subject became ICA positive, the blood samples from the past were analyzed for the IAA, IA2A, GADA and ZnT8A. After 2003, all samples were analyzed for all the autoantibodies [73].

The DIPP data is stored accross different sources and has several problems that can be present in other long-term clinical studies. These problems can be missing data due to patients dropping out of the study, protocol changes, corrections that need to be added to the original data, human errors, etc. Some of these issues can be detected and solved using the interactive data preparation options in the shiny app, while other more complex issues require the domain knowledge and knowledge about data collection protocols in this study. For the latter part, we created an automated preprocessing algorithm customized for this data only. The preprocessing is integrated into the tool in the way that once the data is uploaded in the app, it performs some general data cleaning operations such as deleting duplicated entries, then it checks for column names of the DIPP parameters and performs operations that are customized for these parameters only. The condition for this preprocessing algorithm to deliver clean DIPP data is that the uploaded raw data has the same column names and variable definitions presented in this chapter. If a column name is missing from the data, all the operations that depend on that variable are skipped. Part of this preprocessing algorithm can also be generalized to other medical data sets, for the variables that are more general such as "birth date" or "visit date", also similar rules and dependencies can be later added to better suit other datasets, which is why we have integrated it in the tool.

We have selected 33 parameters from the DIPP data to be used in our tool.These parameters have been identified from previous articles studying T1D in the DIPP study, for example blood samples with diabetes-related autoantibodies

[79], information about virus infections [57, 58, 59, 60], etc. We collected these parameters from various sources, merged them together in one ".csv" table and added a few corrections manually. These are introduced in the following subsection.

5.2.1. The data format

The patients in the DIPP data are identified by a unique combination of numbers and letters that will be referred throughout this thesis as the "patient code". Each medical visit has a visit date, and the combination of "patient code" and "visit date" is used to identify each hospital visit.

The DIPP data from the Oulu University Hospital was stored in four separate datasets from which we extracted the relevant information, matched by patient code and visit date, and dropped the irrelevant columns. The source datasets and their contents are listed below:

Dataset 1: Autoantibody values from blood samples together with some other variables that we are not using.

Dataset 2: Exported SQL table with background information of the patient (weight and height at birth, and in each visit, information about infections, pregnancy duration, breastfeeding, etc.).

Dataset 3: Table containing all the patient codes from the subjects diagnosed with diabetes, birth date and date of diagnosis.

Dataset 4: A curated table containing patient code, visit date and correct heights and weights from visits.

We took the following actions in order to successfully merge all the information in one table.

- In each dataset, dates have been checked and converted to the format dd/mm/yyyy
- Columns that contain the same information have been renamed with a name consistent accross all tables. For example, Dataset 1 had Finnish column names and Dataset 2 had English names for the birth date and visit date information,so we renamed all the columns in English.
- We replaced the visit height and weight from Dataset 2 with the curated ones from Dataset 4, keeping the correct info that we have and replacing the missing/incorrect values with curated values by matching them with patient code and visit date.
- In Dataset 3, we calculated the patient's age at diagnosis from date of diagnosis and birth date of the patient.
- We merged Datasets 1, 2 and 3 based on patient code and visit date.
- We created binary factor variable indicating a positive diagnosis of T1D, called "POS_diabetes".

- We dropped the variables irrelevant for our application. For example, date of diagnosis is dropped since we calculated the patient's age at diagnosis.

The final dataset contains the variables listed in Table 4. Each row contains patient code and visit date, patient data, and visit data. In the DIPP case, patient data and visit data are defined as follows:

- Visit data: Each row contains blood sample autoantibody values from one single visit and other measurements from one single visit like weight and height.

- Patient data: Birth information and other info that is not time variant and not connected to one single visit. Diagnosis information and age when diagnosed.

This data has several important characteristics that have influenced the design of the preprocessing framework. For each numeric autoantibody level in relative units (RU), the DIPP study has a cut-off value for positivity [72]. ICA has a cut-off value of 2 RU, GADA has 5.34 RU, IA2A has 0.42 RU. The IAA autoantibody has been measured with two different methods during the years. There are two variables: "mIAA_1.55" and "mIAA_3.47" that have a cut-off positivity value of 1.55 RU, respectively 3.47 RU. The IAA value is usually present in only one of the two columns, while the other has a missing entry.

Some subjects can have positive autoantibodies transferred from the mother, present at birth, in the cord blood or in the early life, that can last up to one year old [69, 81]. These have been proven not to have an influence in the risk of T1D [82], therefore, they need to be excluded from some analyses. It is a difficult task to differentiate between samples containing transferred autoantibodies and samples containing the subject's own autoantibodies in early life, when the subject might have both.

The patient code is a combination of numbers with a letter, where letter A stands for a child that has been monitored since birth, X stands for their mother, Y stands for their father, and B stands for their sibling. All family members share the same combination of numbers. However, we should note that not all family relationships can be identified in the data, because of the long-term nature of the study. For example, multiple generations from the same families might have been enrolled in the study as children and assigned a patient code with "A", and became parents later, having their own children enrolled as well with a patient code "A". The same person can appear as a child and later as a mother. Also children that are siblings, but each has been enrolled in the study at birth, have different patient codes containing "A" with no way of knowing they are siblings. The same mother might appear multiple times in the dataset under a different code matching each child. We treat each unique patient code as a different person, and only consider family relations between patient codes that match. The mothers and fathers in the dataset do not have follow up visits, they have only one visit with measurements taken at the birth of their child. Not all the children have mothers in the dataset.

The final dataset has 88,939 entries of raw, unclean data, that has several problems:

Table 4. Dataset variables and their explanations

Variable name	**Explanation**
birth_date	Date of birth.
code	Unique patient identification code.
date_of_visit	Visit date.
GADA_5.34 ICA_2 mIAA_3.47 mIAA_1.55 IA2A_0.42	Autoantibody cut-off values measured from blood samples taken each visit.
height weight circle_of_head	Patient measurements taken during the visit.
is_pets	A binary indicator variable for pets in the household (0 = no pets in the household, 1 = pets in the household). [67].
type_of_pets	Description of the pets – free text field.
infections_airway infections_ear infections_fever infections_gastric infections_eye infections_roseola infections_chickenpox infections_hospital_care infections_other infections_entero	A column for each type of infection with a numeric value indicating the number of infections occurred since previous visit.
birth_length birth_weight birth_circle_of_head	Dimensions at birth.
is_mom_t1d is_dad_t1d	Indicator variables for T1D positivity of the patient's mother and father(0=negative, 1=positive, 2=unknown). They contain many missing values.
duration	Pregnancy duration for each child – a free text field with inputs of the form "weeks + days", for example: "37+5", "38" or "38 + 0" (not structured) [80].
breastfeeding_only breastfeeding_ended	Age when the child stopped exclusive breastfeeding and age when the child stopped any breastfeeding.
POS_diabetes	Numeric column: 1 if the child has been diagnosed with T1D and 0 if not (or not yet).
diagnosis_age	Age when the child has been diagnosed with T1D. Contains missing values for the ones who have not been (yet) diagnosed with T1D.

- Columns are in the wrong format, for instance numerical values are stored in "string"-format.
- Some general patient information has been collected during one visit, but it belongs to the patient data and it is not time-varying. For example, the age when breastfeeding has ended appears only in one visit entry, and for all the others it is missing.
- Some patients might have two different values for the same variable (for example two different birth dates or birth weight) due to human error.
- Due to the two previous problems, there are duplicated entries (the same patient code + visit date) appears multiple times, with different info in a patient variable.
- Impossible entries (for example visit date is before birthdate).
- Some entries can be classified into two categories (positive/negative) based on a numeric threshold or formula which is not visible in the raw data.
- Other relevant information is not easily available for analysis and needs to be extracted first from the data.

5.2.2. Preprocessing pipeline

This automated preprocessing pipeline consists of a set of nested functions, applied directly on the data, to tackle with the problems in the raw dataset. It can be split in three stages: data cleaning, data transformation and feature construction.

Data cleaning

This stage contains cleaning and normalization operations. The following list describes in detail and motivates each cleaning action taken on the DIPP data.

1. Variables have inconsistent formats. For example, autoantibody values are stored in character form, although they are numeric values.
    - We converted all variables to appropriate class: numeric, factor , character, date.
2. Birth date is not constant for all visits of the same participant. Some have two different birth dates (most are one day apart. One is a year apart).
    - We replaced missing values in the birth date variable with the value that is present in other visit rows.
    - We chose (randomly) only one birthdate for the ones that have multiple birthdates one day apart.

- For the one patient with two birthdates one year apart, because the visits started later than both birthdates, we chose the birthdate closest to the first visit date.

3. Breastfeeding ending age is inconsistent, with missing values in some visit rows while present in other visit rows for the same participant.
    - We replaced all the missing values with the value present in other visit rows for each participant.
4. Some patients have two or more different breastfeeding ending ages.
    - We replaced all values with the maximum breastfeeding ending age present for each subject.
5. Some visit rows show that exclusive breastfeeding ended later than non exclusive breastfeeding (not possible).
    - For these subjects, we chose the earlier value in both exclusive and non exclusive breastfeeding.
6. Mother diabetes and father diabetes columns are inconsistent and have missing values.
    - We converted to factor columns, with TRUE for the visit rows of patients where it's clear that the mother or father have diabetes (value = 1), and FALSE for the ones with "unknown", "0" or missing value. Here, we assumed that in the context of a T1D study, if the parents would be diagnosed with T1D, they would have definitely mentioned it. If this information is missing or unknown, it is most likely because they don't have T1D.

Data transformation

In this stage, we converted character variables that are difficult to read into categorical variables that are easier to read. Depending on the variable and considering the domain literature, appropriate categorization of the variable has been done, while keeping the original features as well. The following list describes each transformation applied to the DIPP variables.

1. Autoantibody values have threshholds for positive/negative values but they are not obvious in the dataset.
    - Created factor variables for each autoantibody TRUE/FALSE if the value goes over the positive threshold.
2. Pregnancy duration column is in character form, with duration as weeks + days.
    - We converted the variable "duration" into a factor column with three levels: "premature", "normal", "prolonged".

- We performed text parsing for the number of weeks (<37 - premature, >41 - prolonged).

3. The month of birth might have influence on diabetes risk[83].
   - We created month of birth factor column "birth_month" from the birth date.
4. Pets variable is in free text format.
   - We created a visit variable "pets" factor with three levels: "cat/dog", "other", "no pets".
   - If in the type_of_pets we find the following string patterns : "koir" or "kiss", we assign factor "cat/dog" (because in this dataset, most of them appear together in the same entry).
   - If in the type_of_pets there is missing value or empty string "" and in the variable "is_pets" is 0, we assign "no pets".
   - If in the type_of_pets there is anything other than the text patterns "koir" or "kiss", including missing value or empty string "" and in the variable "is_pets" is 1, we assign "other".
   - If in the variable "is_pets" is value 2 or missing and in the type_of_pets is missing or empty string "", we assign NA (we don't know if there is a pet or not) [67].
5. Dataset contains children monitored from birth, their parents and their siblings in the same set, with different letters in the patient code but no other clear distinction between them.
   - We created a variable named "who" with four categories: "child" - for children monitored since birth, "mother", "father", and "sibling" - for the siblings of children monitored from birth, by text parsing the patient code. Here, we know that the children monitored from birth are most likely to have the information from early life.

Feature construction

This stage is the most important and complex. While the tool also enables some user-defined feature construction operations, we have constructed some ready-made features that are relevant for the DIPP study based on domain literature. Some of them are derived from variables that are most prevalent in clinical data, but most of them are specific to the DIPP study. Some of these feature construction techniques can be later generalized to other datasets of the same type. For the new more complex variables that consider autoantibody values, we need to take into account and exclude from calculations the positive autoantibodies transferred from the mother or present in the cord blood. For this, we have assumed that a positive sample has autoantibodies from the mother or cord blood if it meets the following criteria:

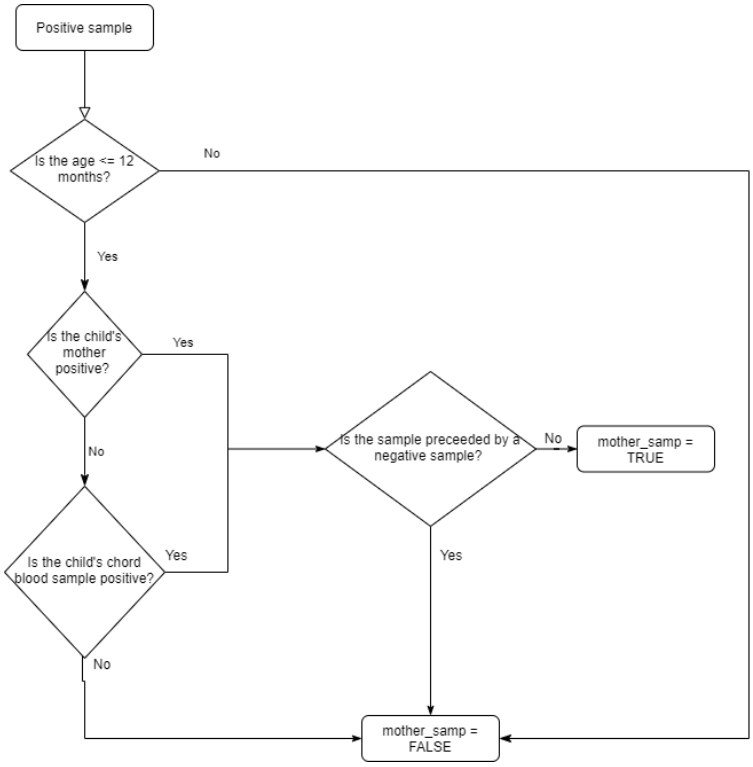


Figure 19. Flowchart of detection of samples with positive autoantibodies transferred from the mother.

1. The child's mother is present in the dataset and has positive autoantibodies OR the child's cord blood sample is present in the dataset and is positive.
2. The sample is from before the age of one year old.
3. The sample is not preceded by a negative sample.

We consider positive, any sample that has one or more positive autoantibodies. We created a variable called "mother_samp" that marks TRUE for the samples meeting the above criteria, and FALSE for all the other samples. A logical scheme of this process is presented in the flowchart in Figure 19.

The following list describes each feature construction technique applied on the DIPP data.

1. Age at the visit is not present in the data.
   - We calculated age at that visit with birth date and visit date.
2. Maximum follow up time is important. Data should be compared from patients with similar follow up times.
   - We created a new variable called "Age_follow_up" that marks the maximum visit age of each patient.
3. Disease progression time is important [73, 74].

- We created a variable "progression_time that marks the time passed between the seroconversion and the diagnosis dates. For the patients who never got diagnosed with T1D the progression_time is NA.

4. Age of the mother at birth could have an impact on analysis [68].
    - We calculated age of mother at birth from the mother's birth date and her (corresponding patient code) child birth date, and created a variable "Mom_birth_age"
5. Blood sample from a visit is positive if any >=1 of the autoantibodies is positive.
    - We created a binary factor visit variable "pos_sample" to mark if the sample(row) is positive in any >=1 antibody.
6. Two consecutive positive samples in any >= 1 antibody for a patient is considered true positivity [73]. For each individual patient code:
    - We excluded positive samples that were identified as transferred from the mother.
    - If at any point there are two or more consecutive TRUE visit rows in any of the four columns with antibodies, we assign a factor variable "POS_antibodies" = TRUE for that patient.
7. Age of seroconversion has an importance in analyses [73, 79]. We found the first positive sample in any >=1 antibodies and retrieved age from that visit as follows:
    - We excluded positive samples that were identified as transferred from the mother.
    - We created a variable called "Age_seroconv" that contains the age of the first positive sample.
    - For the patients who never had a positive antibody sample, Age_seroconv is NA.
8. Seroconversion type is important [73]. Are multiple autoantibodies present at seroconversion and which ones?
    - We created a factor variable to mark whether only one antibody was positive at seroconversion, or multiple, and whether it was IAA or other[79]. Factor levels are: "ICA","GADA","IAA","IA2A","multipositivity IAA" for the ones with IAA combined with other autoantibodies at seroconversion, and "multipositivity" for the ones with 2+ positive antibodies at seroconversion but no IAA.
9. The type of positivity after seroconversion is important [73].

- We excluded positive samples that were identified as transferred from the mother.
- We created a factor variable "positivity_type" with three levels: "single" for patients with at least two consecutive positive samples in only one autoantibody at a time, "multi" for patients with at least two consecutive positive samples in two or more autoantibodies in the same time, and "negative" for patients with one or zero positive samples in any autoantibodies.

10. Antibody titre at seroconversion for each autoantibody is important [73].
    - We retrieved the antibody value from visit rows where visit age = seroconversion age.
    - We created a new titre variable for each autoantibody that contains the value of that autoantibody at seroconversion, regardless of the seroconversion type.
    - Patients who never seroconverted have NA in these variables.

The output of this preprocessing algorithm contains two datatables: Visit data and Patient data. Patient data is separated from the visit data as Visit data is time series data which requires different analyses and further preprocessing, for example, into panel data or survival data. Visit data can contain the patient data as well, but patient data can only contain aggregated visit data (summarized for a certain period), for example, a summarized patient variable that states TRUE if the child has had a pet in the first twelve months of life. This can be done by the user of the data according to the research question. Adding this type of new constructed features to the datatable gives the user a possibility of easily splitting the data into groups or perform comparative analyses, while the original information is still available for checking and/or further processing. The final data variables and their definitions are presented in Tables 5 and 6 for Patient data and Visit data respectively.

5.3. Analysis of DIPP data with ClinFlow

From the preprocessed dataset we have extracted the children and their siblings, using the Filter Data tab in the app, then, using the Outliers tab we have plotted the follow up age against the "POS_diabetes" variable and we selected and deleted all entries that were not diagnosed with T1D and the follow up age was below 180 months old, or fifteen years old. We were left with 1,196 patients, 186 who progressed to T1D and 1,010 who have had follow up visits until at least fifteen years old and have never been diagnosed with T1D.

We previously defined the positivity type as "multi" if the patient had persistent positivity in two or more autoantibodies, "single" if the patient had persistent positivity in only one autoantibody at a time, and "negative" if the patient did not have persistent positivity. Persistent positivity is defined as two or more consecutive positive samples.

Table 5. Patient Data variables and their definitions

Variable name	**Variable Type**	**Variable definition**
code	Factor	Unique patient code
birth_date	Date	Patient birth date in the format of dd/mm/yyyy
birth_length birth_weight birth_circle_of_head	Numeric	Length at birth in cm. Mothers and fathers have NA. Weight at birth in g. Mothers and fathers have NA. Head circumference at birth in cm. Mothers and fathers have NA.
duration	Factor	Gestation period for each child with levels: "premature", "normal", "prolonged"
is_mom_t1d is_dad_t1d	Factor	Parent's diabetes for each child's parents: "TRUE" or "FALSE"
breastfeeding_only breastfeeding_ended	Numeric	Age in months when exclusive breastfeeding respectively any breastfeeding has ended.
POS_diabetes	Factor	TRUE if the child has been diagnosed with T1D. All the others are FALSE.
diagnosis_age	Numeric	Age in months when the child has been diagnosed with T1D. The ones with POS_diabetes = FALSE have NA.
mother_pos	Factor	TRUE if the child has a mother with positive autoantibodies in the data. All the others have FALSE.
Mom_birth_age	Numeric	Age at birth, in years, for the children's mother present in the dataset. If the child's mother is not in the data, they have NA.
POS_antibodies	Factor	TRUE if the patient has had at least two consecutive positive samples in any >=1 autoantibody at any point in their life, excluding the autoantibodies transferred from the mother. Otherwise, FALSE.
birth_month	Factor	The name of month of birth.
Age_seroconv	Numeric	Age at seroconversion, in months, for the patients who had at least one positive sample in any autoantibody, except samples transferred from the mother. The others have NA.
GADA_5.34_titre ICA_2_titre mIAA_3.47_titre mIAA_1.55_titre IA2A_0.42_titre	Numeric	The value of each autoantibody at the first positive sample, for the ones who had at least one positive sample of any autoantibody, except in herited samples. The others have NA.
seroconv_type	Factor	Mentions which autoantibody was positive at seroconversion, or if there were multiple. Levels are: "GADA", "ICA", "IAA", "IA2A", "multipositivity". The ones who never seroconverted have NA.
positivity_type	Factor	"multi", for the ones with POS_antibodies = TRUE, that has two or more positive autoantibodies in the same time, for at least two consecutive samples, "single" for the ones with only one positive autoantibody in at least two consecutive samples. All the others are "negative".
who	Factor	Variable with levels "child", "sibling", "mother", "father".
progression_time	Numeric	Number of moths between the seroconversion date and the diagnosis date, for the ones who have POS_diabetes = TRUE. The others have NA.
Age_follow_up	Numeric	Maximum age in the dataset for each child, in months.

Table 6. Visit Data variables and their definitions

Variable name	**Variable Type**	**Variable definition**
code	Factor	Unique patient code
date_of_visit	Date	Visit date in the format of dd/mm/yyyy
Age	Numeric	Age at the visit.
GADA_5.34 ICA_2 mIAA_3.47 mIAA_1.55 IA2A_0.42	Numeric	Autoantibody values. Samples from before 2003 that had negative ICA have NA in the other autoantibodies.
GADA_POS ICA_POS IAA_POS IA2A_POS	Factor	TRUE if the autoantibody is positive, otherwise FALSE.
pos_samp	Factor	TRUE if one or more autoantibodies are positive in this sample. False if they are all negative.
mother_samp	Factor	TRUE if the positive sample is transferred from the mother. False if otherwise.
height circle_of_head weight	Numeric	Height at the visit in cm. Weight at the visit in kg. Head circumference at the visit in cm. Head circumference is measured until 2 years old, after that it is NA.
infections_airway infections_ear infections_fever infections_gastric infections_eye infections_roseola infections_chickenpox infections_hospital_care infections_other infections_entero	Numeric	Number of infections.
pets	Factor	Levels: "cat/dog" if the patient owns a dog or a cat at the time of the visit. "other" if the patient doesn't have a dog or cat but has other pets, and "no pets" if the patient does not have any pets. NA if it is unknown.

Out of the total number of patients, 178 have progressed to multipositivity, with persistent positivity in two or more autoantibodies at the same time, and 189 have had single positivity - persistent positivity in only one autoantibody at a time. Out of 828 children who were found negative, 33 seroconverted, meaning they had at least one positive sample recorded, but no persistent positivity.

After these operations, the data showed that 74.16% of the children who presented multipositivity and only 5.82% of the ones with single positivity progressed to T1D. As for the ones that tested negative, with only one or no positive samples, 5.19% progressed to T1D. This is quite a high percent, compared to other studies [79], so we decided to investigate them further, and check the correctness and completeness of these entries. Using the Filter Data tab, we subsetted 43 patients that appear negative but they progressed to T1D. Out of these, 25 appear that they never seroconverted to positive autoantibodies, and 18 appear that they seroconverted, but no persistent positivity. Using the Outliers tab, we checked if they have been followed up until diagnosis, or if they dropped out of the study and haven't been followed up until diagnosis. 24 children with gaps of 20-150 months between the last visit and the diagnosis were discovered and eliminated from the analysis (Figure 20). Then, from the Visit table, we plotted each visit age against the diagnosis age, and found six more patients with gaps between the second to last and the last visit of more than 20 months. They were eliminated as well, one by one, using the Outliers Interactive Plot, by selecting them from the plot and checking the entries in the table (Figure 21).

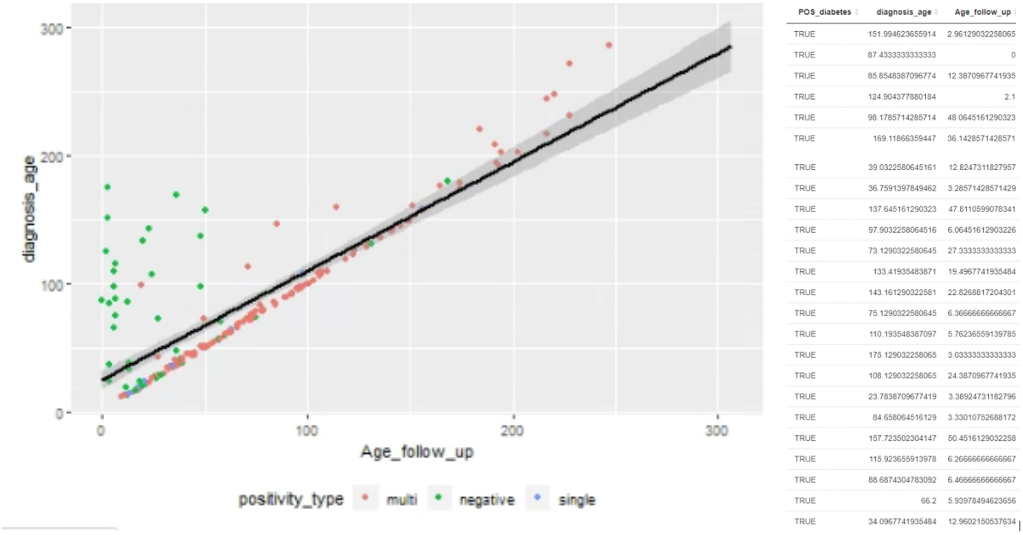
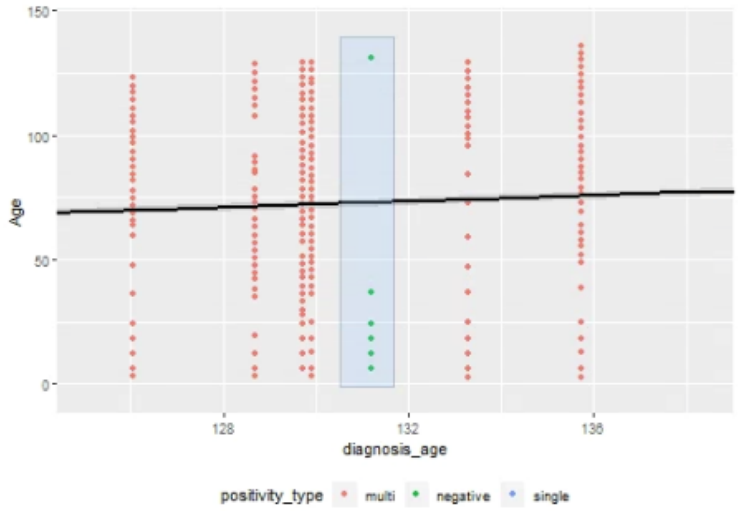


Figure 20. Entries that have had a gap between the last visit and diagnosis (Green points on the left side of the plot), and their corresponding follow up age (months) and diagnosis age (months) table entries.

After these operations, we were left with thirteen patients that have negative autoantibody values and progressed to T1D. They appear as though they progressed in less than one year, and they either haven't been recorded or haven't had multipositivity before diagnosis. We kept these in the analysis. From the 1,166 patients left after the filtering and eliminating the outliers, 392 seroconverted and 156 progressed to T1D. 84.6% of the progressors have had



Age	Age_follow_up
5.95161290322581	131.193548387097
11.8494623655914	131.193548387097
18	131.193548387097
23.9161290322581	131.193548387097
36.6290322580645	131.193548387097
131.193548387097	131.193548387097

Figure 21. Patient visit age plotted against the diagnosis age, of all the patients diagnosed with T1D. Each vertical line represents a patient and each dot is a visit. All patients had multipositivity except for the green one, who was negative. We can observe that this patient had a gap in the visits. Between 36.6 months old until 131 months old this patient had no visits.

multipositivity, 7.1% had single positivity, and 6.4% were recorded negative, with only one positive sample, but no persistent positivity. Only three patients making up 1.9% of the progressors appear as they never seroconverted. From the 1010 non-progressors, 77.8% appear negative, with either no positivity or non-persistent positivity, 17.6% have been recorded with single positivity - persistent positivity in only one autoantibody, and only 4.6% have had multipositivity (Figure 22).

Out of 392 patients who seroconverted, 46% had ICA seroconversion type, followed by 23.6% with multipositivity IAA. 12% had only IAA seroconversion, 9.4% had GADA as the only positive autoantibody at seroconversion, 9% had multipositivity without IAA and none of them seroconverted to only IA2A.

We studied the relationship between autoantibodies and T1D progression with a PCA clustering analysis on a subset of 257 patients who had at least one positive sample in any autoantibody, using the scaled birth dimensions, autoantibody titres at seroconversion, the age at seroconversion and new patient variables created with the Create New Patient Feature tab, that contain the maximum recorded value for each autoantibody during the visits of each patient, excluding the autoantibodies transferred from the mother. We chose these numeric variables due to the fact that they don't contain as many missing values as the rest of the numeric patient variables. Using the Create New Categorical Feature tab, we split the variables used for clustering into categories in order to color the clusters by category using the customization options that allow coloring the points in the plot by a categorical variable, and check whether the autoantibody levels at seroconversion, the seroconversion age, the maximum autoantibody values recorded and the seroconversion type have an influence on T1D progression or multipositivity.

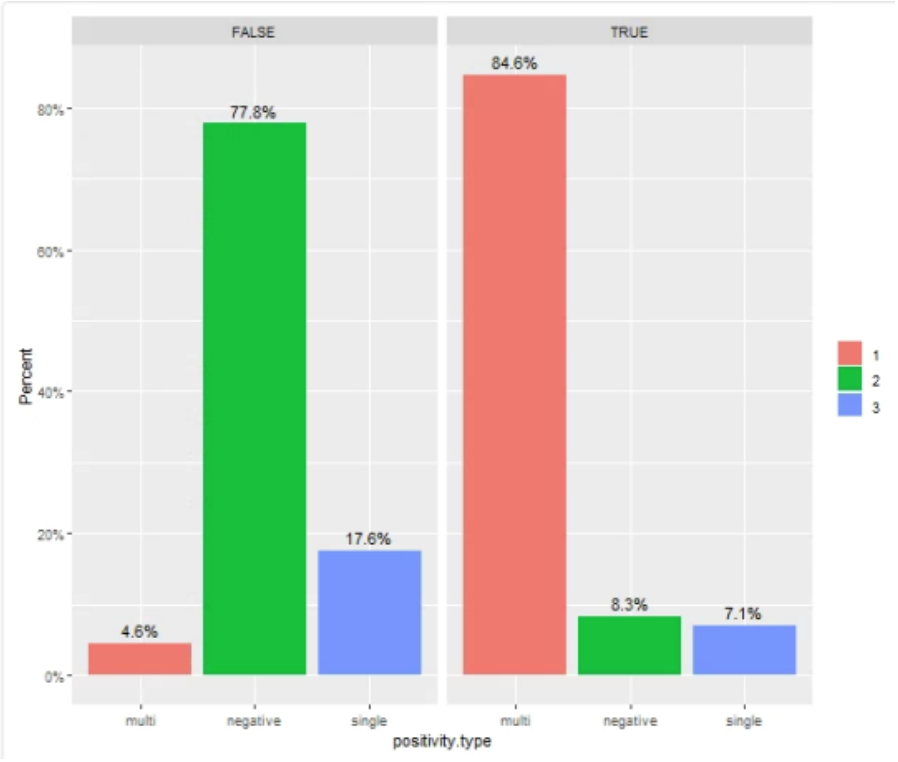


Figure 22. Non progressors (left) vs. progressors(right) by positivity type.

We plotted the first three principal components against each other, and coloured the plot according to different categories. From Figure 23, we can observe that the data is clustered in two main clusters. The shape of the points in the plot stands for progressors - circles and non-progressors - triangles. We coloured the points according to positivity type and we can see theassociation between multipositivity - yellow, and T1D progression. These plots already show that the values of the numeric variables used for clustering have an influence on multipositivity and diabetes progression.

To study how the values affect these clusters, we created categorical variables that split the numeric values into intervals, and we coloured the points according to the interval that they belong to. For the maximum values of GADA, IA2A and IAA autoantibodies we chose cut-off points at the threshold for positivity and at the value of third (highest) quartile for each autoantibody. For ICA, we chose an extra cut-off point at the median, because the ICA values have a much higher range. While the other autoantibody values go up to maximum 1,200 RU, the ICA values can reach up to over 2,600 RU. Figure 24 shows that most people who had multipositivity and progressed to T1D also seroconverted young, before 72 months, or six years old. They also had mostly multipositivity with IAA at seroconversion, followed by multipositivity without IAA, followed by GADA. Most patients who seroconverted first to ICA, did not progress to multipositivity and T1D. The maximum values of autoantibodies seem to affect the same way as long as they are positive, except for ICA. A positive value of ICA (>=2) is found in most non-progressors, however, higher values of ICA are found in progressors (Figure 25).

The PCA clustering shows the relationship between multipositivity, T1D progression, early seroconversion, multipositivity at seroconversion and high autoantibody titres. There is no obvious relationship between the birth dimensions and multipositivity or T1D progression. However, the clustering methods only used a subset of the data, with no missing values in any of the numeric variables. These relationships can be investigated in more detail using the Charts tab, with the whole dataset. The charts will only ignore the entries with missing values in the plotted variables, therefore, they use more data. In the dataset, there are 392 patients who seroconverted. Out of these, 178 progressed to multipositivity, 189 had single positivity and 25 are negative, which means that they had positive samples, but no persistent positivity. The seroconversion age ranges from 1.9 months up to 298.1 months which is approximately 24 years old, with a mean of 80 months or 6.6 years old. 44% of patients who developed multipositivity had multipositivity with IAA at seroconversion, followed by 20% with multipositivity without IAA. Figure 26 shows that patients who progress to multipositivity have an earlier seroconversion. The mean age of seroconversion is 47.4 months, which is approximately 3.6 years old for patients with multipositivity, and 109.4 months or approximately nine years old for patients with single positivity. Also, it appears that patients who have the IAA autoantibody at seroconversion have an earlier age of seroconversion than the ones without IAA. Patients who seroconverted first to ICA have the highest seroconversion age. The mean age for multipositivity with IAA at seroconversion is 28.5 months or 2.4 years old, and for the other types of seroconversion, it is 94.2 months which is approximately 7.8 years old.

The GADA titres at seroconversion have a weak positive correlation with the seroconversion age, for patients who progressed to multipositivity, with a correlation coefficient of $R = 0.29$ and a p value of 0.0001. This means that with a later seroconversion age, there are higher GADA titres for multipositive patients. The IAA titre has a weak negative correlation with the seroconversion age, with a R of -0.37 and a p value of 0.0002 for multipositive patients and a stronger R of -0.68 with $p = 0.003$ for negative patients (Figure 27). We verified this finding further by using the Filter Data tab to check group summary statistics. A group of 90 multipositive patients with GADA seroconversion titres belonging in the lower 75% of the whole population have a mean age of seroconversion of 33.4 months or 2.7 years old. 88 multipositive patients with GADA seroconversion titres in the third quartile of the whole population have a mean seroconversion age of 61.8 months or five years old. 167 multipositive and negative patients with IAA seroconversion titres in the lower 75% of the population have a mean seroconversion age of 59.4 months or five years old, whereas the patients with high IAA titres belonging in the third quartile of the whole population have a mean seroconversion age of 33.8 months or 2.8 years old. There have not been any associations found between the other patient variables, enterovirus infections or pets and multipositivity in this sub-sample of the data.

We also studied the progressors and the association between the autoantibodes and the disease progression time defined as the time that passed between seroconversion and diagnosis, in months. There are 153 progressors in the dataset, with the disease progression time ranging from 0 to 204.58 months

which is approximately 17 years, with a median of 49.7 months, or 4.1 years. The progression time of 0 corresponds to the negative patients, which means they have been diagnosed at their first visit, when they had their first positive sample. The progressors have seroconversion ages between three months old up to 186 months or 15.5 years old, with a median of 24 months old or two years old. We split the seroconversion age into four intervals, with cutoff points at the first quartile, median and third quartile. The lowest seroconversion age interval, between three and thirteen months old, has a mean disease progression time of 43.65 months. The highest seroconversion age interval, between 48 and 186 months, has a mean progression time of 58.8 months.

Patients who seroconverted multipositive with IAA have the most rapid progression time, with a mean of 41.2 months, followed by multipositivity without IAA, with a mean of 54.9 months and IAA seroconversion, with a mean of 58.1 months. Patients who seroconverted first to GADA have a mean progression time of 59.6 months, and the ones who seroconverted first to ICA have the slowest disease progression, with a mean value of 68.1 months. Figure 28 shows the difference in progression time for the seroconversion types and seroconversion age intervals.

All autoantibody titres have a weak negative correlation with the progression time, with correlation coefficient R ranging from -0.22 to -0.29 but with statistically significant p values, except for GADA, that has R = -0.06 and non-significant value of $p = 0.42$ (Figure 29).

These findings underline the importance of studying the association between early multipositive seroconversion combined with high IAA, IA2A and ICA autoantibody titres at seroconversion and a rapid disease progression. The GADA titre at seroconversion and the other patient variables did not show a significant association with the disease progression time.

Summary

The study of Knip et al. [79] of over 7,000 children recruited by the DIPP study [79], found that most progressors had multiple autoantibodies already in the first positive sample. This study found that the appearance of IAA autoantibody had a peak around the second year of life, whereas GADA emerged around the fourth and fifth years of life. A young age of seroconversion is associated with a higher risk of T1D progression. The positivity for multiple autoantibodies is associated with a risk of around 70% for progression to T1D in the following ten years, and single positivity does not lead to T1D progression and is harmless.

The article of Pöllänen et al. [73] on disease progression, including 7,400 children from the DIPP study, found that rapid progressors have a younger age, higher ICA, IAA and IA2A titres at seroconversion, and multipositivity.

In the analysis, we found a strong relationship between persistent multipositivity and progression to T1D, and no relation between single persistent positivity and progression to T1D. An association was found between multipositivity at seroconversion and progression to persistent multipositivity and T1D. The analysis showed that the IAA autoantibody appears earlier then the other autoantibodies, with a mean age of appearance of 2.5 years old. GADA as

the first autoantibody appears later, with a mean age of appearance at 5.4 years old. The age of seroconversion for patients who progressed to multipositivity is on average 5.1 years earlier than for patients with single positivity. These findings seem to support the results of Knip et al [79].

Weak correlations have been found between IAA and GADA titres at seroconversion and the age at seroconversion. Higher titres of GADA seem to be related to a later seroconversion age for patients who progressed to multipositivity, and higher titres of IAA at seroconversion are associated with an early seroconversion age. These findings are not reported in the study of Knip et al. [79], therefore, they will need to be investigated in more detail and validated on larger datasets.

Studying the disease progression time, we found that patients with a younger age and multipositivity at seroconversion had a shorter disease progression time. A weak negative correlation was found between ICA, IAA and IA2A autoantibody titres at seroconversion and disease progression time, but not GADA. These findings support the results of Pöllänen et al. [73].

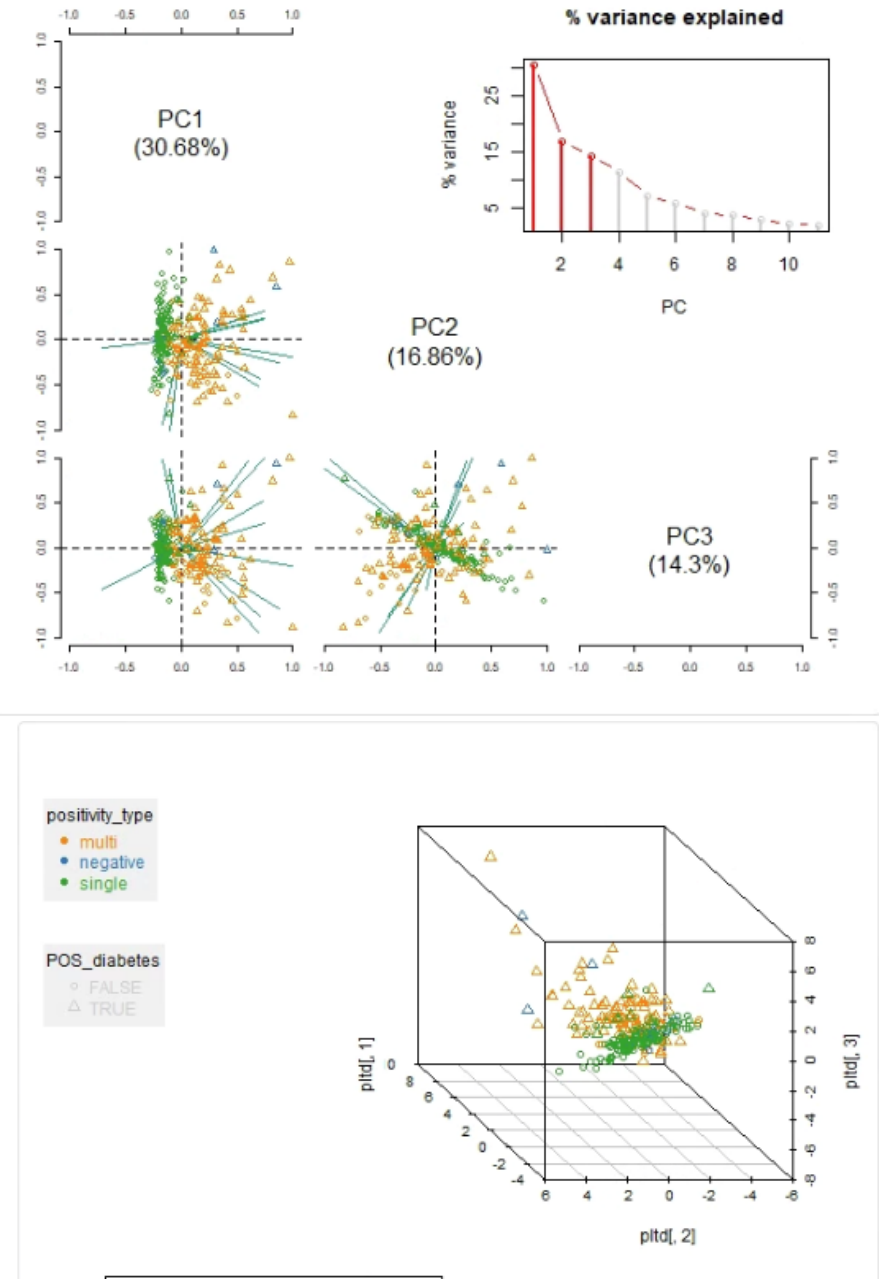


Figure 23. PCA clustering using birth dimensions, autoantibody titres at seroconversion, age at seroconversion and autoantibody maximum values, colored by positivity type and shapes of the points according to T1D progression.

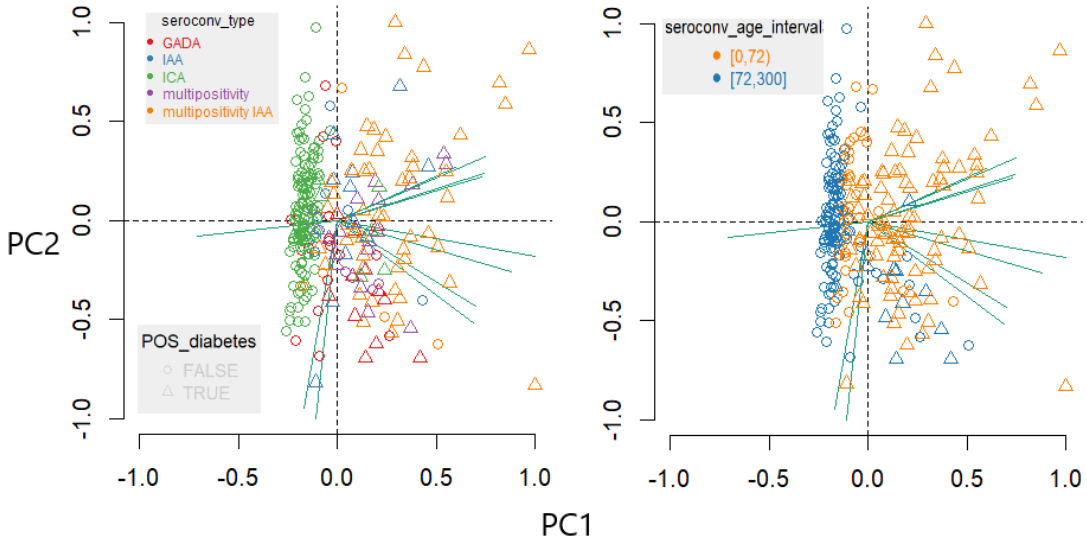


Figure 24. First two principal components plot colored by seroconversion type (left) and seroconversion age (right). Shape of the points according to T1D progression

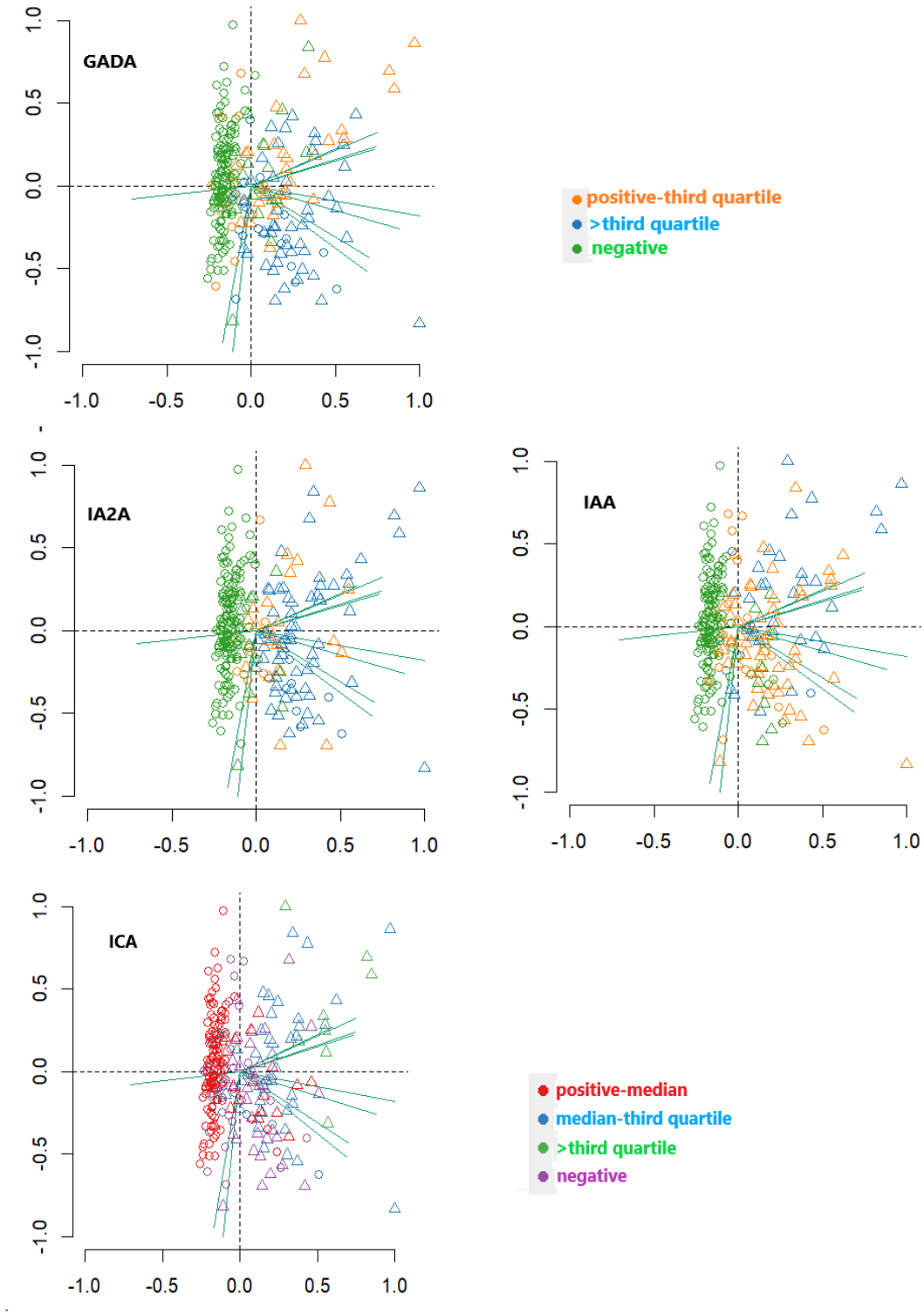


Figure 25. First two principal components plot colored by each autoantibody maximum level. Shape of the points according to T1D progression.

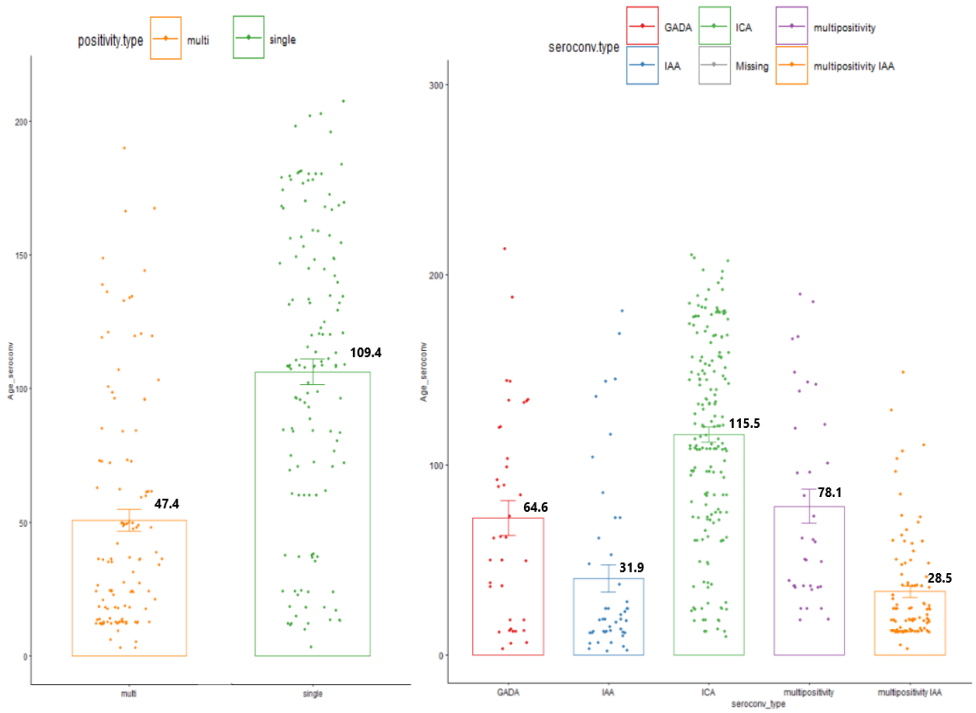


Figure 26. Age at seroconversion (months) for each positivity type (left) and seroconversion type (right). The error bard represent the 95% confidence interval of the mean.

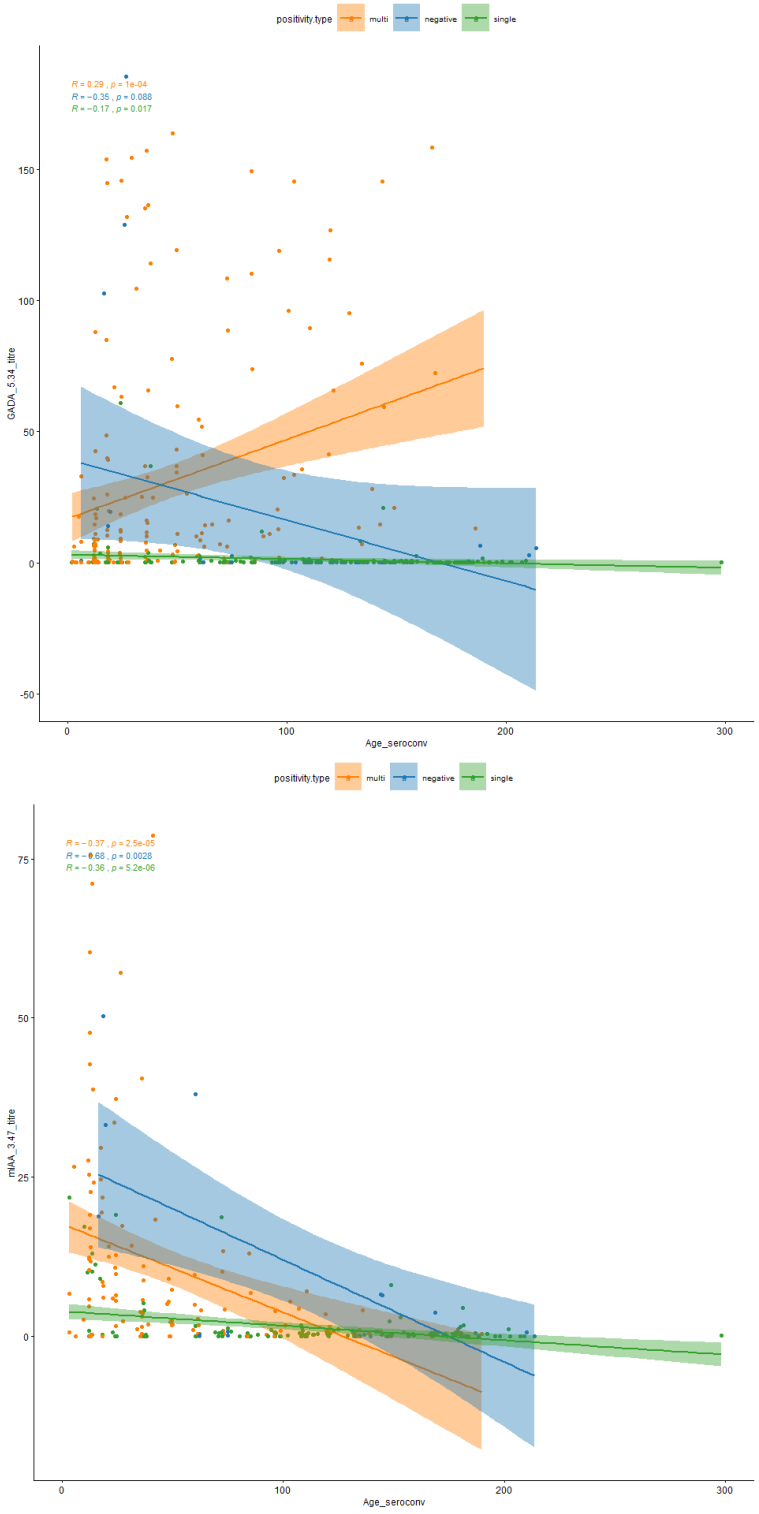


Figure 27. Correlation between the seroconversion age (months) and GADA titre at seroconversion (left) and correlation between the seroconversion age (months) and IAA titre at seroconversion (right), colored by the positivity type.

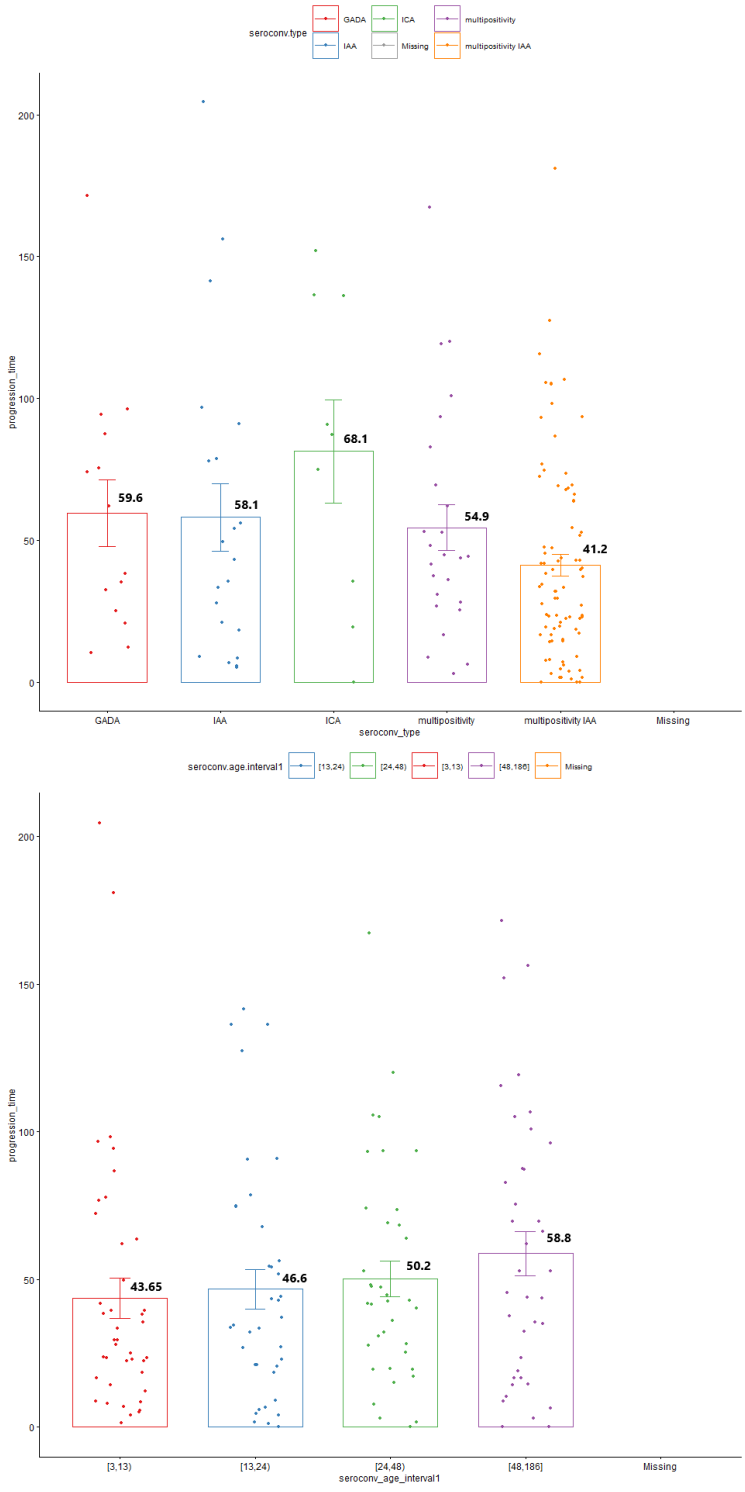


Figure 28. Disease progression time (months) by seroconversion type and seroconversion age intervals. The error bars represent the 95% confidence interval of the mean.

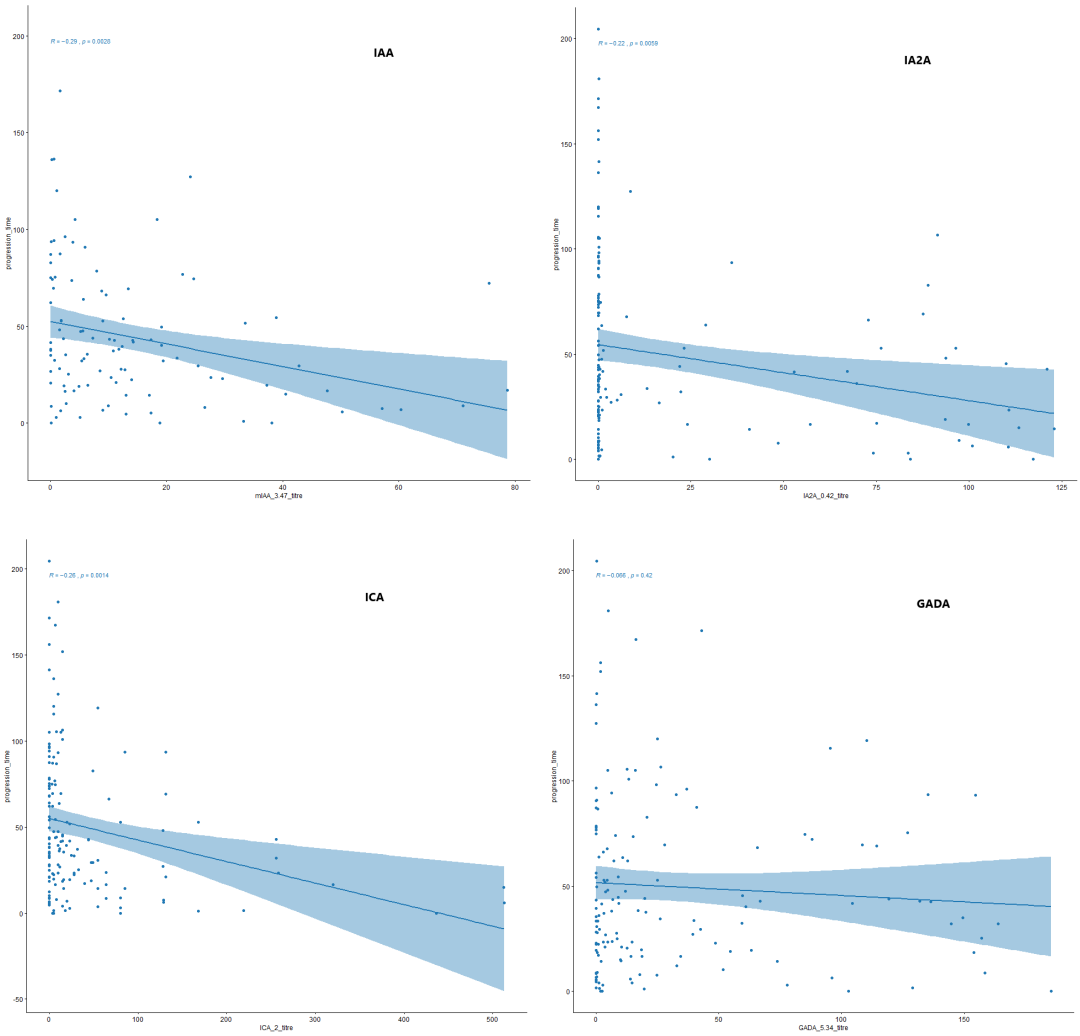


Figure 29. Disease progression time (months) negative correlated with all the autoantibody titres at seroconversion, except GADA.

6. DISCUSSION

The goal of this thesis was to build an interactive tool for processing and exploring clinical data which allows the user to select different subsets of the data based on the user's needs, navigate data, extract new information and perform analyses where statistical software knowledge is not necessary. In order to validate the applicability of this tool, a case study was conducted on the DIPP study data. A preprocessing pipeline was created in order to format the data into a meaningful form while preserving the existing information. While some of the preprocessing techniques used here can be extended to clinical data in general, the whole preprocessing algorithm is customized to the raw DIPP data. The preprocessing was integrated in the tool.

Major findings

The dataset used for this case study was rather limited compared to the dataset used by Knip et al. [79] in their article "Role of humoral beta-cell autoimmunity in type 1 diabetes" and by Pöllänen et al. [73] in their article "Characterisation of rapid progressors to type 1 diabetes among children with HLA-conferred disease susceptibility" which motivated this study. We had access to a smaller subset of the DIPP study data, from Oulu University Hospital, from which we analyzed 1,166 participants. However, the results from the case study conducted using ClinFlow on this subsample of the DIPP data support the findings of the two articles that used larger numbers of patients from the DIPP study, therefore, confirming the general patterns of T1D disease progression mechanism.

A weak positive correlation between GADA seroconversion titres and the age at seroconversion for patients who progressed to multipositivity, and a weak negative correlation between IAA titres at seroconversion and age at seroconversion were found in this case study. These two results were not reported in the reference articles. Further research will be required to validate these findings.

The application has proven useful for easily verifying, filtering and removing the incorrect entries in the data, therefore, removing some of the bias that these entries would have introduced in the analysis. The Filter Data functionality and the reactive properties of the filtered datatable have offered the possibility of easily checking group summary statistics in the data. This is helpful for the user to get a general idea about what kind of analysis results will be obtained even before conducting the actual analysis. It can also be applied for verifying the correctness of the results and checking whether there is any biased results introduced by the removal of missing entries found in the analyzed variables, therefore, having a valuable contribution on the user's analysis methodology choices.

Applicability and Requirements

A first requirement for this tool was to provide the user with a Patient dataset containing the constant variables and a Visit dataset with the time-varying variables. When uploading a dataset, the application detects the constant and time-varying variables and automatically splits the data into Patient and Visit

data. It also removes duplicated entries, provides the user with a preview of the tables and the option of visualizing summary statistics, therefore, meeting this requirement.

A second requirement was to provide the user with an interactive preprocessing interface. The application's Filter Data functionality has proven very useful in the DIPP case study, combined with the Outliers functionality for detecting and removing entries that could have introduced bias in the results. The Filter was also helpful in quickly subsetting groups in the data and verifying their validity. The Create a New Patient Feature functionality makes it easy to analyze periods from the patient's visit history, and The Create a New Categorical Feature functionality allows the user to define groups in the data based on the numeric value of a variable.

In the DIPP case study, these functionalities were used for delivering meaningful graphical representations of the relationships between variables and groups in the data. For example, we created categories in the data in order to color the clustering plots by categories and visualize the distances between these categories in the 2D and 3D cluster space. The new categorical variables also helped in generating mean-by-group bar plots. These plots can be easily downloaded from the application as ".png" or ".svg" images and inserted in reports and presentations. The new patient features created in the DIPP case study allowed us to visualize relationships between the autoantibody values and progression to multipositivity or diabetes. We observed that differences in autoantibody values recorded during the visits do not have a high impact on the outcome, as long as they are positive, a result that supports the prevalent methods of defining the antibody positive thresholds [72].

The user-defined preprocessing in the application has room for improvement, with more user filtering options that include queries involving several variables and possibilities of deriving new information based on a formula that includes multiple variables.

Another part of the user-defined preprocessing interface is the Panel Data creation tool, that allows the user to create a timestamp based on splitting the visit age into intervals and summarize the information for each time point or age interval. Then, the user can download the panel data and use it for different time-series analyses without the need of further preprocessing. This functionality was not used with the DIPP data case study, however, we introduced a preview of how the preprocessed panel data of the Iris dataset [38] in Chapter 4 of this thesis.

The final requirement was a platform for unsupervised learning and visualizations. The various visualization and clustering options in the application offer a dynamic platform for exploring the data. In the DIPP case study, the PCA clustering visualization has proven useful in identifying the numeric variables that influenced the multipositivity and T1D progression. By categorizing these variables and coloring the plot according to different intervals, allowed us to visualize the way in which the points in the clusters are similar. The various charts were used for exploring the relationships in the data in detail, and most of the results were similar to the ones obtained from research done on larger datasets of the same type. For this case study, we reported the results only from the PCA

clustering, but we also checked the other clustering methods on the same data which showed the same patterns.

Overall, the application meets the preprocessing and visualization requirements. The dynamic nature of the application allows the user to try different analysis approaches within a single session. This makes it fast to get a general idea about the dataset and what kind of results could be expected.

Various online resources such as online tutorials, R packages and open source applications from the Shiny gallery [84] had a valuable contribution in the building process. The open source HTPVis Shiny app by Dijun Chen [21] and the "FilterDF" Shiny module from the R package "esquisse" [43] contributed the most in building the visualizations and filter functions.

Limitations

The application's current version can be viewed as a proof-of-concept. ClinFlow requires extensive validation and usability testing before actual deployment. At the moment, the application is customized for the DIPP dataset, with limited generalizability to other clinical datasets. The case study was done on a subsample of the DIPP data, and although the results were compared with the domain articles, the general applicability of the findings would require validation with larger DIPP datasets. Like any clinical dataset, the DIPP data has very specific requirements that needed to be addressed by preprocessing. A big part of successful clinical data preprocessing is domain knowledge combined with knowledge about the data collection, and very few preprocessing operations can be generalized to fit more datasets. While this tool offers an interface for some user defined general preprocessing operations, it does not cover the complex challenges that are specific to the domain and data collection practices of each clinical dataset.

Performance requirements for this application have not been set at the moment. With the DIPP data available for this case study, the application did not display major loading times for executing different operations. However, the performance of the application has not been tested with larger datasets. This might raise the need for code optimizations in order to improve performance.

Future Work

Thorough validation, usability testing and deployment of the tool have been left for the future due to limited time. The application can be locally hosted, making it safe to use from the data privacy point of view. Future work also includes experiments with different datasets and increasing the generalizability to fit more types of clinical data. Some improvement ideas for the functionalities of the application that could be implemented in the future are listed below.

1. Including a general preprocessing pipeline along with user options for the more complex preprocessing operations, which can adapt to various datasets, not just the DIPP dataset.

2. Including filtering options in the Filter Data functionality for more complex filtering operations that use several variables of different types. This would allow for easier subsetting of entries that match more complex criteria.

3. Options to create new features in the data using complex formulas that include multiple variables, not just one variable at a time.

4. Implementing supervised learning methods such as regression or classification models along with the visualizations present in the app.

7. CONCLUSIONS

The main objective of this thesis was to design an interactive application for clinical data analysis with the purpose of helping the clinical researchers gain insight into the patterns found in the medical records of patients, as well as explore the data quality and availability. Common challenges for analysing clinical data were examined such as unclean data, irregularities, and the need for domain knowledge in preprocessing. Based on these challenges, a variety of existing visualization tools were studied and their advantages and limitations were acknowledged. A list of requirements for the new application was comprised, and the tool was built to meet these requirements by integrating dynamic preprocessing and visualization functionalities. The tool was evaluated with a case study on T1D data from the DIPP study. A separate preprocessing pipeline customized for the DIPP data was also integrated in the tool.

ClinFlow was proven to be a useful tool when analysing the clinical data in the case study. The tool provided results that support the existing domain knowledge and it shows interesting prospects for future development as well.

ClinFlow seeks to fill the gap between information technology and clinical research. It aims to provide an interactive interface that allows the clinical researcher to include domain expertise into the analysis process, without the need for statistical programming knowledge. ClinFlow has the potential of becoming a valuable tool in clinical research.

8. REFERENCES

[1] Brailer D. Institute of Medicine (US) Roundtable on Value Science-Driven Health Care Workshop (2010): Clinical Data as the Basic Staple of Health Learning.

[2] Idri A., Benhar H., Fernández-Alemán J. & Kadi I. A systematic map of medical data preprocessing in knowledge discovery. Computer Methods and Programs in Biomedicine 160: 69-85. DOI: https://doi.org/10.1016/j.cmpb.2018.05.007.

[3] Peterkova A. & Michaľčonok G. (2016) Preprocessing Raw Data in Clinical Medicine for a Data Mining Purpose. Research Papers Faculty of Materials Science and Technology Slovak University of Technology 24: 117-122.

[4] Clifton C. (2019), Data mining. `https://www.britannica.com/technology/data-mining/.` Accessed 27.01.2020.

[5] Nura E., Babavalian M.R., Moghadam A.M.E. & Tabar V.K. (2014) Knowledge discovery in medicine: Current issue and future trend. Expert Systems with Applications 41: 4434-4463. DOI: https://doi.org/10.1016/j.eswa.2014.01.011.

[6] Research report: Challenges and opportunities in clinical data management. Pharma Intelligence (2018). URL: `https://www.oracle.com/a/ocom/docs/dc/oracle-clinical-data-report-1809-final-26-sept.pdf?elqTrackId=a3c3795787d24ddb905a0872489fcbd8&elqaid=75274&elqat=2/.` Accessed 10.09.2020.

[7] R Core Team (2017) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL: `https://www.R-project.org/.`, Accessed 01.10.2018.

[8] RStudio, Inc (2013) Easy web applications in R. URL: `http://www.rstudio.com/shiny/.` Accessed 01.02.2020.

[9] Haller M.J. & Schatz D.A. (2016) The DIPP project: 20 years of discovery in type 1 diabetes. Pediatric Diabetes 17: 5-7. DOI: 10.1111/pedi.12398.

[10] MacEachren A.M. (1992) Visualizing Uncertain Information. Cartographic Perspectives 13: 10-19. DOI: 10.14714/CP13.1000. pp. 10–19.

[11] Jensen P., Jensen L. & Brunak S. (2012) Mining electronic health records: towards better research applications and clinical care. Nature reviews. Genetics 13: 395-405. DOI: 10.1038/nrg3208.

[12] Binder H. & Blettner M. (2015) Big Data in Medical Science—a Biostatistical View. Deutsches Ärzteblatt international 112: 137–142. DOI: 10.3238/arztebl.2015.0137.

[13] Monroe M., Lan R., Lee H., Plaisant C. & Shneiderman B. (2013) Temporal Event Sequence Simplification. IEEE Transactions on Visualization and Computer Graphics 19: 2227–36. DOI: 10.1109/TVCG.2013.200.

[14] Plaisant C., Mushlin R., Snyder A., Li J., Heller D. & Shneiderman B. (2003) LifeLines: using visualization to enhance navigation and analysis of patient records. The craft of information visualization 308-312.

[15] Wang T., Plaisant C., Shneiderman B., Spring N., Roseman D., Marchand G., Mukherjee V. & Smith M. (2003) Temporal summaries: Supporting temporal categorical searching, aggregation and comparison. IEEE Transactions on Visualization and Computer Graphics 15: 308-312. DOI: 10.1109/TVCG.2009.187.

[16] Deng Y. & Denecke K. (2014) Visualizing unstructured patient data for assessing diagnostic and therapeutic history. Studies in health technology and informatics 205: 1158–62.

[17] Klimov D. & Shahar Y. (2005) A Framework for Intelligent Visualization of Multiple Time-Oriented Medical Records. AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium 2005: 405-9.

[18] Gotz D., Sun J., Cao N. & Ebadollahi S. (2011) Visual Cluster Analysis in Support of Clinical Decision Intelligence. AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium 2011: 481–90.

[19] Pennington J., Ruth B., Italia M., Miller J., Wrazien S., Loutrel J., Crenshaw E.B. & White P. (2014) Harvest: An open platform for developing web-based biomedical data discovery and reporting applications. Journal of the American Medical Informatics Association : JAMIA 21: 379–83. DOI: 10.1136/amiajnl-2013-001825.

[20] Badgeley M., Khader S., Glicksberg B., Tomlinson M., Levin M., McCormick P., Kasarskis A., Reich D. & Dudley J. (2016) EHDViz: Clinical dashboard development using open-source technologies.. BMJ Open 6: e010579. DOI: 10.1136/bmjopen-2015-010579.

[21] Chen D., Fu L.Y., Hu D., Klukas C., Chen M. & Kaufmann K. (2018) The HTPmod Shiny application enables modeling and visualization of large-scale biological data. Communications Biology 1. DOI:10.1038/s42003-018-0091-x.9.

[22] Gassen (January 2020), Using expand for panel data exploration. URL: `https://joachim-gassen.github.io/ExPanDaR/articles/use_ExPanD.html/`. Accessed 15.02.2020.

[23] Cios K.J. & Moore G. (2002) Uniqueness of medical data mining. Artificial Intelligence in Medicine 26(1-2): 1-24. DOI: 10.1016/S0933-3657(02)00049-0.

[24] Keim D.A. (2002) Information Visualization and Visual Data Mining. IEEE Transactions on Visualization and Computer Graphics 8(1): 1-8.

[25] Harp C. (2013), What is data normalization? URL: https://blog.clinicalarchitecture.com/what-is-data-normalization/. Accessed 1.04.2020.

[26] Motoda H. & Liu H. (2002) Feature selection, extraction and construction. Communication of IICM (Institute of Information and Computing Machinery, Taiwan) 5: 67-72.

[27] Hotelling H. (1933) Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24(6): 417–441. DOI: 10.1037/h0071325.

[28] Hastie T., Tibshirani R. & Friedman J. (2009) The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Springer, second ed.

[29] Yobero C. (2013), Using principal component analysis for clustering. URL: https://rpubs.com/cyobero/pca-clustering/. Accessed 25.02.2020.

[30] van der Maaten L. & Hinton G. (2008) Visualizing Data using t-SNE. Journal of Machine Learning Research 9(2605): 2579-2605.

[31] NCSS Statistical Software 435: 1-17. Multidimensional Scaling. URL: https://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Multidimensional_Scaling.pdf/. Accessed 25.02.2020.

[32] Wehrens R. & Buydens L.M.C. (2007) Self- and super-organizing maps in R: The kohonen package. Journal of Statistical Software 21(5): 1-19. DOI: 10.18637/jss.v021.i05.

[33] Dangeti P. (2017) Statistics for Machine Learning: Techniques for exploring supervised, unsupervised, and reinforcement learning models with Python and R- Chapter 8: Unsupervised Learning. Packt Publishing. https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788295758/. Accessed 25.02.2020.

[34] Peter R.J. (1987) Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics 20: 53–65.

[35] Tibshirani R., Walter G. & Hastie T. (2001) Estimating the number of clusters in a dataset via the gap statistic. Journal of the Royal Statistical Society 63(2): 411-423. DOI: 10.1111/1467-9868.00293.

[36] RStudio Team (2015) RStudio: Integrated Development Environment for R. RStudio, Inc., Boston, MA. URL: http://www.rstudio.com/. Accessed 5.01.2020.

[37] Wickham H. (2016) ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. URL: https://ggplot2.tidyverse.org/. Accessed 01.03.2020.

[38] Fisher A. (1936) The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 179-188.

[39] Kassambara A. (2019) ggpubr: 'ggplot2' Based Publication Ready Plots. URL: https://CRAN.R-project.org/package=ggpubr/. R package version 0.2.4. Accessed 23.03.2020.

[40] Ligges U. & Mächler M. (2003) Scatterplot3d - an r package for visualizing multivariate data. Journal of Statistical Software 8, pp. 1–20. URL: http://www.jstatsoft.org/. Accessed 21.03.2020.

[41] Wei T. & Simko V. (2017) R package "corrplot": Visualization of a Correlation Matrix. URL: https://github.com/taiyun/corrplot/. (Version 0.84). Accessed 23.03.2020.

[42] Tierney N., Cook D., McBain M. & Fay C. (2019) naniar: Data Structures, Summaries, and Visualisations for Missing Data. URL: https://CRAN.R-project.org/package=naniar/. R package version 0.4.2. Accessed 23.03.2020.

[43] Meyer F. & Perrier V. (2020) esquisse: Explore and Visualize Your Data Interactively. URL: https://github.com/dreamRs/esquisse/. R package version 0.3.0.900. Accessed 21.03.2020.

[44] Perrier V., Meyer F. & Granjon D. (2019) shinyWidgets: Custom Inputs Widgets for Shiny. URL: https://CRAN.R-project.org/package=shinyWidgets/. R package version 0.5.0. Accessed 16.02.2020.

[45] Bailey E. (2015) shinyBS: Twitter Bootstrap Components for Shiny. URL: https://CRAN.R-project.org/package=shinyBS/. R package version 0.61. Accessed 16.02.2020.

[46] Attali D. (2020) shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds. URL: https://CRAN.R-project.org/package=shinyjs/. R package version 1.1. Accessed 16.02.2020.

[47] Vaidyanathan R., Xie Y., Allaire J., Cheng J. & Russell K. (2019) htmlwidgets: HTML Widgets for R. URL: https://CRAN.R-project.org/package=htmlwidgets/. R package version 1.5.1. Accessed 16.02.2020.

[48] Chang W. & Borges Ribeiro B. (2018) shinydashboard: Create Dashboards with 'Shiny'. URL: https://CRAN.R-project.org/package=shinydashboard/. R package version 0.7.1. Accessed 16.02.2020.

[49] Stacklies W., Redestig H., Scholz M., Walther D. & Selbig J. (2007) pcamethods – a bioconductor package providing pca methods for incomplete data. Bioinformatics 23: 1164-1167.

[50] Team R., shinyapps.io. https://docs.rstudio.com/shinyapps.io/. Accessed 2.03.2020.

[51] Grace-Martin K., How to diagnose the missing data mechanism. The Analysis Factor, LLC. https://www.theanalysisfactor.com/missing-data-mechanism/. Accessed 3.03.2020.

[52] U.S. FDA Guidance for Industry. E9 Statistical Principles for Clinical Trials (1999) https://www.fda.gov/media/71336/download/. Accessed 3.03.2020.

[53] Oba S., Sato M.A., Takemasa I., Monden M., Matsubara K.I. & Ishii S. (2003) A Bayesian missing value estimation method for gene expression profile data. Bioinformatics 19(16): 2088-2096. DOI: 10.1093/bioinformatics/btg287.

[54] Finnish diabetes association. URL:https://www.diabetes.fi/en/finnish_diabetes_association/diabetes_in_finland/. Accessed 10.09.2020.

[55] Knip M., Korhonen S., Kulmala P., Veijola R., Reunanen A., Raitakari O.T., Viikari J. & Akerblom H.K. (2010) Prediction of type 1 diabetes in the general population. Diabetes Care 33(6): 1206-12. DOI: 10.2337/dc09-1040.

[56] Paschou S.A., Papadopoulou-Marketou N., Chrousos G.P. & Kanaka-Gantenbein C. (2017) On type 1 diabetes mellitus pathogenesis. Endocrine connections 7(1):38-46. DOI: 10.1530/EC-17-0347.

[57] Menser M.A., Forrest J.M. & Bransby R.D. (1978) Rubella infection and diabetes mellitus. The Lancet 1(8055): 57-60. DOI: 10.1016/s0140-6736(78)90001-6.

[58] Hober D. & Sauter P. (2010) Pathogenesis of type 1 diabetes mellitus: interplay between enterovirus and host. Nature Reviews Endocrinology 6(5): 279-89. DOI: 10.1038/nrendo.2010.27.

[59] Yoon J. & Jun H. Viruses cause type 1 diabetes in animals. Annals of the New York Academy of Sciences 1(1079): 138-146. DOI: 10.1196/annals.1375.021.

[60] Hyöty H. (2016) Viruses in type 1 diabetes. Pediatric Diabetes 17(22): 56-64. DOI: 10.1111/pedi.12370.

[61] Virtanen S.M. (2016) Dietary factors in the development of type 1 diabetes. Pediatric Diabetes 17(22): 49-55. DOI: 10.1111/pedi.12341.

[62] Norris J., Barriga K., Klingensmith G., Hoffman M., Eisenbarth G., Erlich H. & Rewers M. (2003) Timing of Initial Cereal Exposure in Infancy and Risk of Islet Autoimmunity . JAMA : the journal of the American Medical Association 290(13):1713-20 DOI: 10.1001/jama.290.13.1713.

[63] Knip M., Åkerblom H., Taji E., Becker D., Bruining J., Castaño L., Danne T., de Beaufort C., Dosch H.M., Dupre J., Fraser W., Howard N., Ilonen J., Konrad D., Kordonouri O., Krischer J., Lawson M., Ludvigsson J.,

Madacsy L. & Wąsikowa R. (2018) Effect of Hydrolyzed Infant Formula vs Conventional Formula on Risk of Type 1 Diabetes: The TRIGR Randomized Clinical Trial. JAMA : the journal of the American Medical Association 319(1): 38-48. DOI: 10.1001/jama.2017.19826.

[64] Virtanen S.M., Uusitalo L., Kenward M.G., Nevalainen J., Uusitalo U., Kronberg-Kippilä C., Ovaskainen M.L., Arkkola T., Niinistö S., Hakulinen T., Ahonen S., Simell O., Ilonen J., Veijola R. & Knip M. (2011) Maternal food consumption during pregnancy and risk of advanced beta-cell autoimmunity in the offspring. Pediatric Diabetes 12(2): 95-9. DOI: 10.1111/j.1399-5448.2010.00668.x.

[65] Lamb M., Myers M., Barriga K., Zimmet P., Rewers M. & Norris J. (2008) Maternal diet during pregnancy and islet autoimmunity in offspring. Pediatric Diabetes 9(2): 135-41. DOI: 10.1111/j.1399-5448.2007.00311.x.

[66] Niinistö S., Takkinen H.M., Uusitalo L., Rautanen J., Nevalainen J., Kenward M., Lumia M., Simell O., Veijola R., Ilonen J., Knip M. & Virtanen S. (2014) Maternal dietary fatty acid intake during pregnancy and the risk of preclinical and clinical type 1 diabetes in the offspring. British Journal of Nutrition 111: 895-903. DOI: 10.1017/S0007114513003073.

[67] Virtanen S., Takkinen H.M., Nwaru B., Kaila M., Ahonen S., Nevalainen J., Niinistö S., Siljander H., Simell O., Ilonen J., Hyöty H., Veijola R. & Knip M. (2014) Microbial Exposure in Infancy and Subsequent Appearance of Type 1 Diabetes Mellitus–Associated Autoantibodies. JAMA Pediatrics 168(8): 755-763. DOI: 10.1001/jamapediatrics.2014.296.

[68] Cardwell C., Stene L., Joner G., Bulsara M., Cinek O., Rosenbauer J., Ludvigsson J., Jané M., Svensson J., Goldacre M., Waldhör T., Jarosz-Chobot P., Gimeno S., Chuang L.M., Parslow R., Wadsworth E., Chetwynd A., Pozzilli P., Brigis G. & Patterson C. (2010) Maternal Age at Birth and Childhood Type 1 Diabetes: A Pooled Analysis of 30 Observational Studies. British Journal of Nutrition 59: 486-494. DOI: 10.2337/db09-1166.

[69] Stanley H., Norris J., Barriga K., Hoffman M., Yu L., Miao D., Erlich H., Eisenbarth G. & Rewers M. (2004) Is Presence of Islet Autoantibodies at Birth Associated With Development of Persistent Islet Autoimmunity? Diabetes Care 27(2): 497-502. DOI: 10.2337/diacare.27.2.497.

[70] Knip M., Luopajärvi K. & Härkönen T. (2017) Early life origin of type 1 diabetes. Seminars in Immunopathology 39(6): 653-667. DOI: 10.1007/s00281-017-0665-6.

[71] Moulder R. & Lahesmaa R. (2016) Early signs of disease in type 1 diabetes. Pediatric Diabetes 17(22): 43-48. DOI: 10.1111/pedi.12329.

[72] Bonifacio E. (2015) Predicting Type 1 Diabetes Using Biomarkers. Diabetes Care 38(6): 989-96. DOI: 10.2337/dc15-0101.

[73] Pöllänen P., Lempainen J., Laine A.P., Toppari J., Veijola R., Paula V., Ilonen J., Siljander H. & Knip M. (2017) Characterisation of rapid progressors to type 1 diabetes among children with HLA-conferred disease susceptibility. Diabetologia 60: 1284-1293. DOI: 10.1007/s00125-017-4258-7.

[74] Pöllanen P., Lempainen J., Laine A.P., Toppari J., Veijola R., Ilonen J., Siljander H. & Knip M. (2019) Characteristics of Slow Progression to Type 1 Diabetes in Children With Increased HLA-Conferred Disease Risk. Journal of Clinical Endocrinology Metabolism 104(11): 5585-5594. DOI: 10.1210/jc.2019-01069.

[75] Gale E., Bingley P., Knip M., Emmett C., Swankie H., Fewell S., Kearsey P., Schober E., Gorus F., Dupre J., Mahon J., Profozic V., Reimers J., Mandrup-Poulsen T., Levy-Marchal C., Jaeger C., Bartsocas C., Vazeou A., Gyorko M. & Weber B. (2004) European Nicotinamide Diabetes Intervention Trial (ENDIT): a randomised controlled trial of intervention before the onset of type 1 diabetes. Lancet 363: 925-931.

[76] Skyler J., Brown D., Chase H., Collier E., Cowie C., Eisenbarth G., Fradkin J., Grave G., Greenbaum C., Jackson R., Kaufman F., Krischer J., Marks J., Palmer J., Ricker A., Schatz D., Winter W., Wolfsdorf J. & Zinman B. (2002) Effects of Insulin in Relatives of Patients with Type 1 Diabetes Mellitus. New England Journal of Medicine 346(22): 1685-1691. DOI: 10.1056/NEJMoa012350.

[77] Ryhänen S., Härkönen T., Siljander H., Näntö-Salonen K., Simell T., Hyöty H., Ilonen J., Veijola R., Simell O. & Knip M. (2011) Impact of Intranasal Insulin on Insulin Antibody Affinity and Isotypes in Young Children With HLA-Conferred Susceptibility to Type 1 Diabetes. Diabetes Care 34(6): 1383-8. DOI: 10.2337/dc10-1449.

[78] Skyler J., Krischer J., Wolfsdorf J., Cowie C., Palmer J., Greenbaum C., Cuthbertson D., Rafkin L., Chase H. & Leschek E. (2005) Effects of oral insulin in relatives of patients with type 1 diabetes: The Diabetes Prevention Trial–Type 1. Diabetes Care 28(5): 1068-76. DOI: 10.2337/diacare.28.5.1068.

[79] Knip M., Siljander H., Ilonen J., Simell O. & Veijola R. (2016) Role of humoral beta-cell autoimmunity in type 1 diabetes. Pediatric Diabetes 17(22): 17-24. DOI: 10.1111/pedi.12386.

[80] Algert C., McElduff A., Morris J. & Roberts C. (2009) Perinatal risk factors for early onset of Type 1 diabetes in a 2000–2005 birth cohort. Diabetic Medicine 26(12): 1193-7. DOI: 10.1111/j.1464-5491.2009.02878.x.

[81] Koczwara K., Bonifacio E. & Ziegler A.G. (2004) Transmission of Maternal Islet Antibodies and Risk of Autoimmune Diabetes in Offspring of Mothers With Type 1 Diabetes. Diabetes 53(1): 1-4. DOI: 10.2337/diabetes.53.1.1.

[82] Hämäläinen A., Ilonen J., Simell O. & Savola K. (2002) Prevalence and fate of type 1 diabetes-associated autoantibodies in cord blood samples from

newborn infants of non-diabetic mothers. Diabetes Metabolism 18(1): 57-63. DOI: 10.1002/dmrr.232.

[83] Kahn H.S. & Morgan T. (2009) Association of Type 1 Diabetes With Month of Birth Among U.S. Youth. Diabetes care 32(11): 2010-5. DOI: 10.2337/dc09-0891.

[84] Shiny apps gallery. RStudio, Inc. `https://shiny.rstudio.com/gallery/`. Accessed 2.03.2020.