A practical guide to Numero

Song Gao*[1], Stefan Mutter[1], Aaron E. Casey[1], and Ville-Petteri Mäkinen†[1,2,3]

[1]South Australian Health And Medical Research Institute, Adelaide, Australia
[2]University of Adelaide, School of Biological Sciences, Adelaide, Australia
[3]Computational Medicine, Institute of Health Sciences, Faculty of Medicine, University of Oulu, Finland

*song.gao@sahmri.com †ville-petteri.makinen@sahmri.com

5 October 2017

Abstract

In textbook examples, multivariable datasets are clustered into distinct subgroups that can be clearly identified by a set of optimal mathematical criteria. However, many real-world datasets arise from synergistic consequences of multiple effects, noisy and partly redundant measurements, and may represent a continuous spectrum of the different phases of a phenomenon. In medicine, complex diseases associated with ageing are typical examples. An individual's data are the result from the mix of genetic and environmental factors that have had cumulative effects over decades, and incidental factors at the time of the measurements. Furthermore, each individual typically has a unique mix of multiple ailments and morbidities that depend on physiology and circumstances. We postulate that population-based biomedical datasets (and many other real-world examples) do not contain an intrinsic clustered structure that would give rise to mathematically well-defined subgroups. From a modeling point of view, the lack of intrinsic structure means that the data form a contiguous cloud in high-dimensional space without abrupt changes in density to indicate subgroup boundaries, hence a mathematical criteria cannot segment the cloud purely by its internal structure. Yet we need data-driven classification and subgrouping to aid decision-making and to facilitate the development of testable hypotheses. For this reason, we developed the Numero package, a more flexible and transparent process that allows human observers to create usable multivariable subgroups even when conventional clustering frameworks struggle.

Package

Numero 0.99.0

Contents

1 Introduction 3
1.1 Limitations of conventional categorization 3
1.2 Self-organizing map 4
2 Terminology 6

3 Example dataset of diabetic kidney disease 7

4 Aims and study design 8

5 Statistical analysis 9

5.1 Importing datasets 9

5.2 Study design and preprocessing 10

5.3 Initializing a self-organizing map 12

5.4 Training the self-organizing map 13

5.5 Best-matching districts and data point layout 14

5.6 Map statistics 16

6 Visualization 17

6.1 Color amplitudes 18

6.2 Color and label assignment 19

6.3 Graphics output 20

7 Results and interpretation 21

7.1 Map quality 21

7.2 Summary of map colorings 23

7.3 Subgroup boundaries 24

7.4 Subgroup statistics 26

8 Concluding remarks 29

References 30

1 Introduction

Pattern recognition and clustering algorithms are the methodological cornerstones of the "big data" paradigm. In biology, high-throughput genomics and detailed imaging techniques are avidly applied to learn the details of how cells work and how diseases develop, and big datasets are expanding at an exponential rate, which also means that biomedical data analysis relies more and more on computational modeling and visualization in addition to the traditional descriptive statistics. The taxonomic tradition in biomedicine to categorize phenomena into distinct easily identifiable boxes remains strong, which explains the popularity of classical algorithms such as principal component analysis and hierarchical clustering as the first and often only choices for visualization and interpretation of the multi-dimensional structure of a complex dataset. However, both methods struggle when the dataset resembles a continuum instead of distinct clusters of data.

In the vignette, we focus on biomedical applications of the *Numero* framework, but there is nothing in the *R* package that is specifically aimed at biology. The choice of diabetic kidney disease as an example reflects our experience in the field, whereas Numero itself can be applied in any analysis problem that involves complex multi-dimensional data.

The document is organized into sections and paragraphs that describe our motivation for developing the library, introduce the concept of the self-organizing map, describe the dataset we use as an example of a biomedical study, go through a complete *R*-script of an analysis pipeline, define metabolic subgroups, interpret the results and discuss the role of the map analyses in publications.

1.1 Limitations of conventional categorization

The conventional notion of qualitative data patterns (e.g. health vs. disease) fits well with clustering algorithms that aim to find discriminatory borders automatically within the data. However, we argue that many biomedical datasets do not have a qualitative structure of regularity, but they instead reflect a multivariable spectrum of causes and consequences where the borderline between health and disease is blurred. For instance, chronic kidney disease is defined according to an internationally accepted threshold of glomerular filtration rate (GFR < 60 mL/min 1.73 m^2, (Levey & Coresh, 2012)), but there is no mathematically identifiable threshold effect in the population-based GFR distribution or any other biomarker or physical characteristic, as demonstrated by the continuous discussion on diagnostic criteria (Delanaye et al., 2012). Therefore, in most cases it is impossible to say exactly when someone develops chronic kidney disease, only that the diagnostic threshold is reached after a gradual decline, after which treatments can be initiated according to consensus guidelines.

Typical clustering analyses rely on algorithms that are tweaked for different application domains to produce classifications that are mathematically optimal, to reproduce an existing gold standard, or to predict future outcomes. We maintain that excessive reliance on mathematical criteria is not useful for datasets without intrinsic clustering structure, since the choice of the criteria will determine the output rather than the data or practical usefulness of the classification. Furthermore, the process that leads to category assignments is often too complicated to understand on a practical level, so the human observer must rely on the "black box" to produce the classification results without access to the inner workings. We propose a half-way solution, where the aim is to simplify the data presentation with statistical verification so that a human observer can determine a suitable subgrouping for a specific purpose, yet with sufficient access to the data patterns to understand the characteristics of the dataset in detail.

A traditional strict classification model will work well, if measurable qualitative differences exist. For instance, type 1 diabetes is an autoimmune form of diabetes that develops in children and adolescents. The condition is severe with a short life expectancy if untreated, so type 1 diabetes can be considered a qualitative example of health versus disease. Consequently, highly accurate diagnostic biomarkers such as glucose, insulin and C-peptide already exist. Even when treated, type 1 diabetes has a profound long-term impact on energy metabolism and it represents a distinct data cluster that is separate from the non-diabetic population.

Unlike type 1 diabetes, common age-associated diseases such as chronic kidney disease, type 2 diabetes, and atherosclerosis are challenging from a clustering perspective: they take decades to develop, they are not immediately life-threatening if left untreated, and there is a wide variation in severity across individuals. Furthermore, the affected individuals often suffer from multiple interacting chronic conditions, making it difficult to isolate specific causes and symptoms . Therefore, the simplistic notion of a qualitative threshold between health and disease becomes problematic. We aim to address these challenges by creating subgroups that are of practical value beyond mathematical criteria, and guided by a human observer with access to understandable presentations of the multivariable data patterns.

Multiple co-occurring and inter-connected phenomena are hallmarks of complex systems and the observable data that can be obtained from them. This presents a challenge to the traditional paradigms of biomedicine. For instance, differential diagnostics cannot cope well with multiple overlapping diseases, or evolving degrees of severity. This motivated us to develop the *Numero* framework in such a way as to enable visual comparisons of multiple overlapping diagnoses and their diagnostic criteria. We expect the *Numero* to be highly valuable in situations where the most important outcome or a set of outcomes is not obvious (e.g. competing risk scenarios). For instance, patients with type 1 diabetes may develop serious injuries to their vasculature over decades, but the affected organs, severity and rate of progression vary. Therefore, predictive models that focus only on a single outcome at a time may miss the big picture. The example of diabetic kidney disease we use in this vignette demonstrates how to use the *Numero* framework to gain insight into the overlaps and longitudinal associations between multiple morbidities.

1.2 Self-organizing map

Expressing multivariable data in visual form is a critical part of any knowledge discovery process, and an extensive number of algorithms have been developed in recent decades. In many cases, the aim is to project a set of multivariable data points into a two-dimensional presentation for human viewing (Figure 1). We built the *Numero* package using the self-organizing map (SOM) algorithm (Kohonen, Schroeder, & Huang, 2001), which is based on only a few simple mathematical rules, does not break down from missing data and can handle a high number of variables. We also developed a method to estimate the statistical significance of the map patterns (V.-P. Mäkinen et al., 2008b). Of note, the modular structure of the library allows users to replace the SOM with any other suitable algorithm for customized analysis pipelines.

Conceptually, the SOM algorithm mimics a human observer who wants to make sense of a set of objects. For instance, Figure 1A depicts schematic drawings of the flowering legume genus Luetzelburgia that grows in South America (Cardoso et al., 2014). Figure 1B shows how a human observer might organize the drawings based on their visual similarities (shape, size and

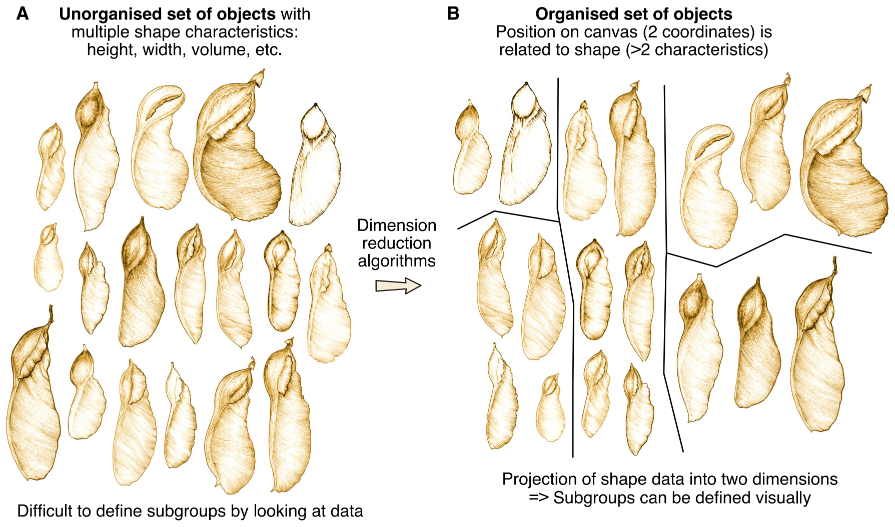

A
Unorganised set of objects with multiple shape characteristics: height, width, volume, etc.
Difficult to define subgroups by looking at data
Dimension reduction algorithms
B
Organised set of objects
Position on canvas (2 coordinates) is related to shape (>2 characteristics)
Projection of shape data into two dimensions
=> Subgroups can be defined visually

Figure 1: A conceptual example
The example shows how to organize objects with multiple features into a two-dimensional layout. The images were obtained from Cardoso, Queiroz, & Lima (2014).

other morphological details). By organization, we refer to the spatial layout of the drawings on the two-dimensional canvas: drawings that look similar are close to each other, whereas drawings that look different are far apart (in most cases). This is how all people, from children to elderly, sort and classify their objects with multiple observable features (= multivariable data points) with the help of a two-dimensional surface (= data map). The same observer then decides how to split the dataset into subgroups based on his or her domain knowledge.

If there are thousands of drawings, manual organization becomes impractical. For this reason, we let the SOM algorithm to do the first organization step, and to visualize the salient patterns within the dataset in a two-dimensional data map. The spatial principle still applies: multivariable data points that have similar values are close to each other, whereas data points that are different are on the opposite sides of the map. The second step of defining subgroups remains the responsibility of the observer. We argue that this type of data-assisted subgrouping is particularly useful in situations where there is no qualitative threshold between health and disease, but a line must be drawn to initiate preventative measures or treatments.

Although there are only 18 drawings in Figure 1, the nature of the dataset resembles many epidemiological studies. Specifically, some of the drawings are very similar, but it is not obvious how they should be classified into subgroups (i.e. our version of the figure can be disputed, a single "correct" visual subgrouping may not exist). If the classification was based on the height of a drawing, the results would look different compared to using the width – some drawings are narrow while being long, whereas others are wide despite being short. This is a naïve example on how the selection of the mathematical criteria for classification has a substantial impact on the results, and illustrates the motivation for our work. We developed the *Numero* library as an alternative tool that helps researchers to define meaningful groupings when pure mathematics cannot provide a conclusive answer.

Previous versions of the software (written for Matlab) were successfully used on a range of metabolomics and other biomedical studies (Bernardi et al., 2010; Kumpula et al., 2010; Kuusisto et al., 2012; V.-P. Mäkinen et al., 2008a, 2008b, 2013; Tukiainen et al., 2008; Würtz et al., 2011). However, the old version used a rectangular SOM, which tends to guide observers into picking four subgroups in the corners even when not supported by data. We created the *Numero* package with a circular implementation of the SOM to remove the limitations from cornered border shapes. Additional technical details and supportive material are available as an online supplement within a previous publication (V.-P. Mäkinen et al., 2012).

2 Terminology

Data point — Here, we define the term *data point* as a single uniquely identifiable row in a spreadsheet of data (with variables as columns). For instance, in the diabetic kidney disease dataset described in the next section, a *data point* refers to a patient (and vice versa) as there is only one row per patient.

Map — A *map* is a general term to describe the two-dimensional canvas onto which the multivariable data points are projected. The concept is analogous to a geographic *map* that indicates where people live, except that the location is not based on geography (i.e. physical distances), but comes from the data (i.e. distances = data-based similarities).

Layout — We make a distinction between what is a map, and what is the *layout* of data points on it. The *layout* is a table of data point locations as coordinates, whereas the map is a more integrated concept that also includes information that is necessary to find the locations of new previously unseen data points, and to draw and paint the map in visual form.

District — A *district* refers to a pre-defined division of the map into uniformly sized areas. The *districts* are created mainly for technical reasons: using *districts* speeds up calculations and enables the estimation of map-related statistics. This is analogous to a real city being divided into *districts* to estimate regional demographics, for instance.

Coloring — The Numero framework always creates a single map. However, the map districts can be painted with different colors. This enables the user to create multiple *colorings* of the map to visualize regional differences. These *colorings* can be made for each variable, which helps to identify which parts of the map are particularly important for a specific phenomenon. Again, this is similar to a real city map where the districts are colored according to the income level of the local residents, or according to the mean age, smoking rates, obesity etc.

Subgroup — We expect that most uses of Numero will result in the subgrouping of a complex dataset. Visually, we define a *subgroup* via a contiguous set of adjacent districts on the map. Consequently, all the data points that are located within the set of districts are the subgroup members.

District profile — The SOM algorithm works through the districts during the optimization of the data point layout on the map. The computational process eventually converges to a stable configuration that is stored as a set of *district profiles*. From a practical point of view, a *district profile* represents the typical average profile that captures the characteristics of the data points within the district. In technical terms, the *district profile* (also known as the prototype) contains the mean weighted data values across all the data points, where the weights are determined by the neighborhood function used in the SOM algorithm.

Best-matching district — The *best-matching district* (BMD, also known as the best-matching unit in the literature) is the district with a profile that is the most similar to a data point when considering all variables simultaneously. The BMD is closely related to the data point layout: the assigned location for a data point is the location of the BMD for that data point.

3 Example dataset of diabetic kidney disease

Diabetic kidney disease is the leading indication for dialysis and kidney transplantation in the developed countries, and carries a substantial risk of premature death due to cardiovascular disease. About one third of individuals with type 1 diabetes will develop diabetic kidney disease during their lifetime. As the age of onset of type 1 diabetes is in childhood or adolescence, these individuals will develop complications at a relatively early age. Therefore, people with type 1 diabetes represent a particularly vulnerable group facing lower quality of life and reduced life span due to kidney damage.

Albuminuria (elevated albumin concentration in urine) is the basis for the clinical classification of diabetic kidney disease. In this example, we applied the threshold of 300mg/24h, if 24h urine collections were done and 0.2 mg/min when overnight urine data were available from the local medical centers that examined the patients. If the threshold was exceeded in at least two out of three consecutive measurements, we assigned the individual in the diabetic kidney disease group. In addition, the FinnDiane Study Group measured urinary albumin excretion rate from a single 24h urine sample in their designated central laboratory. The logarithm of the albumin excretion rate was included in the example dataset.

Our example dataset contains a subset of data from a previous publication (V.-P. Mäkinen et al., 2008b). We created the simplified dataset for educational purposes, but it contains enough information to replicate some of the findings from the original study. The dataset includes 613 individuals of whom 225 individuals had diabetic kidney disease at baseline. In addition, we included information on whether an individual had died after an eight-year follow-up to demonstrate how the study design we chose can be applied to longitudinal data. The available data are summarized in Table 1.

Table 1: Summary of the diabetic kidney disease dataset from the FinnDiane Study
The mean and standard deviation are reported for continuous variables. Abbreviations: urinary albumin excretion rate (AER), triglycerides (TG), high density lipoprotein subclass 2 (HDL2). P-values were estimated by the t-test for continuous variables and by Fisher's test for binary traits.

Trait	No kidney disease	Diabetic kidney disease	P-value
Men / Women	192 / 196	119 / 106	0.45
Age (years)	38.8 ± 12.2	41.7 ± 9.7	0.0012
Type 1 diabetes duration (years)	25.3 ± 10.3	28.6 ± 7.8	<0.001
Log10 of AER (mg/24h)	1.20 ± 0.51	2.72 ± 0.59	<0.001
Log10 of TG (mmol/L)	0.034 ± 0.201	0.159 ± 0.212	<0.001
Total cholesterol (mmol/L)	4.89 ± 0.77	5.35 ± 0.96	<0.001
HDL2 cholesterol (mmol/L)	0.54 ± 0.16	0.51 ± 0.18	0.027
Log10 of serum creatinine (µmol/L)	1.94 ± 0.09	2.14 ± 0.24	<0.001
Metabolic syndrome	90 (23.2%)	114 (50.7%)	<0.001
Macrovascular disease	16 (4.1%)	38 (16.9%)	<0.001
Diabetic retinopathy	133 (34.4%)	178 (79.1%)	<0.001
Died during follow-up	13 (3.4%)	43 (19.1%)	<0.001

4 Aims and study design

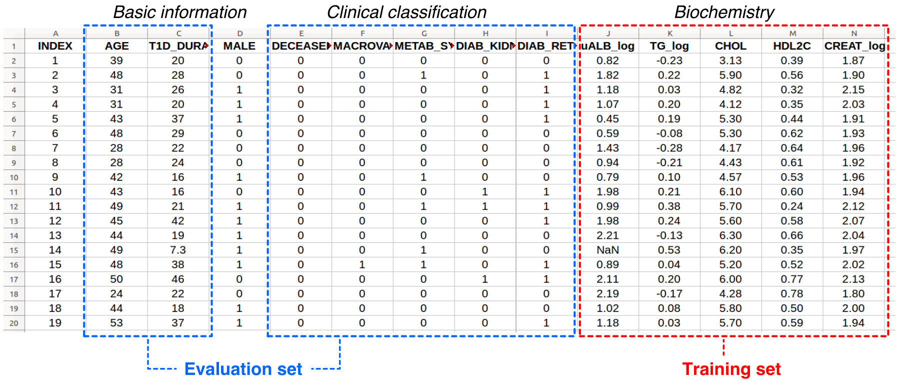
In the original study, we hypothesized that the metabolic profile of an individual with type 1 diabetes at baseline predicts adverse events in the future (V.-P. Mäkinen et al., 2008b). Here, we set two aims to test the same hypothesis in the example dataset:

1. Define metabolic subgroups of type 1 diabetes based on biochemical data.
2. Identify subgroups with high all-cause mortality.

We chose these aims to accommodate a high number of variables and to ensure statistical robustness. Please note that we included only a few variables in the example dataset for pedagogical reasons, but the SOM in the original study was created based on thousands of variables.

The strict separation of Aim 1 and 2 is an example of an unsupervised classification design where the metabolic subgroups are created without using the mortality data. Only after the subgroup modeling has been completed, the deaths during follow-up will be counted within the subgroups. An alternative would be to employ regression or other supervised methods that use all the available data simultaneously to create a predictive model of mortality. While supervised models can achieve high accuracy, they rarely work well outside the dataset they were created for, and they may fail if the outcome to be predicted is poorly defined or biased. For these reasons, we adopted the more robust unsupervised classification design.

We denote the study design as "split-by-variables" since it starts from a spreadsheet with one patient per row and the variables organized into columns, and then assigns one set of variables into the training set, and the remaining variables (e.g. deceased or alive at follow-up) into the evaluation set (Figure 2). Since the evaluation set plays no part in the training of the SOM, we can estimate the statistical significance of the mortality pattern without over-estimating the model accuracy.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	INDEX	AGE	T1D_DURA	MALE	DECEASEI	MACROVA	METAB_SY	DIAB_KIDI	DIAB_RET	uALB_log	TG_log	CHOL	HDL2C	CREAT_log
2	1	39	20	0	0	0	0	0	0	0.82	-0.23	3.13	0.39	1.87
3	2	48	28	0	0	0	1	0	1	1.82	0.22	5.90	0.56	1.90
4	3	31	26	1	0	0	0	0	1	1.18	0.03	4.82	0.32	2.15
5	4	31	20	1	0	0	0	0	1	1.07	0.20	4.12	0.35	2.03
6	5	43	37	1	0	0	0	0	1	0.45	0.19	5.30	0.44	1.91
7	6	48	29	0	0	0	0	0	0	0.59	-0.08	5.30	0.62	1.93
8	7	28	22	0	0	0	0	0	0	1.43	-0.28	4.17	0.64	1.96
9	8	28	24	0	0	0	0	0	0	0.94	-0.21	4.43	0.61	1.92
10	9	42	16	1	0	0	1	0	0	0.79	0.10	4.57	0.53	1.96
11	10	43	16	0	0	0	0	1	1	1.98	0.21	6.10	0.60	1.94
12	11	49	21	1	0	0	1	1	1	0.99	0.38	5.70	0.24	2.12
13	12	45	42	1	0	0	0	0	1	1.98	0.24	5.60	0.58	2.07
14	13	44	19	1	0	0	0	0	0	2.21	-0.13	6.30	0.66	2.04
15	14	49	7.3	1	0	0	1	0	0	NaN	0.53	6.20	0.35	1.97
16	15	48	38	1	0	1	1	0	1	0.89	0.04	5.20	0.52	2.02
17	16	50	46	0	0	0	0	1	1	2.11	0.20	6.00	0.77	2.13
18	17	24	22	0	0	0	0	0	0	2.19	-0.17	4.28	0.78	1.80
19	18	44	18	1	0	0	0	0	0	1.02	0.08	5.80	0.50	2.00
20	19	53	37	1	0	0	0	0	1	1.18	0.03	5.70	0.59	1.94

Figure 2: Application of the split-by-variable study design in the diabetic kidney disease example
Of note, the training set is adjusted for sex differences; hence the 'MALE' column is not formally included in the evaluation set.

5 Statistical analysis

The architecture of the analysis pipeline for the diabetic kidney disease example is detailed in Figure 3. First, we describe how to import data from a tab-delimited spreadsheet into a matrix that is compatible with Numero functions and how to preprocess the data for analysis (Figure 3A-D). Next, we create the SOM based on the training set (Figure 3B,E). The third segment focuses on map statistics and how to color the maps according to regional variation (Figure 3F,G). Lastly, we interpret the results and provide a template on how to manually edit the plots for publications (Figure 3H,I).

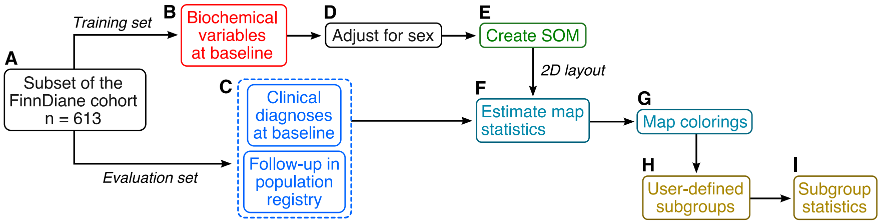


Figure 3: Analysis steps in the diabetic kidney disease example

5.1 Importing datasets

We have included the example dataset in the installation package. To access it, type

```
fname <- system.file("extdata", "finndiane_dataset.txt", package = "Numero")
```

The file contains a tab-delimited spreadsheet where the first column is the patient identity, with the data in the subsequent columns to the left. The first row contains the column headings.

If the data file is very large, or otherwise problematic, the default functions in R such as *read.delim()* may struggle to import the data values correctly. For this reason, we provide the *nroMatrix()* function as a streamlined alternative that guarantees the imported matrix is suitable for the SOM analysis. Specifically, the *nroMatrix()* checks if all identities are unique, forces all elements to a numeric value, and has less computational overhead compared to the default R alternative. On the other hand, *nroMatrix()* lacks several of the automatic filtering features that are included in *read.delim()*, so we recommend a careful inspection of the input file if problems should arise.

To import the example data on diabetic kidney disease, type

```
db <- nroMatrix(file = fname, keyvars = "INDEX")
```

and to show the summary you can type the following command and see its output below:

```
summary(db)
##       AGE           T1D_DURAT          MALE           DECEASED
##  Min.   :15.00   Min.   : 2.59   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:31.00   1st Qu.:19.00   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :39.00   Median :26.00   Median :1.0000   Median :0.00000
##  Mean   :39.86   Mean   :26.53   Mean   :0.5073   Mean   :0.09135
##  3rd Qu.:48.00   3rd Qu.:34.00   3rd Qu.:1.0000   3rd Qu.:0.00000  
##  Max.   :74.00   Max.   :53.00   Max.   :1.0000   Max.   :1.00000  
##                                                                    
##    MACROVASC        METAB_SYNDR      DIAB_KIDNEY     DIAB_RETINO    
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000  
##  Median :0.00000   Median :0.0000   Median :0.000   Median :1.0000  
##  Mean   :0.08809   Mean   :0.3328   Mean   :0.367   Mean   :0.5082  
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000  
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000  
##                                                     NA's   :1       
##     uALB_log          TG_log              CHOL            HDL2C       
##  Min.   :0.3617   Min.   :-0.37160   Min.   : 2.920   Min.   :0.0910  
##  1st Qu.:0.9590   1st Qu.:-0.06550   1st Qu.: 4.470   1st Qu.:0.4120  
##  Median :1.5682   Median : 0.05690   Median : 4.980   Median :0.5200  
##  Mean   :1.7526   Mean   : 0.08023   Mean   : 5.061   Mean   :0.5273  
##  3rd Qu.:2.4900   3rd Qu.: 0.21750   3rd Qu.: 5.600   3rd Qu.:0.6400  
##  Max.   :3.8788   Max.   : 0.90850   Max.   :10.000   Max.   :1.1900  
##  NA's   :28                                                           
##    CREAT_log    
##  Min.   :1.415  
##  1st Qu.:1.909  
##  Median :1.978  
##  Mean   :2.013  
##  3rd Qu.:2.061  
##  Max.   :3.035  
## 
```

5.2 Study design and preprocessing

Here, we adopted the split-by-variable study design (Figure 2), which means that we defined the variables we intend to use for training the SOM beforehand so that we can use the remaining variables to evaluate the relevance of the map patterns in a statistically robust way. We hypothesize that the metabolic phenotype of an individual predicts future adverse outcomes. To investigate the hypothesis, we select all blood and urine biomarkers at baseline as the training set (Aim 1), and then use the remaining columns that contain data on clinical end-points and mortality as the evaluation set (Aim 2). If our hypothesis is correct, we should see a statistically significant regional pattern for mortality on the SOM that we constructed based on the metabolic variables at baseline.

To extract the training columns, use the basic R language feature

```
trdata <- db[,c("uALB_log", "TG_log", "CHOL", "HDL2C", "CREAT_log")]
```

In the data file, the biomarkers are expressed in their physical concentration units, or as log-transformed versions. As a consequence, the standard deviations of the data columns vary, which can bias the SOM to fit better to those biomarkers that have a wide numerical variation. In most cases, it is desirable to standardize the training set before analyses, so that the information content rather than the measurement scale determines the modeling outcome.

Sex difference is another factor to consider when preparing the training set. Men and women display anatomical and metabolic differences, which usually complicate the interpretation of the SOM. For this reason, we recommend using a sex-specific standardization procedure that eliminates the differences. If necessary, separate visualizations can be made afterwards for men and women using the same map, please see V.-P. Mäkinen et al. (2012) for an example.

To separate men and women, type

```
men <- which(db[, "MALE"] == 1)
women <- which(db[, "MALE"] == 0)
```

which centers each column to zero mean, and scales the data to unit variance. Note that the standardization is applied separately for men and women, which means that all differences in the mean and variance between the sexes are removed from the dataset.

You can verify that the training data is zero centered by typing the following commands that will produce the output below:

```
print(summary(trdata[men,]))
##     uALB_log           TG_log              CHOL           HDL2C       
##  Min.   :0.4281   Min.   :-0.3028   Min.   :3.040   Min.   :0.0910  
##  1st Qu.:1.0138   1st Qu.:-0.0269   1st Qu.:4.370   1st Qu.:0.3645  
##  Median :1.7853   Median : 0.0899   Median :4.910   Median :0.4540  
##  Mean   :1.8650   Mean   : 0.1221   Mean   :4.964   Mean   :0.4703  
##  3rd Qu.:2.6100   3rd Qu.: 0.2455   3rd Qu.:5.500   3rd Qu.:0.5700  
##  Max.   :3.8788   Max.   : 0.9085   Max.   :9.400   Max.   :0.9800  
##  NA's   :13                                                         
##    CREAT_log    
##  Min.   :1.716  
##  1st Qu.:1.944  
##  Median :2.004  
##  Mean   :2.058  
##  3rd Qu.:2.093  
##  Max.   :3.035  
##                 
print(summary(trdata[women,]))
##     uALB_log           TG_log               CHOL           HDL2C       
##  Min.   :0.3617   Min.   :-0.37160   Min.   : 2.92   Min.   :0.1350  
##  1st Qu.:0.8662   1st Qu.:-0.10790   1st Qu.: 4.61   1st Qu.:0.4820  
##  Median :1.3424   Median : 0.00855   Median : 5.05   Median :0.5750  
##  Mean   :1.6358   Mean   : 0.03711   Mean   : 5.16   Mean   :0.5861  
##  3rd Qu.:2.3655   3rd Qu.: 0.15762   3rd Qu.: 5.70   3rd Qu.:0.6775  
##  Max.   :3.7209   Max.   : 0.65610   Max.   :10.00   Max.   :1.1900  
##  NA's   :15                                                          
##    CREAT_log    
##  Min.   :1.415  
##  1st Qu.:1.881  
##  Median :1.944  
##  Mean   :1.967  
##  3rd Qu.:2.016  
##  Max.   :2.788  
##                 
```

5.3 Initializing a self-organizing map

Training a SOM requires two steps: i) initialization of the map and ii) iterative optimization of the district profiles. The initialization of district profiles influences the usability of the final map, and several methods including principal component analysis have been proposed (Attik, Bougrain, & Alexandre, 2005). Our experience suggests that the most useful results are usually achieved by creating a limited number of seed profiles that are placed on the edges of the map. The districts in the middle are set automatically in such a way that the transition from one seed to another via the districts in the middle is smooth. The Numero package includes the *nroKmeans()* function which we use here to determine the seed profiles. It is based on the classical k-means algorithm, but our implementation is specifically designed for datasets with missing data and to produce output that is compatible with the other Numero functions.

The minimum number of seeds is three as the triangle is the simplest polygon to cover the map. To create the seed profiles, you can use the command

```
km <- nroKmeans(x = trdata, k = 3)
```

The output is a list that contains the seed profiles, which seeds are closest to each of the rows in the training dataset, and a history of training errors. To show the seeds, type the following command. The seed profile output has the same columns as the training data.

```
print(km$centroids)
##      uALB_log      TG_log     CHOL     HDL2C CREAT_log
## [1,] 1.193738 -0.02385696 4.387166 0.5064555  1.952985
## [2,] 2.796553  0.17891854 5.420209 0.5136950  2.113659
## [3,] 1.132532  0.11468862 5.720743 0.5865437  1.965427
```

Most SOM software is based on rectangular or borderless maps with periodic boundaries. In our experience, the former suffer from an artificial tendency for observers to define four separate subgroups in the corners. The latter do not suffer from boundary artifacts, but are complicated to interpret. For these reasons, we developed a circular map topology for Numero since it does not have corners and yet the well-defined borders limit the visual complexity of the regional patterns.

The preferred size of the map depends on the number of data (small maps for small datasets), however, we advise against using large maps due to their complexity. For most biomedical and epidemiological applications, map radii between two and five will provide enough flexibility and expressive power based on our experience.

To initialize a circular map with a radius of three districts according to the seed profiles, use the command

```
sm <- nroKoho(seeds = km$centroids, radius = 3)
```

This will create the initial matrix of district profiles, and additional topological information that will be required for visualization. To show the district profiles, type the following command that will show how the profiles have the same format as the seeds:

```
print(head(sm$centroids))
##      uALB_log     TG_log     CHOL     HDL2C CREAT_log
## [1,] 1.707607 0.08991674 5.176039 0.5355647  2.010690
## [2,] 2.051065 0.12912937 5.357619 0.5347873  2.044582
## [3,] 1.584102 0.10646482 5.398405 0.5526813  2.002395
## [4,] 1.437547 0.07911748 5.221293 0.5471815  1.986586
## [5,] 1.410435 0.04002411 4.869093 0.5274672  1.979339
## [6,] 1.530999 0.04955830 4.893125 0.5248702  1.990699
```

5.4 Training the self-organizing map

The Kohonen's self-organizing map algorithm was originally developed to mimic the plasticity of neural networks (Kohonen et al., 2001). It scales up well for datasets with a high number of variables, and it can handle missing data values, which is why we chose it as the default method in Numero. To apply the SOM algorithm to the standardized training set, use the command

```
sm <- nroTrain(som = sm, x = trdata)
```

The *nroTrain()* function updates the district profiles in such a way that when the data points are assigned the best-matching districts (BMD), the resulting layout is distributed more evenly across the map. BMDs and the data point layout are discussed in the next section.

The *nroTrain()* function adds a record of the training process within the output list. To plot it on screen, type

```
plot(sm$history)
```

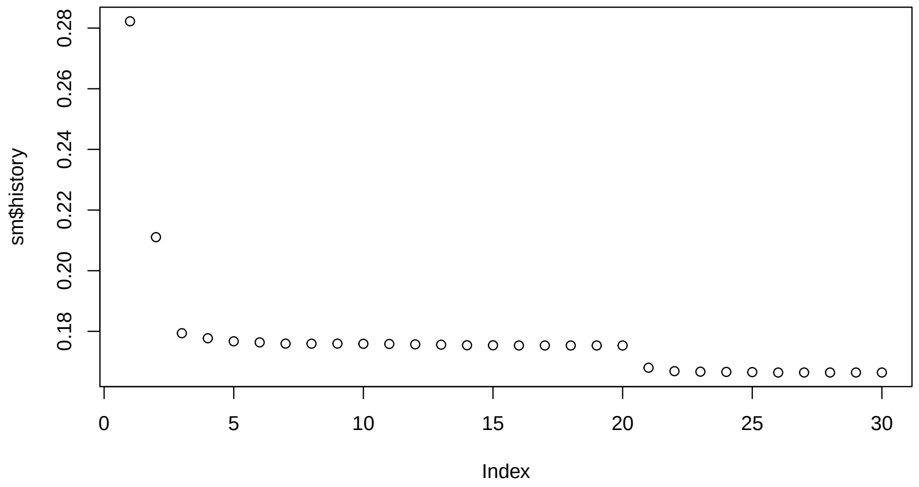


Figure 4: **SOM training history in the diabetic kidney disease example**

The results are shown in Figure 4. The curve shows the mean Euclidean distance between a data point and its best matching district profile for each training cycle. In most cases, the first few cycles account for the largest reductions in the training error. The abrupt reduction that can be observed after the first twenty cycles is part of the training process. For those interested in the technical details, switching from a wide neighborhood function to a narrower one causes the threshold. This is beneficial, since by starting the training process with a wide neighborhood function forces the SOM to adapt to large (and presumably more important) patterns first before adapting to the minor details.

We have included additional tools in the Numero package to assess the internal properties of the SOM given a training dataset. However, the full use of these tools requires visualization functions that are not covered until later in the document. For this reason, we will return to this subject in a dedicated section after introducing map colorings.

5.5 Best-matching districts and data point layout

The application of the SOM algorithm mimics a human researcher who wants to investigate the cohort of type 1 diabetic patients. Suppose the researcher goes through the medical records of all 613 individuals and then organizes the folders on a giant round table in such a way that patients with mutually similar clinical profiles are placed next to each other, whereas patients who are different are on opposite sides of the table. From this organized view, it is then possible to identify sections of the table (= patient subgroups) where there is a high risk of premature death.

To translate the story of the human researcher into a computer program, it is necessary to introduce new concepts and it is also useful to revisit to pre-defined terminology from the beginning of the document. First, the map (representing the giant round table) is divided into districts for technical reasons since the type of manual positioning algorithm a human would apply is unfeasible for large datasets. The districts are also anchors that enable the assignment of the patients onto specific map positions, again important for technical reasons.

The best-matching district (BMD) for a data point is the second important concept. Each data point is compared against a district profile by calculating their pair-wise Euclidean distance. This is repeated across the districts, and after the distances between the data point and all district profiles are calculated, the profile with the shortest distance is chosen as the best match. When the BMDs are located for all data points, the results are collected in a spreadsheet, which we denote as the data point layout. The layout is conceptually equivalent to the spatial configuration of folders on the human researcher's table.

To create a layout, you can use the command

```
matches <- nroMatch(som = sm, x = trdata)
```

The districts are labeled as integers starting from one, and the output data frame contains the BMD labels for each data point. To print a section of the output on screen type the following command. Its output will be a data frame with four columns:

```
print(head(matches))
##   POS       DIST   QUALITY COVER
## 1  31 0.40206211 0.5855206     1
## 2  13 0.08849945 1.3057020     1
## 3  17 0.15857445 1.0241787     1
## 4  30 0.13674617 1.0979176     1
## 5  26 0.20872318 0.8872723     1
## 6  26 0.13426729 1.1069685     1
```

The column POS contains the district identity labels of the BMDs, and DIST shows the Euclidean distance between a data point and the BMD. QUALITY is calculated from the distance by dividing it with the average training error. This provides a scale-independent relative estimate on how well the data points were matched compared to a typical data point in the training set. Finally, COVER shows if a multivariable data point contained missing elements (1 means all elements were usable and 0 means none of the elements contained a numerical value).

Ideally, the data points should be uniformly distributed across districts. While uniformity is rarely observed for real datasets, uneven numbers are usually not a problem unless there are large contiguous groups of districts with very high or very low occupancy. We will revisit the spatial data point distribution in a later section that addresses map quality, but a tabulation of the BMD assignments is a quick way to see if the output is reasonable

```
t <- table(matches$POS)
counts <- data.frame(DISTRICT=names(t), N=as.integer(t))
print(counts, row.names=FALSE)
##  DISTRICT  N
##         1 12
##         2 16
##         3 14
##         4  8
##         5 16
##         6 18
##         7 19
##         8 13
##         9 13
##        10 12
##        11 15
##        12 13
##        13 11
##        14  6
##        15 13
##        16 18
##        17 14
##        18 17
##        19  8
##        20  9
##        21 12
##        22  4
##        23 15
##        24 26
##        25 13
##        26 19
##        27 27
##        28 18
##        29 22
##        30 28
##        31 29
##        32 12
##        33 11
##        34 17
##        35 14
##        36 20
##        37 16
##        38 13
##         39 17
##         40 15
```

If a large number of districts in the above output were devoid of data points, it would indicate that the map did not capture the diversity of the data set, and therefore would not be useful for subgrouping. However, the data points are scattered across all districts in this example, which suggests the layout will be useful.

5.6 Map statistics

Statistical evaluation of whether an observation is likely to occur purely by chance is the cornerstone of biomedical data analysis. In our example, we achieve well-defined statistical analyses via the split-by-variable design and the non-parametric permutation engine that is built into the package. The former ensures that our results are not over-optimistic (no over-fitting) and the latter enables us to avoid restrictive assumptions on the nature of the data generating processes that are often violated in real datasets.

In our example, we investigate if the metabolic profile at baseline indicates the risk of death during follow-up. To estimate statistical significance, it is necessary to find out how much the areas of the map can differ with respect to mortality just by the virtue of random fluctuations. This concept is formally encapsulated by the null hypothesis. Here, the null hypothesis states that the data point layout is not associated with the number of deaths, that is, the location of a patient on the map does not provide any information on how likely the patient is to die in the next eight years. If the null hypothesis is true, then the observed layout and regional patterns of mortality should be within the variation we would expect for random layouts. We use permutation analysis to simulate a high number of random layouts, and then compare the observation with the simulated findings to see if it could have occurred by chance alone (V.-P. Mäkinen et al., 2008b). Within the split-by-variable design, P-values for statistical significance are only meaningful for those variables that are in the evaluation set since, by definition, the variables in the training set will always be strongly associated with the layout. However, it makes sense to evaluate the expected range of regional variation also for the training set, as we will demonstrate later in the vignette. Knowing the randomly expected amplitude of regional patterns (i.e. the basal amplitude) helps us to assess which of the training variables had the strongest influence on the layout. For these reasons, we will apply the permutation analysis to all variables, but only report P-values for the evaluation set.

The function *nroPermute()* repeats the following procedure: i) re-assign best-matching district randomly in accordance with the null hypothesis, ii) recalculate the average district values across the map and iii) summarize the regional variation with a single descriptive statistic. When a sufficient number of cycles has been achieved, the null distribution of the descriptive statistic is analyzed to determine how far, in terms of standard deviations, the observed value is from the mean prediction by the null hypothesis. This distance is reported as the Z-score of regional variation. Furthermore, the function also estimates how frequently a permuted layout produced a regional variation that exceeded the observation (frequency-based P-value).

To go through all the variables in the dataset, we place the *nroPermute()* inside a loop. Simultaneously, we also distinguish between a training variable (no P-value needed and we can use fewer cycles) and an evaluation variable (a maximum of 10,000 cycles to estimate frequency-based P-values). The entire code segment is shown below:

```
stats <- matrix(NA, ncol(db), 5)
rownames(stats) <- colnames(db)
for( vname in colnames(db) ) {

    # Check if a training variable.
    nsim <- NA
    pos <- match(vname, colnames(trdata))
    if(is.na(pos)) nsim <- 10000

    # Estimate the dynamic range of regional variation.
    tmp <- nroPermute(map = sm, x =db[, vname], bmus = matches$POS, n = nsim)
    colnames(stats) <- colnames(tmp)
    stats[vname,] <- as.matrix(tmp)
}
```

To see the resulting five columns, type the following command:

```
print(stats)
##                      Z          P.z P.freq N.data N.cycles
## AGE          4.611190 2.001849e-06  0e+00    613    10000
## T1D_DURAT    3.833326 6.321113e-05  0e+00    613    10000
## MALE         3.305087 4.747344e-04  4e-04    613    10000
## DECEASED     5.070694 1.981842e-07  0e+00    613    10000
## MACROVASC    3.887390 5.066399e-05  0e+00    613    10000
## METAB_SYNDR  5.593069 1.115452e-08  0e+00    613    10000
## DIAB_KIDNEY 10.935264 3.908897e-28  0e+00    613    10000
## DIAB_RETINO  6.827991 4.305610e-12  0e+00    612    10000
## uALB_log    12.813613           NA     NA    585     1000
## TG_log       9.225221           NA     NA    613     1000
## CHOL        17.520242           NA     NA    613     1000
## HDL2C        5.743220           NA     NA    613     1000
## CREAT_log    8.156577           NA     NA    613     1000
```

The first column contains Z-scores that indicate how far the observed regional variation is from the mean expected value if the null hypothesis is true. The second column shows the statistical significance as estimated from the Z-score using the cumulative Gaussian distribution. The third column shows the P-value as estimated by calculating the frequency of observing regional variation for a simulated random layout that exceeds the observed variation. The fourth column shows how many data values were available for the analysis, and the last column shows the number of simulated random layouts that were created.

Please note how the P-values are missing for the training variables. This was achieved by setting the number of cycles to NA for the training variables in the loop. The *nroPermute()* function uses this as an indicator to exclude the P-values for the training set.

6 Visualization

After estimating the map statistics, we now have almost all the results that are required to color the map according to the data patterns: the topological information that is carried within the variable *sm* allows us to draw the districts correctly, the data point layout specifies the locations of the 613 patients on the SOM, and the z-scores tell us how much each training variable influenced the layout, and how the evaluation variables are associated with the layout. The last remaining item to calculate is related to what colors should be assigned to each district according to the average observation among the local residents. To make the color scales comparable between variables of different measurement units, we standardize the dynamic range according to the map statistics.

6.1 Color amplitudes

Assigning a color palette to a set of values is not much different from photography. When a photo is taken, the intensity of light is converted into numbers by the digital camera, and then converted back to light on the viewing screen. If there is too much light, the photo gets overexposed, which means that most pixels show up as "burned" since the intensity is beyond their dynamic range (i.e. light saturates the sensor). If the photo is underexposed, most pixels will show a zero signal (i.e. the light is below the detection limit). In principle, the SOM colors work the same way: we aim to set up an optimal color assignment so that the colorings with very high regional variation do not over-expose too much, while the colorings with less regional variation can still show differences between districts despite under-exposure.

In the *Numero* framework, a photo corresponds to a map coloring (please revisit Terminology if necessary), light intensity is analogous to statistical significance (captured by z-scores), and the dynamic range is the gap between the lowest and highest district averages. Importantly, the "camera settings" are kept constant to ensure all colorings remain visually comparable. Ideally, the camera would be set up so that the full dynamic range of every coloring could be expressed within the available color palette. However, this approach is usually impractical as interesting detail could be lost for variables that show statistically modest but biologically critical variation. The following code segments introduce a procedure that we have found to produce high quality results in most datasets.

First, we separate the z-scores of training and evaluation variables. This is important since we can expect the training set to show higher regional variation than the evaluation set, and it is typically more useful to avoid under-exposing the evaluation variables even if it means over-exposing the training variables. The map statistics loop was constructed in such a way that missing P-values indicate the training variables:

```r
trmask <- which(is.na(stats[, "P.z"]))
evmask <- which(stats[, "P.z"] >= 0.0)
```

In the next step, we apply a heuristic rule to set the base z-score, which fills the role of the constant camera setting:

```r
zbase <- 0.5*(mean(stats[trmask, "Z"], na.rm = TRUE) +
              max(stats[evmask, "Z"], na.rm = TRUE))
zbase <- max(zbase, 2.0)
```

The last line ensures that if none of the map colorings show significant regional variation (a high Z-score), the colorings will stay washed out.

Finally, the *Numero* function that assigns the colors, which we will use in the next section, requires a ratio to indicate the available dynamic range for colors for each variable, which we denote as color amplitudes:

```
amplitudes <- stats[, "Z"] / zbase
amplitudes <- pmax(amplitudes, 0.02)
```

The last line is to ensure that a minimum positive amplitude is available for all variables for a more consistent visual presentation.

6.2 Color and label assignment

For clarity, this section focuses on how to produce a map coloring for the prevalence of diabetic kidney disease across the map districts rather than start with a loop that goes through all variables (which we will introduce later). We first choose the variable:

```
vname <- "DIAB_KIDNEY"
```

The color of a district depends on the estimated mean value across its local resident data points. To calculate the district values, use the command

```
plane <- nroAggregate(map = sm, x = db[, vname], bmus = matches$POS)
```

In the SOM literature, the set of district mean values for a variable are typically referred to as the component plane, hence the name of the output. We now have all the materials to assign colors to each district based on i) the amplitude for kidney disease, which tells how much "exposure" the camera provides, and ii) the component plane, which gives the dynamic range and district means:

```
colrs <- nroColorize(values=plane, amplitude=amplitudes[vname])
```

The output is a list of colors in a format that matches the values in the component plane and can be used in subsequent Numero functions in the pipeline.

Due to the standardization by z-scores, the colors are not directly relatable to the original measurement units, or to the original binary coding of diabetic kidney disease. For this reason, text labels that indicate the actual mean values for selected districts are a useful visual addition to the final map plot. To create a set of labels for the map coloring, use the command

```
labls <- nroLabel(map=sm, values=plane)
```

The final piece of information for visualization is now completed and the code below shows the information for the first few districts:

```
print(head(data.frame(VALUE=plane, COLOR=colrs, LABEL=labls)))
##       VALUE   COLOR  LABEL
## 1 0.3362211 #cbeeff
## 2 0.5350108 #ffc778
## 3 0.2700085 #8ed9ff +0.270
## 4 0.1227043 #559eff
## 5 0.1191208 #559dff
## 6 0.2411225 #76d0ff
```

6.3 Graphics output

The *Numero* package contains the basic functions to save the map colorings in the Scalable Vector Graphics (SVG) format. Vector graphics is advantageous here due to the smaller size of the files and the high visual quality that can be achieved in printed media. Macs and most Linux distributions have integrated user interface support for SVG files, which helps when viewing multiple files simultaneously. Internet browsers are also able to render SVG, and they can be used in systems without an integrated support.

To annotate the plots beyond the file name alone, it is possible to add a title text and, for an evaluation variable such diabetic kidney disease, it is useful to indicate the P-value as well:

```
pval <- stats[vname, "P.z"]
ttxt <- sprintf("%s, P = %.2e", vname, pval)
```

We now have all the necessary information and intermediate results to create the SVG code that specifies how the coloring looks:

```
smfig <- nroCircus(map = sm, colors = colrs, labels = labls, title = ttxt)
```

The output from *nroCircus()* contains a string of SVG code and the dimensions of the graphical object. To save the plot into an SVG file, you can use the commands

```
fpath <- paste(vname, ".svg", sep="")
nroFigure(file = fpath, scene = smfig)
```

It is often tedious to create the map plots one-by-one, so we have therefore included the code segment below to allow users to print a large number of colorings automatically:

```
for( vname in colnames(db) ) {

     # Estimate district values.
     plane <- nroAggregate(map = sm, x = db[, vname], bmus = matches$POS)

     # Determine district colors.
     colrs <- nroColorize(values = plane, amplitude = amplitudes[vname])

     # Determine which district labels should be shown.
     labls <- nroLabel(map = sm, values = plane)

     # Create a vector graphics object.
     pval <- stats[vname, "P.z"]
     ttxt <- vname # only the name if training variable
     if(is.na(pval) == FALSE) { # add p-value for evaluation variables
         ttxt <- sprintf("%s, P = %.2e", vname, pval);
     }
     smfig <- nroCircus(map = sm, colors = colrs, labels = labls, title = ttxt)

     # Save figure.
     fpath <- paste(vname, ".svg", sep = "")
     nroFigure(file = fpath, scene = smfig)
}
```

7 Results and interpretation

In the final section, we summarize and discuss the results of the SOM analysis. As previously mentioned, the big open problems in biomedicine and public health are typically characterized by multiple synergistic risk factors that produce a gradual decline in biological functions over time. For this reason, the observed data patterns are not likely to be self-explanatory, but will require additional analyses and contextual assessment with respect to how the original data was collected and what are the clinically impactful findings.

Important note on reproducibility: The *Numero* framework uses optimized code that reduces memory footprint and computational burden. For this reason, different computers, particularly 32-bit vs. 64-bit architectures, may produce map patterns that have been flipped, mirrored, rotated or otherwise transformed when compared with the figures in the vignette. This is a technical limitation due to machine precision, not an unintentional mistake in the code.

7.1 Map quality

Before delving into the characteristics of diabetic kidney disease, it is prudent to examine the SOM for potential problems with the data. The Numero package provides three different quality metrics: i) the data point histogram reveals problems of misrepresentation between the data points and the district profiles, ii) the coverage map shows systematic patterns of missing data that may influence the results, and iii) the matching quality indicates subgroups of data points that may have been modeled poorly.

The data point histogram shows the average number of data points within districts across the map. To calculate it, use the command

```
plane <- nroAggregate(map = sm, bmus = matches$POS)
```

The output format is equivalent to what was used for the map colorings, so the same code sequence is applied to create an SVG figure:

```
colrs <- nroColorize(values = plane, palette = "fire")
labls <- nroLabel(map = sm, values = plane)
smfig <- nroCircus(map = sm, colors = colrs, labels = labls,
                   title = "Data point histogram")
nroFigure(file = "histogram.svg", scene = smfig)
```

To make a distinction between diagnostic and other colorings, we used a different color palette for the *nroColorize()* function. The final plot is shown in Figure 5A. While there were noticeable differences between the districts, there was a sufficient data point count everywhere on the map and, based on our experience from previous studies, it is unlikely that the results were adversely affected due to sparse representation.

The second quality metric reflects the pattern of missing data on the map. The code template is again the same, with minor modifications:

```
nroAggregate(map = sm, bmus = matches$POS, x = matches$COVER)
colrs <- nroColorize(values = plane, palette = "fire")
labls <- nroLabel(map = sm, values = plane)
smfig <- nroCircus(map = sm, colors = colrs, labels = labls,
                   title = "Data coverage")
nroFigure(file = "coverage.svg", scene = smfig)
```

Please note the changes to *nroAggregate()*, title and file name. The results are shown in Figure 5B. The example dataset contained only a few elements with missing values. Therefore, the coverage was close to 100% across the map.

The quality metrics for data point matching helps to identify subsets or individual data points that are not captured by the district profiles. There are two ways to show matching quality, either by coloring the map according to the mean matching errors for districts, or by examining the matching errors of individual data points (also referred to as quantization errors or model residuals). The familiar code template was adapted to produce the coloring:

```
plane <- nroAggregate(map = sm, bmus =  matches$POS, x = matches$QUALITY)
colrs <- nroColorize(values = plane, palette = "fire")
labls <- nroLabel(map = sm, values = plane)
smfig <- nroCircus(map = sm, colors = colrs, labels = labls,
                   title = "Matching quality")
nroFigure(file = "quality.svg", scene = smfig)
```

The results are shown in Figure 5C. Again, some regional differences are expected, but there were no indications of serious problems. In particular, the relative quality even in the worst region was close to the average training quality (i.e. close to one).

To examine the quality of individual data points, we sorted the data frame *matches* according to the QUALITY column to reveal which data points show the lowest matching quality:

```
sorted <- order(matches$QUALITY)
print(matches[sorted[1:20],])
##     POS      DIST   QUALITY COVER
## 420  40 1.6030118 0.1881186   1.0
## 425  40 1.3351857 0.2216711   1.0
## 591  40 0.8182970 0.3380272   1.0
## 136  33 0.5659048 0.4545246   1.0
## 613  31 0.5356713 0.4740970   1.0
## 39   40 0.5325607 0.4762068   1.0
## 74   40 0.5322260 0.4764349   1.0
## 60   23 0.5082468 0.4933682   1.0
## 398  39 0.4935227 0.5043756   1.0
## 147  40 0.4650760 0.5270956   1.0
## 344  35 0.4628206 0.5289848   1.0
## 95   34 0.4623557 0.5293759   1.0
## 165  37 0.4545521 0.5360283   1.0
## 79   35 0.4478536 0.5418734   1.0
## 58   39 0.4434066 0.5458248   1.0
## 426  39 0.4423753 0.5467494   1.0
## 551  40 0.4423045 0.5468130   1.0
## 180  32 0.4283005 0.5596887   1.0
## 66   40 0.4170404 0.5704898   0.8
## 104  23 0.4137421 0.5737330   1.0
```

Data points that show the poorest quality may represent failed measurements or unusual biology, however, it seems that in this example there are no extreme outliers, which suggests that all data points were usable.

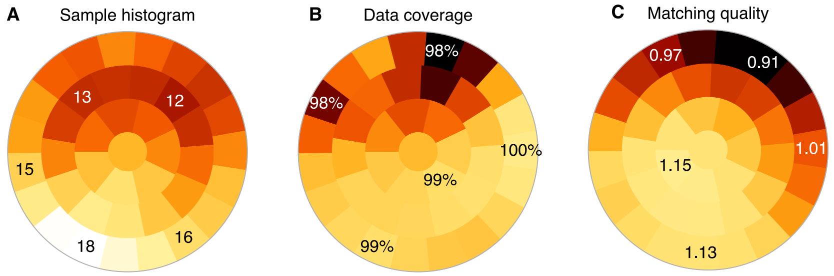


Figure 5: Map quality metrics in the diabetic kidney disease example

7.2 Summary of map colorings

After statistical analyses and visualization, the work folder will be filled with numerous SVG-files. Each of them contains a map coloring. To summarize the raw material into easily legible and clear documentation, we recommend using *Inkscape*, a freely available editing software that uses SVG as its native format.

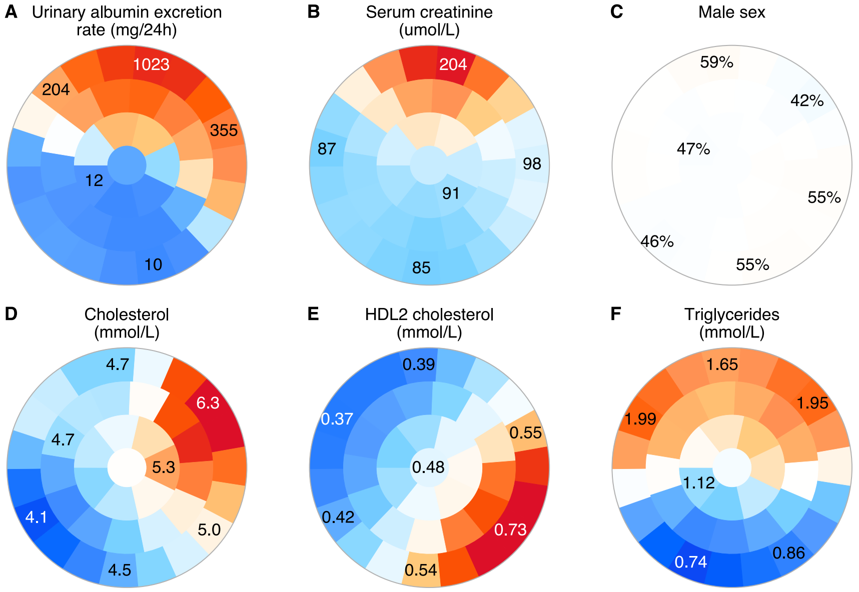


Figure 6: Map colorings for the training set
The colorings for urinary albumin, serum creatinine and triglycerides in Plots A,B and F were created based on the logarithmic values, whereas the numbers show absolute concentrations after reversing the logarithmic transformation.

Figure 6 shows the map colorings for the training set. Urinary albumin was substantially higher for a subset of individuals located on the top part of the map compared to the lower part (1023 mg/24h versus 10 mg/24h, Figure 6A), and a similar pattern was found for serum creatinine (Figure 6B). Therefore, the individuals with the most advanced stages of chronic kidney disease were located in the upper districts on the SOM. Please note also the dramatic difference in the color intensity between the training set (Figure 6A,B,D-F) and male sex (Figure 6C), which indicates that sex variation was successfully eliminated by the adjustments earlier in the pipeline. Specifically, the observed variation of male-female ratios across the map was within the expected range of random fluctuations ($P > 0.05$).

The patterns for the lipids were more complicated. Cholesterol showed a pattern of high concentrations in the upper right area and low concentration in the bottom left (6.3 mmol/L versus 4.1 mmol/L, Figure 6D), whereas HDL2 cholesterol was the highest in the bottom-right and the lowest in the upper-left (0.73 mmol/L versus 0.37 mmol/L, Figure 6E). Triglycerides showed a general pattern of high concentrations on the upper part of the map (Figure 6F). Of note, the coloring was based on the logarithm of triglyceride concentration, whereas the numbers in the plot indicate absolute concentrations.

We recommend using the SOM together with conventional approaches such as linear correlations, for broader understanding of the nature of the dataset. For instance, cholesterol and triglycerides were correlated ($r = 0.43$, $P < 0.001$), however, the SOM colorings suggest that the correlation may not apply to all individuals; particularly those in the upper-left area with high triglycerides did not seem to follow the linear trend (Figure 6E,F). Other dimension reduction methods such as principal component analysis may work better in datasets where there are clear clusters (a typical SOM analysis may miss the clustering structure) and, again, we recommend using multiple conceptually different methods to achieve robust conclusions. We did not observe any obvious clustering structure in the kidney disease dataset (results not shown).

7.3 Subgroup boundaries

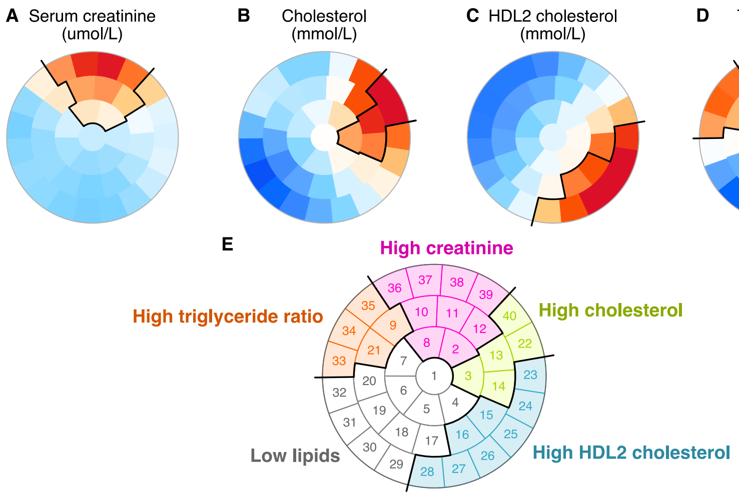


Figure 7: Non-overlapping metabolic subgroups
They were defined according to the regional patterns of the training set. The numbers in Plot E are the unique district identifiers.

The aims of the example study were i) to define and describe metabolic subgroups of type 1 diabetes, and ii) to investigate how the subgroups are associated with mortality. The ability to choose subgroups boundaries on the map while simultaneously observing multiple variables is the main strength of the Numero framework. Furthermore, the intensity of the colorings guides the process towards selecting criteria that have the strongest statistical support. To illustrate the process, we used Figure 6 as the starting point to define interesting subgroups and we then compared the subgroups using standard statistical tests. While we admit that our choices for the subgroup boundaries were subjective, we also argue that any observer can dispute those choices and provide an alternative by examining the figures. Therefore, the transparency of the methodology allows collective objectivity that is superior to strict "black box" classifiers, especially when the data patterns overlap and involve multiple outcomes.

The boundaries that capture the main metabolic features of the study cohort are shown in Figure 7. We started from the region with the highest creatinine (a marker of reduced kidney function), since we expect it to be the most clinically important subgroup with respect to mortality (Figure 7A). Next, we split the region with high total or HDL2 cholesterol into two (Figure 7B,C). Please note that the boundaries may not fit exactly with any specific variable, since we also required that the subgroups have to be mutually exclusive. This is the part where there are no perfect mathematical solutions due to overlaps and multi-morbidity. Lastly, we divided the remaining map area based on the triglyceride concentration (Figure 7D). The final subgrouping with district labels is shown in Figure 7E.

The second aim of the study was to compare the subgroups with respect to mortality and clinical diagnoses. Graphical comparisons of the metabolic subgroup boundaries and selected map colorings are shown in Figure 8. Mortality was the highest in the top section of the map (34% in eight years, Figure 8A), and the same region was also characterized by greater than 90% prevalence of diabetic kidney disease (Figure 8B). As expected the High Creatinine Subgroup captured this segment of the study population. In addition, a few districts with increased mortality (up to 22%) and kidney disease prevalence (up to 72%) were found within the High Cholesterol Subgroup, and similar spill-over was observable also in the High Triglyceride Ratio Subgroup. On the other hand, the Low Lipids Subgroup showed the lowest rates of death or complications across all the plots in Figure 8.

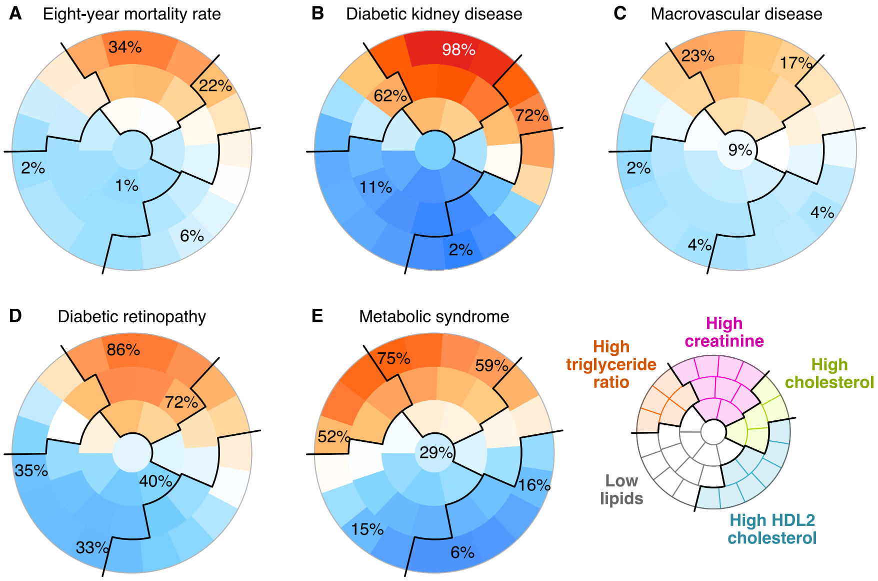


Figure 8: **Overlay of metabolic subgroup boundaries on selected map colorings of clinical endpoints**

The metabolic syndrome is a clinical entity to describe the co-occurrence of obesity, diabetes, high blood pressure and abnormal blood lipids that is often observed in people at risk of cardiovascular death [ref]. Triglycerides and HDL cholesterol comprise the lipid component of the metabolic syndrome, which explains the similar yet different patterns with respect to cardiovascular disease (Figure 8C,E). In particular, over half of the individuals in the High Triglyceride Ratio Subgroup have the metabolic syndrome.

7.4 Subgroup statistics

The district identifiers in Figure 7E enable us to assign the data points to the five metabolic subgroups. Please note that these results have been generated with a 64-bit machine, and they may be different from the results from 32-bit architectures due to the lower machine precision. If you notice problems, please redefine the subgroups yourself and update the *R*-code accordingly.

To select all study participant who were assigned into in the High Creatinine Subgroup, for example, collect the corresponding subgroup districts with the command

```
subgrp <- c(2,8,10,11,12,36,37,38,39)
```

The identifiers can then be combined with the data point layout to find those rows that belong to the High Creatinine Subgroup:

```
rows <- which(match(matches$POS, subgrp) > 0)
```

To show the characteristics of the subgroup on the screen, you can type

```
print(summary(db[rows,]))
##       AGE          T1D_DURAT          MALE           DECEASED
##  Min.   :21.00   Min.   : 5.30   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:33.00   1st Qu.:22.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :40.00   Median :27.00   Median :1.0000   Median :0.0000
##  Mean   :41.14   Mean   :27.42   Mean   :0.5481   Mean   :0.1926
##  3rd Qu.:49.00   3rd Qu.:33.50   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :65.00   Max.   :50.00   Max.   :1.0000   Max.   :1.0000
##
##    MACROVASC       METAB_SYNDR      DIAB_KIDNEY      DIAB_RETINO
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :1.0000   Median :1.0000
##  Mean   :0.1333   Mean   :0.4889   Mean   :0.8741   Mean   :0.7407
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##
##     uALB_log         TG_log             CHOL           HDL2C
##  Min.   :1.944   Min.   :-0.2218   Min.   :4.650   Min.   :0.0910
##  1st Qu.:2.468   1st Qu.: 0.0607   1st Qu.:5.300   1st Qu.:0.4025
##  Median :2.917   Median : 0.1875   Median :5.700   Median :0.5500
##  Mean   :2.882   Mean   : 0.1883   Mean   :5.681   Mean   :0.5429
##  3rd Qu.:3.239   3rd Qu.: 0.3043   3rd Qu.:6.000   3rd Qu.:0.6500
##  Max.   :3.879   Max.   : 0.9085   Max.   :7.200   Max.   :1.1900
##  NA's   :7
##    CREAT_log
##  Min.   :1.415
##  1st Qu.:1.971
##  Median :2.068
##  Mean   :2.125
##  3rd Qu.:2.219
##  Max.   :2.814
##
```

As before, going through all the subgroups and all variables would be tedious, so we prepared a code loop for the subgroup comparisons. The subgroups were copied from Figure 7E:

```
subgroups <- list()
subgroups[["HighCreat"]] <- c(2,8,10:12,36:39)
subgroups[["HighChol"]] <- c(3,13,14,22,40)
subgroups[["HighHDL2"]] <- c(15,16,23:28)
subgroups[["HighTGRatio"]] <- c(9,21,33:35)
```

and the healthiest Low Lipids Subgroup can be used as the control group:

```
subgrp <- c(1,4:7,17:20,29:32)
controls <- which(match(matches$POS, subgrp) > 0)
```

The descriptive statistics for selected variables can now be screened within a loop. We only output a few results as an example:

```
for(name in names(subgroups)) {
    subgrp <- subgroups[[name]]
    rows <- which(match(matches$POS, subgrp) > 0)

    # T-test for continous traits.
    for(vname in c("AGE", "T1D_DURAT")) {
        x <- na.omit(db[controls, vname])
        y <- na.omit(db[rows, vname])
        stats <- t.test(y, x)

        #only print some results
        if(name == "HighCreat" & vname == "T1D_DURAT"){
          cat("\n", name, ", ", vname, ":\n", sep = "")
          cat("Difference:", (mean(y) - mean(x)), "\n")
          cat("CI:", as.double(stats$conf.int), "\n")
          cat("P-value:", stats$p.value, "\n")
        }
    }

    # Fisher's test for binary traits.
    for(vname in c("DECEASED", "DIAB_KIDNEY", "METAB_SYNDR")) {
        x <- na.omit(db[controls, vname])
        y <- na.omit(db[rows, vname])
        bits <- c(0*x, 0*y + 1)
        stats <- fisher.test(c(x, y), bits)
        #only print some results
        if(name == "HighCreat"){
          cat("\n", name, ", ", vname, ":\n", sep="")
          cat("Odds ratio:", stats$estimate, "\n")
          cat("CI:", as.double(stats$conf.int), "\n")
          cat("P-value:", stats$p.value, "\n")
        }
    }
}
##
## HighCreat, T1D_DURAT:
## Difference: 2.807227
## CI: 0.8813819 4.733073
## P-value: 0.004404905
##
## HighCreat, DECEASED:
## Odds ratio: 5.352828
## CI: 2.329739 13.46866
## P-value: 1.194232e-05
##
## HighCreat, DIAB_KIDNEY:
## Odds ratio: 53.28781
## CI: 26.80858 111.6468
## P-value: 1.172956e-48
##
## HighCreat, METAB_SYNDR:
## Odds ratio: 3.854552
## CI: 2.337002 6.423992
## P-value: 1.707418e-08
```

Table 2: Comparison of metabolic subgroups in individuals with type 1 diabetes
For continuous variables, differences with respect to the Low Lipids Subgroup are reported. For binary traits, odds ratios with the Low Lipids Subgroup are reported. In addition, 95% confidence intervals are reported for the descriptive statistics. P-values were estimated by the t-test for continuous variables and by the Fisher's test for binary traits.

Variable	Subgroup	Difference or odds ratio	P-value
Deceased at follow-up	High creatinine	15.2 (5.6, 51.8)	<0.001
	High cholesterol	5.6 (1.5, 23.3)	0.0049
	High HDL2 cholesterol	3.5 (1.1, 13.3)	0.027
	High triglyceride ratio	2.7 (0.5, 12.8)	0.22
Diabetic kidney disease	High creatinine	159 (62, 486)	<0.001
	High cholesterol	8.8 (4.4, 18.1)	<0.001
	High HDL2 cholesterol	2.9 (1.6, 5.5)	<0.001
	High triglyceride ratio	5.2 (2.6, 10.5)	<0.001
Metabolic syndrome	High creatinine	5.9 (3.5, 9.9)	<0.001
	High cholesterol	3.1 (1.7, 5.9)	<0.001
	High HDL2 cholesterol	0.27 (0.11, 0.58)	<0.001
	High triglyceride ratio	8.9 (4.7, 17.2)	<0.001

Selected findings are listed in Table 2. As expected, the High Creatinine Subgroup had the highest odds ratio for dying within the follow-up period (OR 15.2, $P < 0.001$) and the highest for diabetic kidney disease (OR 159, $P < 0.001$) when compared against the Low Lipids Subgroup. However, the prevalence of the metabolic syndrome was the highest in the High Triglyceride Ratio Subgroup (OR 8.9, $P < 0.001$), albeit the confidence intervals overlapped with the High Creatinine Subgroup. This partially shared pattern was also observed in the original study in 2008 (V.-P. Mäkinen et al., 2008b).

8 Concluding remarks

Now that the SOM analyses have been completed, how should these findings be reported in a journal article, and what is the take-home message? Our first recommendation is not to abandon conventional statistics when using the *Numero* framework – the two are complementary. In the kidney disease example, we recommend starting with the description of the study cohort and age- and sex-adjusted comparisons between established clinical categories (e.g. Table 1 is a basic first step). This will give most readers in the field an understanding of the basic nature of the dataset.

Next, we recommend drawing Kaplan-Meier mortality curves for diabetic kidney disease, retinopathy, and metabolic syndrome, and apply Cox regression to investigate associations with mortality in a multivariate context (or other well established statistical methods). Again, the biomedical readership will appreciate using methodology that is familiar to them. These analyses work best for datasets with only a few variables and a well-defined hypothesis, but they are not well suited to identifying non-linear subgroups, synergies across a high number of variables, or multi-morbidity from several correlated yet diverse clinical end-points. Hence the machine learning audience will probably want more.

The third section of the article should involve the SOM to identify features that cannot be detected by the standard tools. Even if nothing new was discovered, we still recommend adding the SOM as a supplement, since it gives a comprehensive window into the data, and it is particularly useful to detect non-random patterns of missing data, the effects of censoring in longitudinal studies and opportunities to detect outliers. For readers and reviewers, sophisticated visualizations that capture the nature of the cohort and give an accurate description of the structural strengths and weaknesses will be highly appreciated – it is better science.

As to the take-home message, we propose the following: if people with type 1 diabetes can achieve such metabolic control that their serum lipids are low, they are likely to be resilient against diabetic complications and mortality. We and others have made similar observations before, so this is not novel, however, it shows how the SOM lead to the expected conclusions and it gives us confidence that the approach is robust for high-dimensional big data that is out of reach of conventional tools.

References

Attik, M., Bougrain, L., & Alexandre, F. (2005). Self-organizing map initialization. In W. Duch, J. Kacprzyk, E. Oja, & S. Zadrożny (Eds.), *Artificial neural networks: Biological inspirations – icann 2005: 15th international conference, warsaw, poland, september 11-15, 2005. proceedings, part i* (pp. 357–362). doi:10.1007/11550822_56

Bernardi, L., De Barbieri, G., Rosengård-Bärlund, M., Mäkinen, V.-P., Porta, C., & Groop, P.-H. (2010). New method to measure and improve consistency of baroreflex sensitivity values. *Clinical Autonomic Research*, *20*(6), 353–361. doi:10.1007/s10286-010-0079-1

Cardoso, D. B. O. S., Queiroz, L. P. de, & Lima, H. C. de. (2014). A taxonomic revision of the south american papilionoid genus luetzelburgia (fabaceae). *Botanical Journal of the Linnean Society*, *175*(3), 328–375. doi:10.1111/boj.12153

Delanaye, P., Schaeffner, E., Ebert, N., Cavalier, E., Mariat, C., Krzesinski, J.-M., & Moranne, O. (2012). Normal reference values for glomerular filtration rate: What do we really know? *Nephrology Dialysis Transplantation*, *27*(7), 2664–2672. doi:10.1093/ndt/gfs265

Kohonen, T., Schroeder, M. R., & Huang, T. S. (Eds.). (2001). *Self-organizing maps* (3rd ed.). Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Kumpula, L. S., Makela, S. M., Mäkinen, V.-P., Karjalainen, A., Liinamaa, J. M., Kaski, K., . . . Ala-Korpela, M. (2010). Characterization of metabolic interrelationships and in silico phenotyping of lipoprotein particles using self-organizing maps. *The Journal of Lipid Research*, *51*(2), 431–439. doi:10.1194/jlr.D000760

Kuusisto, S. M., Peltola, T., Laitinen, M., Kumpula, L. S., Mäkinen, V.-P., Salonurmi, T., . . . Ala-Korpela, M. (2012). The interplay between lipoprotein phenotypes, adiponectin, and alcohol consumption. *Annals of Medicine*, *44*(5), 513–522. doi:10.3109/07853890.2011.611529

Levey, A. S., & Coresh, J. (2012). Chronic kidney disease. *The Lancet*, *379*(9811), 165–180. doi:10.1016/S0140-6736(11)60178-5

Mäkinen, V.-P., Forsblom, C., Thorn, L. M., Waden, J., Gordin, D., Heikkila, O., . . . on behalf of the FinnDiane Study Group. (2008a). Metabolic Phenotypes, Vascular Complications, and Premature Deaths in a Population of 4,197 Patients With Type 1 Diabetes. *Diabetes*, *57*(9), 2480–2487. doi:10.2337/db08-0332

Mäkinen, V.-P., Soininen, P., Forsblom, C., Parkkonen, M., Ingman, P., Kaski, K., . . . Ala-Korpela, M. (2008b). 1H NMR metabonomics approach to the disease continuum of diabetic complications and premature death. *Molecular Systems Biology*, *4*. doi:10.1038/msb4100205

Mäkinen, V.-P., Soininen, P., Kangas, A. J., Forsblom, C., Tolonen, N., Thorn, L. M., . . . Finnish Diabetic Nephropathy Study Group. (2013). Triglyceride-cholesterol imbalance across lipoprotein subclasses predicts diabetic kidney disease and mortality in type 1 diabetes: The FinnDiane Study. *Journal of Internal Medicine*, *273*(4), 383–395. doi:10.1111/joim.12026

Mäkinen, V.-P., Tynkkynen, T., Soininen, P., Peltola, T., Kangas, A. J., Forsblom, C., . . . Groop, P.-H. (2012). Metabolic Diversity of Progressive Kidney Disease in 325 Patients with Type 1 Diabetes (the FinnDiane Study). *Journal of Proteome Research*, *11*(3), 1782–1790. doi:10.1021/pr201036j

Tukiainen, T., Tynkkynen, T., Mäkinen, V.-P., Jylänki, P., Kangas, A., Hokkanen, J., . . . Ala-Korpela, M. (2008). A multi-metabolite analysis of serum by 1H NMR spectroscopy: Early systemic signs of Alzheimer's disease. *Biochemical and Biophysical Research Communications*, *375*(3), 356–361. doi:10.1016/j.bbrc.2008.08.007

Würtz, P., Soininen, P., Kangas, A. J., Mäkinen, V.-P., Groop, P.-H., Savolainen, M. J., . . . Ala-Korpela, M. (2011). Characterization of systemic metabolic phenotypes associated with subclinical atherosclerosis. *Mol. BioSyst.*, *7*(2), 385–393. doi:10.1039/C0MB00066C