Creating a Multipollutant Exposure Metric with sommix

John L. Pearce

August 31, 2020

Load package dependencies

library(kohonen)#self-organizing map algorithim  
library(MASS)#Sammons Mapping  
library(e1071) #k-nearest neighbor class assignments  
library(fpc) #Cluster validation statistics  
library(cluster)  
library(colorspace) #Color options  
library(spdep) #Spatial Statistics  
library(corrplot) #Correllogram

Load sommix from local repository

devtools::load\_all(".")

Load example data

data(data2)

## Overview

This vignette illustrates how the R package sommix can be used for development of exposure indices that improve understanding of co-exposure patterning and serve as a multipollutant exposure metric in subsequent study of health effects. The objective of this documentation is to illustrate the functionalities within the sommix package using an established synthetic data example. Please send any comments or suggestions to [pearcejo@musc.edu](mailto:pearcejo@musc.edu).

## Self-Organizing Maps

SOM is an artificial neural network (ANN) that applies competitive learning with an integrated neighborhood function in order to discover and ‘map’ representative features within data sets (Kohonen 2013). The ‘map’ is a low-dimensional representation that is used to illustrate discovered profiles in a spatially organized way. This unique ‘self-organizing’ feature ensures that the resulting mapping provides an arrangement of categories where neighboring profiles are similar. This allows for a larger number of classes to be more easily understood, offering users the ability to represent exposures as a high-resolution continuum of categorizations, a key distinction from the discrete clusters/factors often produced through more traditional methods (Austin, Coull et al. 2012, Agay-Shay, Martinez et al. 2015, Berg, Nost et al. 2017).

## Data

sommix includes data from a recent NIEHS Mixtures Workshop, which was obtained here: <https://www.niehs.nih.gov/about/events/pastmtg/2015/statistical/index.cfm>

Here, our example will be based on Data Set 2 as is provides mixture simulated data with an environmentally relevant complex correlation pattern. This data file is intended to represent data from a cross-sectional study of 14 biomarkers (e.g., PCBs, dioxins, furans) from biomonitoring data potentially associated with a biomarker of effect (e.g., ALT as a marker of liver toxicity). Three covariates are also included, one binary and two continuous. Specifics are provided below.

Number of records: 500 Variables: Y, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, Z1, Z2, Z3 Y = The outcome, a continuous variable. X1 - X14 = Fourteen exposure biomarkers. Each is a continuous variable. Z1-Z2: Potential confounders that are continuous. Z3: A potential confounder that is binary.

summary(data2$X1)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.1774 0.5392 1.0303 1.0234 1.5003 2.8881

## Selecting Training Data

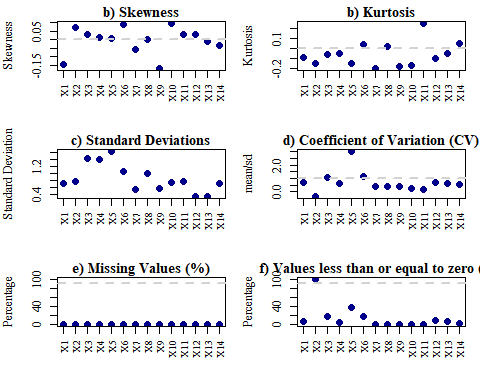
Learning methodologies within sommix center on the exposure matrix and thus we subset our training data to only include exposure variables.

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.1774 0.5392 1.0303 1.0234 1.5003 2.8881

## The first step of our analysis is to assess our variables independently.

The var.eval() function is designed to enhance understanding of the distribution of each variable in your data, the relative variability, and the proportion of missing and zero values.

var.eval(data.trn)



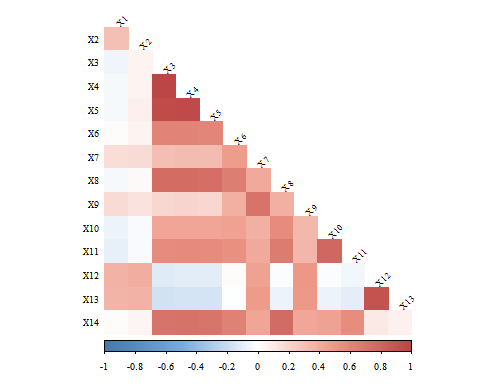
## VARIABLE N AVG SD MN Q1 MED Q3 MX  
## X1 X1 500 1.02344 0.69935 -1.17736 0.53919 1.03029 1.50030 2.88810  
## X2 X2 500 -2.12069 0.76380 -4.32176 -2.63082 -2.13741 -1.58591 0.19261  
## X3 X3 500 1.32455 1.41453 -2.32918 0.40810 1.33222 2.30523 5.43346  
## X4 X4 500 2.34760 1.39365 -1.36095 1.42284 2.32149 3.33643 6.36153  
## X5 X5 500 0.53662 1.62139 -4.21636 -0.50924 0.50358 1.74140 4.84849  
## X6 X6 500 0.89094 1.04041 -2.12254 0.22553 0.83286 1.55046 3.75578  
## X7 X7 500 1.32104 0.55167 -0.35618 0.96743 1.32698 1.69157 2.93515  
## X8 X8 500 2.69835 0.99938 -0.26837 2.02689 2.69163 3.34416 5.91717  
## X9 X9 500 1.32355 0.56422 -0.32806 0.94584 1.33902 1.75688 2.97870  
## X10 X10 500 3.14123 0.74826 1.07168 2.62353 3.13492 3.64544 5.26064  
## X11 X11 500 5.18841 0.76868 2.62432 4.66439 5.21260 5.69728 7.80171  
## X12 X12 500 0.48056 0.34245 -0.48087 0.23698 0.48274 0.70849 1.49435  
## X13 X13 500 0.55532 0.34864 -0.37097 0.31474 0.55198 0.79976 1.64925  
## X14 X14 500 1.31182 0.71880 -1.28514 0.83152 1.30958 1.82559 3.55268  
## IR CV CVrank  
## X1 0.96111 0.6833327 5  
## X2 1.04491 -0.3601658 14  
## X3 1.89712 1.0679325 3  
## X4 1.91359 0.5936488 7  
## X5 2.25064 3.0214863 1  
## X6 1.32493 1.1677666 2  
## X7 0.72414 0.4176028 10  
## X8 1.31728 0.3703671 11  
## X9 0.81104 0.4262929 9  
## X10 1.02190 0.2382061 12  
## X11 1.03289 0.1481533 13  
## X12 0.47151 0.7126061 4  
## X13 0.48501 0.6278182 6  
## X14 0.99407 0.5479410 8

## The second step of our analysis is to assess correlation among our variables.

The cor.eval() function is designed to enhance understanding of correlation among our variables.

cor.eval(data.trn)

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13  
## X1 NA NA NA NA NA NA NA NA NA NA NA NA NA  
## X2 0.31 NA NA NA NA NA NA NA NA NA NA NA NA  
## X3 -0.05 0.05 NA NA NA NA NA NA NA NA NA NA NA  
## X4 -0.04 0.06 0.99 NA NA NA NA NA NA NA NA NA NA  
## X5 -0.04 0.08 0.94 0.95 NA NA NA NA NA NA NA NA NA  
## X6 0.02 0.05 0.61 0.62 0.60 NA NA NA NA NA NA NA NA  
## X7 0.17 0.17 0.32 0.32 0.32 0.47 NA NA NA NA NA NA NA  
## X8 -0.03 0.03 0.76 0.76 0.74 0.66 0.42 NA NA NA NA NA NA  
## X9 0.17 0.14 0.20 0.20 0.19 0.38 0.72 0.38 NA NA NA NA NA  
## X10 -0.07 -0.02 0.43 0.44 0.43 0.45 0.39 0.58 0.34 NA NA NA NA  
## X11 -0.08 -0.03 0.58 0.59 0.57 0.55 0.42 0.67 0.35 0.79 NA NA NA  
## X12 0.38 0.39 -0.11 -0.10 -0.10 0.02 0.45 -0.01 0.50 -0.01 -0.05 NA NA  
## X13 0.36 0.38 -0.16 -0.15 -0.16 -0.01 0.48 -0.07 0.50 -0.06 -0.10 0.91 NA  
## X14 0.02 0.05 0.72 0.72 0.71 0.63 0.44 0.76 0.42 0.45 0.56 0.11 0.06  
## X14  
## X1 NA  
## X2 NA  
## X3 NA  
## X4 NA  
## X5 NA  
## X6 NA  
## X7 NA  
## X8 NA  
## X9 NA  
## X10 NA  
## X11 NA  
## X12 NA  
## X13 NA  
## X14 NA  
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



## Scaling Training Data

The third step of our analysis is to standardize our data as learning methodologies within sommix often perform best when data are on the same scale.

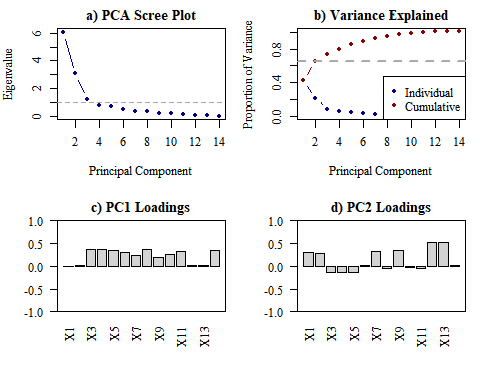
data.trn.sc<-scale(data.trn, center=TRUE, scale=TRUE)  
summary(data.trn.sc[,1])

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.146920 -0.692425 0.009799 0.000000 0.681863 2.666282

## The fourth step of our analysis is to assess variance-covariance structure among our variables with principal component analysis (PCA).

The pca.eval() function is designed to enhance understanding of variance-covariance structure among our variables. The function relies on prcomp().

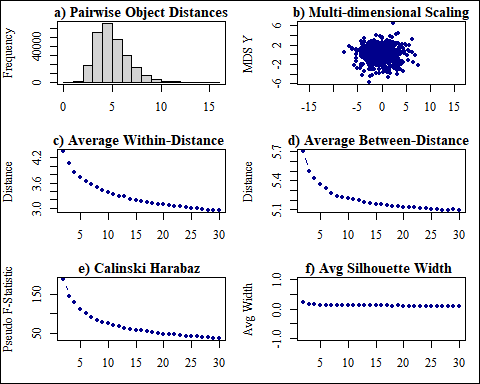
pca.eval(data.trn.sc)



## The fifth step of our analysis is to assess grouping structure among our variables using a variety of strategies applied in cluster analysis.

The grp.eval() function is designed to enhance understanding of grouping structure among our variables using a variety of cluster analysis strategies.

grp.eval(data.trn.sc, kmn=2, kmx=30, iter.max=1000)



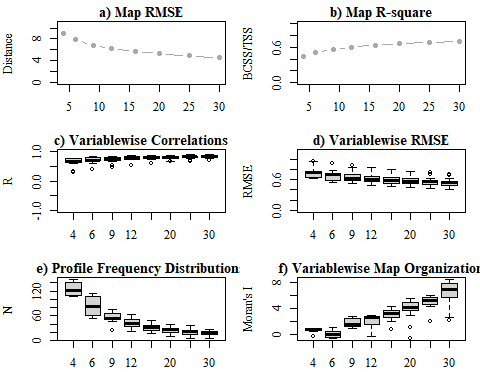
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## K WITHIN BETWEEN CalH SilW  
## 1 2 4.337725 5.697546 187.78531 0.22255388  
## 2 3 4.056800 5.495225 145.15619 0.16742379  
## 3 4 3.856194 5.421961 128.18963 0.16004185  
## 4 5 3.733166 5.361670 111.58470 0.14238664  
## 5 6 3.635727 5.323883 100.23241 0.13400493  
## 6 7 3.554429 5.268378 91.27587 0.12930363  
## 7 8 3.495191 5.242857 83.46469 0.11859484  
## 8 9 3.428560 5.232960 78.80190 0.12512296  
## 9 10 3.370056 5.218312 74.70195 0.12420185  
## 10 11 3.329894 5.205133 70.43022 0.12168506  
## 11 12 3.284923 5.192650 67.05985 0.12125732  
## 12 13 3.270203 5.177876 62.38521 0.10950740  
## 13 14 3.215447 5.163086 60.85605 0.11342740  
## 14 15 3.189289 5.157235 58.34285 0.11688370  
## 15 16 3.165478 5.155735 55.93325 0.11426736  
## 16 17 3.139885 5.148097 53.92461 0.11290208  
## 17 18 3.120063 5.137583 51.71941 0.10774506  
## 18 19 3.093168 5.135467 50.40495 0.11105495  
## 19 20 3.084239 5.126361 47.89210 0.10767913  
## 20 21 3.062838 5.124600 46.72993 0.10590632  
## 21 22 3.046117 5.123705 45.52236 0.10731887  
## 22 23 3.047307 5.111817 43.21028 0.09391259  
## 23 24 3.017212 5.114295 42.76357 0.10315816  
## 24 25 2.992656 5.107767 42.02680 0.09910739  
## 25 26 2.984856 5.105528 40.67028 0.09682229  
## 26 27 2.968410 5.099509 39.67102 0.09557427  
## 27 28 2.959214 5.096736 38.74834 0.09519755  
## 28 29 2.953780 5.100003 37.58920 0.09169040  
## 29 30 2.948627 5.095167 36.50890 0.08947902

## The six step of our analysis is to assess how differing map sizes influence our exposure metric.

The map.size.eval() function is designed to enhance understanding of different map sizes can be used to capture different characteristics of our exposure data. In this example, we seek to build an exposure index that has the benefits of both PCA and clustering approaches. More specifically, we desire an exposure index that explains variability as well as PCA but retains the interpretability of clustering profiles. To achieve, we examine our prior outputs and see that the first two PCs explain ~60% of the variabity in our data, which we set as a target for our SOM. Examination of our group structure evaluation reveals little evidence of structuring and thus we focus on Average within-distance as this gives us the best indication of class cohesiveness.

map.size.eval(data.trn.sc, kmx=30, nstart=5, iter.max=5000)



## $MOD.FIT  
## SOMX SOMY k R2 QE RMSE  
## 1 2 2 4 0.4370145 7.895703 8.972168  
## 2 3 2 6 0.5035446 6.967612 7.855646  
## 3 3 3 9 0.5634162 6.119136 6.852419  
## 4 4 3 12 0.6031507 5.564973 6.182030  
## 5 4 4 16 0.6365041 5.103149 5.713014  
## 6 5 4 20 0.6591661 4.779316 5.303524  
## 7 5 5 25 0.6835375 4.442470 4.874659  
## 8 6 5 30 0.7004984 4.203967 4.619706  
##   
## $VAR.FIT  
## VAR.NAM SOMX SOMY k R.P RMSE.P MAE.P  
## 1 X1 2 2 4 0.32 0.9472 0.7668  
## 2 X2 2 2 4 0.31 0.9516 0.7621  
## 3 X3 2 2 4 0.78 0.6210 0.4946  
## 4 X4 2 2 4 0.79 0.6177 0.4884  
## 5 X5 2 2 4 0.78 0.6258 0.4919  
## 6 X6 2 2 4 0.66 0.7516 0.6071  
## 7 X7 2 2 4 0.67 0.7428 0.5934  
## 8 X8 2 2 4 0.76 0.6441 0.5188  
## 9 X9 2 2 4 0.65 0.7620 0.6020  
## 10 X10 2 2 4 0.59 0.8070 0.6485  
## 11 X11 2 2 4 0.68 0.7316 0.5724  
## 12 X12 2 2 4 0.64 0.7656 0.6115  
## 13 X13 2 2 4 0.65 0.7608 0.5908  
## 14 X14 2 2 4 0.73 0.6861 0.5504  
## 15 X1 3 2 6 0.39 0.9194 0.7442  
## 16 X2 3 2 6 0.39 0.9199 0.7368  
## 17 X3 3 2 6 0.84 0.5433 0.4279  
## 18 X4 3 2 6 0.84 0.5403 0.4216  
## 19 X5 3 2 6 0.82 0.5681 0.4514  
## 20 X6 3 2 6 0.71 0.7002 0.5506  
## 21 X7 3 2 6 0.71 0.7063 0.5579  
## 22 X8 3 2 6 0.81 0.5844 0.4698  
## 23 X9 3 2 6 0.70 0.7119 0.5700  
## 24 X10 3 2 6 0.62 0.7868 0.6324  
## 25 X11 3 2 6 0.71 0.7046 0.5598  
## 26 X12 3 2 6 0.71 0.7091 0.5624  
## 27 X13 3 2 6 0.70 0.7164 0.5690  
## 28 X14 3 2 6 0.78 0.6301 0.5056  
## 29 X1 3 3 9 0.53 0.8460 0.6765  
## 30 X2 3 3 9 0.47 0.8815 0.7103  
## 31 X3 3 3 9 0.85 0.5212 0.4147  
## 32 X4 3 3 9 0.86 0.5181 0.4066  
## 33 X5 3 3 9 0.83 0.5538 0.4345  
## 34 X6 3 3 9 0.72 0.6979 0.5474  
## 35 X7 3 3 9 0.75 0.6563 0.5247  
## 36 X8 3 3 9 0.82 0.5777 0.4720  
## 37 X9 3 3 9 0.74 0.6726 0.5390  
## 38 X10 3 3 9 0.68 0.7335 0.5814  
## 39 X11 3 3 9 0.77 0.6406 0.5003  
## 40 X12 3 3 9 0.79 0.6148 0.4819  
## 41 X13 3 3 9 0.80 0.6022 0.4852  
## 42 X14 3 3 9 0.78 0.6217 0.4960  
## 43 X1 4 3 12 0.55 0.8351 0.6748  
## 44 X2 4 3 12 0.55 0.8314 0.6620  
## 45 X3 4 3 12 0.87 0.4880 0.3888  
## 46 X4 4 3 12 0.88 0.4773 0.3790  
## 47 X5 4 3 12 0.86 0.5032 0.4020  
## 48 X6 4 3 12 0.74 0.6733 0.5329  
## 49 X7 4 3 12 0.77 0.6344 0.5026  
## 50 X8 4 3 12 0.83 0.5570 0.4505  
## 51 X9 4 3 12 0.76 0.6541 0.5270  
## 52 X10 4 3 12 0.74 0.6689 0.5406  
## 53 X11 4 3 12 0.80 0.5977 0.4695  
## 54 X12 4 3 12 0.81 0.5917 0.4566  
## 55 X13 4 3 12 0.81 0.5879 0.4659  
## 56 X14 4 3 12 0.80 0.6024 0.4791  
## 57 X1 4 4 16 0.60 0.7975 0.6247  
## 58 X2 4 4 16 0.62 0.7808 0.6199  
## 59 X3 4 4 16 0.89 0.4591 0.3593  
## 60 X4 4 4 16 0.89 0.4581 0.3599  
## 61 X5 4 4 16 0.88 0.4809 0.3774  
## 62 X6 4 4 16 0.75 0.6632 0.5273  
## 63 X7 4 4 16 0.79 0.6076 0.4879  
## 64 X8 4 4 16 0.84 0.5469 0.4402  
## 65 X9 4 4 16 0.76 0.6483 0.5170  
## 66 X10 4 4 16 0.77 0.6400 0.5119  
## 67 X11 4 4 16 0.80 0.5967 0.4693  
## 68 X12 4 4 16 0.85 0.5276 0.4154  
## 69 X13 4 4 16 0.84 0.5375 0.4223  
## 70 X14 4 4 16 0.81 0.5869 0.4660  
## 71 X1 5 4 20 0.66 0.7539 0.5898  
## 72 X2 5 4 20 0.69 0.7226 0.5668  
## 73 X3 5 4 20 0.89 0.4591 0.3629  
## 74 X4 5 4 20 0.89 0.4576 0.3597  
## 75 X5 5 4 20 0.88 0.4737 0.3659  
## 76 X6 5 4 20 0.77 0.6418 0.5119  
## 77 X7 5 4 20 0.80 0.6024 0.4812  
## 78 X8 5 4 20 0.85 0.5282 0.4284  
## 79 X9 5 4 20 0.78 0.6285 0.4974  
## 80 X10 5 4 20 0.81 0.5909 0.4716  
## 81 X11 5 4 20 0.82 0.5787 0.4563  
## 82 X12 5 4 20 0.83 0.5533 0.4384  
## 83 X13 5 4 20 0.85 0.5336 0.4171  
## 84 X14 5 4 20 0.82 0.5661 0.4534  
## 85 X1 5 5 25 0.67 0.7404 0.5817  
## 86 X2 5 5 25 0.70 0.7113 0.5631  
## 87 X3 5 5 25 0.91 0.4243 0.3418  
## 88 X4 5 5 25 0.91 0.4191 0.3335  
## 89 X5 5 5 25 0.89 0.4546 0.3628  
## 90 X6 5 5 25 0.78 0.6197 0.4924  
## 91 X7 5 5 25 0.82 0.5739 0.4593  
## 92 X8 5 5 25 0.85 0.5189 0.4147  
## 93 X9 5 5 25 0.81 0.5841 0.4698  
## 94 X10 5 5 25 0.81 0.5841 0.4657  
## 95 X11 5 5 25 0.83 0.5546 0.4410  
## 96 X12 5 5 25 0.86 0.5158 0.4110  
## 97 X13 5 5 25 0.85 0.5184 0.4173  
## 98 X14 5 5 25 0.83 0.5645 0.4454  
## 99 X1 6 5 30 0.71 0.7041 0.5527  
## 100 X2 6 5 30 0.72 0.6915 0.5553  
## 101 X3 6 5 30 0.91 0.4110 0.3322  
## 102 X4 6 5 30 0.91 0.4059 0.3213  
## 103 X5 6 5 30 0.90 0.4305 0.3320  
## 104 X6 6 5 30 0.78 0.6267 0.4990  
## 105 X7 6 5 30 0.84 0.5473 0.4308  
## 106 X8 6 5 30 0.86 0.5169 0.4170  
## 107 X9 6 5 30 0.82 0.5670 0.4488  
## 108 X10 6 5 30 0.82 0.5662 0.4527  
## 109 X11 6 5 30 0.83 0.5506 0.4328  
## 110 X12 6 5 30 0.86 0.5087 0.4098  
## 111 X13 6 5 30 0.87 0.4953 0.3916  
## 112 X14 6 5 30 0.83 0.5499 0.4432  
##   
## $MAP.FREQ  
## SOMX SOMY k N FREQ  
## 1 2 2 4 108 21.6  
## 2 2 2 4 134 26.8  
## 3 2 2 4 147 29.4  
## 4 2 2 4 111 22.2  
## 5 3 2 6 87 17.4  
## 6 3 2 6 53 10.6  
## 7 3 2 6 78 15.6  
## 8 3 2 6 115 23.0  
## 9 3 2 6 60 12.0  
## 10 3 2 6 107 21.4  
## 11 3 3 9 52 10.4  
## 12 3 3 9 56 11.2  
## 13 3 3 9 24 4.8  
## 14 3 3 9 51 10.2  
## 15 3 3 9 76 15.2  
## 16 3 3 9 75 15.0  
## 17 3 3 9 54 10.8  
## 18 3 3 9 66 13.2  
## 19 3 3 9 46 9.2  
## 20 4 3 12 22 4.4  
## 21 4 3 12 30 6.0  
## 22 4 3 12 41 8.2  
## 23 4 3 12 25 5.0  
## 24 4 3 12 56 11.2  
## 25 4 3 12 41 8.2  
## 26 4 3 12 63 12.6  
## 27 4 3 12 60 12.0  
## 28 4 3 12 45 9.0  
## 29 4 3 12 37 7.4  
## 30 4 3 12 40 8.0  
## 31 4 3 12 40 8.0  
## 32 4 4 16 48 9.6  
## 33 4 4 16 17 3.4  
## 34 4 4 16 20 4.0  
## 35 4 4 16 33 6.6  
## 36 4 4 16 24 4.8  
## 37 4 4 16 41 8.2  
## 38 4 4 16 33 6.6  
## 39 4 4 16 28 5.6  
## 40 4 4 16 25 5.0  
## 41 4 4 16 27 5.4  
## 42 4 4 16 32 6.4  
## 43 4 4 16 36 7.2  
## 44 4 4 16 21 4.2  
## 45 4 4 16 37 7.4  
## 46 4 4 16 29 5.8  
## 47 4 4 16 49 9.8  
## 48 5 4 20 28 5.6  
## 49 5 4 20 30 6.0  
## 50 5 4 20 29 5.8  
## 51 5 4 20 32 6.4  
## 52 5 4 20 29 5.8  
## 53 5 4 20 17 3.4  
## 54 5 4 20 30 6.0  
## 55 5 4 20 23 4.6  
## 56 5 4 20 31 6.2  
## 57 5 4 20 16 3.2  
## 58 5 4 20 28 5.6  
## 59 5 4 20 23 4.6  
## 60 5 4 20 39 7.8  
## 61 5 4 20 35 7.0  
## 62 5 4 20 18 3.6  
## 63 5 4 20 9 1.8  
## 64 5 4 20 23 4.6  
## 65 5 4 20 20 4.0  
## 66 5 4 20 25 5.0  
## 67 5 4 20 15 3.0  
## 68 5 5 25 11 2.2  
## 69 5 5 25 15 3.0  
## 70 5 5 25 7 1.4  
## 71 5 5 25 36 7.2  
## 72 5 5 25 6 1.2  
## 73 5 5 25 16 3.2  
## 74 5 5 25 26 5.2  
## 75 5 5 25 24 4.8  
## 76 5 5 25 25 5.0  
## 77 5 5 25 19 3.8  
## 78 5 5 25 18 3.6  
## 79 5 5 25 25 5.0  
## 80 5 5 25 26 5.2  
## 81 5 5 25 26 5.2  
## 82 5 5 25 28 5.6  
## 83 5 5 25 23 4.6  
## 84 5 5 25 24 4.8  
## 85 5 5 25 25 5.0  
## 86 5 5 25 13 2.6  
## 87 5 5 25 25 5.0  
## 88 5 5 25 15 3.0  
## 89 5 5 25 23 4.6  
## 90 5 5 25 19 3.8  
## 91 5 5 25 13 2.6  
## 92 5 5 25 12 2.4  
## 93 6 5 30 12 2.4  
## 94 6 5 30 17 3.4  
## 95 6 5 30 19 3.8  
## 96 6 5 30 8 1.6  
## 97 6 5 30 14 2.8  
## 98 6 5 30 18 3.6  
## 99 6 5 30 13 2.6  
## 100 6 5 30 16 3.2  
## 101 6 5 30 26 5.2  
## 102 6 5 30 16 3.2  
## 103 6 5 30 22 4.4  
## 104 6 5 30 6 1.2  
## 105 6 5 30 17 3.4  
## 106 6 5 30 15 3.0  
## 107 6 5 30 23 4.6  
## 108 6 5 30 21 4.2  
## 109 6 5 30 24 4.8  
## 110 6 5 30 14 2.8  
## 111 6 5 30 18 3.6  
## 112 6 5 30 23 4.6  
## 113 6 5 30 23 4.6  
## 114 6 5 30 24 4.8  
## 115 6 5 30 18 3.6  
## 116 6 5 30 5 1.0  
## 117 6 5 30 15 3.0  
## 118 6 5 30 11 2.2  
## 119 6 5 30 22 4.4  
## 120 6 5 30 12 2.4  
## 121 6 5 30 14 2.8  
## 122 6 5 30 14 2.8  
##   
## $MAP.STRUCTURE  
## VAR.NAM SOMX SOMY k MI MI\_P GC GC\_Pval  
## 1 X1 2 2 4 -0.3194838 6.253202e-01 -0.7437821 7.714959e-01  
## 2 X2 2 2 4 0.8921535 1.861553e-01 0.9940883 1.600899e-01  
## 3 X3 2 2 4 0.8348286 2.019071e-01 0.6585096 2.551054e-01  
## 4 X4 2 2 4 0.8337920 2.021991e-01 0.6443772 2.596654e-01  
## 5 X5 2 2 4 0.8359100 2.016028e-01 0.6597356 2.547118e-01  
## 6 X6 2 2 4 0.8574729 1.955918e-01 0.5121380 3.042772e-01  
## 7 X7 2 2 4 0.4131069 3.397641e-01 0.0830636 4.669005e-01  
## 8 X8 2 2 4 0.9005457 1.839149e-01 0.5991886 2.745236e-01  
## 9 X9 2 2 4 0.6760665 2.494992e-01 0.5782562 2.815456e-01  
## 10 X10 2 2 4 0.9586177 1.688757e-01 0.6152343 2.692000e-01  
## 11 X11 2 2 4 0.9084630 1.818168e-01 0.5770026 2.819688e-01  
## 12 X12 2 2 4 0.6173988 2.684859e-01 0.3972151 3.456044e-01  
## 13 X13 2 2 4 0.7458015 2.278937e-01 0.5069892 3.060812e-01  
## 14 X14 2 2 4 0.9287561 1.765078e-01 0.6090386 2.712494e-01  
## 15 X1 3 2 6 1.0858674 1.387688e-01 0.8945196 1.855220e-01  
## 16 X2 3 2 6 0.4043386 3.429819e-01 0.3261572 3.721527e-01  
## 17 X3 3 2 6 0.4134513 3.396380e-01 -0.3270540 6.281865e-01  
## 18 X4 3 2 6 0.4275969 3.344723e-01 -0.3247920 6.273308e-01  
## 19 X5 3 2 6 0.4386439 3.304598e-01 -0.3329555 6.304161e-01  
## 20 X6 3 2 6 -0.2386132 5.942972e-01 -0.8148400 7.924181e-01  
## 21 X7 3 2 6 -0.5963415 7.245264e-01 -0.6960778 7.568099e-01  
## 22 X8 3 2 6 -0.2322350 5.918222e-01 -0.7640945 7.775945e-01  
## 23 X9 3 2 6 -0.5273966 7.010409e-01 -0.3306701 6.295531e-01  
## 24 X10 3 2 6 -0.3899211 6.517026e-01 -0.9625340 8.321093e-01  
## 25 X11 3 2 6 -0.5227028 6.994094e-01 -1.0988101 8.640745e-01  
## 26 X12 3 2 6 0.4596899 3.228694e-01 0.4240252 3.357737e-01  
## 27 X13 3 2 6 0.6352461 2.626340e-01 0.6245354 2.661380e-01  
## 28 X14 3 2 6 -0.2707444 6.067062e-01 -0.6690404 7.482651e-01  
## 29 X1 3 3 9 1.9501580 2.557865e-02 1.5150336 6.488195e-02  
## 30 X2 3 3 9 1.9440718 2.594339e-02 1.6978612 4.476698e-02  
## 31 X3 3 3 9 0.9898811 1.611161e-01 1.4852894 6.873359e-02  
## 32 X4 3 3 9 0.9773443 1.641994e-01 1.4784725 6.964067e-02  
## 33 X5 3 3 9 0.9681589 1.664825e-01 1.4737971 7.026812e-02  
## 34 X6 3 3 9 1.6081762 5.389830e-02 2.0581404 1.978833e-02  
## 35 X7 3 3 9 2.4271611 7.608749e-03 2.6365259 4.187991e-03  
## 36 X8 3 3 9 1.2620598 1.034636e-01 1.7986940 3.603355e-02  
## 37 X9 3 3 9 2.3924250 8.368724e-03 2.5628813 5.190376e-03  
## 38 X10 3 3 9 1.5113242 6.535294e-02 1.8606416 3.139740e-02  
## 39 X11 3 3 9 1.5071280 6.588892e-02 2.0148532 2.196002e-02  
## 40 X12 3 3 9 2.7698265 2.804308e-03 2.7900544 2.634959e-03  
## 41 X13 3 3 9 2.6976062 3.492000e-03 2.6472318 4.057685e-03  
## 42 X14 3 3 9 1.3412887 8.991337e-02 1.8367249 3.312526e-02  
## 43 X1 4 3 12 1.0447140 1.480776e-01 0.7269802 2.336190e-01  
## 44 X2 4 3 12 -0.3280910 6.285786e-01 -0.3359924 6.315617e-01  
## 45 X3 4 3 12 2.6446965 4.088211e-03 2.8651808 2.083856e-03  
## 46 X4 4 3 12 2.6477892 4.051002e-03 2.8465061 2.210094e-03  
## 47 X5 4 3 12 2.6917020 3.554422e-03 2.8489348 2.193294e-03  
## 48 X6 4 3 12 2.9792075 1.444975e-03 3.1274630 8.816101e-04  
## 49 X7 4 3 12 2.9635385 1.520620e-03 3.0000013 1.349892e-03  
## 50 X8 4 3 12 2.6257804 4.322529e-03 2.8425784 2.237511e-03  
## 51 X9 4 3 12 2.8121205 2.460803e-03 2.6860531 3.615080e-03  
## 52 X10 4 3 12 0.7977968 2.124942e-01 1.0285777 1.518391e-01  
## 53 X11 4 3 12 1.5367822 6.217333e-02 1.8039594 3.561883e-02  
## 54 X12 4 3 12 2.5073067 6.082755e-03 2.5524819 5.347923e-03  
## 55 X13 4 3 12 2.4323514 7.500574e-03 2.3899288 8.425821e-03  
## 56 X14 4 3 12 2.8682958 2.063447e-03 3.0328782 1.211167e-03  
## 57 X1 4 4 16 2.0269559 2.133346e-02 1.5235424 6.381154e-02  
## 58 X2 4 4 16 0.7027620 2.411020e-01 0.6534620 2.567292e-01  
## 59 X3 4 4 16 3.5523115 1.909312e-04 3.9294028 4.257854e-05  
## 60 X4 4 4 16 3.5642744 1.824320e-04 3.9297565 4.251596e-05  
## 61 X5 4 4 16 3.6316531 1.408057e-04 3.9840260 3.387873e-05  
## 62 X6 4 4 16 4.3528756 6.718169e-06 4.5038836 3.336138e-06  
## 63 X7 4 4 16 2.7928175 2.612558e-03 2.8647877 2.086444e-03  
## 64 X8 4 4 16 3.8954333 4.901166e-05 4.1822493 1.443196e-05  
## 65 X9 4 4 16 2.4181976 7.798802e-03 2.4618347 6.911417e-03  
## 66 X10 4 4 16 2.5896101 4.804235e-03 2.6413210 4.129172e-03  
## 67 X11 4 4 16 3.2956393 4.909900e-04 3.4683853 2.617980e-04  
## 68 X12 4 4 16 3.0032058 1.335758e-03 3.1083378 9.407144e-04  
## 69 X13 4 4 16 2.7910998 2.626463e-03 2.8672425 2.070328e-03  
## 70 X14 4 4 16 3.8028725 7.151397e-05 4.0699668 2.350992e-05  
## 71 X1 5 4 20 -0.5346155 7.035421e-01 -0.4992768 6.912078e-01  
## 72 X2 5 4 20 1.1006487 1.355248e-01 0.8246098 2.047966e-01  
## 73 X3 5 4 20 4.8783133 5.349843e-07 5.0434769 2.285740e-07  
## 74 X4 5 4 20 4.8721933 5.518304e-07 5.0085773 2.741692e-07  
## 75 X5 5 4 20 4.8477775 6.242615e-07 4.9593365 3.536718e-07  
## 76 X6 5 4 20 4.9056657 4.655550e-07 5.0154997 2.644790e-07  
## 77 X7 5 4 20 3.8359114 6.254976e-05 4.3620847 6.441449e-06  
## 78 X8 5 4 20 5.1763913 1.131094e-07 5.3410665 4.620069e-08  
## 79 X9 5 4 20 3.7389871 9.238159e-05 4.1440405 1.706197e-05  
## 80 X10 5 4 20 2.8941400 1.900993e-03 2.8269541 2.349653e-03  
## 81 X11 5 4 20 4.3427014 7.037069e-06 4.2354158 1.140646e-05  
## 82 X12 5 4 20 3.9316725 4.217846e-05 4.5096593 3.246591e-06  
## 83 X13 5 4 20 3.8745169 5.341821e-05 4.4000477 5.411354e-06  
## 84 X14 5 4 20 5.4902189 2.007180e-08 5.5977611 1.085689e-08  
## 85 X1 5 5 25 4.3865590 5.757896e-06 4.5493280 2.690876e-06  
## 86 X2 5 5 25 1.9445448 2.591489e-02 1.9430258 2.600652e-02  
## 87 X3 5 5 25 5.4835700 2.084136e-08 5.8314832 2.746842e-09  
## 88 X4 5 5 25 5.5175592 1.718702e-08 5.7901481 3.516219e-09  
## 89 X5 5 5 25 5.6492489 8.057520e-09 5.8537872 2.402516e-09  
## 90 X6 5 5 25 5.3821045 3.681001e-08 5.7379134 4.792505e-09  
## 91 X7 5 5 25 4.5355248 2.873021e-06 4.9879729 3.050806e-07  
## 92 X8 5 5 25 5.9920796 1.035872e-09 6.2738654 1.760965e-10  
## 93 X9 5 5 25 4.2891523 8.967818e-06 4.4545337 4.203787e-06  
## 94 X10 5 5 25 5.0191383 2.595189e-07 5.0550384 2.151521e-07  
## 95 X11 5 5 25 5.9062885 1.749505e-09 6.1012667 5.261557e-10  
## 96 X12 5 5 25 5.0871223 1.817687e-07 5.6238340 9.338255e-09  
## 97 X13 5 5 25 4.6736584 1.479406e-06 5.2871523 6.211761e-08  
## 98 X14 5 5 25 5.8891497 1.940938e-09 6.1717709 3.376464e-10  
## 99 X1 6 5 30 2.6007802 4.650601e-03 3.2848934 5.101049e-04  
## 100 X2 6 5 30 2.1015673 1.779560e-02 1.5776214 5.732631e-02  
## 101 X3 6 5 30 7.7044935 6.568164e-15 7.1317144 4.956318e-13  
## 102 X4 6 5 30 7.7346946 5.182579e-15 7.1416152 4.612026e-13  
## 103 X5 6 5 30 7.8558509 1.985336e-15 7.2772550 1.703405e-13  
## 104 X6 6 5 30 8.3190807 4.432410e-17 7.9203866 1.183866e-15  
## 105 X7 6 5 30 6.1255908 4.517386e-10 6.4095415 7.297895e-11  
## 106 X8 6 5 30 8.1657887 1.596709e-16 7.6380771 1.102449e-14  
## 107 X9 6 5 30 5.7315458 4.975973e-09 6.1007386 5.278974e-10  
## 108 X10 6 5 30 5.4258926 2.883281e-08 5.0843371 1.844561e-07  
## 109 X11 6 5 30 7.4373417 5.136579e-14 6.9040762 2.526559e-12  
## 110 X12 6 5 30 6.2777630 1.717394e-10 6.2456115 2.110727e-10  
## 111 X13 6 5 30 6.2318299 2.305089e-10 6.3479942 1.090702e-10  
## 112 X14 6 5 30 8.3614325 3.098063e-17 7.8133128 2.785206e-15

Based on our targets we see that a SOM with 12 profiles explains ~60 of the variability (Panel B) in our data. We use this size as our example below.

## Finding the Optimal Seed

Starting points have been shown to influence results from techniques such as SOM and k-means and thus we identify an optimal seed value that provides the lowest overall quantization error (a.k.a. mean absolute error).

somx=4  
somy=3  
opt.seed<-som.seed(data.trn.sc, nstart=10, iter.max=5000, somx, somy, grid.topo="rectangular")  
print(opt.seed)

## [1] 14

## Fitting a Biomarker SOM

Here we fit a 4x3 SOM to our synthetic biomarker data using the som() function provided by the kohonen package. We note that there are multiple SOM packages in R; however, the kohonen som function provides greater flexibiity in implementation of the algorithm. For more detail on kohonen see: <https://cran.r-project.org/web/packages/kohonen/index.html>

set.seed(opt.seed)  
bio.som<-som(data.trn.sc, grid=somgrid(somx,somy, "rectangular"),   
 rlen=5000, dist.fcts="euclidean", mode="online",  
 alpha=c(0.05,0.01)) #learning parameter alpha  
print(summary(bio.som))

## SOM of size 4x3 with a rectangular topology and a bubble neighbourhood function.  
## The number of data layers is 1.  
## Distance measure(s) used: euclidean.  
## Training data included: 500 objects.  
## Mean distance to the closest unit in the map: 2.295.  
## NULL

## Summarizing SOM Output

Summarizing the output of a SOM is critical for interpretation. Aspects of interest here are overall model fit, potential sample size, class cohesion and seperation, and the representativeness (i.e., relative importance) of each component of the map.

som.summ<-som.fit.summ(bio.som)  
print(names(som.summ))

## [1] "MODEL\_STATS" "COORDINATES" "FREQUENCIES"   
## [4] "PROFILES" "PREDICTIONS" "PROFILE\_DISTANCES"   
## [7] "COMPONENT\_EVALUATION" "MAP\_ORGANIZATION" "COMPONENT\_STATISTICS"

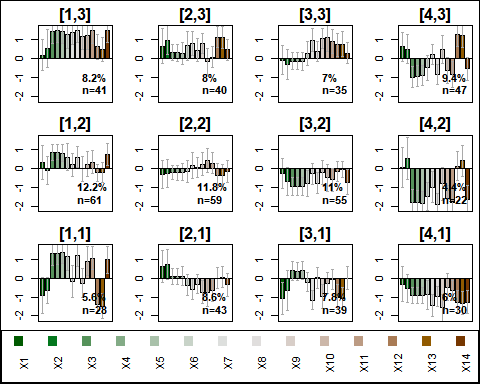
print(som.summ$FREQUENCIES)

## SOM\_ID XY X Y N FREQ  
## 1 1 11 1 1 28 5.6  
## 5 2 21 2 1 43 8.6  
## 6 3 31 3 1 39 7.8  
## 7 4 41 4 1 30 6.0  
## 8 5 12 1 2 61 12.2  
## 9 6 22 2 2 59 11.8  
## 10 7 32 3 2 55 11.0  
## 11 8 42 4 2 22 4.4  
## 12 9 13 1 3 41 8.2  
## 2 10 23 2 3 40 8.0  
## 3 11 33 3 3 35 7.0  
## 4 12 43 4 3 47 9.4

## Plotting Results

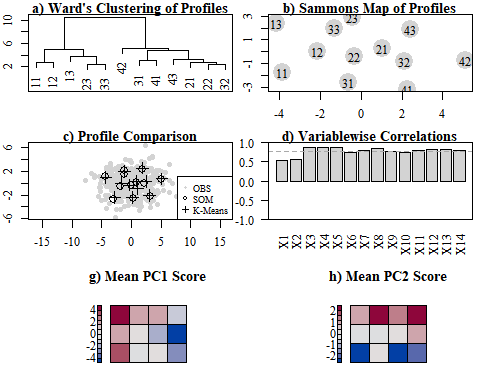
We present a 4X3 SOM that illustrates 12 (synthetic) Biomarker Mixture Types (BMTs) that capture the range of exposure scenarios that occurred in the synthetic data. Each BMT is characterized by a profile which reflects a defined exposure category, where bar plots illustrate the magnitudes via the mean (± standard deviations) of biomarker concentrations. The y-axis is presented on a normalized scale and where zero reflects the mean. (This is a standard approach used to provide comparative measures across differing chemical classes and reporting units.) Labels reflect SOM grid coordinates and are presented above each profile in brackets [x,y] and the relative frequencies (%) and the within-class sample sizes (n) of classification assignments are in the upper right corner.

map.bar.plot(bio.som, label.loc="bottomright")



## Assessing Map Fit

map.fit.plots(bio.som)

 Here we provide a range of plots designed to improve understanding of our SOM model and facilitate comparision to more traditional techiques. Panel a) presents a hierarchical clustering dendrogram of SOM profiles that allows us to assess map organization. Panel b) presents a Sammons mapping of SOM profiles that allows us to more accurately assess differences among profiles (via distance). Panel c) compares SOM profiles to k means. Panel d) provides a barplot of profile variable correlations that allow us to see which variables are best captured by the SOM. Panels g/h) plot the average PCA scores for PC1 and PC2 over the SOM grid.

## Constructing an exposure metric

Given the unique information provided by SOM, we are able to develo an exposure metric that contains two ways to assess our results in subsequent mixtures studies. First, SOM profile assignment results to create a categorical exposure variable that can be used to assess effects (95% confidence intervals) for each profile. Second, we are able take advantage of the organized arrangement provided by SOM as the neighborhood similarity allows us to treat results as an exposure continuum. Here, we use the SOM coordinate information for that can then be used as a trend surface model (e.g. bivariate smooth term) in subsequent analyses. This ‘spatialization’ allows us to explore a joint-dose response function for multiple exposures while alleviating challenges imposed by smaller sample sizes.

bmt.exp<-exp.metric(bio.som)  
head(bmt.exp)

## SOM\_ID OBS XY X Y DISTANCE  
## 1 7 1 32 3 2 2.439463  
## 2 5 2 12 1 2 2.545448  
## 3 6 3 22 2 2 1.766833  
## 4 3 4 31 3 1 3.572333  
## 5 11 5 33 3 3 2.587199  
## 6 5 6 12 1 2 2.595177

Next steps involve illustrating how to integrate SOM results in linear and generalized modeling frameworks.

The End.