Visualization of medication trajectories among patients with type 2 diabetes to optimize treatment selection

Understanding the common trajectories of current treatment patterns from a multimorbidity population

Author: Shadee Albronda
Solis-ID: 0959464
Mail: s.m.albronda@students.uu.nl

Academic year: 2021 – 2022
Study program: MSc Applied data science
Educational institution: Utrecht University
Heidelberglaan 8
3584 CS Utrecht
The Netherlands

Thesis project 14 ECTS
Title project: Medication trajectories visualization
Start/End date: April 25, 2022 - July 1, 2022
Supervisor: Daniala Weir, PhD
Second examiner: Helga Gardarsdottir, PhD

Host institute
Division of Pharmacoepidemiology and Clinical Pharmacology
Department of Pharmaceutical Sciences
Utrecht University
Universiteitsweg 99, 3584 CG Utrecht

Thursday, 30 June 2022

Contents

Abstract

Multimorbidity and polypharmacy are strongly linked to diabetes. Multimorbidity complicates prescription choices because of the various drug combinations. Aiming to optimize drug prescription choices in multimorbidity, a first step is to characterize, visualize and understand the current trajectories of treatment patterns among these patients.

The overall objective is to visualize longitudinal medication trajectories among patients with type 2 diabetes. The aim is to generate a visualisation that describes the complexity of these trajectories, while limited in size. The research aims to inform the optimization of drug prescription choices by providing information on common prescriptions, their sequence and time intervals.

Medication history vectors, per patient, were designed as a sequence containing all BNF chapters of chronic prescriptions. Clustering these vectors identified 11 groups of common trajectory sequences. All clusters contain similar medications but show differences over time. This implies that there're 11 common trajectories which contain the same BNF chapters but their sequences over time differ.

Overall a visualisation was reached that proved useful to visualize medication trajectories over time, while capturing as much complexity possible.

1 Introduction

1. 1 Motivation and context

The co-occurrence of multiple chronic diseases within one person is known as multimorbidity.[1,2] These patients are often prescribed multiple prolonged medications simultaneously, so called polypharmacy.[2,3] Multimorbidity and polypharmacy are strongly linked to diabetes. A quarter of the diabetes patients even experience more than 3 chronic conditions. For type 2 diabetes the prevalence of polypharmacy is estimated ranging between 57%-99%. [2,3] Comorbidities of diabetes often requires treatment using multiple medications, leading to various possible drug combinations. These patients with diabetes using multiple medications endure a poorer health compared to patients with only diabetes and are at high risk to encounter adverse drug events.[4] Multimorbidity complicates prescription because recommended medications may lead to adverse drug effects, resulting in uncertainty of the optimal prescription choice.[5] Most current guidelines apply to the management of a single disease. The optimal care for patients with multiple conditions is often unknown and single disease guidelines could even be harmful for these patients.

Aiming to optimize drug prescription choices in multimorbidity, a first step is to characterize, visualize and understand the current trajectories of treatment patterns among these patients.[6] Polypharmacy and multimorbidity results in a complex patient population. Analysis, visualisation and understanding medication patterns over time for these complex patients, is no easy task.[6] Some previous studies analysed diabetes specific medication trajectories and others focused on whole drug regimens, nevertheless it remained challenging to characterize these regimens meaningfully, reflecting its complexity while keeping methods clear and compact.[6,7]

A previous study, by *Giannoula et al.*, presented a time-analysis for large-scale comorbidity studies.[6] Their aim was on identification of a method to reveal complex time-dependant disease patterns. Time sequences of ordered disease diagnoses (disease-history vectors) were grouped according to the temporal patterns they share using unsupervised clustering. This showed that the temporal assessment of such trajectories could be exploited in order to discover disease patterns. These disease patterns could facilitate the prediction of the course of a disease given previous diagnosis.[6] This system retrieves trajectories that the patients often follow (common trajectories) by clustering them into groups. These clusters describe the common trajectories itself and the variability within the commonly followed patterns. In the research from *Giannoula et al.* dynamic time warping distance was used to create these clusters of common trajectories, based on the similarity of their sequences. The dynamic time warping (DTW) algorithm aims to cluster trajectories based on temporal characteristics they share. DTW is a technique for measuring similarities between two sequences that may vary in time or speed.[6] It's applied with success to multiple pattern recognition applications and recently on patient disease trajectories.[6] Applying this approach to chronic medication trajectories could help to cluster into groups which can represent the common medication trajectories of its population. Analysing medication trajectories within a time-dependent context is promising to provide better understanding of the progression of specific medication trajectories in complex patients with type 2 diabetes. This method could be used to investigate the most common medication trajectories and their time-dependent characteristics. The approach can be summarised as the extraction of medication-history vectors from each patient and temporal analysation over the population. *Giannoula et al.* achieved a visualisation of the trajectories, of a comorbidities population, showing six clusters which represent the common disease trajectories.[6] Each cluster was visualised separate using a network plot containing nodes which represent classes of disease diagnoses. The nodes are drawn at a relative size to the frequency of appearance of the group of diseases. Time is visualised by arrows between nodes representing disease diagnosis over time. Additional diagnoses in the same node are shown as cyclic arrows.[6] These arrows provide some information of the diagnoses of disease groups over a few timepoints but don't show the amount of time between parts of the trajectory.

Some visualisations of complex medication trajectories contain much information, resulting in large figures which are hard to read. Other visualizations, like the one described above, are more abstract but fail to describe detailed change over time. These figures are clear and uncomplicated but only contain a few cross-sectional points in time combined in one visualisation. These previous visualizations aren't sufficient to provide detailed and clear insight into the common prescriptions over time and so lack clinical applicability. The clinical understanding of these models are an important component of learning how to better optimize drug regimens for complex patients in the future. Insight in the most common medication trajectories could provide knowledge which medicines are often prescribed, their sequence and intervals. When identified for a certain cohort this information could help describe the progress of chronic disease. Some medication can worsen other conditions it isn't intended for, leading to adverse drug effects. The common trajectories and their variability could be taken into account, when prescription choices are made, to optimize medication selection and reduce adverse drug effects. In the future these models could be capable to describe guidelines in multimorbidity disease management which are currently unknown. These guidelines can then form a custom treatment plan targeting a specific population.

1.2 Objectives

The overall objective is to visualize longitudinal medication trajectories among patients with type 2 diabetes. The aim is to generate a complete visualisation that's capable of describing the complexity of these trajectories, while also limited in size. The research aims to inform the optimization of drug prescription choices by providing information on common prescriptions, their sequence and time intervals.

To build toward this, the first sub-objective is to reshape the dataset to a representation of prescriptions over time for each patient. Next is identification and visualization of the common sequences of medications over time. The second sub-objective is to include timing and duration of medication use, and the third is to generate a cross sectional visualization. These visualizations are describing the overall common trajectories over time and additionally at one point in time.

2 Data

The Clinical Practice Research Datalink (CPRD) controls general practice data and provides one of the largest primary care datasets of longitudinal records. Since 1987 a small database grew to become the CPRD in 1993. Anonymised electronic health record data is routinely collected from general practices who agreed to providing data. All patients registered with participating practices are included in the dataset, unless they requested exclusion.[8] The CPRD GOLD database is widely used and the high data quality is stated in various studies.[8,9] The dataset finds strength in its large size, in 2014 it included 79 million person-years of follow-up of data from 674 practices.[8] Data quality is enhanced by the Quality and Outcomes Framework, a system for the performance management for GPs, which encourages completeness in recording certain variables. In primary care data the quality is often variable because data is collected during routine consultations, not intended for research. Data quality checks are therefore advised.[8]

A specific cohort was cut from the CPRD GOLD dataset version 2.5 during a previous study on use of oral antidiabetics and risk of sudden cardiac arrest. Selection was based on having at least one oral antidiabetic after 2013. This pre-existing dataset is re-used for this study on the visualization of its medication trajectories. From this dataset three tables are used: therapy, product, and patients. The therapy table contains details of all drug prescriptions and prescribed products. The latter is not in the focus of this study and can be excluded. Patients may have more than one row in this table, one for each therapeutic event. The product table contains information on prescribed medications *(Appendix A)*. Products in the table contain a BNF code which describes the product in detail. The British

National Formulary (BNF) lists over 70,000 standard medicines prescribed in the UK. The BNF code is used as a unique identifier for prescribed medicines. These BNF codes give information about the drugs indications, dosages and size effects. The codes are in a hierarchy, the first two characters represent the BNF chapter and tell the class of the drug, for example BNF chapter 04 (Central Nervous System). The BNF is then further subdivided, for example Antidepressant Drugs, within chapter 04 section 03 of the BNF. The last few characters provide details like form and strength.[10]

2.1 Data exploration results

Two target tables are described in *Appendix A.* The therapy table contains a total of 277,530,649 rows but not all are unique. These duplicate rows should be removed prior analysis. The column describing the number of treatment days often is 0, providing no treatment duration. Prescription dates are available to potentially calculate treatment durations. *Table 1* contrasts some data characteristics pre- and post-filtering. Corresponding filtering steps are described subsequently. The total number of prescriptions was summed for each year and found to peak amid the year 2000 and 2020 *(Fig. 1)*.

Table 1. *Properties of the raw data and filtered data*
An SQL script was used in data exploration, see Appendix B.1. Pre-filtering describes the raw data and Post-filtering the remaining data after the first filtering steps. Post-filtering only includes prescriptions after the year 200, on repeat schedule, having at least 90 treatment days.

Total number of:	**Pre-filtering**	**Post-filtering**
Patients	356,590	345,463
Percentage female	45.83%	45.40%
Prescriptions	277,530,649	4,660,519
Unique prescriptions	277,036,024	4,660,519

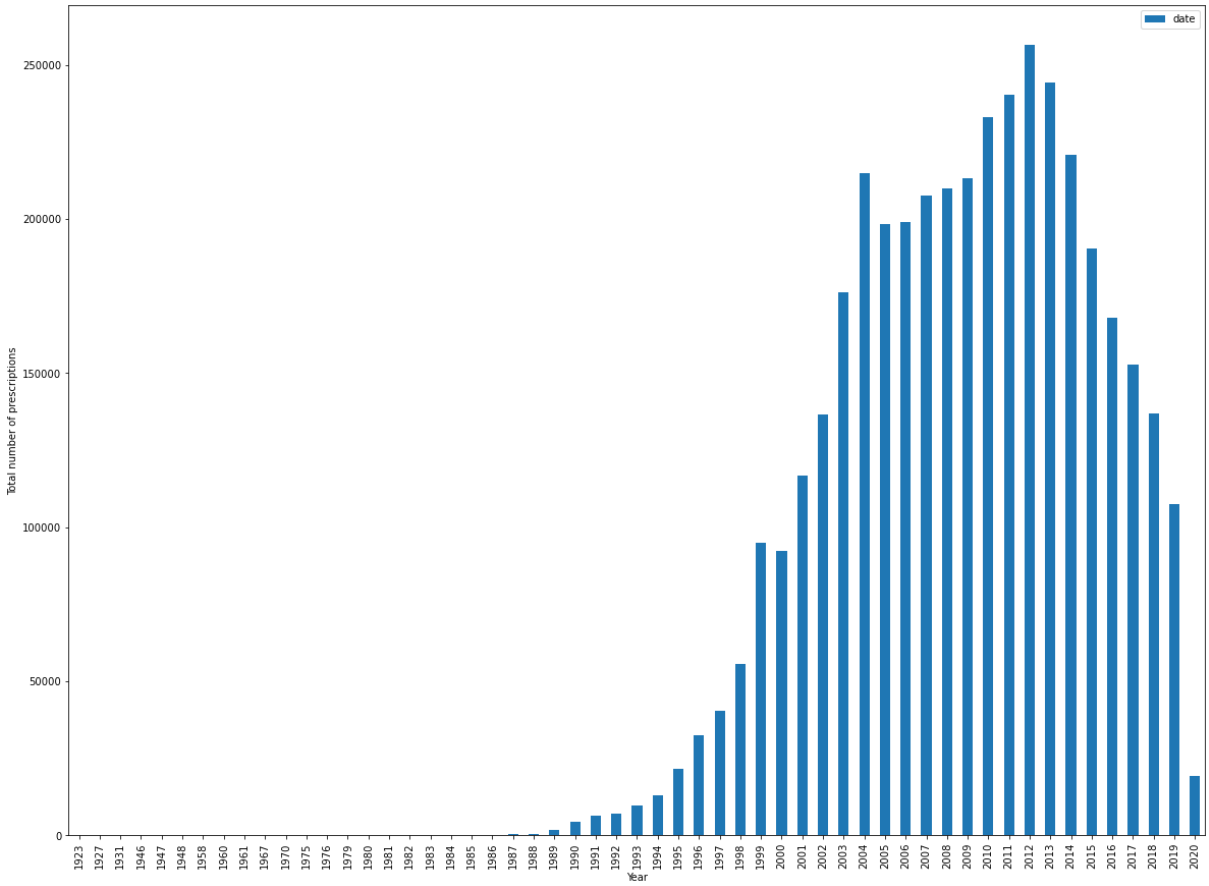

Figure 1. *Total number of medication prescriptions each year*
Only prescriptions on a repeat schedule with a treatment duration of at least 90 days are included. For methods see Appendix B2.

2.2 Data preparation

A challenge regarding this analysis is the size of the dataset. To create a vector of medication prescriptions of each patient over time, the data needs multiple restructuring and extraction steps. To achieve this two tables were merged to obtain both the prescription information from the therapy table, and the full BNF code of the medicine from the product table. The therapy table exceeds a size of 33 GB. Relational database management systems like MySQL are not designed to run complicated queries against big datasets that exceed 2 GB.[11] The analysis are performed on a single computer and no server containing multiple cores is presently available. This means that queries and scripts can't be executed in parallel making the first stages a time-consuming process. The Python library Pandas was found capable handling large datasets but available RAM wasn't sufficient to merge the tables at once. A SQLite database is limited to 281 terabytes.[12] This generous capacity motivated the choice to employ SQLite for data preparation. A database containing the tables required for analysis was created *(Fig 3: 1.Preparation)*. This allowed to obtain and restructure the target data, to wished format, using SQL queries. The therapy and product tables were joined on the product code while only preserving prescriptions on a repeat schedule. The number of treatment days was calculated from the last and first date of prescription using SQL. The assumption that chronic drug treatment is at least 90 days, and never ends, was used. Only these chronic medications were then extracted and saved for further analysis. To speed-up the first steps, the dataset size was minimized keeping only the essential columns and tables *(Appendix B)*. Based of *Figure 1*, it was agreed to only save prescriptions upward from the year 2000. The BNF codes and event dates are needed for subsequent analysis so only prescriptions having these values available were included.

2.3 Ethical and legal considerations

Individual patient specific data provides the details to create the desired medication trajectories sequences over time. The use of individual patient data is crucial to reach the goal of this study and can't be avoided. To conserve patient privacy the data was provided in a completely anonymised format.[13,14] Patient data is stored at the department and can only be analysed from this location to avoid the spread of data over multiple systems.

3 Methods

3.1 Study population

The study population was selected on the prescription of at least one oral antidiabetic after 2013. This resulted in a prevalent cohort defined as a group of individuals who have diabetes type 2 at some point during their follow-up period. Medications could already be prescribed before a patient entered the study, meaning there's no information on the first date of prescription for these prevalent users. Given the selection criterion there could be antidiabetic prescriptions at baseline. Patients having pre-existing antidiabetic prescriptions don't provide information on the first instance of the treatment course of diabetes. Another consequence of pre-existing prescriptions is the uncertainty in treatment duration. This led the decision to create both a prevalent and incident cohort. The prevalent cohort was created containing all recorded prescriptions. The incident cohort only describes new use of medications. New use is defined by no prescription for a given medication in the first year recorded for each patient.

3.2 Translation of the research question to a data science question

Data storage and extraction choices are the starting foundation of this analysis and affect the speed of the entire process. The capability to handle big data was therefore selected a key property in method

selection. This translates to answering the question: which methods qualify to process a dataset this size with reasonable running times. This question was answered at the data preparation step and a SQLite database was selected.

Generating visualisations over time requires significant restructuring of the data. Patients are recorded for varying follow-up time in years having a median (25th and 75th percentile) of 12.0 (7.0-16.0) years. This variability complicates the generation of medication vectors. This gives rise to a data science question regarding the restructuring of data. Creation of medication history vectors allows various possibilities based on assumptions made. The vector design choices directly affect the visualization options and limitations. Dividing vectors into separate chunks reflecting some range in time allows for visualization in temporal parts which enables to zoom-in on different time regions and its prescriptions. Visualizations should reflect the gradual course over time that characterize chronic diseases. The visualization of prescriptions over only a few points in time wouldn't reflect this chronic progress. A large compression of prescriptions over time in few time-points prevents detailed interpretation of change over time. In contrast a too small compression complicates visualisation resulting in large figures. Compression into ranges of one and three years were tested and the one year range visualisations were found most suitable. Aiming on clinical applicability, a user-friendly representation of the trajectories of treatment patterns is valued. This limits visualisation options and only well-known and uncomplicated visualization types were considered. To allow detailed interpretation, change over time should be instantly visible. A figure having time on the X-axis divided in parts of 1 year would fit this condition and provides an intuitive design that's inline with the research objectives.

3.3 Motivated selection of method(s) for analysis

Working towards generating a medication history vector for each patient, the data was rebuilt to a format of one line per patient *(Appendix B.3)*. From this a medication history vector was made for each patient *(Fig 3: 2.Restructure)*. This vector is represented as a sequence of all recorded prescriptions BNF codes in chronological order *(Appendix B.4)*. Only the first occurrence of each BNF code was included in the vectors. These whole BNF code vectors resulted in a high number of 296,826 unique trajectories. These data measurements led to a problematic and slow subsequent analysis. Therefore, a second collection of vectors was created containing only the first occurrence of the BNF chapters 1 to 10 *(Fig 3: 3.Filter)*. Only the first level of the BNF code was used in generating these vectors aiming to gain a compact dataset, speeding up subsequent analysis.

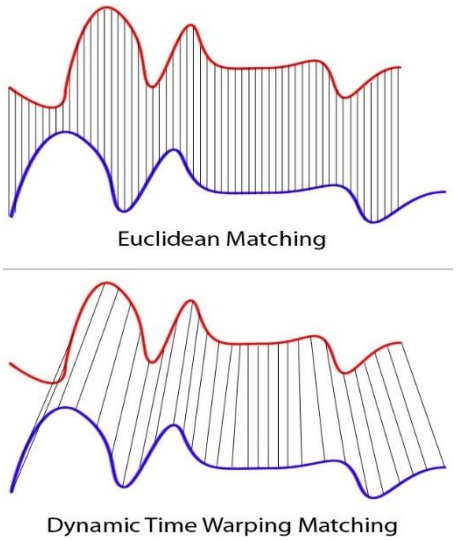

Figure 2. *Dynamic time warping distance vs Euclidean distance. DTW calculates an optimal match between two sequences with certain restrictions and rules. Sequences having different follow-up durations can be compared because the whole length of both sequences is used when comparisons are made.*
Adapted from Portilla, Heintz and Lee, 2022 [19]

Clustering

To identify the common medication trajectories of the population, clustering can be used to group the trajectories based on temporal patterns they share. The resulting clusters could then represent the most common trajectories. The trajectories vary in total follow-up time, number of prescriptions, and sequence. Dynamic time warping can compare temporal sequences that don't sync up perfectly. Meaning, it's capable to find groups of similar sequences even when they differ in number of prescriptions and follow-up time *(Fig. 2)*. Because of this desirable property, DTW was used to generate a pairwise distance matrix, containing the DTW distance, of all pairs of unique trajectories. The DTW distance refers to the length of the optimal alignment between two given trajectories.

K-medoids combined with DTW distance is a commonly used method for time-series clustering. Unlike the k-means method, k-medoids updates the cluster centre using the median cluster member itself instead of the overall mean position. This results in less sensitivity to outliers when using the k-medoids method.[15,16] The high number of 296,826 unique trajectories *(Table 2)* emphasizes the variability in this raw dataset. High sensitivity to outliers could result in many clusters which would make visualisation unfeasible. The low complexity of the k-medoids algorithm results in a fast clustering process.[16] This property is deemed most important in clustering this dataset and led to the selection of the k-medoids algorithm. The DTW distance matrix was used as input for the k-medoids algorithm, this clustered all trajectories into groups of similar sequences by using their pairwise DTW distance. K-medoids clustering was done for a range of 1 to 24 clusters *(Fig 3: 4.Cluster)*. For each number of clusters a total distortion score was computed by taking the sum of squared distances from each point to its assigned centre. When clustering, the aim is to minimize the distance between points in a cluster, the distortion score measures this distance. Distortion scores over the number of clusters were then plotted *(Fig. 4)*. The number of clusters after which the distortion score didn't decrease significantly anymore (the elbow of the graph) was selected as the optimal number of clusters. Based of the created graph, a number of 11 clusters was chosen because the distortion starts dropping more slowly after 11 clusters.

Visualisation

The total follow-up time was calculated, for each patient, from its first and last recorded prescription dates. This total time range was divided into parts of 1 year containing all recorded prescriptions for each year. For each cluster, the total number of prescriptions per BNF chapter was calculated separately for each time range of 1 year. So, per BNF chapter and follow-up year, the number of prescriptions from all trajectories within a cluster were summed up. This resulted in 11 data frames, one for each cluster, describing the raw prescription count for every follow-up year of the BNF chapters 1-10. These 11 data frames were generated for both the prevalent and incident cohort *(Fig. 3: 5.Results)*. All data frames were then used to generate both a prevalent and incident heatmap for each cluster *(Fig. 7, 8)*. In addition to these raw heatmaps, normalised heatmaps were created by using z-score normalization over every row (BNF chapter). The z-score measures how many standard deviations a value is away from its mean. It allows to determine how usual or unusual a value is in a distribution. In a normal distribution, over 99% of values fall within 3 standard deviations from the mean. If a z-score is larger than 3 the value can be considered unusual.[17] For every BNF chapter, the mean number of prescriptions per follow-up year and its standard deviation was calculated. Then the z-score was calculated, for every value, by subtracting the raw prescription count from its mean followed by division over its standard deviation. Z-score normalization results in each row having a mean *(μ)* of 0 and a standard deviation *(δ)* of 1. This process accounts for the difference in the total number of prescriptions between the different BNF chapters. The resulting normalized heatmaps *(Fig. 9, 10)*, visualise the fluctuations over time within the BNF chapters. Z-score normalisation is described in the formula below.

$$z\ score\ = \frac{(x - \mu)}{\delta}$$

Raw and z-score normalized heatmaps were made for both cohorts. The raw heatmaps provide insight in the fluctuations in number of prescriptions over time, for each BNF chapter, by comparing over the X-axis. In addition it also shows the number of prescriptions of each BNF chapter relative to the others when compared over the Y-axis. A disadvantage of these raw visualizations arises when the difference in number of prescriptions between some BNF chapters is big. This results in dark rows, for some underrepresented BNF chapters, whereby changes over time are undetectable. The z-score normalized heatmaps aim to solve this by normalizing, over the BNF chapters, such that the mean of each row is 0 and the standard deviation is 1. This normalization helps to interpret the change over time for each

BNF chapter separately, but prescription counts between BNF chapters can't be compared. In addition to the heatmaps, bar-charts describing the distribution of BNF-chapters within clusters were added. These multiple visualizations were selected to show the results from different angles, allowing for both within- and between-cluster comparison.

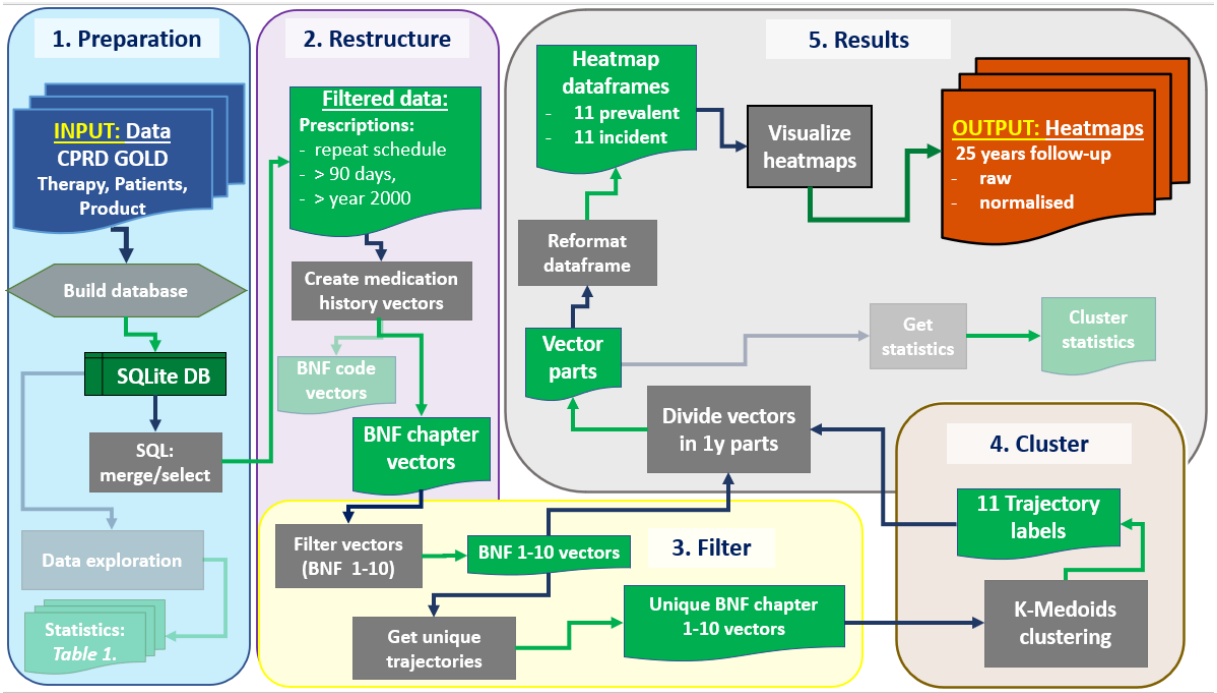

Figure 3. *The global workflow*
The initial input data is shown in blue and the goal output in red. The performed processes are in grey and files generated during analysis are in green.

3.4 Motivated settings for selected method(s)

At the filtering stage, prescriptions were selected based on some criterion. These criterion and the motivations behind their selection are described subsequently. The choice was made to only include prescriptions from BNF chapters 1-10. These chapters were considered most informative in describing medication trajectories in patients with diabetes type 2. The remaining chapters, 11 – 23, mostly consist of non-medications and prescriptions like topical creams. The exclusion of chapter 11 – 23, also results in less unique medication trajectories which is helpful to speed up the clustering algorithm. The assumption that chronic drug treatment is at least 90 days, and never ends, was made to simplify the creation of medication history vectors and its visualizations. Prescriptions of short duration could cause trajectory changes being too frequent to be visible in a heatmap, making visualizations unclear. The focus of this study is on chronic medication usage. Many chronic conditions require lifelong treatment which led the choice to simplify trajectories by assuming these prescriptions never end. The cut-off of 90 days was selected because acute treatment is expected not to exceed 3 months. Chronic medications are also expected to be on a repeat schedule because of its prolonged usage, which added this feature as condition in data selection. The exclusion of all prescriptions before the year 2000 reduces the amount of data while still including the most recent data. This speeds up subsequent analysis and possibly improves data quality by excluding irrelevant and outdated data.

During visualisation, the heatmaps were rated on detail and clarity at different settings and the optimal were selected. Only 25 years of follow-up is visualized in the heatmaps. After 25 years the number of prescriptions per BNF chapter becomes very low, and is invisible in the heatmaps. Limiting the follow-up years, results in a shorter X-axis, making the heatmaps easier to read.

4 Results

4.1 Number of (unique) trajectories

The data were filtered and multiple versions of medication history vectors were created for each patient. The size and variability of these versions is described in *Table 2*. It shows that, when prescriptions are represented by their BNF chapter, the dataset ends up with less overall and unique trajectories compared to whole BNF code representation. This reduction in total number of trajectories results from exclusion of trajectories containing less than 5 prescriptions for both the BNF code and BNF chapter vectors. When only BNF chapter 1 to 10 prescriptions are selected, there is a drop in the number of unique trajectories from 134,497 to 44,497, reducing data variability. The total number of trajectories stays equal to previous version because no minimal prescription number was defined for trajectories in this version. The incident cohort is the smallest in size (193,138) and variability (27,832) due to the selection of incident prescriptions only and removal of trajectories that end up empty.

Table 2. *Number of (unique) trajectories*
First three columns represent the whole (prevalent) dataset. The amount unique trajectories reduce when drugs are represented by their BNF chapter only and reduce even more when only chapters 1 to 10 are maintained. The incident cohort has less prescriptions due to the removal of drugs recorded in the first year of follow-up and ends up with less unique and informative trajectories overall. For each BNF code/chapter only the first occurrence is maintained in the vector. Vector version 3 was used for clustering the trajectories into groups and the generation of the prevalent visualisations. Vector version 4 was used for the visualisations of the incident cohort.

Vector version	**Vector version 1: Whole BNF code**	**Vector version 2: BNF chapter**	**Vector version 3: BNF chapter 1 - 10**	**Vector version 4: BNF chapter 1 – 10**
Cohort	Prevalent	Prevalent	Prevalent	Incident
# Trajectories	300,264	200,075	200,075	193,138
# Unique trajectories	296,826	134,397	44,495	27,832

4.2 Clustering

To find the optimal number of clusters (k), the elbow method was used. *Figure 4* shows the results from running k-medoids, on the medication vectors, for a range of 1 to 25 clusters. The Y-axis shows the total Within-cluster Sum of Square (WSS score), also called distortion, and X-axis the number of clusters. As the number of clusters increases, the WSS value starts to decrease. The WSS value is largest for only one cluster and rapidly drops when more are added. The graph shows that after 11 clusters, the WSS score doesn't decrease significantly anymore. This point is defined as the elbow of the graph and represents the optimal number of clusters for this dataset. At 8 clusters the graph shows another possible elbow which change in slope is about as sharp as the one located at 11 clusters *(Appendix C.2)*. It's clear that more than 11 clusters are inappropriate values as they are not the elbow of the graph, which is where the slope changes sharply.

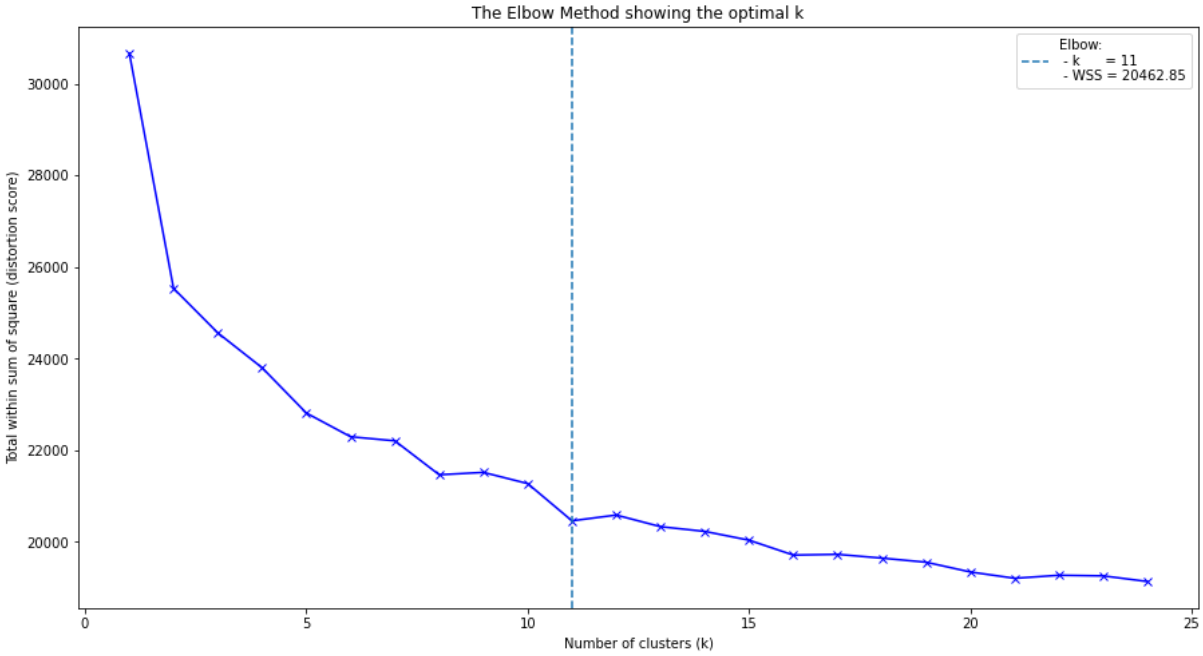

Figure 4. *The elbow method*
The x-axis shows the number of clusters and y-axis the explained variance as a function of the number of clusters. The WSS (distortion) score is defined as the sum of the squares of the distances of all points (to its centre) within a cluster.

Cluster properties

The sizes of clusters, together with some characteristics, are described in *Table 3*. Cluster 2 is the biggest in size, containing 69,315 trajectories which is 35.89% of the total dataset. Cluster 3 is the smallest and contains only 3,129 (1.62%) patients. Even though clusters show a big size difference, the person years follow-up is stable over all clusters. The percentage of females seems to be a bit higher in cluster 6, compared to the others, but doesn't seem to differ much. All clusters seem to have comparable properties and none diverge to the extent that it defines a cluster.

Table 3. *Cluster properties*
The number of patients directly reflects the number of trajectories in a cluster, as each patient holds one trajectory. Within cluster age is described by the median age and its 25th and 75th percentiles in parentheses. Person years follow-up represents the sum of total follow-up years divided by the number of patients.

Cluster	0	1	2	3	4	5	6	7	8	9	10	Total
# Patients	19,863	6,273	69,315	3,129	6,692	5,496	9,750	29,751	14,215	5,089	23,565	193,138
% Female	49.42	55.51	45.14	57.21	53.80	47.07	59.11	44.54	54.68	53.47	48.26	48.38
Median age (25th - 75th)	77.0 (67.0 - 85.0)	78.0 (69.0 - 85.0)	73.0 (63.0 - 82.0)	76.0 (67.0 - 84.0)	76.0 (68.0 - 84.0)	75.0 (66.0 - 84.0)	78.0 (69.0 - 85.0)	75.0 (66.0 - 84.0)	77.0 (68.0 - 85.0)	77.0 (69.0 - 85.0)	75.0 (65.0 - 83.0)	75.0 (65.0 - 83.0)
Average person years follow-up	11.89	12.73	10.98	14.93	15.57	13.16	13.98	11.54	13.14	14.06	13.01	12.17

4.3 BNF chapter distribution

The distribution of BNF chapters within clusters is shown in *Fig. 5* for the prevalent, and *Fig. 6* for the incident cohort. Both figures closely resemble each other, indicating that there isn't much of a difference in the class of medications both cohorts get prescribed over the entire follow-up period. The exclusion of BNF chapters prescribed in the first follow-up year did not result in BNF chapters being

excluded from the whole incident cohort. For both cohorts the BNF chapter distribution between clusters is also similar. Even though the trajectories were clustered according to sequence patterns, there seem to be no notable differences in the class of medications prescribed in each group. Clusters could likely be defined by their differences in prescription sequence, grouping together trajectories with comparable sequence of the BNF chapters over time.

All 10 selected BNF chapters are represented in every cluster. Oral antidiabetic prescriptions belong to BNF chapter 6, the endocrine system. BNF chapter 6 is the only chapter that covers more than 10 percent of every cluster *(Appendix C.3)*. BNF chapter 3, 5, 7 and 8 are visibly present in lower numbers than the remaining chapters. On the other hand, prescriptions from BNF chapters 1, 2, 4 and 6 are more commonly prescribed in this population.

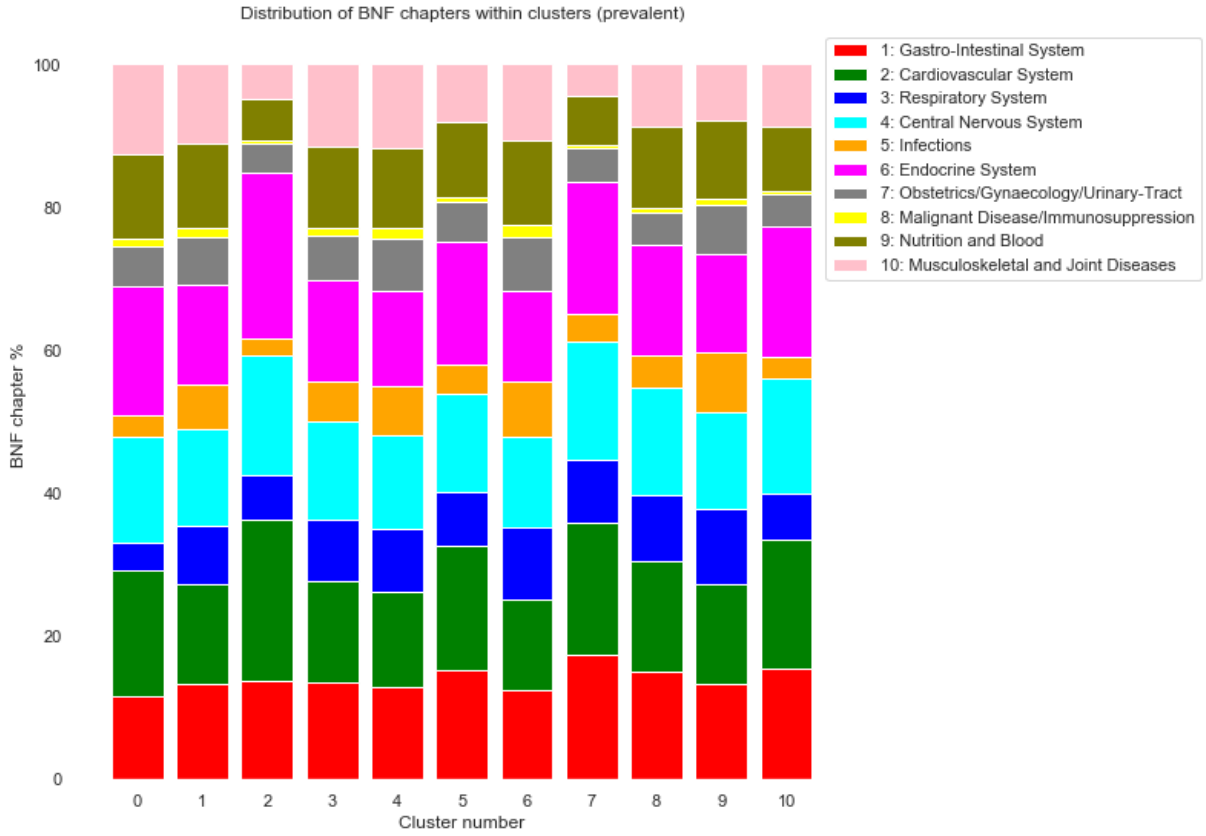

Figure 5. *The distribution of BNF chapters 1-10 within clusters for the prevalent cohort*

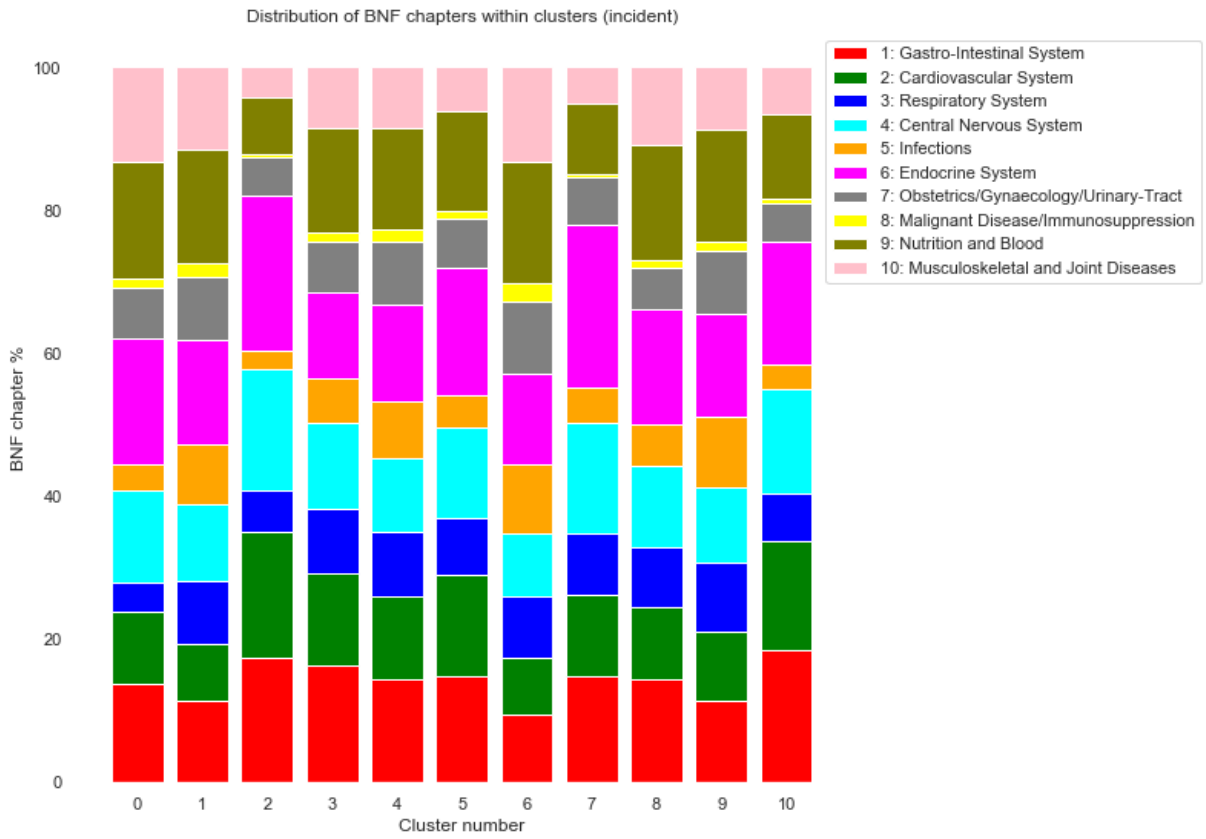

Figure 6. *The distribution of BNF chapters 1-10 within clusters for the incident cohort*

4.4 Heatmaps

The treatment patterns over time were visualised separately for each cluster. Each cluster represents a group of similar trajectories, that's a treatment pattern over time that's common for patients in this population. The raw and normalized heatmaps are shown in *Fig. 7 and 9* for the prevalent cohort, and *Fig. 8 and 10* for the incident cohort. The *Figures 7-10*, each consist of 11 different heatmaps, one for each cluster. Their X-axis represents time in follow-up years and Y-axis the BNF chapters 1 to 10. Each square in the heatmaps describes the total amount of prescriptions for a specific BNF chapter, in a specific year of follow-up. Note that the legends, next to the raw heatmaps, differ in scale due to the highly variable cluster sizes. The raw heatmaps *(Fig. 7, 8)*, show areas which have a relatively low number of prescriptions in a dark colour and areas with a high prescription count in a bright colour. Locations having 0 prescriptions are represented by black squares. The normalized heatmaps *(Fig. 9, 10)*, show the number of standard deviations each value is away from the mean of its row. Both a positive z-score of 3 as well as a negative of -3, imply that the value is three standard deviations away from its mean. The mean of a row is defined as the mean number of prescriptions, per year, for a specific BNF chapter. Values around the rows mean are coloured white and have a z-score close to zero. Negative z-scores are shown blue and represent values that are below the mean prescription count. Positive z-scores are coloured red, these values are above the mean prescription count of its row.

Prevalent and incident differences

In the prevalent cohort most BNF chapters are present from an early point in time, but in a few clusters a single BNF chapter shows up after a few follow-up years. In cluster 9 it even takes 17 years for BNF chapter 3 to appear *(Fig. 7)*. In the incident cohort none of the clusters have new BNF chapters

prescribed later in the trajectory. This implies no new use of medications, at BNF chapter level, and indicates that drug regimens are relatively stable after a certain period of time *(Fig 9.)*. The presence of new BNF chapters in the raw prevalent heatmaps and its absence in the raw incident heatmaps, result from the different structures of the vectors representing the cohorts. During generation of the incident vectors, all BNF chapters that showed up in the first year of follow-up were excluded from the entire trajectory. These could be pre-existing prescriptions, meaning there is no information on the first prescription date. Their exclusion results in a change in trajectory sequence compared to the prevalent cohort. The pre-existing medications, that are present in the prevalent cohort, repeatedly seem to be stable for some time before new medications are prescribed. This is indicated by the black blocks present in *Figure 7*, these represent 0 prescriptions for some BNF-chapters at the start of follow-up. These black blocks are expected to arise when pre-existing prescriptions are stable for multiple follow-up years before incident prescriptions occur. This is the case, for example, at BNF chapter 2 in cluster one of the prevalent cohort *(Fig. 7)*. In this specific case, the first 10 years of follow-up only show BNF chapters that all already were prescribed at the first year of follow-up. These pre-existing BNF chapters then show an increase in number of prescriptions over time. This increase is caused by new incident users but no new BNF chapters are prescribed, until after multiple years a new BNF chapter appears. This visualises that incident BNF chapters are often prescribed after multiple years of repeated pre-existing prescriptions.

Between cluster differences

The normalized incident heatmap *(Fig. 10)*, is characterized by a overall pattern within all clusters. This pattern shows red coloured values at the start of follow-up, which then change to white over time and turn blue near the end of follow-up. This pattern suggests that, the number of prescriptions overall is higher than average in the first few years of follow-up (red) and lower than average near the end (blue). Although all clusters show this same pattern overall, the clusters do have distinct differences in their change over time. Each cluster shows a white line between the red and blue regions, but it is located at a different time point in each cluster. This shows that, the amount of time to reach the highest prescription rate (darkest red) and then decrease to its average prescription rate (white), varies for each cluster. All clusters, except cluster two, mostly show an increase in number of prescriptions over the first few years of follow-up, marked by red getting darker *(Fig. 10)*. The incident heatmaps show that cluster two received most prescriptions at the first follow-up year, followed by a decrease directly after the first year. Therefore, cluster two seems to have more notable differences compared to the other clusters. Cluster two contains most trajectories of all clusters, and so is the biggest in size *(Table 3)*. This cluster diverges from all others regarding both its pattern over time and size. These observations together suggest that there is a large group of similar trajectories that diverge a bit more from the rest. This group of patients is characterized by a high number of prescriptions at the start of follow-up, followed by a fast decline in prescription numbers.

Some small regions in the incident heatmaps are not consistent with the overall pattern and therefore stand out. One of these regions is located at cluster three, in the row of BNF 8 *(Fig. 10)*. The number of prescriptions belonging to this chapter shows a blue colour in the first follow-up year. This implies that the chapter is prescribed in numbers below its average at the start of follow-up. This is inconsistent with the overall pattern which is mostly red in early follow-up. Another remarkable region is located at incident heatmap *(Fig. 10)* year one of BNF chapter 9 in the clusters zero, two, five and seven. This location is coloured dark red in multiple clusters, which implies that BNF chapter 9 is most frequently prescribed at the first follow-up year. Cluster three and four, on the other hand, show the opposite pattern. These clusters are coloured blue at the first year of BNF chapter 9, meaning it is prescribed below average.

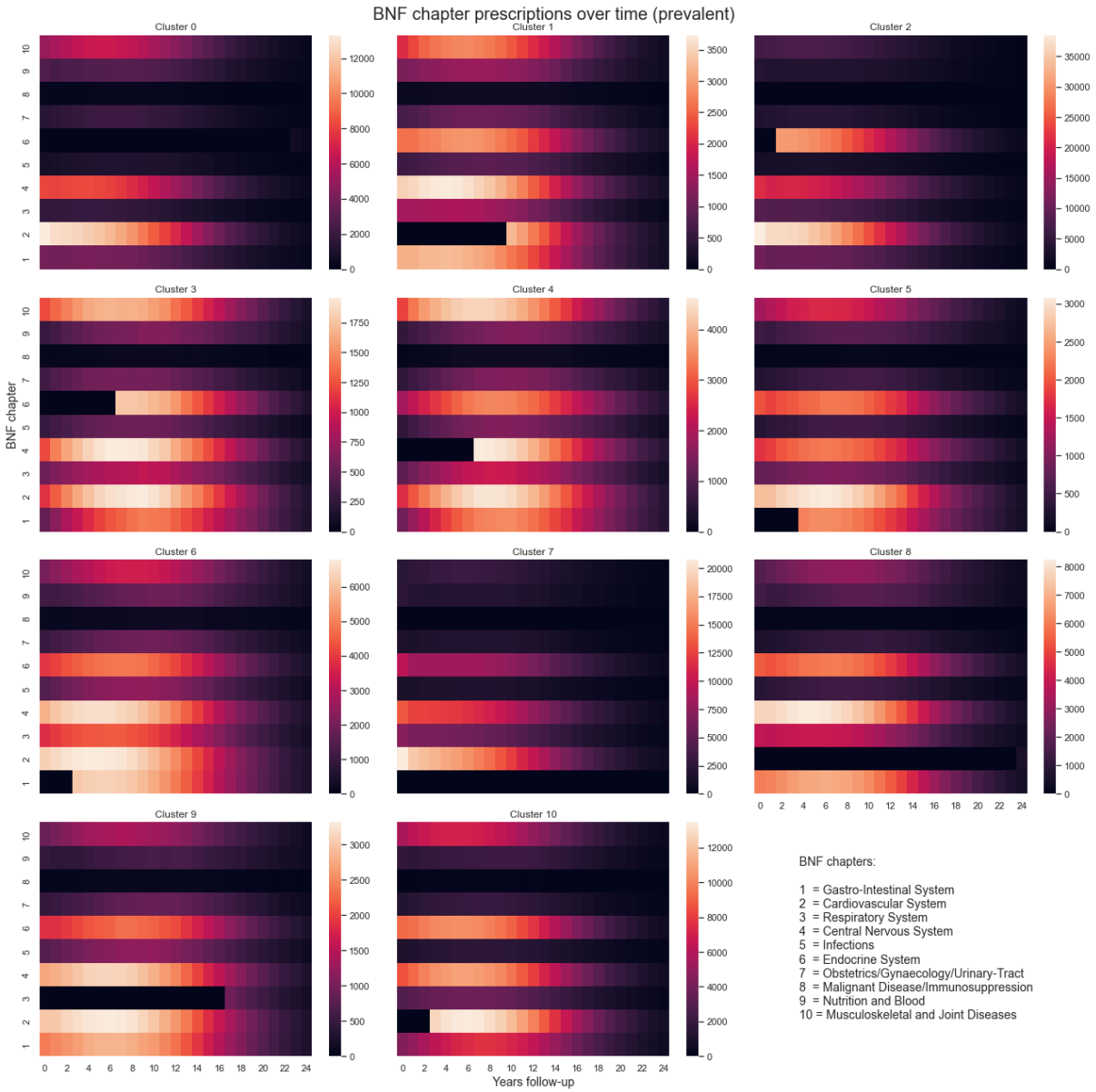

Figure 7. *Raw prescription heatmaps for the prevalent cohort*
Each cluster is visualized in an individual heatmap showing the raw number of medication prescriptions belonging to BNF chapters 1-10 over time in steps of 1 year.

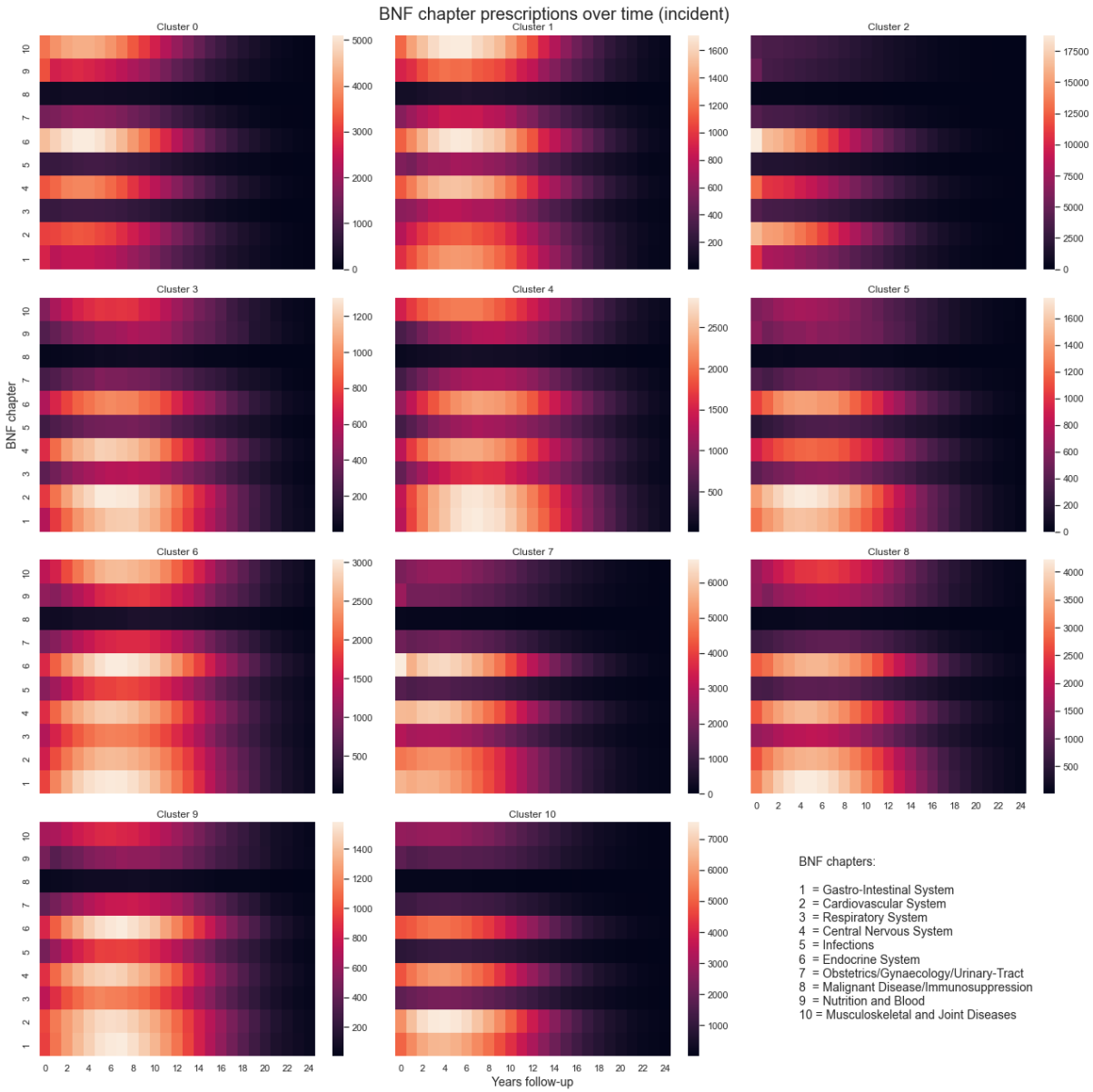

Figure 8. Raw prescription heatmaps for the incident cohort

Each cluster is visualized in an individual heatmap showing the number of medication prescriptions belonging to BNF chapters 1-10 over time in steps of 1 year.

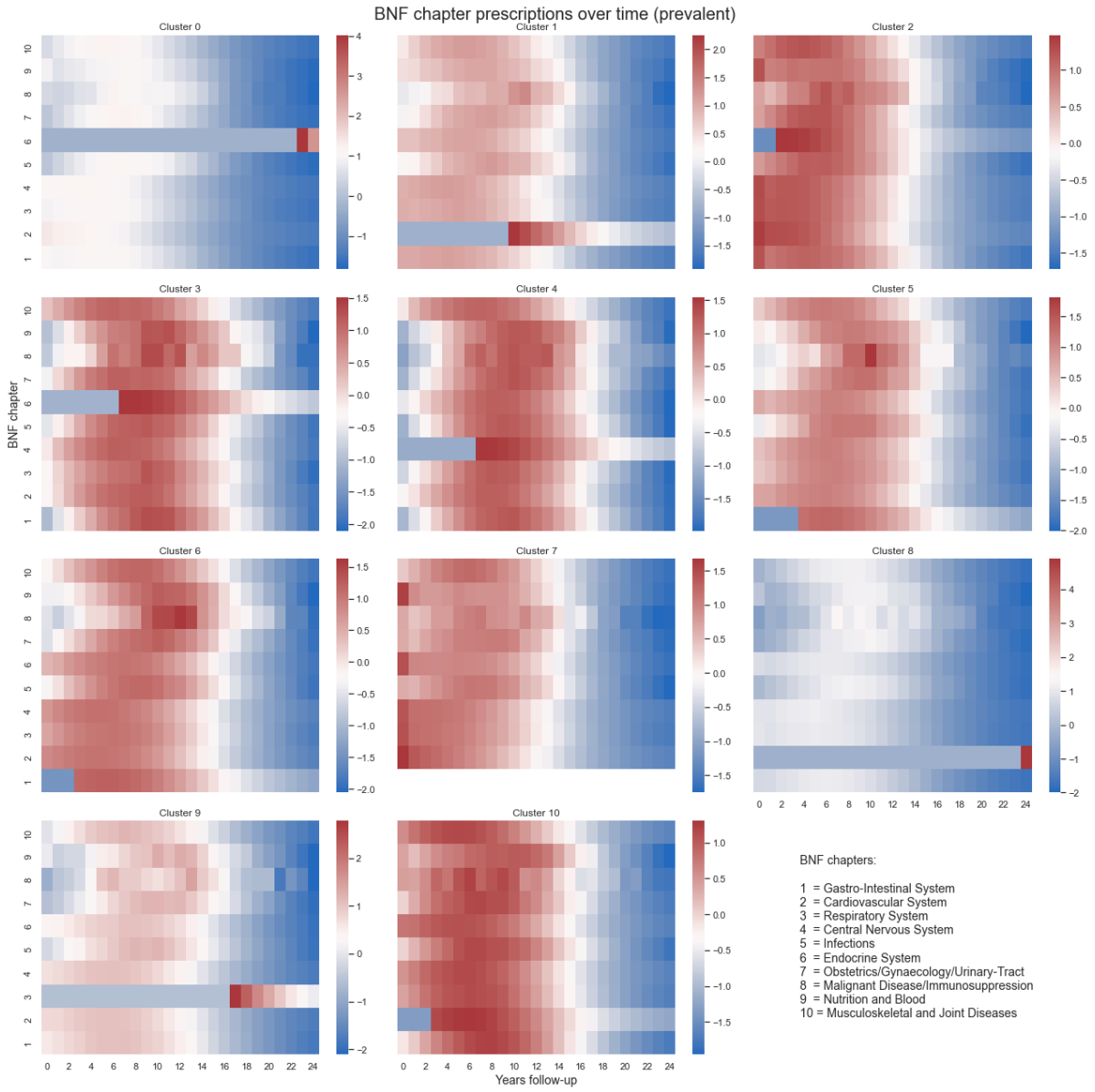

Figure 9. *Normalized prescription heatmaps for the prevalent cohort*
Each cluster is visualized in an individual heatmap showing the z-score normalized medication prescriptions belonging to BNF chapters 1-10 over time in steps of 1 year.

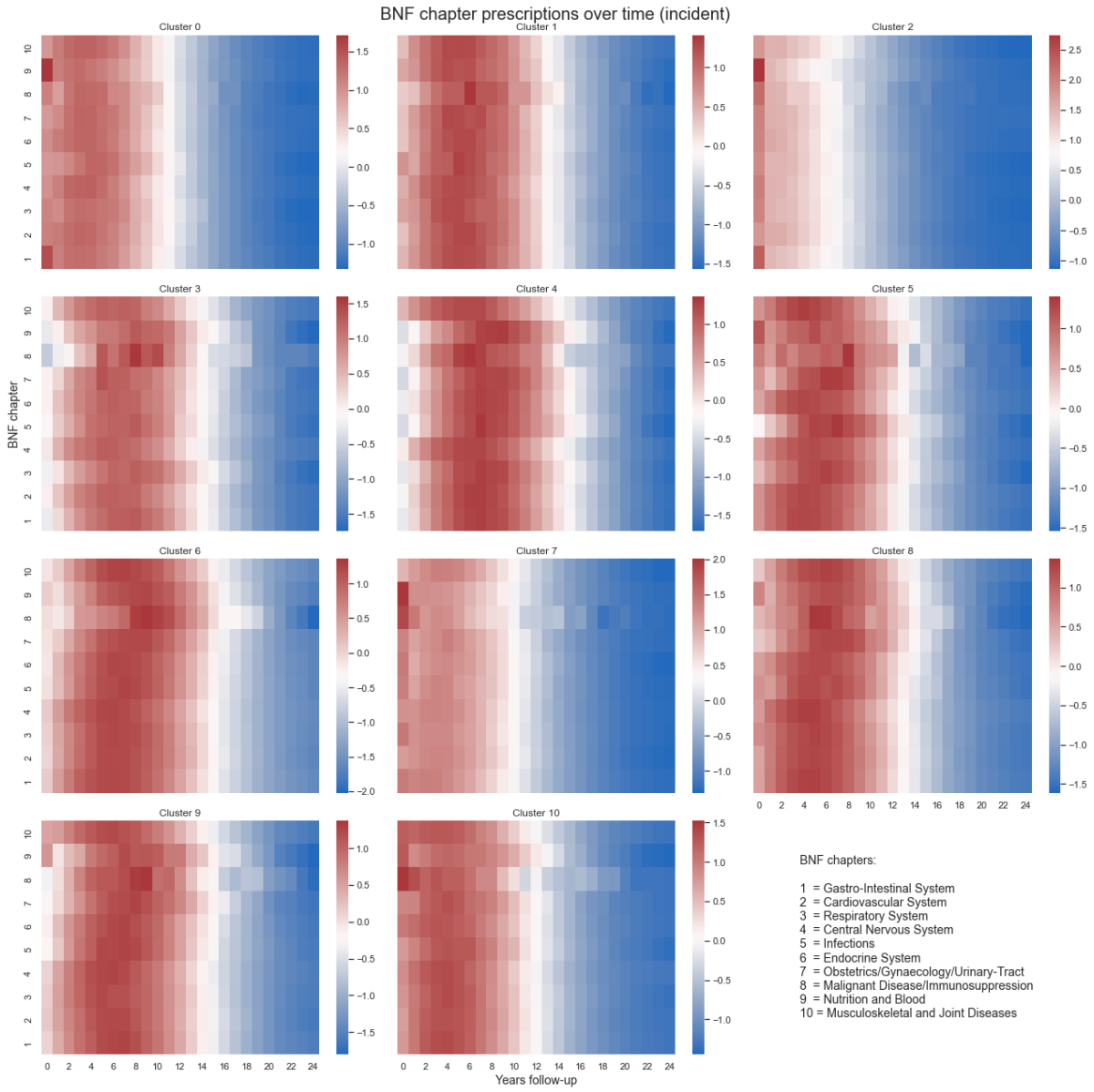

Figure 10. *Normalized prescription heatmaps for the incident cohort*
Each cluster is visualized in an individual heatmap showing the z-score normalized medication prescriptions belonging to BNF chapters 1-10 over time in steps of 1 year.

5 Discussion

5.1 Visualisation design

The visualisations that were generated *(Fig. 5 - 10)*, all together aim on describing the whole complexity of the medication trajectories. The raw heatmaps *(Fig. 7, 8)*, visibly show trajectories change over time (X-axis) and the relative number of prescriptions for each BNF chapter (Y-axis). BNF chapters having a relatively low number of prescriptions compared to the others are to dark to read. For these dark rows, the raw heatmaps are unable to describe the change over time but they do show that these chapters are prescribed in lower numbers overall within a cluster. By reflecting raw numbers, these heatmaps succeeded to provide a global overview of the medication trajectories over time. The Z-score normalised heatmaps *(Fig. 9, 10)* were added to clarify fluctuations over time separately for each BNF chapter. Using the Z-score for each row resulted in a mean of 0 and standard deviation of 1 for each BNF chapter. This compensated for the differences in total number of prescriptions between BNF chapters and resulted in a visible change over time for each chapter. While the raw heatmaps provide a global overview of the trajectories over time, the normalized heatmaps expand this visualization by adding focus on the change over time within each chapter. The bar-charts *(Fig. 5, 6)* show the distribution of BNF chapters within the clusters. These three different visualization types have showed to reinforce each other by focusing on different data aspects. Together they extend a global overview (raw heatmap) with a detailed change over time (normalized heatmap) and information on which BNF chapters are commonly prescribed (bar-chart). The selected visualization design, combining these three figures, showed useful in the pursue of a visualization that captures the trajectories in their full extent.

5.2 Compared to previous research

No similar visualizations, of trajectories over time, were found in previously published research. A similar study by *Giannoula et al*, on identifying temporal patterns in disease trajectories, published a visualisation containing both information on diagnose sequence over time and frequency of disease appearance.[6] This visualisation contains no concrete definition of time. Time was visualised by arrows between diagnoses and length of arrows doesn't reflect the amount of time. Their visualisation describes the most common sequences by showing one time step (arrow) between each diagnose, without giving an indication about the amount of time between diagnoses. From this it's possible to retrieve the common diagnose sequences but not information on the total follow-up time of trajectories, or the time between diagnoses. This type of visualization isn't capable to fully describe the complexity of trajectories because, firstly, it lacks a defined timeline. Secondly, it generalizes all trajectories within a whole cluster into a single sequence without allowing for variations. This visualization is sufficient for providing an overview of the most common trajectories and their sequence but isn't capable to reflect the variability of trajectory sequences and time intervals. The visualisations designed in this study aimed to extend this previous visualisation, by *Giannoula et al*, to describe the sequences while also reflecting trajectory variability over a clearly defined timeline. The heatmap visualisations that were generated *(Fig. 7, 8, 9, 10)*, combined with the bar-charts *(Fig. 5, 6)*, successfully visualised medication trajectories with an increased complexity in comparison to previously published visualisations.

Per BNF chapter and follow-up year, the number prescriptions from all trajectories within a cluster were summed up. These numbers are the base of the heatmaps by providing the value of each square in the visualization. Each square in the heatmap is representing the prescription count of its corresponding BNF chapter and follow-up year. Therefore the heatmaps allow the projection of precise prescription counts over clearly defined timepoints for each class of medications. These counts can be described as the, within cluster, prescription counts grouped by medication class over time-points ranging one year. This grouping directly implements information on the common sequence

itself, patient variability and time intervals into one visualization by reflecting each group in a separate square in the heatmap.

5.3 Between cluster differences

Cluster 2 is characterized by a high number of prescriptions direct at the start of follow-up, followed by a fast decline in prescription numbers. All other clusters first show a short increase in numbers until the highest prescription rate is reached after some time, followed by a decrease. Cluster 2 differs from the rest regarding this pattern over time and also its notably bigger size. These observations together suggest that there is a large group of similar trajectories that diverges more from the rest. The trajectories belonging cluster 2 share the same pattern having a high prescription rate at the start of follow-up which stabilizes fast. Patients in this cluster could be characterized by a stable pattern after their treatment schedule is first selected at the start of follow-up. This implies that the largest group of diabetes type 2 patients is represented by cluster 2, and they receive most of their prescriptions in the first year of follow-up. The other groups of common trajectories all show an increase over the first few years of follow-up before the highest number of prescriptions is reached. Although they share this overall pattern, the normalized heatmap shows clusters do differ in prescription rate over time for the different BNF chapters. The trajectories were clustered in groups based on their sequences over time. Clustering aimed for the creation of groups of similar trajectories, therefore the heatmaps were expected to differ between clusters.

BNF chapter 8 contains malignant disease and immunosuppression medications. The raw incident heatmap *(Fig. 8)* shows a black line for this chapter in all clusters, therefore its change over time is undetectable. This is a limitation of the raw heatmap which results from BNF chapter 8 having a low total number of prescriptions compared to the other chapters. The normalized heatmap *(Fig. 10)* accounts for the size differences between the BNF chapters, therefore change over time for BNF 8 becomes detectable. The low number of total prescriptions for BNF 8 could be an explanation for the differing pattern of BNF 8 in cluster three *(Fig. 10)*. A low number of total prescriptions can allow fluctuations over time to arise more easily in the normalized heatmaps. BNF 8 is less common and appears in less trajectories on average. As a result, its change over time in the incident heatmap originates from less trajectories than the more common BNF chapters, this allows for less variability and more pronounced fluctuations. The more commonly prescribed BNF chapters seem to be a logical result from the cohort selection. The selection of patients with diabetes doesn't focus on certain diseases like cancer because it isn't directly linked to diabetes. The low prescription numbers of BNF 8 are therefore expected in this population. In addition, the overall fluctuations over time could be even more emphasized in cluster three because it is the smallest in size *(Table 3)*.

The incident normalized heatmap *(Fig. 10)* shows that year one of BNF chapter 9 differs between clusters. Cluster number zero, two, five, seven and ten show this location coloured dark red. In cluster three and four the same location is coloured blue. BNF chapter 9 represents medications for nutrition and blood related treatments. It includes prescriptions like vitamins, foods and anaemias treatments. The opposite representations of this location in different clusters could imply that it's common for diabetes patients to have nutrient deficiencies at the start of follow-up, but there is also a group of patients that develops these deficiencies at a later moment. Patients with diabetes are known to be prone to nutrient deficiencies and some medications used to treat diabetes can increase nutrient requirements.[18] These observations could possibly explain why BNF chapter 9 is well represented in each cluster, with most clusters having at least 10 percent of prescriptions from this chapter. Most clusters suggest that BNF 9 is prescribed well above average at the first follow-up year. If nutrition deficiencies would be caused by diabetes medications, BNF 9 would show an increase directly after an increase in BNF 6, because chapter 6 includes diabetes medications. The heatmaps don't clearly show this pattern and therefore don't suggest that diabetes medications could cause nutrition deficiencies.

5.4 Common trajectories

Using clustering, the trajectories were grouped based on temporal patterns they share. These clusters represent the most common trajectories in the cohort. Using DTW distance and k-medoids clustering 11 clusters were identified. This suggests that there're 11 different treatment patterns that are often followed by patients with diabetes type 2. The number of 11 clusters was selected through visual inspection of the distortion score over the number of clusters used *(The elbow method: Fig. 4)*. Using this graph, the elbow was identified at 11 clusters. Choosing the elbow of the graph is always a subjective choice. The same graph *(Fig. 4)* shows a sharp change in slope (elbow) at 11, 6 and 8 clusters resulting in multiple candidate elbows. It's clear that more than 11 clusters are inappropriate for this cohort because they don't show a clear elbow. The choice was made to select the elbow at 11 clusters because diabetes patients are strongly linked to multimorbidity and various treatment patterns. The complexity of this cohort led to the assumption that the actual number of common trajectories is likely to be the elbow at the highest number of clusters.

The distortion score is above 30,000 when all trajectories are in one cluster and reduces as more clusters are used *(Fig. 4)*. When 11 clusters are used the score is reduced to 20,463, meaning that around 32% of the variability is explained by grouping the trajectories in 11 clusters. An addition to this analysis could be the testing of other clustering methods using the elbow graph. A clustering algorithm that gets to a lower distortion could imply a more accurate clustering of trajectories. The dataset size was found to be a limitation in selection of clustering methods because of the time it takes to run these algorithms. To find an appropriate number of clusters, the clustering algorithm was repeatedly executed for a range of clusters. The generation of a pairwise distance matrix for 44,495 unique trajectories took 10 minutes for this dataset. This is an acceptable time but the limited 16 GB RAM available still causes termination of this process when the number trajectories is increased to a certain point. More available RAM would increase the number of trajectories that can be included in the distance matrix and subsequent clustering. The k-medoids algorithm, for a range of 24 clusters, took 15 hours and 23 minutes to complete. Although k-medoids is a bit slower than k-means, it's described as a low complexity clustering algorithm. Using a clustering algorithm of higher complexity than k-medoids isn't advised based on these findings. Other clustering algorithms could be considered when more computing power is available. A server system that has multiple processing units (cores) is capable to execute multiple instructions in parallel, this shortens execution time and could be a valuable addition to this analysis.

It could be possible that the distortion can't be reduced any further because of the variability of trajectories. This would suggest that no more variability could be explained by clustering the trajectories into groups and the current 11 clusters are already near optimal. Using more clusters would always result in a reduce in distortion score but could lead to overfitting. All patients in the cohort were selected on having at least one oral antidiabetic prescription and so all selected patients have diabetes type 2. Before analysis, only patients having at least 5 chronic medication prescriptions were selected. This filtering step results in the selection of multimorbidity patients which likely have multiple chronic conditions. Patients vary in which chronic conditions they have, resulting in various possible treatment patterns making trajectories complex. Trajectories belonging to the same cluster therefore are expected to still have significant variability in their sequences, and so the distortion score isn't expected to get to a low number.

5.5 BNF code

For each patient, a vector was made which describes their prescriptions over time. The BNF chapter was used to represent its prescription and only the first occurrence of each chapter was added to the vector. The BNF code describes a medication classification system of multiple levels, with each level adding more detail about the medication. The use of the second level BNF code in addition to the BNF chapter could be a valuable addition to this analysis. Re-doing this whole analysis using second level

BNF codes leads to different results and possibly new insights. Second-level analysis is expected to have longer and more unique vectors because there are more groups to which prescriptions can belong. Because of the increase in the number of unique trajectories, clustering is expected to be problematic and this analysis would probably not be possible to execute on a single computer. Showing the prescription count for all second level BNF codes over time in a heatmap would result in large figures due to the high number of possible two-level BNF codes. Aiming on clear figures, visualisation of second-level BNF vectors in heatmaps is still advised to use BNF chapter counts. The addition of a few relevant second level BNF codes to the Y-axis could be possible. The second level BNF code 6.1 describes drugs used in diabetes and could be a valuable addition to the heatmap. The bar-carts could be changed to show the distribution of the second level BNF codes. This adjustment could provide more detailed information on the commonly prescribed medications, a the second level BNF, for this cohort.

6 Conclusion

The overall objective of this research is to visualize longitudinal medication trajectories among patients with type 2 diabetes.

Medication history vectors, per patient, were designed as a sequence containing all BNF chapters of chronic prescriptions. These vectors were divided into parts containing all prescriptions in a one year range. This design proved to be useful for visualization over time using heatmaps. Clustering these vectors could identify 11 groups of common trajectory sequences from these vectors.

Three types of visualizations using a raw heatmap, normalized heatmap and bar-chart were made. The heatmaps were successful in the visualisation of trajectories over time by showing the prescription count per BNF chapter, for each follow-up year. The distribution of prescriptions from BNF chapters 1-10, within each cluster, was visualised cross-sectionally using bar charts. These showed that all clusters are similar in the percentages they get prescribed, from each chapter, over the total follow-up time. Although all clusters were found to have similar distributions, the heatmaps show distinct differences between clusters changes over time. This implies that there’re 11 common trajectories which contain the same BNF chapters but their sequences over time differ.

The largest cluster showed the highest prescription rate at first year of follow-up, followed by decline. All other clusters first increase in number prescriptions in the beginning of follow-up until a peak is reached and then decrease. This implies that the most common treatment pattern, for this cohort, mostly receives new prescriptions (based on BNF chapter) in the first year of follow-up followed by a stable trajectory. This could be explained by the chronic nature of diabetes. After diabetes medications are prescribed, these treatments are expected to continue for a long time which could stabilize the trajectories over time.

The generation of separate visualisations for a prevalent and incident cohort, distinguished pre-existing prescriptions from BNF chapters that were first prescribed during the follow-up period. These showed that prevalent trajectories are often stable for multiple years before new BNF chapters are prescribed.

Altogether this study has succeeded in identifying a useful method to visualize medication trajectories over time, while capturing as much complexity possible. The three visualisation types combined show a promising visualisation that’s one step closer to describing the trajectories in their full extent. It should be taken into account that medications were represented by BNF chapter. Implementation of the second level BNF codes provides more detail and could result in increased insights. So although

the visualisations are useful, they don’t provide detailed insight in the common medication trajectories of patients with type 2 diabetes.

7 Acknowledgements

Words cannot express my gratitude to Daniala Weir for her time, patience and feedback. Her generously provided knowledge and expertise created a great learning experience for me these last ten weeks. I experienced a honest, calm and to the point communication style which I sincerely enjoyed. I want to thank David Liang for his accessibility, extended feedback and invested time. I’m grateful that Helga Gardarsdottir accepted to take on the role as second examiner and for her time.

8 References

1. Boyle JP, Thompson TJ, Gregg EW, Barker LE, Williamson DF. Projection of the year 2050 burden of diabetes in the US adult population: Dynamic modeling of incidence, mortality, and prediabetes prevalence. *Population Health Metrics*. 2010;8. doi:10.1186/1478-7954-8-29

2. Formiga F, Agustí A, José AS. Polypharmacy in elderly people with diabetes admitted to hospital. *Acta Diabetologica*. 2016;53(5):857-858. doi:10.1007/s00592-015-0818-9

3. Bauer S, Nauck MA. Polypharmacy in people with Type 1 and Type 2 diabetes is justified by current guidelines-a comprehensive assessment of drug prescriptions in patients needing inpatient treatment for diabetes-associated problems. *Diabetic Medicine*. 2014;31(9):1078-1085. doi:10.1111/dme.12497

4. AL-Musawe L, Martins AP, Raposo JF, Torre C. The association between polypharmacy and adverse health consequences in elderly type 2 diabetes mellitus patients; a systematic review and meta-analysis. *Diabetes Research and Clinical Practice*. 2019;155. doi:10.1016/j.diabres.2019.107804

5. *COMMENTS AND RESPONSES Is Long-Term Use of Antismoking Drugs Consistent with Public Health Goals or Pharmaceutical Marketing Goals?*; 2008. www.annals.org

6. Giannoula A, Gutierrez-Sacristán A, Bravo Á, Sanz F, Furlong LI. Identifying temporal patterns in patient disease trajectories using dynamic time warping: A population-based study. *Scientific Reports*. 2018;8(1). doi:10.1038/s41598-018-22578-1

7. Hanauer DA, Ramakrishnan N. Modeling temporal relationships in large scale clinical associations. *Journal of the American Medical Informatics Association*. 2013;20(2):332-341. doi:10.1136/amiajnl-2012-001117

8. Herrett E, Gallagher AM, Bhaskaran K, et al. Data Resource Profile: Clinical Practice Research Datalink (CPRD). *International Journal of Epidemiology*. 2015;44(3):827-836. doi:10.1093/ije/dyv098

9. Gulliford MC, Sun X, Charlton J, et al. Serious bacterial infections and antibiotic prescribing in primary care: Cohort study using electronic health records in the UK. *BMJ Open*. 2020;10(2). doi:10.1136/bmjopen-2020-036975

10. Lisa French. Prescribing Data: BNF Codes. Published April 24, 2017. Accessed June 13, 2022. https://www.bennett.ox.ac.uk/blog/2017/04/prescribing-data-bnf-codes/

11. Tiger Technologies. How large and busy can a MySQL database be? Published 2022. Accessed May 10, 2022. https://support.tigertech.net/mysql-size

12. sqlite. Appropriate Uses For SQLite. Published 2022. Accessed May 19, 2022. https://www.sqlite.org/whentouse.html

13. Rodgers LR, Weedon MN, Henley WE, Hattersley AT, Shields BM. Cohort profile for the MASTERMIND study: Using the Clinical Practice Research Datalink (CPRD) to investigate stratification of response to treatment in patients with type 2 diabetes. *BMJ Open*. 2017;7(10). doi:10.1136/bmjopen-2017-017989

14. Wolf A, Dedman D, Campbell J, et al. Data resource profile: Clinical Practice Research Datalink (CPRD) Aurum. *International Journal of Epidemiology*. 2019;48(6):1740-1740G. doi:10.1093/ije/dyz034

15. Chen Y, Liu X, Li X, et al. Delineating urban functional areas with building-level social media data: A dynamic time warping (DTW) distance based k-medoids method. *Landscape and Urban Planning*. 2017;160:48-60. doi:10.1016/j.landurbplan.2016.12.001

16. scikit-learn-extra. Clustering with KMedoids and Common-nearest-neighbors. Published 2019. Accessed June 7, 2022. https://scikit-learn-extra.readthedocs.io/en/stable/modules/cluster.html#:~:text=The%20complexity%20of%20K%2DMedoids,O%20(%20N%20K%20T%20)%20.

17. Gopal S, Patro K, Kumar Sahu K. *Normalization: A Preprocessing Stage*. www.kiplinger.com,

18. *Potential Micronutrient Deficiency Lacks Recognition in Diabetes*. http://statistics.defra.gov.uk/esg/publications/nfs/datase

19. Portilla R, Heintz B. Understanding Dynamic Time Warping. Published January 14, 2022. Accessed June 27, 2022. https://databricks.com/blog/2019/04/30/understanding-dynamic-time-warping.html

Appendices

Appendix A: Data

A description of most columns in the dataset is shown in *Table 4* and *Table 5*.

Table 4. *Dataset description of the therapy table*

Column name	*Field name*	*Description*	*Type*
Patient Identifier	patid	Encrypted unique identifier given to a patient in CPRD GOLD	TEXT
Event Date	eventdate	Date associated with the event, as entered by the GP	DATE
System Date	sysdate	Date the event was entered into Vision	DATE
Consultation Identifier	consid	Identifier that allows information about the consultation to be retrieved, when used in combination with pracid	INTEGER
Product Code	prodcode	CPRD unique code for the treatment selected by the GP	INTEGER
DMD Code	drugdmd	The mapped drug DMD code	TEXT
Staff Identifier	staffid	Identifier of the practice staff member entering the data. A value of 0 indicates that the staffid is unknown	INTEGER
Dosage Identifier	dosageid	Identifier that allows dosage information on the event to be retrieved. Use the Common Dosages Lookup to obtain the anonymised dosage text and extracted numerical information such as daily dose.	TEXT
BNF Code	bnfcode	Code representing the chapter & section from the British National Formulary for the product selected by GP	INTEGER
Total Quantity	qty	Total quantity entered by the GP for the prescribed product	INTEGER
Number of Days	numdays	Number of treatment days prescribed for a specific therapy event	INTEGER
Number of Packs	numpacks	Number of individual product packs prescribed for a specific therapy event	INTEGER
Pack Type	packtype	Pack size or type of the prescribed product	INTEGER
Issue Sequence Number	issueseq	Number to indicate whether the event is associated with a repeat schedule. Value of 0 implies the event is not part of a repeat prescription. A value ³ 1 denotes the issue number for the prescription within a repeat schedule	INTEGER
As Required	prn	Indicates if the prescription is to be supplied ‘as required’. Field available to GPs from end 2020.	BOOLEAN

Table 5. *Dataset description of the product file*

Column name	*Description*	*Type*
prodcode	CPRD unique code for the treatment selected by the GP	INTEGER
dmdcode	Unique product identifier from the NHS Dictionary of Medicines and Devices (dm+d) – the NHS standard dictionary for products licensed in the UK	TEXT
gemscriptcode	Gemscript product code for the corresponding product name - should be treated as a string field as it contains leading '0''s	TEXT
productname	Product name as entered at the practice	TEXT
drugsubstance	Drug substance	TEXT
strength	Strength of the product	TEXT
formulation	Form of the product e.g. tablets, capsules etc	TEXT
route	Route of administration of the product	TEXT
bnfcode	British National Formulary (BNF) code	TEXT
bnfchapter	British National Formulary (BNF) chapter	TEXT

Appendix B.1: Data exploration script

```
"""
Shadee Albronda
updated: 19/05/2022

Data exploration SQL script
Diabetes CPRD GOLD dataset

Install:
  - sqlite3
"""
import sqlite3

def Count_rows():
    print("Count total number of rows in therapy table")
    query = "SELECT COUNT(*) FROM therapy;"
    cur.execute(query)
    result = cur.fetchall()
    print(result)

    print("Count number of unique rows in therapy table")
    query2 = "SELECT COUNT(*) FROM (SELECT DISTINCT * FROM therapy)"
    cur.execute(query2)
    result2 = cur.fetchall()
    print(result2)

    print("Count total number of rows in product table")
    query3 = "SELECT COUNT(*) FROM product;"
    cur.execute(query3)
    result3 = cur.fetchall()
    print(result3)

    print("Count number of unique rows in product table")
    query4 = "SELECT COUNT(*) FROM (SELECT DISTINCT * FROM product)"
    cur.execute(query4)
    result4 = cur.fetchall()
    print(result4)
```

```
def Count_Female():
    query = "SELECT COUNT(*) FROM (SELECT DISTINCT * FROM patients)"
    cur.execute(query)
    result = cur.fetchall()
    print("number of patients: ", result[0][0])
    query2 = "SELECT COUNT(*) FROM (SELECT DISTINCT * FROM patients WHERE
gender = 2)"
    cur.execute(query2)
    result2 = cur.fetchall()
    print("number of females: ", result2[0][0])
    perc_female = int(result2[0][0]) / int(result[0][0])
    print("percentage female: ", round(perc_female,4) * 100)

def post_filter():
    query3 = "SELECT COUNT(*) FROM (SELECT DISTINCT pat_id FROM Chronic4) "
\
             "LEFT JOIN patients ON pat_id = patients.patid " \
             "WHERE patients.gender = 2;"
    cur.execute(query3)
    result3 = cur.fetchall()
    print("number of females: ", result3[0][0])

    query2 = "SELECT COUNT(*) FROM (SELECT DISTINCT pat_id FROM Chronic4)"
    cur.execute(query2)
    result2 = cur.fetchall()
    print("number of patients: ", result2[0][0])

    perc_female = int(result3[0][0]) / int(result2[0][0])
    print("percentage female: ", round(perc_female,4) * 100)

    query = "SELECT COUNT(*) FROM (SELECT DISTINCT * FROM Chronic4)"
    cur.execute(query)
    result = cur.fetchall()
    print("number of prescriptions: ", result[0][0])

con =
sqlite3.connect("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/SQL_Database
/SQL_DB.db")
cur = con.cursor()
#Count_rows()
#Count_Female()
post_filter()
```

Appendix B.2: Data exploration script2

```
#!/usr/bin/env python
# coding: utf-8
"""
Shadee Albronda
updated: 30/05/2022
Data exploration script2: Generate bar plot of prescription count over time
Diabetes CPRD GOLD dataset
Install:
  - pandas
"""
import pandas as pd

fileR="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Data_Expl_Dates.tx
t"
df = pd.read_csv(fileR, usecols=[0], names=["date"])
print(df.head)
df["date"] = df["date"].astype("datetime64")
ax = df.groupby(df["date"].dt.year).count().plot(kind="bar", figsize=(20,15))
```

```
ax.set_xlabel("Year")
ax.set_ylabel("Total number of prescriptions")
```

Appendix B.3: Data preparation SQL script

```
"""
Shadee Albronda
updated: 30/05/2022
Data preparation SQL script
Diabetes CPRD GOLD dataset

Install:
  - sqlite3
"""
import sqlite3
import csv

def index_db():
    "Index the therapy table"
    query2 = "CREATE INDEX therapy_index ON therapy(patid, eventdate,
prodcode);"
    cur.execute(query2)

def Join_tables():
    """
    Join the therapy table with the product table
    - Only keep prescriptions within a repeat schedule (therapy.issueseq >
0)
    - Only prescriptions upward from the year 2000
(substr(therapy.eventdate, 7, 4) >= 2000)
    Input tables: therapy, product
    Output table: Merged_therapy_product4
    """
    print("Join the therapy table with the product table")
    query1 = """CREATE TABLE Merged_therapy_product4 AS
    SELECT therapy.patid, therapy.eventdate, therapy.prodcode,
therapy.bnfcode, therapy.numdays,
     therapy.issueseq, product.productname, product.drugsubstance,
product.bnfchapter, product.bnfcode
    FROM therapy LEFT JOIN product ON therapy.prodcode = product.prodcode
    WHERE therapy.issueseq > 0 AND substr(therapy.eventdate, 7, 4) >=
2000;"""
    cur.execute(query1)

def update_date():
    # Update the date
    print("update date format from dd/mm/yyyy to yyyy-mm-dd")
    q1 = "UPDATE Merged_therapy_product4 SET eventdate = substr(eventdate,
7, 4) || '-' || substr(eventdate, 4,2) || '-' || substr(eventdate, 1,2)"
    cur.execute(q1)

def Chronic_only():
    """
    Create table keeping only the chronic medications used > 90 days
    Input table: Merged_therapy_product4
    Output table: Chronic4
    """
    queryD = "DROP TABLE IF EXISTS Chronic4"
    cur.execute(queryD)
    print("Only keep the chronic medicines, used > 90 days")
```

```
    query2 = "CREATE TABLE Chronic4 AS SELECT * FROM (" \
             "SELECT patid, `bnfcode:1`, MIN(eventdate) AS date_first, " \
             "MAX(eventdate) AS date_last, julianday(MAX(eventdate)) -
julianday(MIN(eventdate)) AS num_days " \
             "FROM Merged_therapy_product4 " \
             "WHERE eventdate IS NOT NULL AND `bnfcode:1` IS NOT NULL " \
             "GROUP BY patid, `bnfcode:1`) WHERE num_days > 90"
    cur.execute(query2)

def write_file():
    # Write Chronic table to file
    print("writing table to file")
    query2 = "SELECT * FROM Chronic4"
    cur.execute(query2)
    result = cur.fetchall()
    f =
open("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Chronic4.cs
v", 'w')
    writer = csv.writer(f)
    writer.writerows(result)
    f.close()

def test(table_name):
    col = "*"
    query2 = "SELECT " + col + " FROM " + table_name + " LIMIT 3"
    cur.execute(query2)
    result = cur.fetchall()
    print(result)
    query3 = "PRAGMA table_info(" + table_name + ")"
    cur.execute(query3)
    result2 = cur.fetchall()
    print(result2)

# Create a SQL connection to our SQLite database
con =
sqlite3.connect("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/SQL_Database
/SQL_DB.db")
cur = con.cursor()

#index_db()
#Join_tables()
#test("Merged_therapy_product4")
#update_date()
#test("Merged_therapy_product4")
#Chronic_only()
#test("Chronic4")
#write_file()

# Close the connection
con.close()
```

Appendix B.3: Concat script

```
#!/usr/bin/env bash
# Shadee Albronda
# Updated: 13/06/22
# Goal: Concat information to one line for each patient
# Diabetes CPRD GOLD dataset

fileW="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Chronic_On
eLine_Patient4.csv"
```

```
fileR="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Chronic4.csv"

cat ${fileR} | wc -l > num_lines.txt
cat ${fileR} | sort | uniq | wc -l >> num_lines.txt
cat ${fileR} | awk -F "," '{print $1}' | sort | uniq > ${fileW}
cat ${fileW} | wc -l >> num_lines.txt
```

Appendix B.4: Medication history vector creation script

```
#!/usr/bin/env python
# coding: utf-8
"""
Shadee Albronda
updated: 23/05/2022
Diabetes CPRD GOLD dataset

Goal: Make medication history vectors for each patient and write to file.
        (Only keep trajectories with atleast 5 prescriptions)
Output: Med_His_Vecs_Whole.txt
Output format: patient id, gender, list of prescriptions, list of
prescription dates
"""
import pandas as pd

def Create_lookup_table():
    # Make pandas dataframe to serve as look-up table
    col_names = ["patid", "bnf_code", "date_first"]
    df =
pd.read_csv("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Chronic4.csv",
                    usecols = [0,1,2], names=col_names)
    #print(df.head(10))
    return df

def Create_Vectors_Whole(df):
    # Open file to write medication history vectors to
    f =
open("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Vecs_Whole2_V6.1.txt", "w")
    f2 =
open("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Vecs_Whole2_V6.2.txt", "w")

    patient = 1
    with
open("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Chronic_OneLine_Patient4.csv", 'r') as p_file:
        for line in p_file:
            line = line.strip()
            if len(line) > 0:
                print("Patient " + str(patient))
                line = line.split(",")
                id1 = line[0]
                df2 = df.loc[df['patid'] == int(id1)]
                df3 = df2.dropna(axis=0, subset=['patid', 'bnf_code',
'date_first'])
                df3 = df3.sort_values(by="date_first") #Sort by date
                date_vec = df3['date_first'].tolist()
                rslt_df = df3.loc[df3['bnf_code'] != '00000000']
                bnf_c_vec = rslt_df['bnf_code'].tolist()
```

```
                string = id1 + "|" + str(bnf_c_vec) + "|" + str(date_vec)
                string = str(string.encode('utf-8'))
                if len(bnf_c_vec) > 4:
                    f.write(string + "\n")
                patient += 1

                df_gr = df3
                df_gr['bnf_chapt'] = df3['bnf_code'].str.slice(0,2)
                df_gr = df_gr.loc[df_gr['bnf_chapt'] != '00']
                df_gr2 = df_gr.groupby(['patid', 'bnf_chapt'],
as_index=False)['date_first'].min()
                df_gr3 = df_gr2.sort_values(by="date_first")  # Sort by
date
                bnf_c_vec2 = df_gr3['bnf_chapt'].tolist()
                date_vec2 = df_gr3['date_first'].tolist()
                string2 = id1 + "|" + str(bnf_c_vec2) + "|" +
str(date_vec2)
                string2 = str(string2.encode('utf-8'))
                if len(bnf_c_vec2) > 4:
                    f2.write(string2 + "\n")
    f.close()

df = Create_lookup_table()
Create_Vectors_Whole(df)
```

Appendix B.5: Medication history vector filter script

```
#!/usr/bin/env bash
#Shadee Albronda
#updated: 05/06/2022
#Diabetes CPRD GOLD dataset
#Goal: Filter medication history vectors to only keep the prescriptions
from BNF chapter 1 - 10

fileR="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Vec
s_Whole2_V6.2.txt"
fileW="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Vec
s_Whole2_V6.3.txt"

while read -r line;
do
   #echo "$line";
   id=$(echo ${line} | awk -F "|" '{print $1}')
   vec1=$(echo ${line} | awk -F "|" '{print $2}' | tr -d "[" | tr -d "]" |
tr -d "," | tr -d "'")
   vec2=$(echo ${line} | awk -F "|" '{print $2}' | tr -d "[" | tr -d "]" |
tr -d "," | tr -d "'" | egrep -o '01|02|03|04|05|06|07|08|09|10')
   dates=$(echo ${line} | awk -F "|" '{print $3}' | tr -d "[" | tr -d "]" |
tr -d "," | tr -d "'")
   echo ${id},${vec1},${vec2},${dates} >> ${fileW}
done < ${fileR}
```

Appendix B.6: Get all unique trajectories script

```
#!/usr/bin/env bash
#Shadee Albronda
#updated: 05/06/2022
#Diabetes CPRD GOLD dataset
#Goal: Get all unique trajectories and write to file
```

```
fileR="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Vecs_Whole2_V6.3.txt"
fileW="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Uniq_traj6.txt"

cat ${fileR} | awk -F "," '{print $3}' | sort | uniq > ${fileW}
cat ${fileW} | wc -l > Num_Uniq_Traj.txt
cat ${fileR} | wc -l >> Num_Uniq_Traj.txt
```

Appendix B.7: Clustering script

```
"""
Shadee Albronda
01/06/2022
Install:
  numpy
  dtaidistance
  sklearn
  sklearn_extra

Goal:  Cluster all trajectory vectors into groups/clusters of similar
trajectories.

Trajectory properties:
    - trajectory = a medication history vector for each patient in
chronological order
    - contains a sequence of prescriptions represented by their BNF-chapter
    - for each unique chapter, only the first occasion is included in the
vector
    - vectors contain atleast 5 different prescriptions
"""
import numpy as np
from dtaidistance import dtw
from sklearn.decomposition import PCA
from sklearn.preprocessing import MinMaxScaler
import time
from sklearn_extra.cluster import KMedoids

start = time.time()
def Get_DM():
    print("Get the distance matrix using DTW")
    file =
"O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Uniq_traj6.txt"
    num_lines = 44495
    num_cols = 48
    matrix = np.zeros((num_lines,num_cols), dtype=float)
    traj_list = []
    num = -1
    with open(file) as f:
        for line in f:
            num +=1
            line2 = line.strip().split(" ")
            line3 = list(map(int, line2))
            traj_list.append(line)
            for field in range(len(line3)):
                #print(field)
                matrix[num,field:field+1] =  line3[field]
    f.close()
```

```
    # PCA to reduce data
    scaler = MinMaxScaler()
    data_rescaled = scaler.fit_transform(matrix)
    pca = PCA(n_components = 0.95) #95% of variance
    pca.fit(data_rescaled)
    reduced = pca.transform(data_rescaled)

    dm = dtw.distance_matrix_fast(reduced)
    return dm, traj_list, reduced

dm, traj_list, r_matrix = Get_DM()
end = time.time()
print("The time of execution is :", end-start)

start = time.time()
def Cluster_KMedoids_ElbowMethod(dm):
    distortions = []
    K = range(1,25)
    for k in K:
        print(k)
        clustering = KMedoids(n_clusters=k, random_state=0,
metric='precomputed')
        res4 = clustering.fit(dm)
        distortions.append(clustering.inertia_)
    return distortions, K
distortions, K = Cluster_KMedoids_ElbowMethod(dm)

print(distortions)
end = time.time()
print("The time of execution is :", end-start)

import matplotlib.pyplot as plt

def Plot_Distortion(distortions, K):
    plt.figure(figsize=(16,8))
    plt.plot(K, distortions, 'bx-')
    plt.xlabel('k')
    plt.ylabel('Distortion')
    plt.title('The Elbow Method showing the optimal k')
    plt.show()

Plot_Distortion(distortions, K)

start = time.time()
def Cluster_KMedoids(dm):
    clustering = KMedoids(n_clusters=11, random_state=0,
metric='precomputed')
    res4 = clustering.fit(dm)
    labels4 = res4.labels_
    return labels4
labels4 = Cluster_KMedoids(dm)

print(labels4)
end = time.time()
print("The time of execution is :", end-start)

print(len(labels4))
print(len(set(labels4)))
print(labels4)

def Write_File(labels3, traj_list):
```

```
    f =
open("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/traj_labels
.txt", "w")
    for i in range(len(labels3)):
        label = labels3[i]
        vec = traj_list[i]
        string = str(label) + "," + str(vec)
        #print(string)
        f.write(string + "\n")
    f.close()

Write_File(labels4, traj_list)
```

Appendix B.8: Create medication history vectors in parts of 1 year time

```
#%%
#!/usr/bin/env python
# coding: utf-8
"""
Shadee Albronda
updated: 28/05/2022
Diabetes CPRD GOLD dataset

Goal: Make medication history vectors for each patient over time and write
to file.
    resulting file: Med_His_Vecs.txt
    file format: patient id, [[vector1], [vector2], [vector..]]
    file description: one line for each patient followed by lists of
medicines used in a 1 year timerange.
"""
from datetime import datetime
import pandas as pd
import math

#%%
def Create_lookup_table():
    # Make pandas dataframe to serve as look-up table
    col_names = ["patid", "bnf_code", "date_first"]
    df =
pd.read_csv("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Chro
nic4.csv",
                     usecols = [0,1,2], names=col_names)
    #print(df.head(10))
    return df

df = Create_lookup_table()

#%%
def Create_lookup_table2():
    # Make pandas dataframe to serve as look-up table for vector cluster
labels
    col_names = ["label", "trajectory"]
    df =
pd.read_csv("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/traj
_labels.txt",
                     usecols = [0,1], names=col_names)
    print(df.head(10))
    return df

df_label = Create_lookup_table2()
```

```
#%%
def Create_Vectors(df, df_label):
    # Open file to write medication history vectors to
    f =
open("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Vecs
_Parts_1y.txt", "w")
    patient = 1
    with
open("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Vecs
_Whole2_V6.3.txt", 'r') as p_file:
        for line in p_file:
            line = line.strip()
            if len(line) > 0:
                print("Patient " + str(patient))
                line = line.split(",")
                id1 = line[0][2:]
                vec = line[2]
                df2 = df.loc[df['patid'] == int(id1)]
                df3 = df2.dropna(axis=0, subset=['patid', 'bnf_code',
'date_first']) # rows with unknown drug substance are excluded
                df_gr = df3
                df_gr['bnf_chapt'] = df3['bnf_code'].str.slice(0,2)
                df_gr = df_gr.loc[(df_gr['bnf_chapt'] !=
'00')&(df_gr['bnf_chapt'] != 'Su')&(df_gr['bnf_chapt'] != 'Or')
                                  &(df_gr['bnf_chapt'] !=
'In')&(df_gr['bnf_chapt'] != 'Pe')&(df_gr['bnf_chapt'] != 'Le')&
                                 (df_gr['bnf_chapt'] !=
'Cu')&(df_gr['bnf_chapt'] != 'Ni')]

                df_gr = df_gr[df_gr['bnf_chapt'].str.match('[0-9][0-9]')==
True]

                df_gr['bnf_chapt2'] = df_gr['bnf_chapt'].astype('int')
                df_gr_f = df_gr.loc[df_gr['bnf_chapt2'] < 11]

                df_gr2 = df_gr_f.groupby(['patid', 'bnf_chapt2'],
as_index=False)['date_first'].min()
                df_gr3 = df_gr2.sort_values(by="date_first")  # Sort by
date
                first_date = df_gr3["date_first"].min()
                last_date = df_gr3["date_first"].max()
                last_date2 = datetime.strptime(last_date,'%Y-%m-%d')
                first_date2 = datetime.strptime(first_date, '%Y-%m-%d')
                delta = last_date2 - first_date2
                tot_time = math.ceil(delta.days / 365) # 1 year
                time_points = pd.date_range(start=first_date2,
end=last_date2, periods=tot_time+1)

                df1 = df_label.loc[df_label["trajectory"] == vec]
                label1 = df1["label"].tolist()
                vec2 = df1["trajectory"]

                all_vecs = []
                for t in range(tot_time):
                    max_date = time_points[t+1]
                    df4 = df_gr3.loc[df_gr3['date_first'] <= str(max_date)]
                    vec = df4['bnf_chapt2'].isnull() == False
                    if vec.empty == False:
                        vec = df4['bnf_chapt2'].tolist()
                        all_vecs.append(vec)
                    elif vec.empty == True:
```

```
                        all_vecs.append('')
                string = id1 + "," + str(label1) + "," + str(all_vecs) + "
\n"
                string = str(string.encode('utf-8'))
                #print(string)
                f.write(string)
                patient +=1
    f.close()

Create_Vectors(df, df_label)
```

Appendix B.9: Reformat vector files script

```
#!/usr/bin/env bash
#Shadee Albronda
#updated: 05/06/2022
#Diabetes CPRD GOLD dataset
#Goal: Change file lay-out to prepare data for visualization

fileR="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Vec
s_Parts_1y.txt"
fileW="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedVecs_Pa
rts_1y.txt"
fileR2="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedHis_Ve
cs_Parts.txt"
fileW2="O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedVecs_P
arts_3y.txt"

nl=$'\n'
echo "File1"
cat ${fileR} | sed 's/, / /g' | sed "s/'b'/ /g" | sed 's/ \\n
/'"\\${nl}"'/g' | sed "s/b'//g" \
| sed "s/ \\\n'//g" | tr "," "|" | tr " " "," > ${fileW}
echo "File2"
cat ${fileR2} | sed 's/, / /g' | sed "s/'b'/ /g" | sed 's/ \\n
/'"\\${nl}"'/g' | sed "s/b'//g" \
| sed "s/ \\\n'//g" | tr "," "|" | tr " " "," > ${fileW2}
```

Appendix B.10: Make heatmap data frame script

```
#%%
#!/usr/bin/env python
# coding: utf-8
"""
Shadee Albronda
updated: 09/06/2022
Diabetes CPRD GOLD dataset
Goal: Visualization of clustering results for incidence medication use only
"""
import pandas as pd
import numpy as np
import ast
#%%
def read_files():
    col_names = ["patid", "label", "vector"]
    df_1y =
pd.read_csv("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedV
ecs_Parts_1y.txt",
```

```
                    header=None, names=col_names, index_col=0, sep="|")
    print(df_1y.head(5))
    return df_1y
#read_files()

#%%
from collections import Counter
def counts(df_all, label):
    df_all = df_all.fillna('None')
    n_colls = df_all.shape[1]
    dfcc3 = pd.DataFrame()
    dfcc4 = pd.DataFrame()
    for i in range(n_colls):
        string_c = ' '.join(df_all[i])
        list_c = string_c.split(" ")
        counts = Counter(list_c)
        dfcc = pd.DataFrame.from_dict(counts,
orient='index').reset_index().transpose()
        stringW2 = "Time_point"
        stringW = str(i)
        for chap, count in counts.items():
            if chap != "None":
                stringW = stringW + "," + str(count)
                stringW2 = stringW2 + "," + str(chap)
                dfcct = pd.DataFrame({str(chap) : count}, index=[i])
                result = i in dfcc3.index
                if result == False:
                    dfcc3 = pd.concat([dfcc3, dfcct], axis=0, join='inner')
                elif result == True:
                    dfcc3 = pd.merge(dfcc3, dfcct, left_index=True,
right_index=True, how='inner')
        dfcc4 = pd.concat([dfcc4, dfcc3], axis=0, join='outer')

    print(dfcc4)
    loc =
"O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Heatmap_Datafram
e_1y_Incident" + str(label) + ".txt"
    dfcc4.to_csv(loc)
    return dfcc4

#%%
def per_cluster(df, lab):
    df_1 = df[df['label'] == lab]
    num_lines = df_1.shape[0]
    num_cols = 85
    df_1['vector2'] = df_1['vector'].apply(ast.literal_eval)
    matrix = np.zeros((num_lines,num_cols), dtype=float)
    df_all = pd.DataFrame()

    for index, row in df_1.iterrows():
        line1 = row['vector']
        line1 = line1.replace("[[", " ")
        line1 = line1.replace("]]", " ")
        line1 = line1.replace(",", " ")
        line2 = line1.strip().split("] [")
        len_incident = len(line2[0])
        lijst2 = line2[1::]
        lijst3 = [i[len_incident:] for i in lijst2]
        lijst4 = list(filter(None, lijst3))
        cols = len(lijst4)
        col_list = [*range(0, cols, 1)]
```

```
        ids = index
        df2 = pd.DataFrame([lijst4], columns=col_list, index=[ids])
        df_all = df_all.append(df2)
    #print(df_all)
    return df_all

#%%
df_1y = read_files()
label_list = df_1y.label.unique()

for lab in label_list:
    print("Label: " + lab)
    df_all = per_cluster(df_1y, lab)
    dfcc4 = counts(df_all, lab)

#print(df_all)
```

Appendix B.11: Make heatmaps script

```
#%%
"""
Shadee Albronda
updated: 09/06/2022
Diabetes CPRD GOLD dataset
Goal: Visualization of clustering results for incidence medication use only
"""
import pandas as pd
import numpy as np
import seaborn as sns
from scipy.stats import zscore
import matplotlib.pyplot as plt

#%%
def read_files(loc):
    df = pd.read_csv(loc, header=0, index_col=0, sep=",")
    df = df[['10', '9', '8', '7', '6', '5', '4', '3', '2', '1']].fillna(0)
    # Only keep the first 25 years of follow-up
    df = df.head(25)
    #print(df.head(5))
    return df
#%%
def Heatmap_raw(df, title):
    df_transposed = df.T
    p = sns.heatmap(df_transposed, annot=False)
    p.set(ylabel = "BNF chapter", xlabel = "Years follow-up", title =
title)
    p.set_yticklabels(['1: Gastro-Intestinal System','2: Cardiovascular
System','3: Respiratory System'
                       ,'4: Central Nervous System','5: Infections','6:
Endocrine System'
                       , '7: Obstetrics/Gynaecology/Urinary-Tract'
                       , '8: Malignant Disease/Immunosuppression', '9:
Nutrition and Blood',
                      '10: Musculoskeletal and Joint Diseases'],
rotation=0)
    p.set_xticklabels(['1','2','3','4','5','6', '7', '8', '9', '10',
                      '11', '12', '13', '14', '15', '16', '17', '18', '19'
                       , '20', '21', '22', '23', '24', '25'], rotation=90)
    return p
#%%
from scipy.stats import zscore
```

```
def Heatmap_Zscore(df, title):
    df_transposed = df.T
    df_Z = df_transposed.apply(zscore, axis=1)
    print(df_Z)
    pz = sns.heatmap(df_Z, annot=False, cmap="vlag")
    pz.set(ylabel = "BNF chapter", xlabel = "Years follow-up", title =
title)
    pz.set_yticklabels(['1: Gastro-Intestinal System','2: Cardiovascular
System','3: Respiratory System'
                        ,'4: Central Nervous System','5: Infections','6:
Endocrine System'
                        , '7: Obstetrics/Gynaecology/Urinary-Tract'
                        , '8: Malignant Disease/Immunosuppression', '9:
Nutrition and Blood',
                       '10: Musculoskeletal and Joint Diseases'],
rotation=0)
    pz.set_xticklabels(['1','2','3','4','5','6', '7', '8', '9', '10',
                      '11', '12', '13', '14', '15', '16', '17', '18', '19'
                       , '20', '21', '22', '23', '24', '25'], rotation=90)
    return pz

#%%
loc =
"O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Heatmap_Datafram
e_1y_Incident["
title = "Cluster "

sns.set(rc={'axes.facecolor':'white', 'figure.facecolor':'white'})
for i in range(11):
    plt.figure(figsize=(26,8))
    loci = loc + str(i) + "].txt"
    titi = title + str(i)
    df = read_files(loci)
    titi = titi
    p = Heatmap_raw(df, titi)
    plt.show()
    fig = p.get_figure()

fig.savefig('C:/Users/shade/OneDrive/Bureaublad/ADS_Thesis/Results/Incident
_Heatmap_raw_C' + str(i) + ".png")
    plt.figure(figsize=(26,8))
    pz = Heatmap_Zscore(df, titi)
    plt.show()
    fig2 = pz.get_figure()

fig2.savefig('C:/Users/shade/OneDrive/Bureaublad/ADS_Thesis/Results/Inciden
t_Heatmap_Z_C' + str(i) + ".png")
```

Appendix B.12: Merge heatmaps script

```
#%%
import pandas as pd
import numpy as np
import seaborn as sns
from scipy.stats import zscore
import matplotlib.pyplot as plt
#%%
def read_files(loc):
    df = pd.read_csv(loc, header=0, index_col=0, sep=",")
    df = df[['10', '9', '8', '7', '6', '5', '4', '3', '2', '1']].fillna(0)
    # Only keep the first 25 years of follow-up
```

```
    df = df.head(25)
    #print(df.head(5))
    return df
#%%
def Heatmap_raw(df, title, axs, i, i2):
    df_transposed = df.T
    p = sns.heatmap(df_transposed, annot=False, ax=axs[i2,i])
    axs[i2,i].set_title(title)
    if i != 0:
        axs[i2,i].axes.yaxis.set_visible(False)
    if i2 != 3:
        axs[i2,i].axes.xaxis.set_visible(False)
    if i == 2 and i2 == 2:
        axs[i2,i].axes.xaxis.set_visible(True)
    return p
#%%
from scipy.stats import zscore
def Heatmap_Zscore(df, title, axs, i, i2):
    df_transposed = df.T
    df_Z = df_transposed.apply(zscore, axis=1)
    #print(df_Z)

    pz = sns.heatmap(df_Z, annot=False, cmap="vlag", ax=axs[i2,i])
    axs[i2,i].set_title(title)
    if i != 0:
        axs[i2,i].axes.yaxis.set_visible(False)
    if i2 != 3:
        axs[i2,i].axes.xaxis.set_visible(False)
    if i == 2 and i2 == 2:
        axs[i2,i].axes.xaxis.set_visible(True)
    return pz
#%%
#loc =
"O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Heatmap_Datafram
e_1y_Incident["
loc =
"O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/Heatmap_Datafram
e_1y["
title = "Cluster "
#fig_title = 'BNF chapter prescriptions over time (incident)'
fig_title = 'BNF chapter prescriptions over time (prevalent)'
sns.set(rc={'axes.facecolor':'white', 'figure.facecolor':'white'})

fig2, axs2 = plt.subplots(4, ncols=3, figsize=(18,18))
fig, axs = plt.subplots(4, ncols=3, figsize=(18,18))
i2 = 0
i3 = 0
for i in range(11):
    if i3 > 2:
        i2 +=1
    if i3 > 2:
        i3 = 0
    loci = loc + str(i) + "].txt"
    titi = title + str(i)
    df = read_files(loci)
    titi = titi
    print(i3, i2)
    p = Heatmap_raw(df, titi, axs2, i3, i2)
    pz = Heatmap_Zscore(df, titi, axs, i3, i2)
    i3 +=1
```

```
axs2[3,2].axes.xaxis.set_visible(False)
axs2[3,2].axes.yaxis.set_visible(False)
p.set_xlabel("Years follow-up", fontsize=14)
axs2[1,0].set_ylabel("BNF chapter", fontsize=14)
axs2[1,0].axes.yaxis.set_visible(True)
axs[3,2].axes.xaxis.set_visible(False)
axs[3,2].axes.yaxis.set_visible(False)
pz.set_xlabel("Years follow-up", fontsize=14)
axs[1,0].set_ylabel("BNF chapter", fontsize=14)
axs[1,0].axes.yaxis.set_visible(True)

string1 = "BNF chapters: \n\n\
1  = Gastro-Intestinal System \n\
2  = Cardiovascular System \n\
3  = Respiratory System \n\
4  = Central Nervous System \n\
5  = Infections \n\
6  = Endocrine System \n\
7  = Obstetrics/Gynaecology/Urinary-Tract \n\
8  = Malignant Disease/Immunosuppression \n\
9  = Nutrition and Blood \n\
10 = Musculoskeletal and Joint Diseases \n"

fig.suptitle(fig_title, fontsize=20)
fig2.suptitle(fig_title, fontsize=20)
plt.figtext(0.72, 0.06, string1, fontsize=14)
plt.tight_layout()
```

Appendix B.13: Get statistics script

```
"""
Shadee Albronda
updated: 09/06/2022
Diabetes CPRD GOLD dataset
Goal: Get within cluster statistics
"""
#%%
import pandas as pd
import numpy as np
import seaborn as sns

#%%
def read_files(loc):
    df = pd.read_csv(loc, header=None, index_col=0, sep="|")
    df = df[df[2].apply(lambda x: len(x) > 2)] # remove patients with empty
trajectories
    num_patients = len(df)
    df2 = pd.DataFrame()
    df2['label'] = df[1]
    df2['vector'] = df[2].str.split('\],\[').str[-1]
    df2['vector'] = df2['vector'].replace({'\]':''}, regex=True)
    df2['vector'] = df2['vector'].replace({'\[':''}, regex=True)
    df2['vector'] = df2['vector'].replace({',':' '}, regex=True)
    df2['label'] = df2['label'].replace({'\]':''}, regex=True)
    df2['label'] = df2['label'].replace({'\[':''}, regex=True)
    df2 = df2.astype({'label': 'int32'})
    df2 = df2.astype({'vector' : 'category'})
    print(df2)
    return df2
loc =
"O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedVecs_Parts_1y
```

```
.txt"
df2 = read_files(loc)

#%%
def make_plot1(df2):
    df3 = pd.DataFrame()
    df3['label'] = df2['label']
    print(df3)
    ax = df3.groupby(df3['label']).size().plot(kind="bar", figsize=(20,15))
    ax.set_xlabel("")
    ax.set_ylabel("")
make_plot1(df2)
#%%
import matplotlib.pyplot as plt
from collections import Counter
from collections import OrderedDict

df3 = df2.groupby(df2['label'], as_index = False).agg({'vector': ' '.join})
df4 = pd.DataFrame()
df4['vector'] = df3['vector'].str.split(' ', expand=False)
#print(df4)

plt.figure(figsize=(9,9))
for i, row in df4.iterrows():
    botm = 0
    list1 = row['vector']
    c = Counter(list1)
    vals = c.values()
    for key, val in c.items():
        plt.bar(str(i), val, bottom=botm, label=key)
        plt.xlabel("Cluster number")
        plt.ylabel("Prescription count")
        botm = botm + val

def legend_without_duplicate_labels(figure):
    handles, labels = plt.gca().get_legend_handles_labels()
    by_label = dict(zip(labels, handles))
    by_label2 = OrderedDict(sorted(by_label.items(), key=lambda t:
int(t[0])))
    figure.legend(by_label2.values(), by_label2.keys(), loc='upper right')
    plt.title("The total number of prevalent prescriptions, colored by BNF-
chapter, for each cluster")
    plt.show()
legend_without_duplicate_labels(plt)
#%%
def read_files_incident(loc):
    df = pd.read_csv(loc, header=None, index_col=0, sep="|")

    df = df[df[2].apply(lambda x: len(x) > 2)] # remove patients with empty
trajectories
    num_patients = len(df)
    print(df.head(5))
    df2 = pd.DataFrame()
    df2['label'] = df[1]
    df2['prevalent'] = df[2].str.split('\],\[').str[0]
    df2['prevalent'] = df2['prevalent'].replace({'\]':''}, regex=True)
    df2['prevalent'] = df2['prevalent'].replace({'\[':''}, regex=True)
    df2['prevalent'] = df2['prevalent'].replace({',':' '}, regex=True)
    df2['vector'] = df[2].str.split('\],\[').str[-1]
    df2['vector'] = df2['vector'].replace({'\]':''}, regex=True)
    df2['vector'] = df2['vector'].replace({'\[':''}, regex=True)
```

```
    df2['vector'] = df2['vector'].replace({',':' '}, regex=True)
    df2['label'] = df2['label'].replace({'\]':''}, regex=True)
    df2['label'] = df2['label'].replace({'\[':''}, regex=True)

    def calculation(val):
        val2 = val[1] + " "
        return val[0].replace(val2,'').strip()
    df2['incident'] = df2[['vector','prevalent']].apply(calculation,
axis=1)
    df2 = df2.astype({'label': 'int32'})
    df2 = df2.astype({'vector' : 'category'})
    df2 = df2.astype({'incident' : 'category'})
    df2 = df2.astype({'prevalent' : 'category'})
    print(df2)
    return df2

loc =
"O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedVecs_Parts_1y
.txt"
df2 = read_files_incident(loc)

#%%
import matplotlib.pyplot as plt
from collections import Counter
from collections import OrderedDict
sns.set(rc={'axes.facecolor':'white', 'figure.facecolor':'white'})

df3 = df2.groupby(df2['label'], as_index = False).agg({'incident': '
'.join})
df4 = pd.DataFrame()
df4['incident'] = df3['incident'].str.split(' ', expand=False)

plt.figure(figsize=(9,9))
for i, row in df4.iterrows():
    #print(i)
    botm = 0
    list1 = row['incident']
    c = Counter(list1)
    del c[""]
    c = OrderedDict(sorted(c.items(), key=lambda t: int(t[0])))
    vals = c.values()
    total = sum(c.values())
    colors = ['black', 'red', 'green', 'blue', 'cyan', 'orange', 'magenta',
'grey', 'yellow', 'olive', 'pink']
    for key, val in c.items():
        if key != "":
            perc = (val / total) * 100
            i_key = int(key)
            plt.bar(str(i), perc, bottom=botm, label=key,
color=colors[i_key])
            plt.xlabel("Cluster number")
            plt.ylabel("BNF chapter %")
            botm = botm + perc

def legend_without_duplicate_labels(figure):
    handles, labels = plt.gca().get_legend_handles_labels()
    by_label = dict(zip(labels, handles))
    by_label2 = OrderedDict(sorted(by_label.items(), key=lambda t:
int(t[0])))
    leg_n = ['1: Gastro-Intestinal System','2: Cardiovascular System','3:
Respiratory System'
```

```
                        ,'4: Central Nervous System','5: Infections','6:
Endocrine System'
                        , '7: Obstetrics/Gynaecology/Urinary-Tract'
                        , '8: Malignant Disease/Immunosuppression', '9:
Nutrition and Blood',
                       '10: Musculoskeletal and Joint Diseases']
    figure.legend(by_label2.values(), leg_n, loc='upper right',
bbox_to_anchor=(1.47, 1))
    plt.title("Distribution of BNF chapters within clusters (incident)")

legend_without_duplicate_labels(plt)

#%%
import matplotlib.pyplot as plt
from collections import Counter
from collections import OrderedDict
import seaborn as sns
sns.set(rc={'axes.facecolor':'white', 'figure.facecolor':'white'})

df3 = df2.groupby(df2['label'], as_index = False).agg({'vector': ' '.join})
df4 = pd.DataFrame()
df4['vector'] = df3['vector'].str.split(' ', expand=False)
#print(df4)

plt.figure(figsize=(9,9))
for i, row in df4.iterrows():
    #print(i)
    botm = 0
    list1 = row['vector']
    c = Counter(list1)
    del c[""]
    c = OrderedDict(sorted(c.items(), key=lambda t: int(t[0])))
    vals = c.values()
    total = sum(c.values())
    colors = ['black', 'red', 'green', 'blue', 'cyan', 'orange', 'magenta',
'grey', 'yellow', 'olive', 'pink']
    for key, val in c.items():
        if key != "":
            perc = (val / total) * 100
            i_key = int(key)
            plt.bar(str(i), perc, bottom=botm, label=key,
color=colors[i_key])
            plt.xlabel("Cluster number")
            plt.ylabel("BNF chapter %")
            botm = botm + perc

def legend_without_duplicate_labels(figure):
    handles, labels = plt.gca().get_legend_handles_labels()
    by_label = dict(zip(labels, handles))
    by_label2 = OrderedDict(sorted(by_label.items(), key=lambda t:
int(t[0])))
    leg_n = ['1: Gastro-Intestinal System','2: Cardiovascular System','3:
Respiratory System'
                        ,'4: Central Nervous System','5: Infections','6:
Endocrine System'
                        , '7: Obstetrics/Gynaecology/Urinary-Tract'
                        , '8: Malignant Disease/Immunosuppression', '9:
Nutrition and Blood',
                       '10: Musculoskeletal and Joint Diseases']
    figure.legend(by_label2.values(), leg_n, loc='upper right',
bbox_to_anchor=(1.47, 1))
```

```
    plt.title("Distribution of BNF chapters within clusters (prevalent)")
    plt.show()
legend_without_duplicate_labels(plt)
#%%
import sqlite3
def df_to_sql(df2):
    df2.index.names = ['patid']
    print(df2)
    con =
sqlite3.connect("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/SQL_Database
/SQL_DB.db")
    cur = con.cursor()
    cur.execute('CREATE TABLE IF NOT EXISTS clusters (patid number, label
number, prevalent text, vector text, incident text)')
    con.commit()
    df2.to_sql('clusters', con, if_exists='replace', index = True)
    con.close()
df_to_sql(df2)
#%%
def merge_df():
    con =
sqlite3.connect("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/SQL_Database
/SQL_DB.db")
    cur = con.cursor()
    print("join dataframe with patients table")
    query = "SELECT * FROM clusters"
    cur.execute(query)
    result = cur.fetchall()
    #print(result)
    query3 = "PRAGMA table_info(clusters)"
    cur.execute(query3)
    result2 = cur.fetchall()
    print(result2)
    query2 = "SELECT clusters.patid, label, gender FROM clusters LEFT JOIN
patients ON clusters.patid = patients.patid"
    cur.execute(query2)
    cols = [column[0] for column in cur.description]
    result3 = pd.DataFrame.from_records(data = cur.fetchall(), columns =
cols)
    query4 = "SELECT clusters.patid, label, gender, yob, mob, deathdate
FROM clusters LEFT JOIN patients ON clusters.patid = patients.patid"
    cur.execute(query4)
    cols = [column[0] for column in cur.description]
    result4 = pd.DataFrame.from_records(data = cur.fetchall(), columns =
cols)
    print(result4)
    con.close()
    return(result3, result4)
result3, result4 = merge_df()
#%%
def cluster_stats(result3):
    # Get number and percentage of female and male per cluster
    df = result3
    df_female = df[df['gender'] == 1]
    df_male = df[df['gender'] == 2]

    df1 = pd.DataFrame()
    df1['count'] = df.groupby(['label', 'gender']).size()
    #print(df1)
    df2 = pd.DataFrame()
    df2['size'] = result3.groupby(['label']).size()
```

```
    nums = pd.DataFrame()
    nums['num_males'] = df_male.groupby(['label']).size()
    nums['num_females'] = df_female.groupby(['label']).size()

    df3 = pd.merge(df1, df2, on=["label", "label"], how='left')
    df3['percentage'] = round(df3['count'] / df3['size'] * 100, 2)
    print(df3)
    print(df3['count'].sum())
    print("# female: ", nums['num_females'].sum())
    print("# male: ", nums['num_males'].sum())
    print("% female: ", nums['num_females'].sum() / df3['count'].sum() *
100)
    print("% male: ", nums['num_males'].sum() / df3['count'].sum() * 100)
    return df
gender_df = cluster_stats(result3)
#%%
def cluster_stats2(result4):
    # Get min/max/sum/average/median age per cluster
    df = result4
    df_lives = df[df['deathdate'].isnull()]
    df_died = df[~df['deathdate'].isnull()]
    df_lives['age'] = 2022 - df_lives['yob']
    df_died['age'] =  df_died['deathdate'].str[6:].astype(int) -
df_died['yob']
    result = pd.concat([df_lives, df_died])
    print(result)
    result2 = result.groupby(['label']).agg({'label' : ['size'], 'age' :
['min', 'max', 'sum', 'mean', 'median']})
    print(result2)
    print(result['age'].mean())
    print(result['age'].min())
    print(result['age'].max())
cluster_stats2(result4)
#%%
def cluster_stats3():
    col_names = ["patid", "label", "vector"]
    df_1y =
pd.read_csv("O:/BETA/Instituut/UIPS/PECP/Students/Diabetes/Shadee_temp/MedV
ecs_Parts_1y.txt",
                        header=None, names=col_names, index_col=0, sep="|")
    df = df_1y
    df['lijst'] = df_1y['vector'].str.split('\],\[', expand=False)
    df['Length'] = df['lijst'].str.len()
    df = df[df['Length'] > 1]
    print(df)
    df_g = df.groupby(['label']).agg({'label' : ['count'], 'Length' :
['min', 'max', 'sum', 'mean', 'median']})
    print(df_g)
    df_s = pd.DataFrame()
    df_s['sum'] = df.groupby(['label']).agg({'Length' : ['sum']})
    print(df_s['sum'].sum())

    tot_follow_up = df['Length'].sum()
    print('total follow-up years: ', tot_follow_up)
    print('mean follow-up years: ', df['Length'].mean())
    print('median follow-up years: ', df['Length'].median())
    print('min follow-up years: ', df['Length'].min())
    print('max follow-up years: ', df['Length'].max())
cluster_stats3()
```

Appendix C.1: Bar plots stratified by gender or age

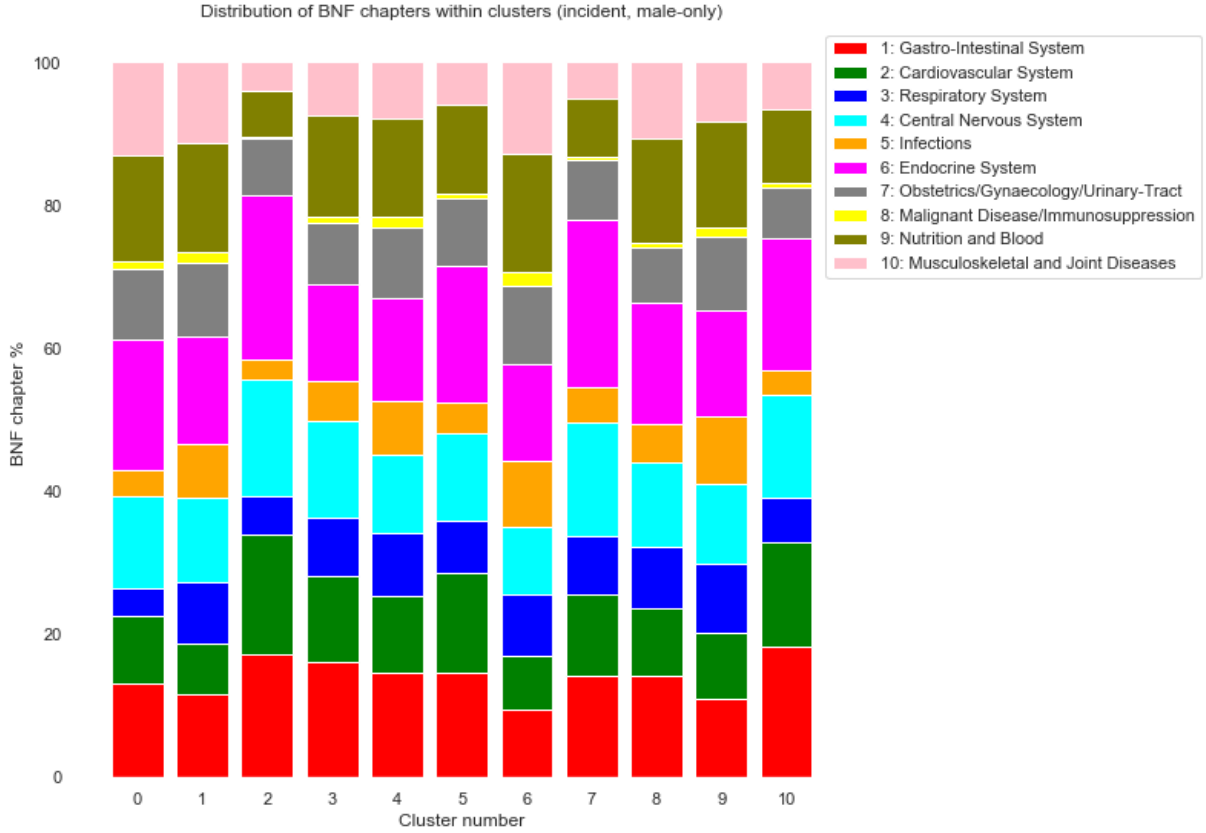

Figure 11. *The distribution of BNF chapters 1-10 within clusters for the incident, male only cohort*

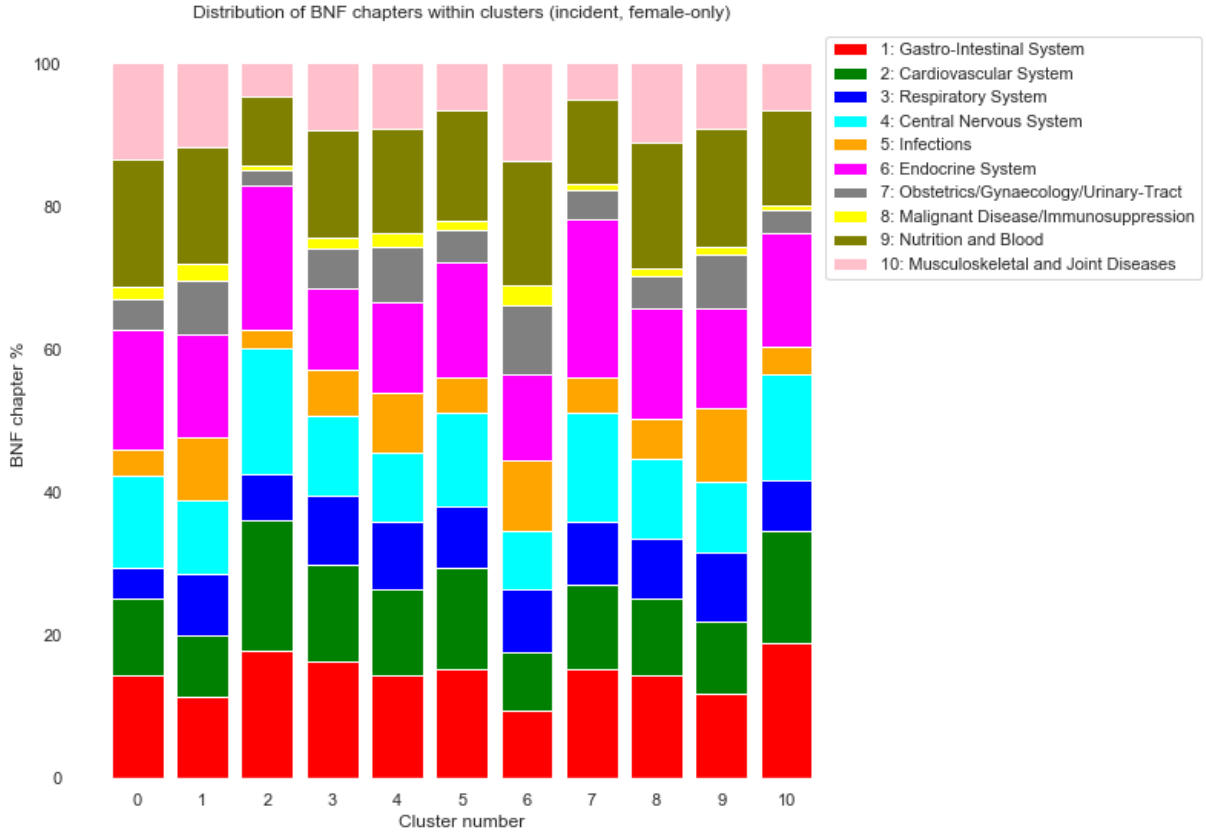

Figure 12. *The distribution of BNF chapters 1-10 within clusters for the incident, female only cohort*

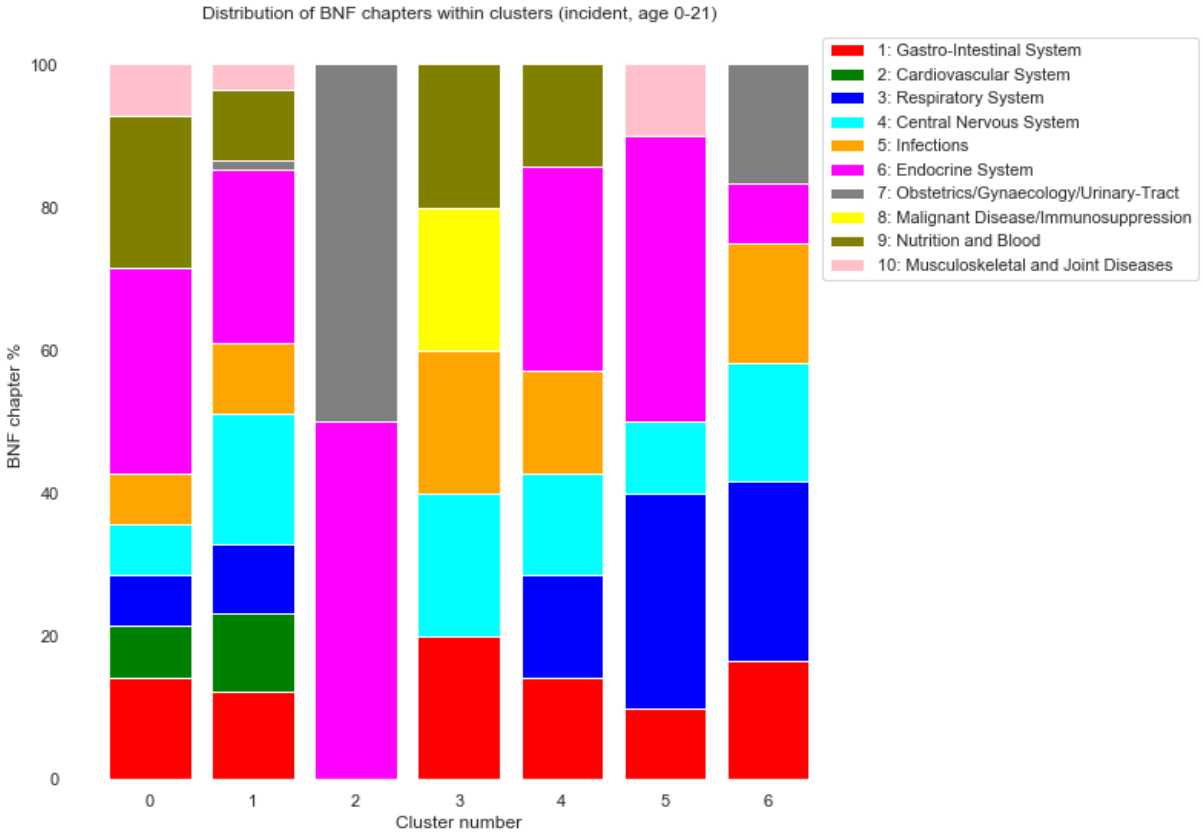

Figure 13. *The distribution of BNF chapters 1-10 within clusters for the incident, young only cohort*

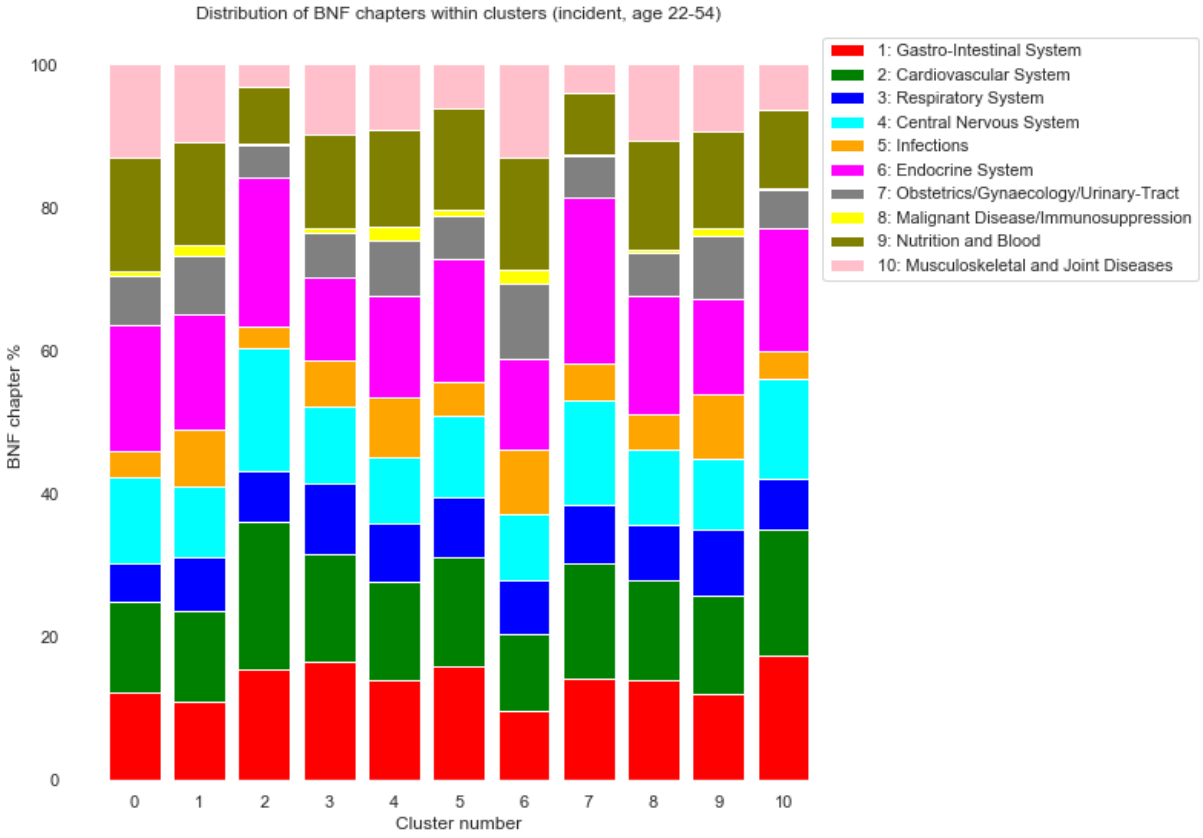

Figure 14. *The distribution of BNF chapters 1-10 within clusters for the incident, medium age only cohort*

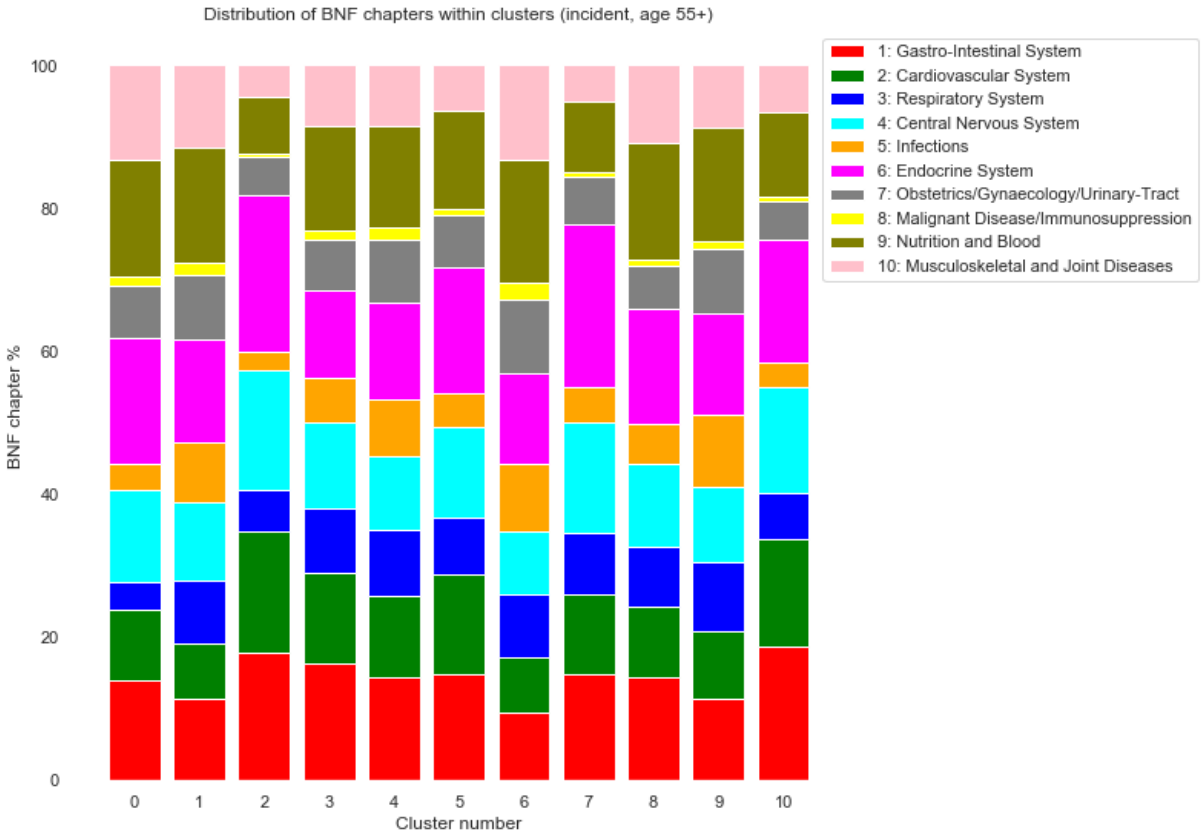

Figure 15. *The distribution of BNF chapters 1-10 within clusters for the incident, 55+ only cohort*

Appendix C.2: Normalised elbow method

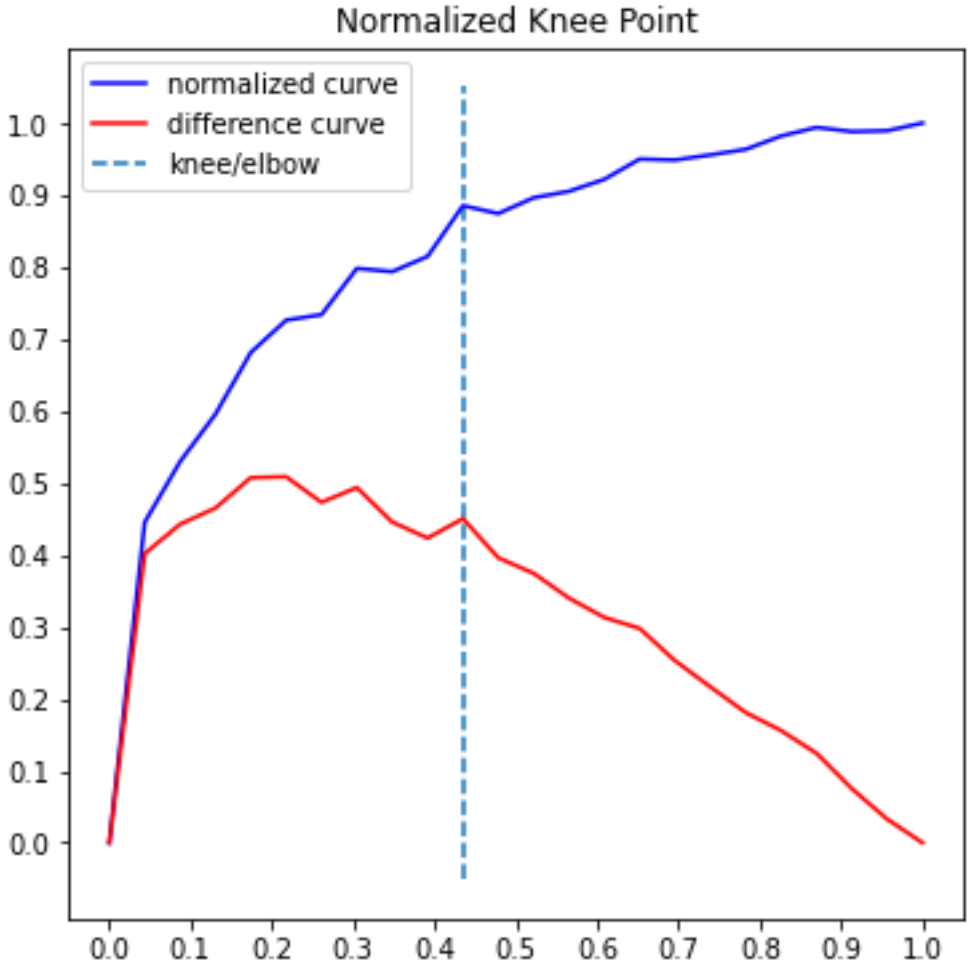

Figure 16. *Normalised distortion scores*

Appendix C.3: Percentage of BNF chapters within clusters

Table 6. *The percentage of BNF chapters 1-10 within clusters of the incident cohort*

	BNF 1	BNF 2	BNF 3	BNF 4	BNF 5	BNF 6	BNF 7	BNF 8	BNF 9	BNF 10
Cluster 1	10.07	13.81	4.1	12.83	3.6	17.57	7.17	1.3	16.26	13.28
Cluster 2	7.95	11.49	8.63	10.9	8.33	14.54	8.85	1.86	15.95	11.48
Cluster 3	17.45	17.53	5.86	16.92	2.63	21.69	5.26	0.46	7.93	4.26
Cluster 4	12.93	16.33	8.92	12.07	6.18	12.21	7.02	1.18	14.58	8.6
Cluster 5	11.42	14.52	9.22	10.15	8.09	13.4	8.71	1.8	14.13	8.56
Cluster 6	14.12	14.95	7.87	12.67	4.64	17.65	7.04	0.96	13.87	6.25
Cluster 7	7.84	9.54	8.73	8.75	9.53	12.7	10.18	2.46	17.0	13.27
Cluster 8	11.55	14.77	8.48	15.54	4.88	22.78	6.58	0.56	9.78	5.08
Cluster 9	10.18	14.35	8.37	11.44	5.63	16.12	5.92	0.95	16.14	10.91
Cluster 10	9.59	11.51	9.72	10.53	9.89	14.27	8.84	1.21	15.73	8.72
Cluster 11	15.23	18.6	6.53	14.65	3.53	17.17	5.36	0.6	11.7	6.63