

# Tool for Visual Cluster Analysis and Consensus Clustering

Christian Permann

Faculty of Computer Science, University of Vienna,  
Währinger Straße 29, 1090 Vienna

TBD.04.2020

# Introduction

## Clustering:

- ▶ Grouping data-points such that their underlying relationships are reflected
- ▶ Gaining knowledge through this grouping

The process of clustering is not done when a solution is computed,  
but when the researcher involved:

“... **evaluated**, **understood** and **accepted** the patterns.” (Chen and Liu [2])

## Challenges:

- ▶ Many possibilities for clustering:
  - ▶ Algorithms/Parameters/Assumptions
- ▶ Choice and interpretation of solution is difficult

## Related Work: Clustering

There is a vast amount of clustering techniques, including:

- ▶ Partition-based methods (KMeans-like algorithms)
- ▶ Hierarchy-based methods (e.g. Joining of Sets/Linking)
- ▶ Density-based methods (e.g. DBSCAN/OPTICS)
  - ▶ Many more...

## Related Work: Visual Frameworks

- ▶ ClusterVision
  - ▶ Ranking solutions according to a combination of quality metrics
  - ▶ Choosing from the highest ranked ones
- ▶ VISTA
  - ▶ In-depth analysis of individual solutions
  - ▶ Possibilities for relabeling of points (ClusterMap)
- ▶ Simple Visualizations
  - ▶ Included in most data-analysis tools
  - ▶ Scatter plots, bar charts, etc.

## Related Work: Consensus Clustering

Combining clustering results may yield a better solution:

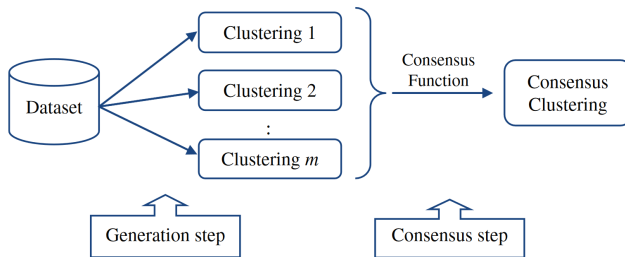


Figure 1: Workflow for generating consensus clusterings [5, p. 340]

## Idea of our Tool: Facilitating clustering exploration

How can we assist users in exploring clustering results?

- ▶ Visualizing individual results
  - ▶ Scatter plot (matrices)/kernel density estimation
  - ▶ Dimensionality reduction
- ▶ Visualizing similarities between results
  - ▶ OPTICS meta-clustering
  - ▶ Heat maps
  - ▶ Multi-Dimensional-Scaling to approximate solution space

## Idea of our Tool: Gathering more Information

Can we gain additional knowledge from multiple computed solutions?

- ▶ Previous frameworks only try to select the best one
  - ▶ Additional information lost
  - ▶ Difficult to objectively identify best one
- ▶ Consensus clustering
  - ▶ Can combine solutions or groups of solutions

Idea:

- ▶ Combine group of robust solutions into one

# The Tool

Three main parts:

- ▶ Data-View
  - ▶ Loading/Saving/Creating data
  - ▶ Cleaning up data
  - ▶ Visualizing data
- ▶ Workflow-View
  - ▶ Creating clustering workflows
  - ▶ Defining parameters
- ▶ Meta-View
  - ▶ Visualizing clusterings and meta-clusterings
  - ▶ Selecting or creating final results (& consensus clustering)

Aim: Facilitating use through clear separation



# The Tool: Data-View

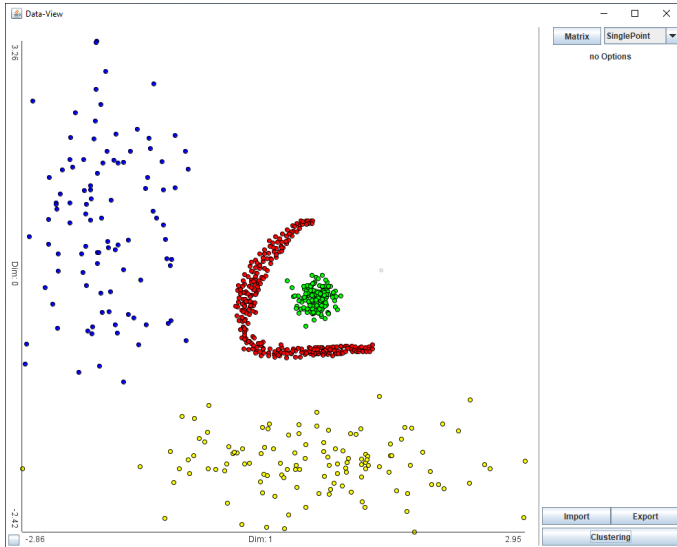


Figure 2: Data-View

# The Tool: Data-View - Scatter Plot Matrix

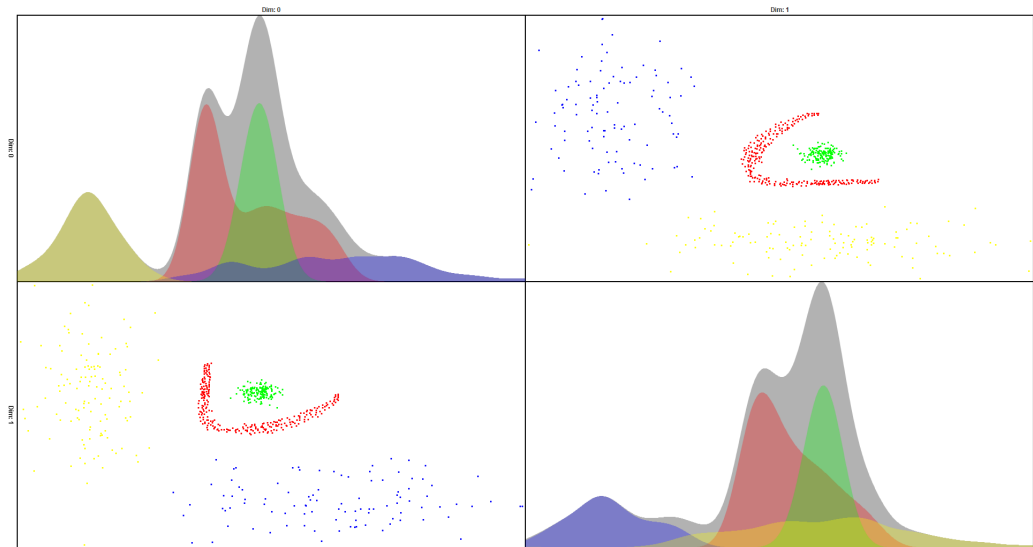


Figure 3: Scatter Plot Matrix

# The Tool: Workflow-View

The screenshot shows the 'Workflow-View' window of a data mining tool. At the top left, there is an 'Add' button and a dropdown menu currently showing 'DBScan'. Below this, a 'Workflow:' section lists three steps, each with an 'X' icon to its left: 'LloydKMeans: k(LB:2 UB:10) Samples each(3)', 'MacQueenKMeans: k(LB:2 UB:10) Samples each(4)', and 'DBScan: minPTS(LB:3 UB:20) Epsilon(LB:0.2 UB:2.0 Samples(100))'. To the right of the workflow list, there are input fields for 'minPTS' (lower bound: 1, upper bound: 1), 'epsilon' (lower bound: 1, upper bound: 1), and 'Samples' (1). At the bottom left, there are input fields for 'MinPts:' (2) and 'Seed:' (5), along with checkboxes for 'Add ground truth', 'Keep trivial solutions', and 'Add trivial solutions' (which is checked). At the bottom center, there is a 'Variation of Information' dropdown menu and a 'Waiting' button. At the bottom right, there are buttons for 'Execute Workflow', 'Confirm', 'Load Wf', and 'Save Wf'.

Workflow-View

Add DBScan

Workflow:

- X LloydKMeans: k(LB:2 UB:10) Samples each(3)
- X MacQueenKMeans: k(LB:2 UB:10) Samples each(4)
- X DBScan: minPTS(LB:3 UB:20) Epsilon(LB:0.2 UB:2.0 Samples(100))

minPTS  
lower bound: 1  
upper bound: 1

epsilon  
lower bound: 1  
upper bound: 1

Samples: 1

MinPts: 2 Seed: 5 ☐ Add ground truth

Eps: -1 ☐ Keep trivial solutions ☒ Add trivial solutions

Variation of Information

Waiting Execute Workflow

Confirm Load Wf Save Wf

Figure 4: Workflow-View

# The Tool: Meta-View

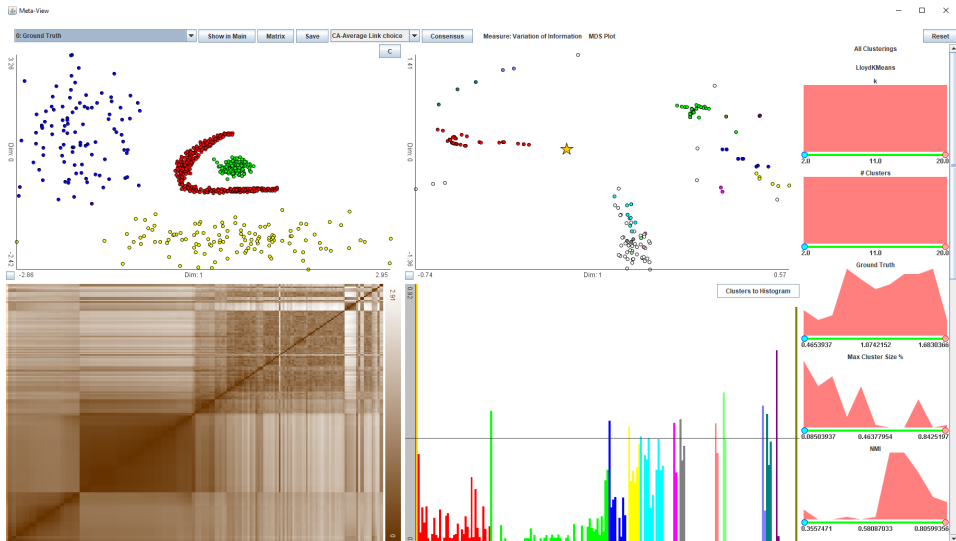


Figure 5: Meta-View

# Implementation

Used tools:

- ▶ Java 1.8, utilizing Streams for parallelization
- ▶ Libraries:
  - ▶ ELKI [1] - Clustering
  - ▶ WEKA [3] - IO
  - ▶ Java Smile [4] - Additional Methods
- ▶ Swing's JComponents and overriding the *draw()* method

Ease of extension:

- ▶ All selectable methods provide simple interfaces

# Tests: Introduction

We want to show that with our tool we can:

- ▶ Produce solutions better than any individual clustering result
- ▶ Obtain solutions unobtainable by single methods
- ▶ Find multiple alternative solutions which can be analyzed to find a fitting choice

And do so in a straightforward and useful way:

- ▶ Letting a user test our tool
- ▶ Also showing real world test data-sets

# Tests: Better than individual Solutions

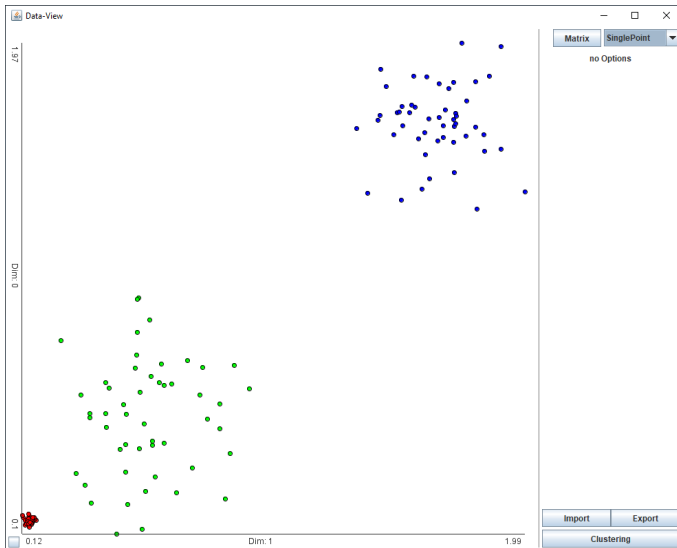


Figure 6: Synthetic Data with Ground Truth

## Tests: Better than individual Solutions

Best individual result when sampling Lloyd's k-Means algorithm with  $k = 2 \dots 20$  and 6 samples per  $k$ :

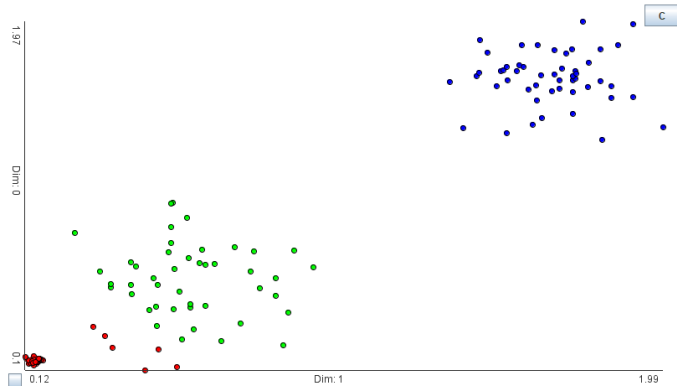


Figure 7: Result of best k-Means run for example Data-Set

Combining all solutions finds the ground truth exactly (without defining  $k$ )



# Tests: Unobtainable Solutions

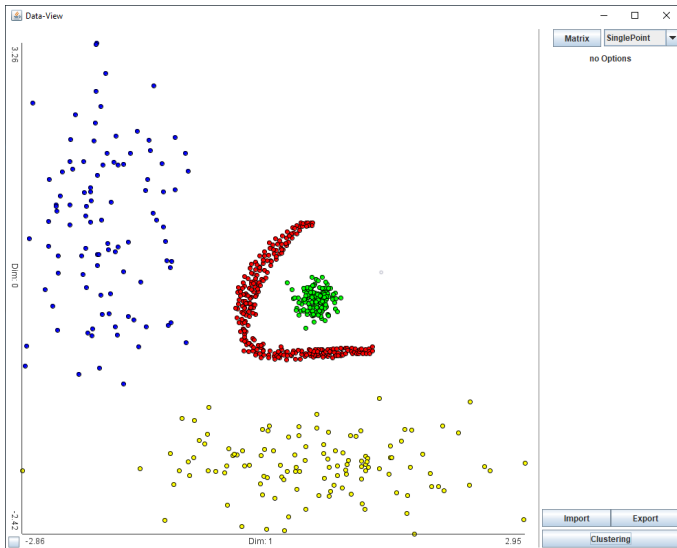


Figure 8: Synthetic Data with Ground Truth

# Tests: Unobtainable Solutions

Workflow:

- X LloydKMeans: k{LB:2 UB:20} Samples each{5}
- X DBScan: minPTS{LB:5 UB:5} Epsilon{LB:0.01 UB:0.5} Samples{100}

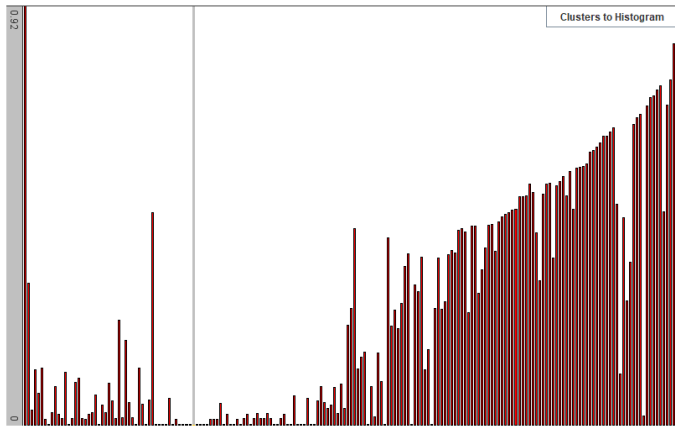
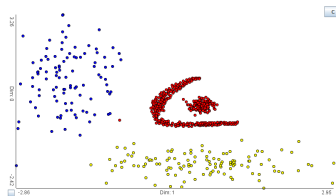
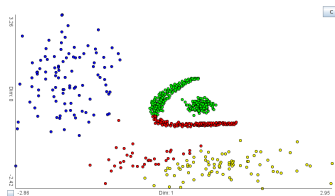


Figure 9: OPTICS reachability plot

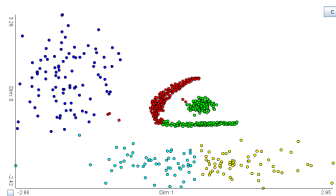
# Tests: Unobtainable Solutions



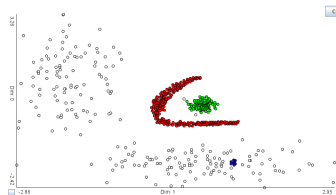
(a) K-Means with  $k = 3$



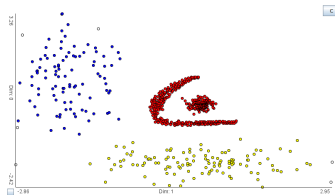
(b) K-Means with  $k = 4$



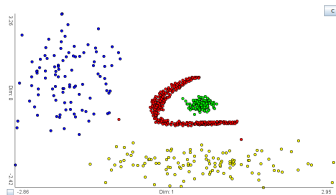
(c) K-Means with  $k = 5$



(d) DBSCAN, best single Result



(e) Selected robust Clustering



(f) Consensus Result

Figure 10: Single Clustering results for Data-Set

# Tests: Multiple Solutions

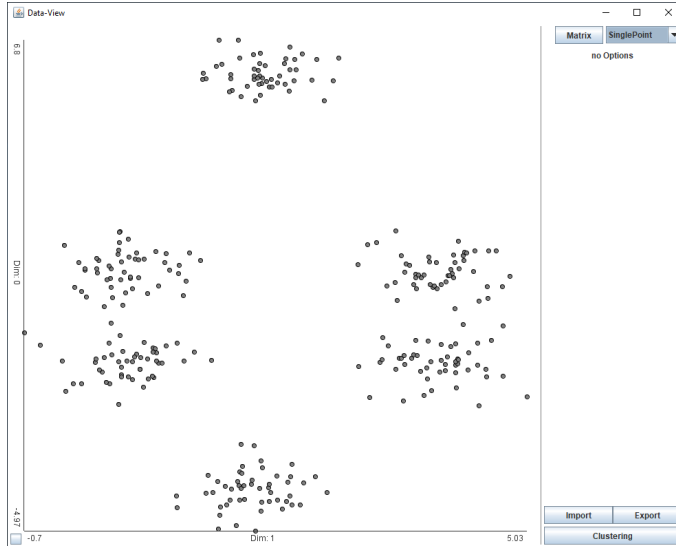


Figure 11: Example Data-Set with unknown Labels

## Tests: Multiple Solutions

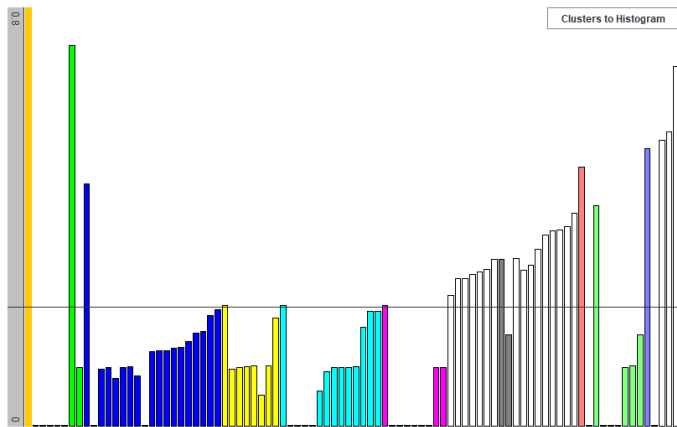
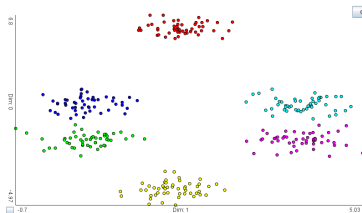
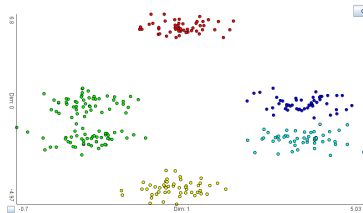


Figure 12: OPTICS Plot

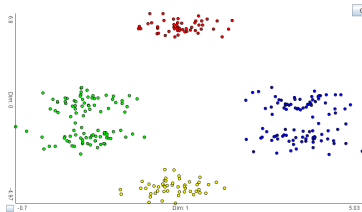
# Tests: Multiple Solutions



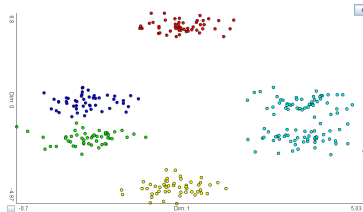
(a) Blue Cluster (19 base clu.)



(b) Light-blue Cluster (14 base clu.)



(c) Pink Cluster (9 base clu.)



(d) Yellow Cluster (8 base clu.)

Figure 13: Consensus Clustering results

# Tests: User & Real world data

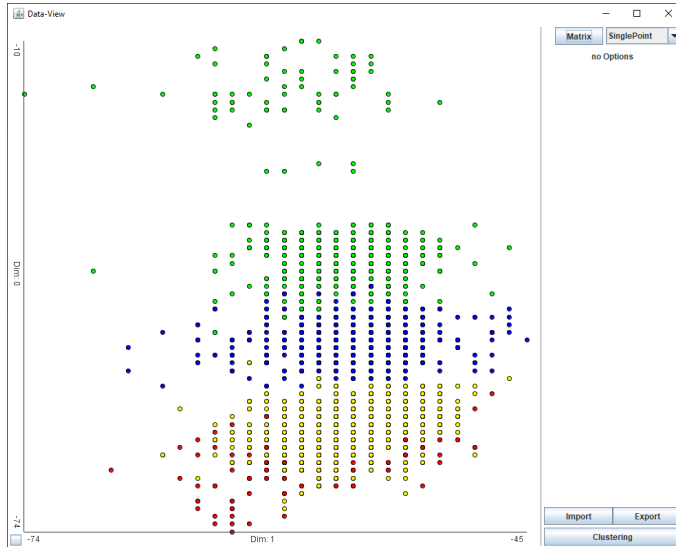


Figure 14: WiFi Localization Data-Set with first two Dimensions shown

## Tests: User & Real world data

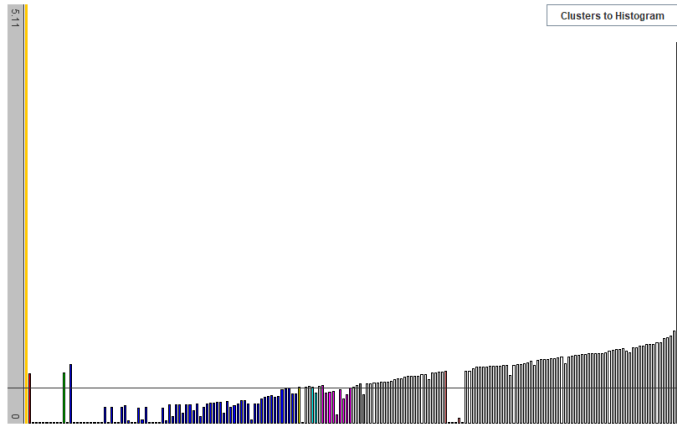
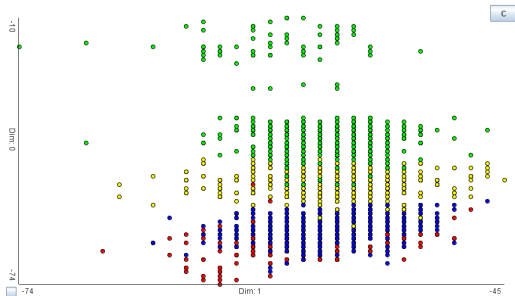


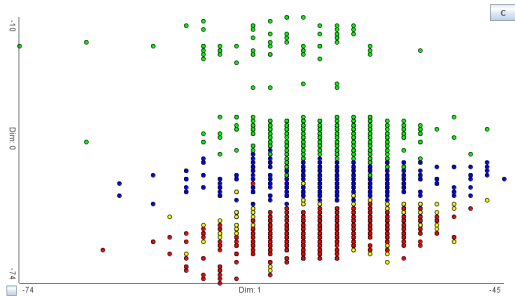
Figure 15: OPTICS Plot for WiFi Localization Data-Set



## Tests: User & Real world data



(a) User Consensus Result:  
NMI: 0.9142  
blue (largest) Meta-Cluster



(b) Best Result of single Clustering Run  
NMI: 0.8904

Figure 16: Clustering results for WiFi Localization Data-Set

# Future Work

Further evaluating usability:

- ▶ Additional study on usability
- ▶ Gathering information on which parts are especially useful
- ▶ Evaluating alternative views and functionality




Research on consensus clustering:

- ▶ Analysis of generation/selection mechanisms
- ▶ Evaluation of selection criteria (is there a better choice than robustness)



# Conclusion

- ▶ We created a new visual tool for cluster analysis:
  - ▶ Visualizing clusterings on a meta-level
  - ▶ Showing groups of robust clusterings
  - ▶ Allowing to find solutions using consensus clustering
- ▶ We showed:
  - ▶ Robust groups indicate good results
  - ▶ Combined results facilitate choice and can be better than any individual result
- ▶ Link to the tool:
  - ▶ [https://github.com/chris9182/Visual\\_Cluster\\_Exploration](https://github.com/chris9182/Visual_Cluster_Exploration)

# References I

-  Elke Achtert, Hans-Peter Kriegel, and Arthur Zimek. “ELKI: A Software System for Evaluation of Subspace Clustering Algorithms”. In: *Proceedings of the 20th International Conference on Scientific and Statistical Database Management*. SSDBM '08. Hong Kong, China: Springer-Verlag, 2008, 580–585. ISBN: 9783540694762. DOI: 10.1007/978-3-540-69497-7\_41. URL: [https://doi.org/10.1007/978-3-540-69497-7\\_41](https://doi.org/10.1007/978-3-540-69497-7_41).
-  Keke Chen and Ling Liu. “A visual framework invites human into clustering process”. In: Aug. 2003, pp. 97 –106. ISBN: 0-7695-1964-4. DOI: 10.1109/SSDM.2003.1214971.
-  Mark Hall et al. “The WEKA Data Mining Software: An Update”. In: *SIGKDD Explor. Newsl.* 11.1 (Nov. 2009), 10–18. ISSN: 1931-0145. DOI: 10.1145/1656274.1656278. URL: <https://doi.org/10.1145/1656274.1656278>.

## References II

-  *Smile - Statistical Machine Intelligence and Learning Engine.*  
<http://haifengl.github.io/>. Accessed: 2020-01-30.
-  Sandro Vega-Pons and José Ruiz-Shulcloper. “A Survey of Clustering Ensemble Algorithms.”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 25 (2011), pp. 337–372. DOI: 10.1142/S0218001411008683.