opticskxi: OPTICS K-Xi Density-Based Clustering

Thomas Charlon
University of Geneva

Abstract

Density-based clustering methods are well adapted to the clustering of high-dimensional data and enable the discovery of core groups of various shapes despite large amounts of noise. The **opticskxi** R package provides a novel density-based cluster extraction method, OPTICS k-Xi, and a framework to compare k-Xi models using distance-based metrics to investigate datasets with unknown number of clusters. This article first introduces density-based algorithms with simulated datasets, then presents and evaluates the k-Xi cluster extraction method. Finally, the models comparison framework is described and experimented on 2 genetic datasets to identify groups and their discriminating features. The k-Xi algorithm is a novel OPTICS cluster extraction method that specifies directly the number of clusters and does not require fine-tuning of the steepness parameter as the OPTICS Xi method. Combined with a framework that compares models with varying parameters, the OPTICS k-Xi method can identify groups in noisy datasets with unknown number of clusters.

Keywords: DBSCAN, OPTICS, density-based clustering, hierarchical clustering.

1. Introduction

Density-based clustering methods detect groups of similar observations based on their distance to a given number of their nearest neighbors. In contrast with other clustering methods as k-means or Gaussian mixture models, they do not expect the observed data to follow Gaussian or other parametric distributions and they can thus detect groups of various shapes.

In this article, density-based clustering algorithms are first presented on simulated datasets using the **dbscan** package (Hahsler and Piekenbrock 2016), and limitations due to clusters of varying densities and fine-tuning of parameters are described. A novel cluster extraction algorithm, OPTICS k-Xi, is then presented and evaluated on the datasets. Finally, a framework to compare multiple k-Xi models with varying parameters is detailed and experimented on 2 genetic datasets of Schizophrenia and Crohn's disease patients, to enable further investigation of the best models and identify genetic signatures of core groups.

1.1. DBSCAN

DBSCAN (Ester, Kriegel, Sander, Xu *et al.* 1996) is a well-known density-based clustering algorithm with 3 parameters: a distance matrix, a number of neighbors, and a reachability distance threshold.

The algorithm first searches for core points in the distance matrix, *i.e.* points that have distances from at least a given number of points, the number of neighbors, smaller than the

reachability threshold. If one is found, the core point and its neighbors form a cluster, and if additional core points are found within its neighbors, the cluster is expanded to also include their neighbors, iteratively until no core points are discovered in the neighbors. Additional clusters are then formed similarly for each core point not yet assigned and finally remaining non-assigned points are considered noise.

In the **dbscan** package, by default the euclidean distance is used, with 5 neighbors, and the distance threshold must be fine-tuned. In these simulated datasets from the **factoextra** and **dbscan** R packages, points are organized by various shapes along with some noise, on 2 variables. With specific distance thresholds, DBSCAN successfully detects the shapes and the noise (Figure 1).

```
R>   library('opticskxi')
R>   data('multishapes')
R>   dbscan_shapes <- dbscan::dbscan(multishapes[1:2], eps = 0.15)
R>   gg_shapes <- cbind(multishapes[1:2], Clusters = dbscan_shapes$cluster) %>%
+      ggpairs(group = 'Clusters')
R>   data('DS3', package = 'dbscan')
R>   dbscan_ds3 <- dbscan::dbscan(DS3, minPts = 25, eps = 12)
R>   gg_ds3 <- cbind(DS3, Clusters = dbscan_ds3$cluster) %>%
+      ggpairs(group = 'Clusters')
R>   cowplot::plot_grid(gg_shapes, gg_ds3, nrow = 2,
+                       labels = c('(a)', '(b)'), label_x = 0.9)
```

However, DBSCAN uses a fixed distance threshold and thus can not detect clusters of varying densities. In this simulated Gaussian data with 2 large clusters and 2 smaller, more dense, clusters, DBSCAN detects either the pair of large or small clusters, depending on the distance threshold, but can not detect all 4 clusters (Figure 2).

```
R>   n <- 1e3
R>   set.seed(0)
R>   multi_gauss <- cbind.data.frame(
+      x = c(rnorm(n / 2, -3), rnorm(n / 4, 3), rnorm(n / 4, 3, .2)),
+      y = c(rnorm(n * .75), rnorm(n / 8, 1, .2), rnorm(n / 8, -1, .2)))
R>   dbscan_gauss <- dbscan::dbscan(multi_gauss, minPts = 30, eps = .5)
R>   gg_mgauss <- cbind(multi_gauss, Clusters = dbscan_gauss$cluster) %>%
+      ggpairs(group = 'Clusters')
R>   gg_mgauss_small <- dbscan::dbscan(multi_gauss, minPts = 30, eps = .2) %$%
+      cbind(multi_gauss, Clusters = cluster) %>% ggpairs(group = 'Clusters')
R>   cowplot::plot_grid(gg_mgauss, gg_mgauss_small, nrow = 2,
+      labels = c('(a)', '(b)'), label_x = .9)
```

1.2. OPTICS

OPTICS (Ankerst, Breunig, Kriegel, and Sander 1999) is another density-based algorithm that produces an ordering and a distance profile of observations, similar to a tree-like dendogram,

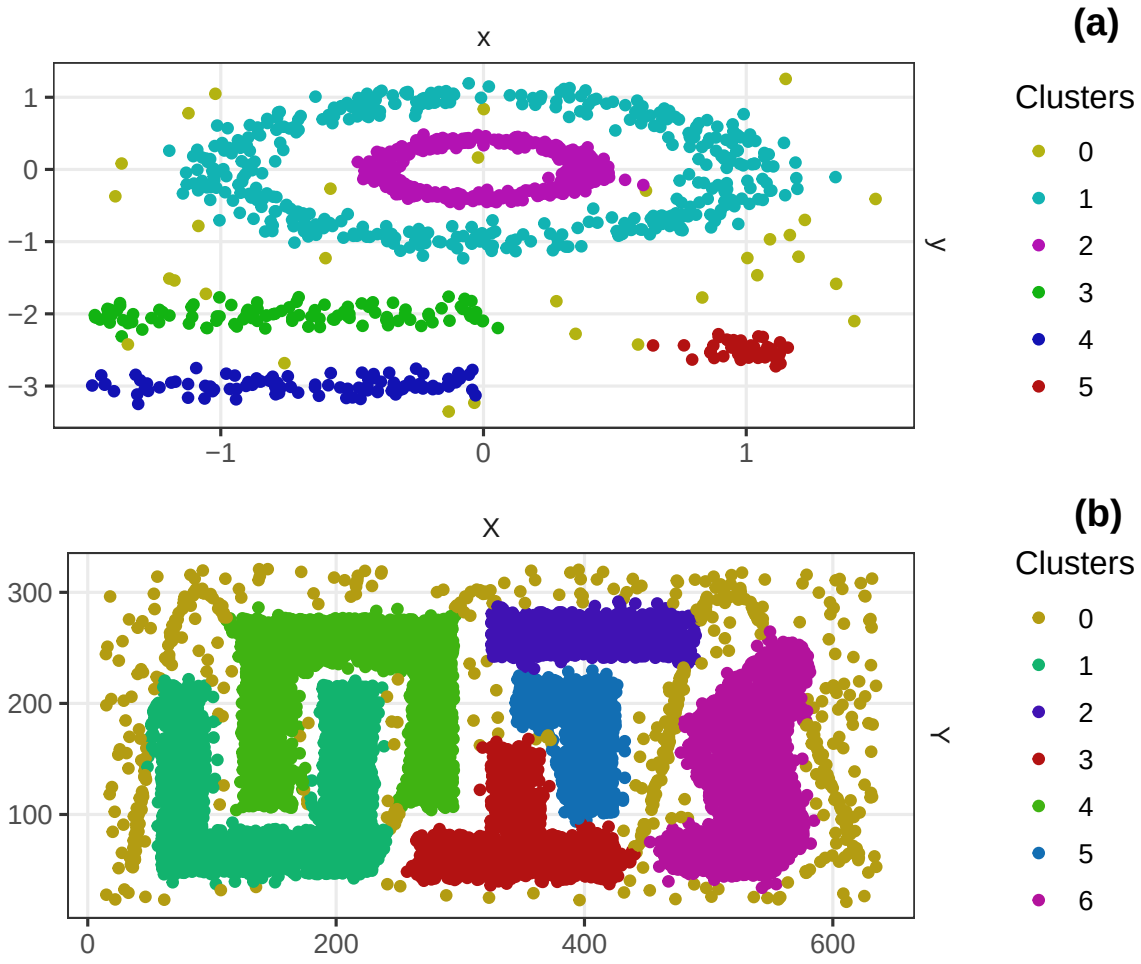

Figure 1: DBSCAN clusterings of various shapes and noise (cluster 0). (a): Multishapes dataset. (b): DS3 dataset.

and enables the detection of clusters of varying densities with the cluster extraction method OPTICS Xi (Ankerst *et al.* 1999).

OPTICS uses at least 2 parameters, a distance matrix and a number of neighbors, and produces a distance profile that reveals the density structure of the dataset and can be used to extract clusters. The algorithm iteratively explores point neighborhoods in the order of lowest to highest core distance, *i.e.* the maximum distance from a point to a given number of its nearest neighbors, and returns the orders and the reachability distances of successive points, *i.e.* the maximum between the core distance of the point and the distance from it to the previous point. Low reachability-distances regions, or valleys, thus represent clusters and are separated by peaks, *i.e.* points with high reachability distances.

OPTICS can be used to fine-tune the distance threshold in DBSCAN, as the DBSCAN method is equivalent to a horizontal threshold on the reachability plot.

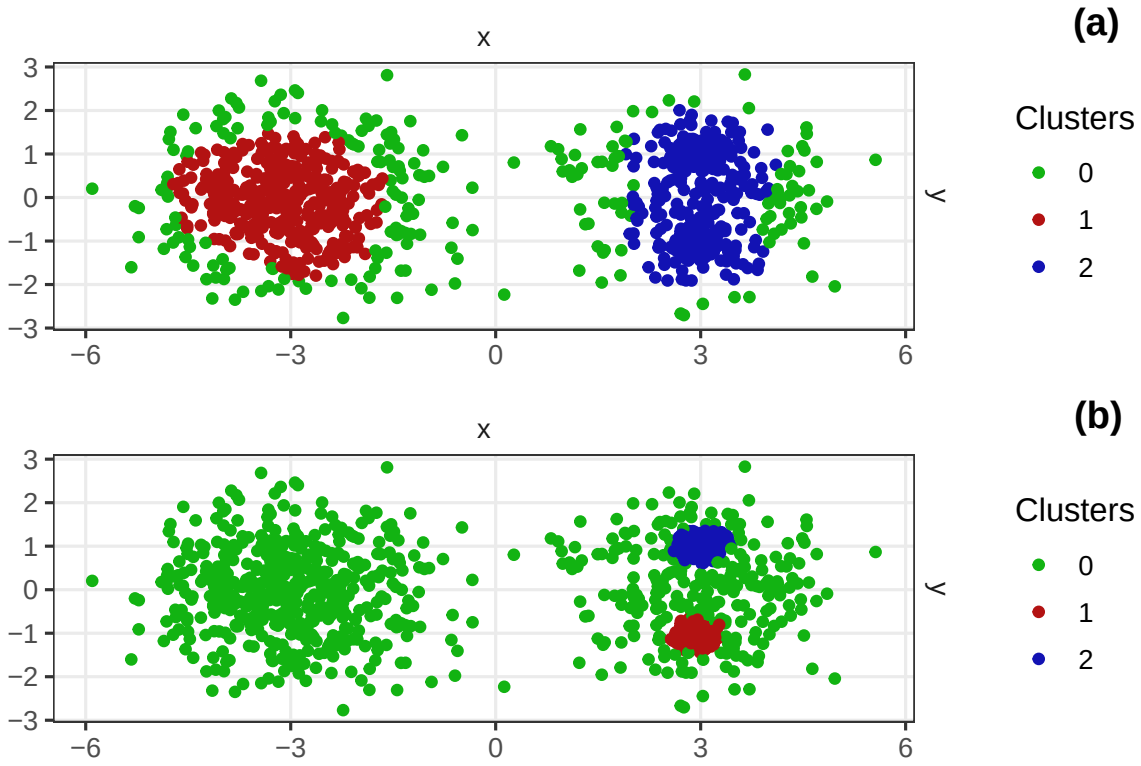

Figure 2: DBSCAN clustering of hierarchical Gaussian clusters. (a): Detection of large clusters. (b): Detection of small clusters.

1.3. OPTICS Xi

To extract clusters of varying densities from OPTICS profiles, the OPTICS Xi (Ankerst *et al.* 1999) algorithm uses a steepness threshold. The differences between reachability distances of successive points are first computed. Then, for each absolute distance difference above the given threshold, all adjacent points with a smaller reachability distance form a cluster.

Two cases are distinguished when forming a cluster, steep down or steep up areas, in which the reachability distance differences that delimit the cluster are negative or positive, *i.e.* when the first observation has a reachability higher or lower than the second observation, respectively.

- In a steep down area, both observations that produce the large distance difference are part of the cluster, and all successive points with a reachability distance smaller than the first observation, are part of the cluster.
- In a steep up area, only the first observation is part of the cluster, and all previous points with a reachability distance smaller than the second observation, and the adjacent previous point, are part of the cluster.

OPTICS Xi thus detects clusters of varying densities, possibly hierarchical, although the Xi steepness parameter must be fine-tuned. In the simulated hierarchical Gaussian data, OPTICS Xi successfully detects both the large and small clusters with Xi = 0.03 (Figure 3).

```
R>   optics_gauss <- dbscan::optics(multi_gauss, minPts = 30)
R>   xi_gauss <- dbscan::extractXi(optics_gauss, xi = 0.03)
R>   ggplot_optics(optics_gauss, groups = xi_gauss$cluster)
```

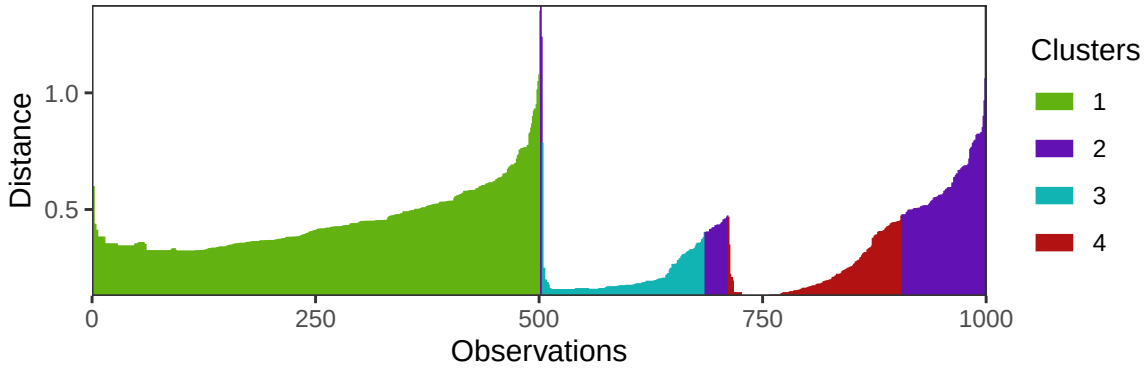

Figure 3: OPTICS profile of the hierarchical Gaussian clusters, colored by OPTICS Xi clustering.

2. OPTICS k-Xi

The **opticskxi** package provides a variant OPTICS cluster extraction algorithm, k-Xi, that specifies directly the number of clusters and does not require fine-tuning a parameter. Instead of using a fixed distance difference threshold OPTICS Xi, the k-Xi algorithm iteratively investigates the largest differences until at the given number of clusters are defined.

2.1. Algorithm

For each successive largest difference, OPTICS k-Xi will attempt to form a cluster of all adjacent points with a smaller reachability-distance, distinguishing steep down and up areas similarly as OPTICS Xi (detailed above). If the newly formed cluster contains less observations than the `pts` parameter, or if it reduces the size of a previously formed cluster below the `pts` parameter, the new cluster is discarded and the next largest difference is considered.

The algorithm then stops when the number of clusters has reached the `n_xi` parameter, or when the number of largest differences considered has reached the `max_loop` threshold, by default 50. The `pts` parameter is set by default to the `minPts` parameter used to compute the OPTICS profile, and avoids introducing small clusters due to nearby large distance differences.

2.2. Results

In the hierarchical Gaussian data, OPTICS k-Xi successfully detects the large and small clusters (Figure 4).

```
R>   kxi_gauss <- opticskxi(optics_gauss, n_xi = 4, pts = 100)
R>   ggplot_optics(optics_gauss, groups = kxi_gauss)
```

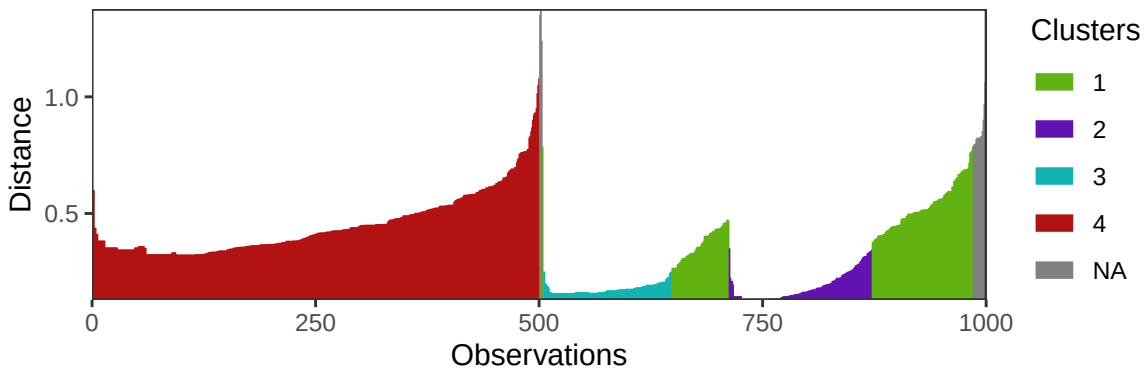

Figure 4: OPTICS profile of the hierarchical Gaussian clusters, colored by OPTICS k-Xi clustering.

In the multishapes and the DS3 datasets, OPTICS k-Xi also successfully detects the shapes, but the noise is included in the largest cluster (Figure 5).

```
R>   gg_shapes_optics <- dbscan::optics(multishapes[1:2]) %>%
+      ggplot_optics(groups = opticskxi(., n_xi = 5, pts = 30))
R>   gg_ds3_optics <- dbscan::optics(DS3, minPts = 25) %>%
+      ggplot_optics(groups = opticskxi(., n_xi = 6, pts = 100))
R>   cowplot::plot_grid(gg_shapes_optics, gg_ds3_optics, nrow = 2,
+      labels = c('(a)', '(b)'), label_x = .9)
```

3. Models comparisons by distance-based metrics

To explore complex datasets where clusters are not well defined, k-Xi models with various distances, number of points, and number of clusters may be investigated and compared. Furthermore, in datasets with many variables, dimension reduction methods as principal component analysis (PCA) or independent component analysis (ICA) may be required prior to the clustering to summarize information.

The **opticskxi** package provides a framework to efficiently compare multiple k-Xi models with varying parameters and to extract and visualize the models with highest metrics, to enable further investigation of the clusters of the best models.

3.1. Framework

The main function, `opticskxi_pipeline`, inputs a data frame of k-Xi clustering parameters and returns corresponding clustering results and their distance-based metrics.

Parameters are specified using a data frame with the following columns:

- `dim_red`: Optional dimension reduction: 'PCA', 'ICA' (using the **fastICA** package (Marchini, Heaton, and Ripley 2017))
- `n_dim_red`: Optional number of components of the dimension reduction

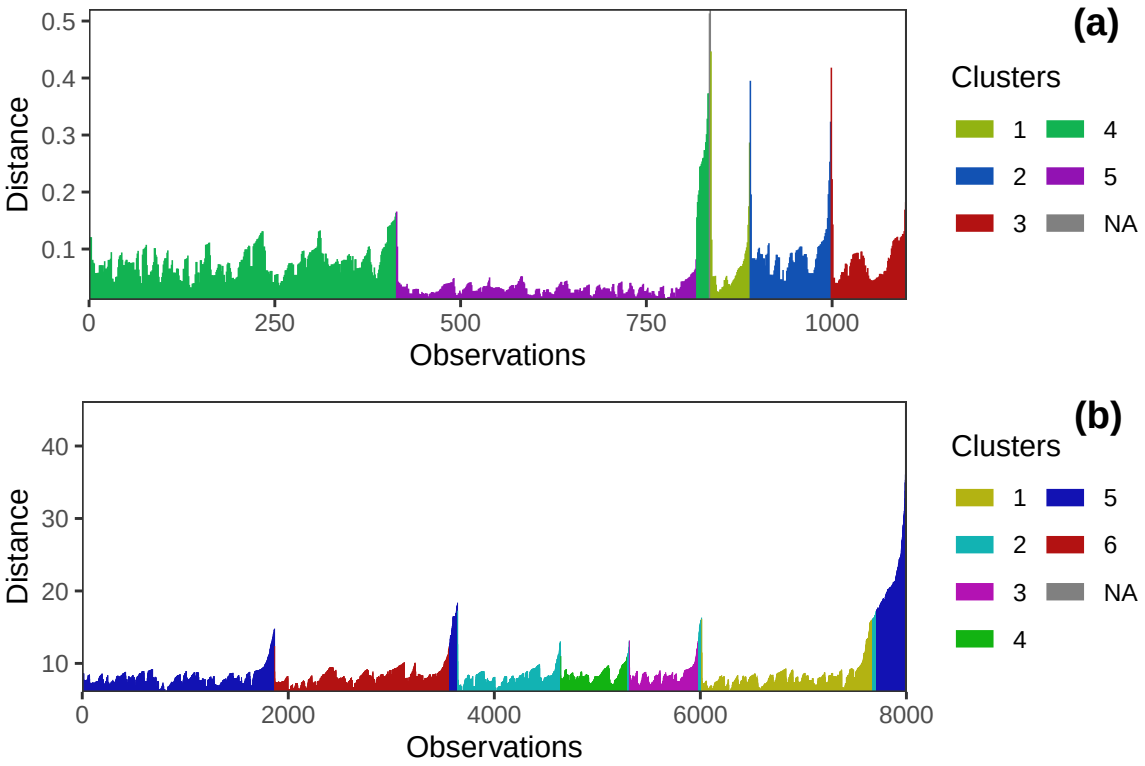

Figure 5: OPTICS profiles colored by OPTICS k-Xi clusterings. (a): Multishapes dataset. (b): DS3 dataset.

- `dist`: Distance, one of the 11 distances from the **amap** package (Lucas 2019)
- `pts`: Number of points for OPTICS (`minPts`) and k-Xi
- `n_xi`: Number of clusters for k-Xi

To efficiently compute multiple k-Xi models with varying dimension reductions and parameters, the framework proceeds step-by-step, by first computing the unique dimension reduction matrices required, then unique distance matrices, unique OPTICS models, and finally k-Xi cluster extractions.

Distance-based metrics are then measured for each model using the **fpc** package (Hennig 2019). The following metrics are stored: `avg.silwidth`, `bw.ratio`, `ch`, `pearsongamma`, `dunn`, `dunn2`, `entropy`, `widestgap`, `sindex`.

Finally, clusters and metrics are binded to the input parameter data frame that defines the unique parameters of each model, in the columns `clusters` and `metrics`.

Three functions can be used directly on the results data frame to extract specific models and investigate the distance profiles and the clusterings:

- `get_best_kxi`: Subset the data frame by specifying a metric and one or more ranks, in decreasing order of the metric.

- `ggplot_kxi_metrics`: Plot metrics of the top ranked k-Xi models, by default the 8 models with highest average silhouette width. Additional metrics can be displayed next to the one used for ranking, by default the between-within ratio.
- `gtable_kxi_profiles`: Plot OPTICS profiles of the top ranked k-Xi models, by default the 4 models with highest average silhouette width.

3.2. Schizophrenia patients and controls

In this dataset from the **gap** R package (Zhao, colleagues with inputs from Kurt Hornik, and Ripley 2015), 6 alleles from the chromosome 6 gene *HLA* were measured from 94 schizophrenia patients and 177 controls.

All combinations of the following OPTICS k-Xi parameters are computed:

- Distance: Manhattan, Euclidean, absolute Pearson, absolute correlation
- Number of clusters: 3 to 5
- Number of points: 20, 30, 40

The 8 best models by average silhouette width are first visualized and reveal that all use Manhattan or Euclidean distances and half use a `pts` parameter of 20 (Figure 6).

```
R>   data('hla')
R>   m_hla <- hla[-c(1:2)] %>% scale
R>   df_params_hla <- expand.grid(n_xi = 3:5, pts = c(20, 30, 40),
+      dist = c('manhattan', 'euclidean', 'abscorrelation', 'abspearson'))
R>   df_kxi_hla <- opticskxi_pipeline(m_hla, df_params_hla)
R>   ggplot_kxi_metrics(df_kxi_hla, n = 8)
```

The OPTICS profiles of the 4 best models are then visualized and reveal that the two best models only differ by their number of clusters, 3 or 4, and that the third and fourth models have hierarchical clusters (Figure 7).

```
R>   gtable_kxi_profiles(df_kxi_hla) %>% plot
```

The second best model is then selected to investigate the model with 4 clusters. To assess if patients are significantly enriched in each group, standardized Pearson residuals are computed using `chisq.test`. One disease is enriched or depleted in one group if the residual is above or below 2, respectively (Friendly 1994).

Results show cluster 4 is enriched in Schizophrenia patients ($residual = 3.98$): 58% of individuals are patients, although the distribution in the complete dataset is 34%; and that cluster 2 is enriched in controls ($residual = 2.45$) (Table 1).

```
R>   best_kxi_hla <- get_best_kxi(df_kxi_hla, rank = 2)
R>   clusters_hla <- best_kxi_hla$clusters
R>   hla$id %<>% `levels<-`(c('Controls', 'Sch. patients'))
R>   residuals_table(clusters_hla, hla$id) %>% print_vignette_table('HLA')
```

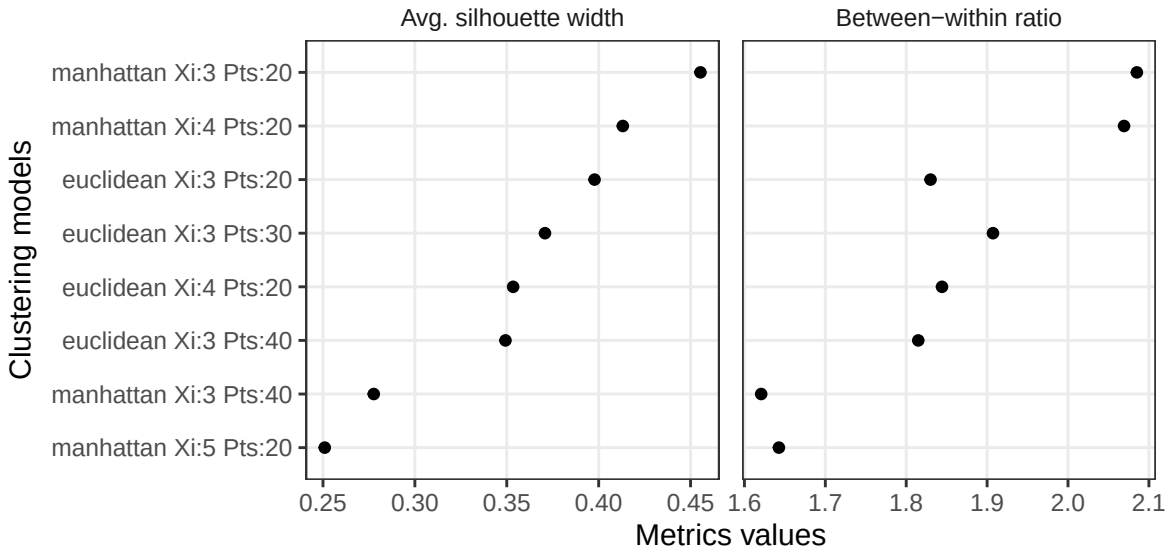

Figure 6: Distance-based metrics of the 8 best k-Xi clusterings of the HLA dataset, ranked by decreasing average silhouette width. All the models use Manhattan or Euclidean distances and half use a `pts` parameter of 20.

	Controls	Sch. patients	Total	Residuals: Controls	Residuals: Sch. patients
1	56	27	83	1.22	-1.22
2	19	3	22	2.45	-2.45
3	23	9	32	1.19	-1.19
4	24	34	58	-3.98	3.98
NA	55	21	76	NA	NA
Total	177	94	271	NA	NA

Table 1: Contingency table of disease status and k-Xi clustering of the HLA dataset, with standardized Pearson residuals.

The groups are finally visualized using PCA dimension reduction, and the contributions of variables are displayed to identify the contributions of the genetic markers. On PCA, the genetic markers most discriminating cluster 4 from other observations are *DQB.a2*, *DQR.a2*, and *DQA.a2* (Figure 8).

```
R>   fortify_pca(m_hla, sup_vars = data.frame(Clusters = clusters_hla)) %>%
+      ggpairs('Clusters', ellipses = TRUE, variables = TRUE)
```

3.3. Crohn's disease patients and relatives

In this other dataset from the **gap** package (Zhao *et al.* 2015), 212 single nucleotide polymorphisms (SNPs) from chromosome 5 (5q31) were measured from 129 Crohn's disease patients and their 2 parents.

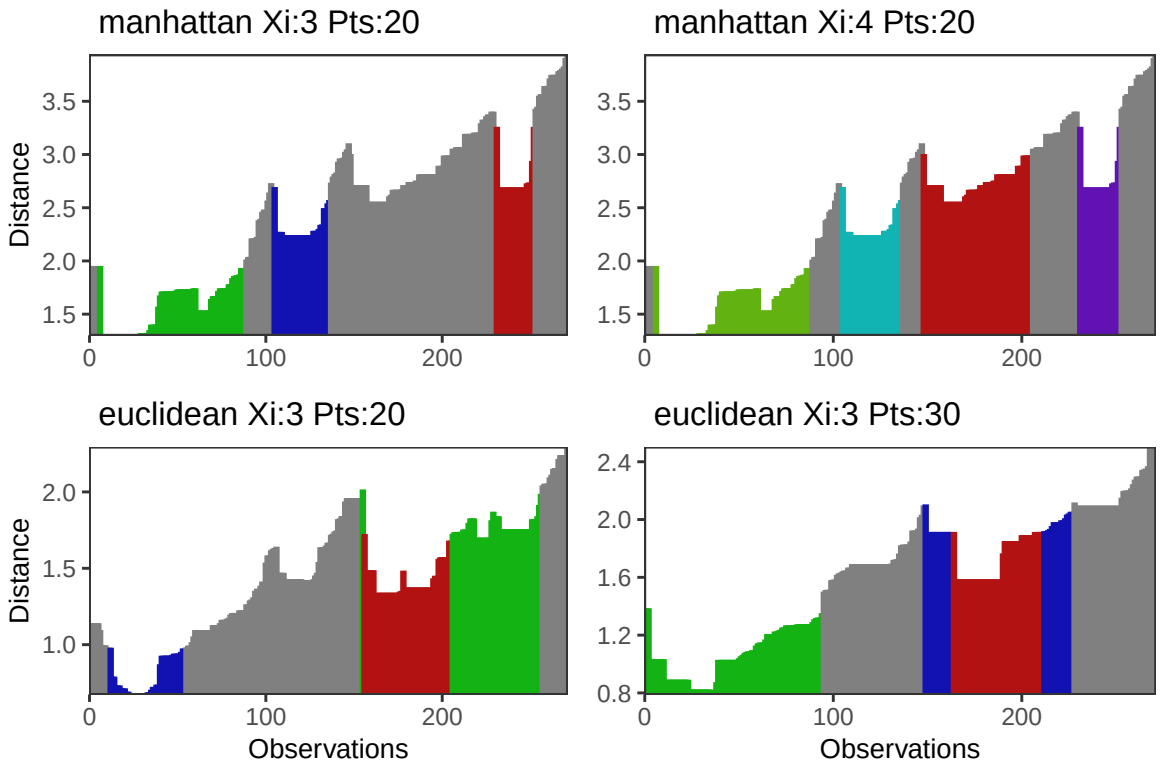

Figure 7: OPTICS profiles of the 4 best k-Xi clusterings of the HLA dataset, ranked by decreasing average silhouette width. The two best models only differ by their number of clusters, 3 or 4, and the third and fourth models have hierarchical clusters.

Since the number of variables is high, dimension reduction methods are applied to the clustering. All combinations of the following OPTICS k-Xi parameters are computed:

- Dimension reduction: PCA, ICA
- Number of dimension reduction components: 4, 6, 8
- Distance: Euclidean, absolute Pearson, absolute correlation
- Number of clusters: 3 to 5
- Number of points: 30, 40, 50

The 8 best models by average silhouette width mostly use the Absolute Pearson or Absolute Correlation distances and 3 clusters (Figure 9).

```
R>   data('crohn')
R>   m_crohn <- crohn[-c(1:6)] %>% scale
R>   df_params_crohn <- expand.grid(n_xi = 3:5, dim_red = c('PCA', 'ICA'),
+      dist = c('euclidean', 'abscorrelation', 'abspearson'),
+      pts = c(30, 40, 50), n_dimred_comp = c(4, 6, 8))
```

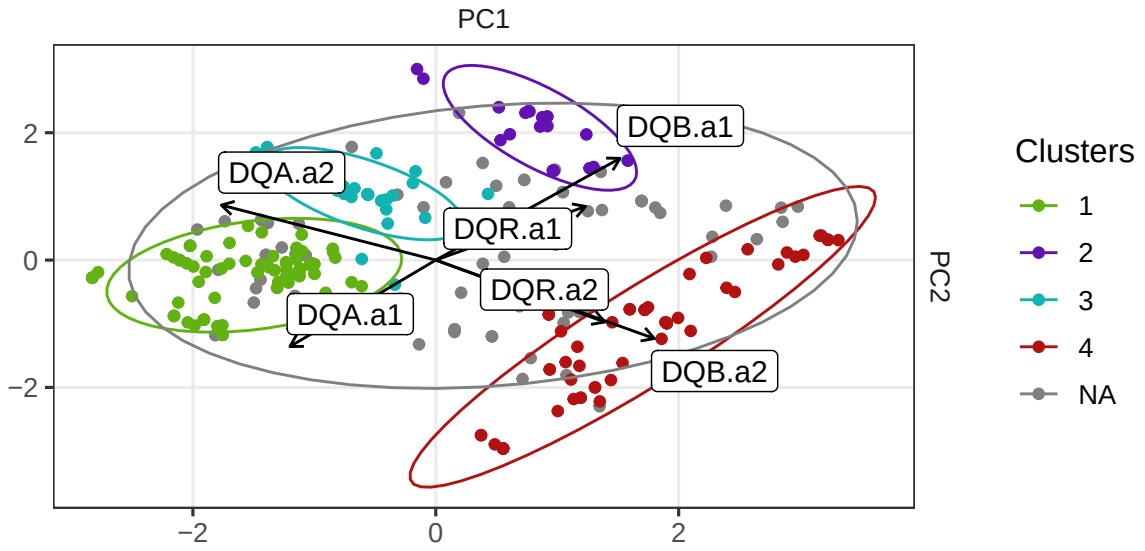

Figure 8: PCA of the HLA dataset colored by k-Xi clustering with 95% confidence ellipses.

```
R>   df_kxi_crohn <- opticskxi_pipeline(m_crohn, df_params_crohn)
R>   ggplot_kxi_metrics(df_kxi_crohn)
```

The OPTICS profiles of the 4 best models reveal that all have hierarchical clusters (Figure 10).

```
R>   gtable_kxi_profiles(df_kxi_crohn) %>% plot
```

In the best model, cluster 2 is enriched in Crohn's disease patients ($residual = 3.73$): 46% of individuals are patients, although only 37% in the complete dataset; and cluster 3 is enriched in controls ($residual = 3.05$) (Table 2).

```
R>   best_kxi_crohn <- get_best_kxi(df_kxi_crohn, rank = 1)
R>   clusters_crohn <- best_kxi_crohn$clusters
R>   crohn$crohn %<>% factor %>% `levels<-`(c('Controls', 'Crohn patients'))
R>   residuals_table(clusters_crohn, crohn$crohn) %>%
+      print_vignette_table('Crohn')
```

The groups are then visualized using the corresponding dimension reduction, ICA with 4 components, which reveals the hierarchical structure of clusters 1 and 2, discriminated from cluster 3 mostly by the second and third components (Figure 11).

```
R>   ica <- fortify_ica(m_crohn, n.comp = 4,
+      sup_vars = data.frame(Clusters = clusters_crohn))
R>   ggpairs(ica, 'Clusters', axes = 1:4, ellipses = TRUE, level = .75) %>%
+      plot
```

The dimension reduction visualization is then focused on the second and third components to reveal the variables with strong contributions (Figure 12).

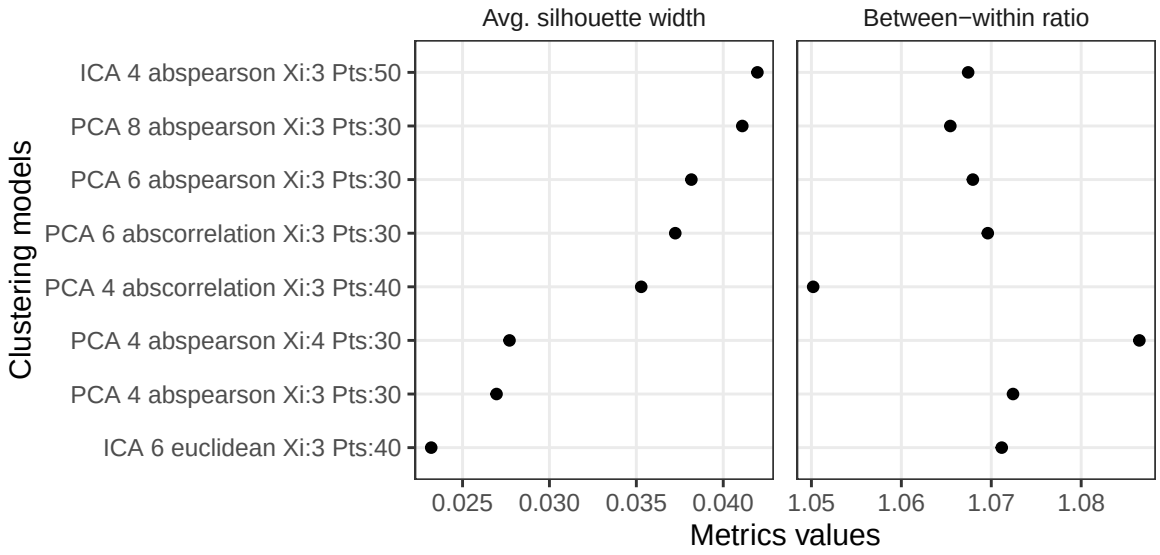

Figure 9: Distance-based metrics of the 8 best k-Xi clusterings of the Crohn dataset, ranked by decreasing average silhouette width.

	Controls	Crohn patients	Total	Residuals: Controls	Residuals: Crohn patients
1	45	17	62	1.08	-1.08
2	51	44	95	-3.73	3.73
3	49	10	59	3.05	-3.05
NA	98	73	171	NA	NA
Total	243	144	387	NA	NA

Table 2: Contingency table of disease status and k-Xi clustering of the Crohn dataset, with standardized Pearson residuals.

```
R>   ggpairs(ica, 'Clusters', axes = 2:3, ellipses = TRUE, variables = TRUE,
+      n_vars = 3)
```

4. Conclusions

The OPTICS k-Xi algorithm attempts directly to define a given number of clusters and does not require fine-tuning of a steepness parameter as OPTICS Xi. Combined with a framework to compare models with varying parameters, the k-Xi method can identify core groups in noisy datasets with an unknown number of clusters.

Recent density-based algorithms as HDBSCAN (Campello, Moulavi, and Sander 2013) also enable to detect clusters of varying densities and to specify directly the number of clusters to define. In contrast with OPTICS k-XI which iteratively attempts to define clusters until the specified number is reached, HDBSCAN can provide any given number of clusters, based on a hierarchical structure. Future work may thus include comparing OPTICS k-Xi with

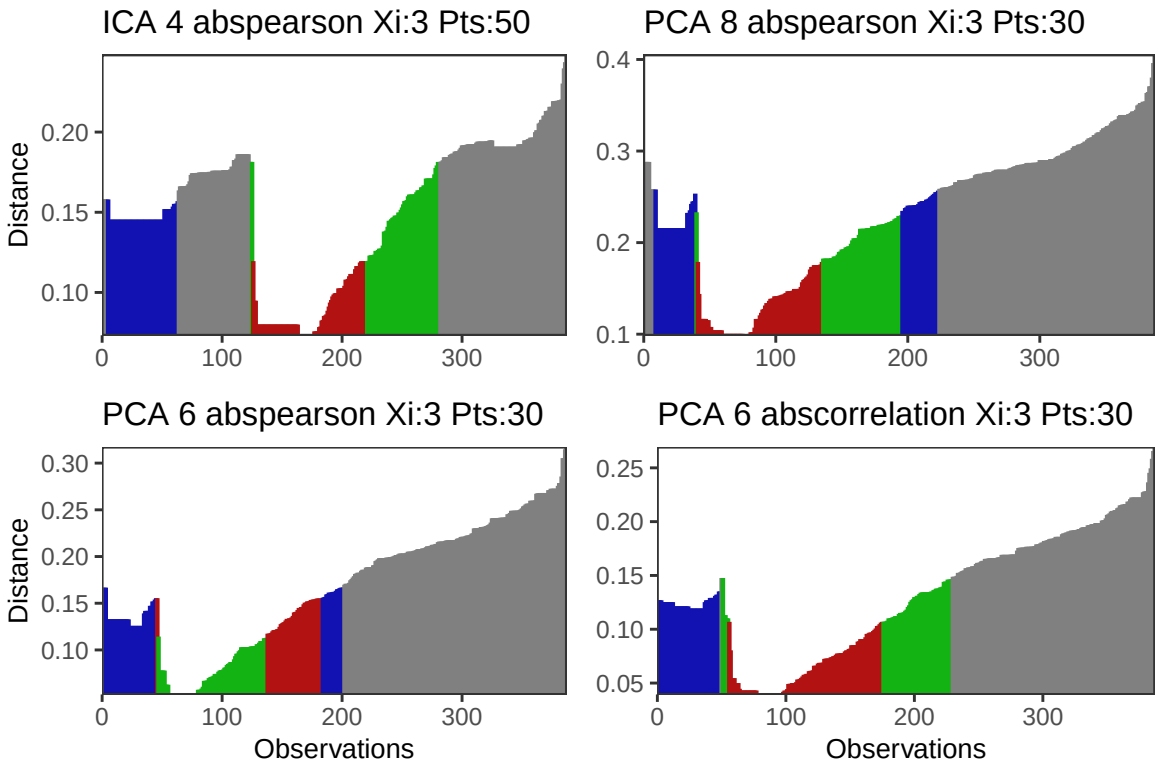

Figure 10: OPTICS profiles of the 4 best k-Xi clusterings of the Crohn dataset, ranked by decreasing average silhouette width.

more recent density-based algorithms as HDBSCAN, and expanding the models comparison framework to include other density-based algorithms.

5. Acknowledgements

This work was inspired by Jérôme Wojcik (Precision for Medicine) and Sviatoslav Voloshynovskiy (University of Geneva).

References

Ankerst M, Breunig MM, Kriegel HP, Sander J (1999). "OPTICS: Ordering Points to Identify the Clustering Structure." In *ACM Sigmod Record*, volume 28, pp. 49–60. ACM.

Campello RJ, Moulavi D, Sander J (2013). "Density-Based Clustering Based on Hierarchical Density Estimates." In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172. Springer.

Ester M, Kriegel HP, Sander J, Xu X, *et al.* (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In *Kdd*, volume 96, pp. 226–231.

Friendly M (1994). "Mosaic Displays for Multi-Way Contingency Tables." *Journal of the American Statistical Association*, **89**(425), 190–200.

Hahsler M, Piekenbrock M (2016). ***dbscan**: Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms*. R package version 0.9-8.2.

Hennig C (2019). ***fpc**: Flexible Procedures for Clustering*. R package version 2.2-2, URL https://CRAN.R-project.org/package=fpc.

Lucas A (2019). ***amap**: Another Multidimensional Analysis Package*. R package version 0.8-17, URL https://CRAN.R-project.org/package=amap.

Marchini JL, Heaton C, Ripley BD (2017). ***fastICA**: FastICA Algorithms to Perform ICA and Projection Pursuit*. R package version 1.2-1, URL https://CRAN.R-project.org/package=fastICA.

Zhao JH, colleagues with inputs from Kurt Hornik, Ripley B (2015). ***gap**: Genetic Analysis Package*. R package version 1.1-16, URL https://CRAN.R-project.org/package=gap.

Affiliation:

Thomas Charlon
Stochastic Information Processing group
Department of Computer Science
University of Geneva
1227 Carouge, Switzerland
E-mail: charlon@protonmail.com

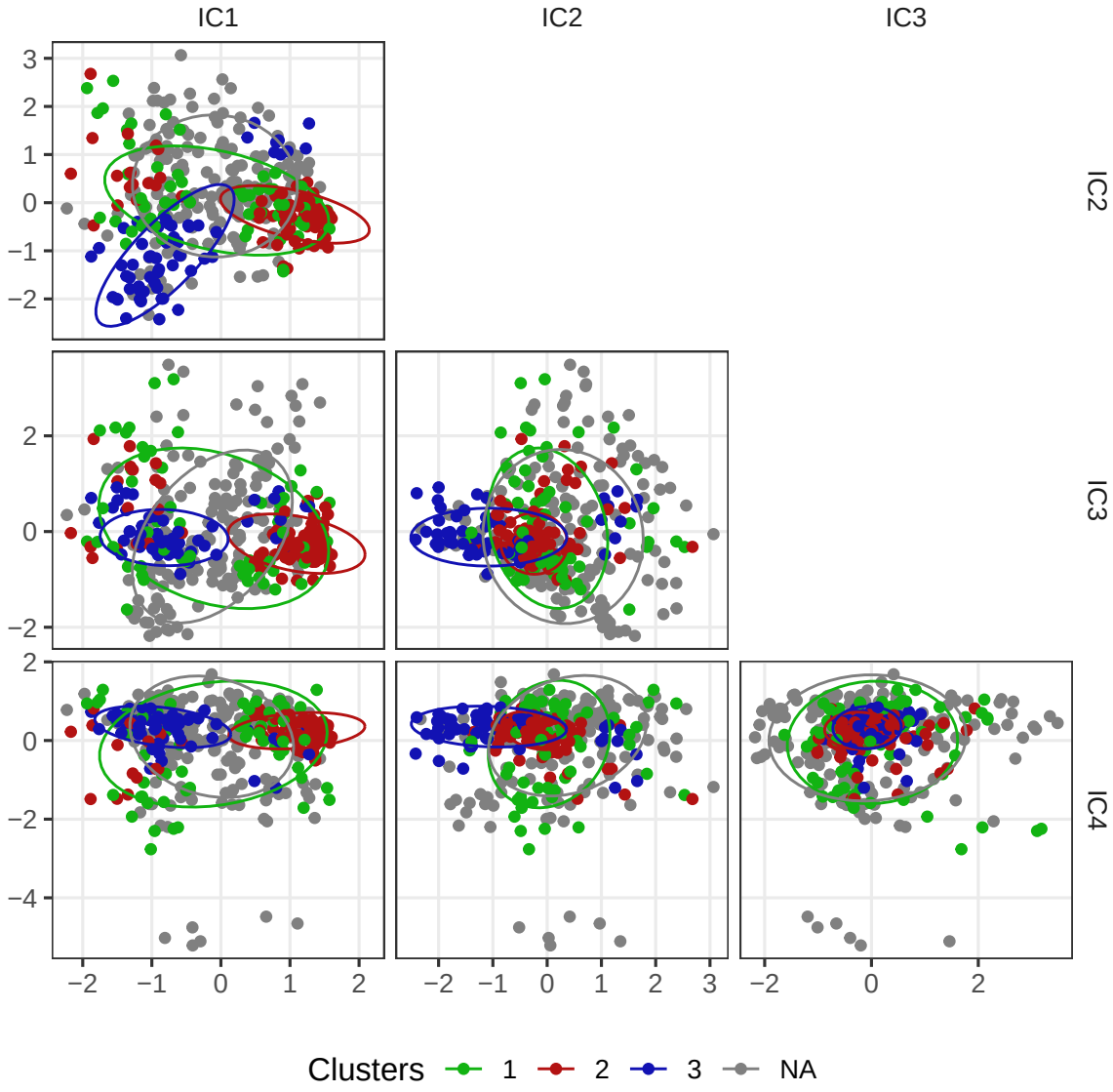

Figure 11: ICA with 4 axes of the Crohn dataset, colored by k-Xi clustering with 75% confidence ellipses. Clusters 1 and 2 are hierarchical and are discriminated from cluster 3 mostly by the second and third components.

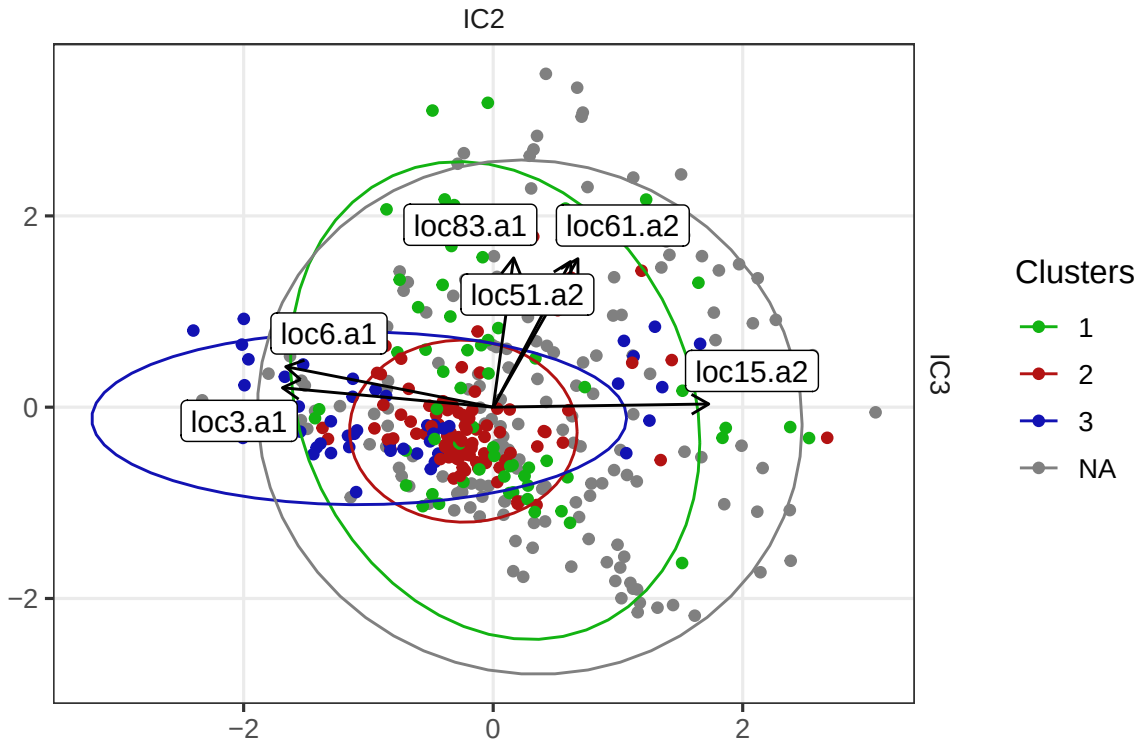

Figure 12: ICA with 2 axes of the Crohn dataset, colored by k-Xi clustering with 95% confidence ellipses.