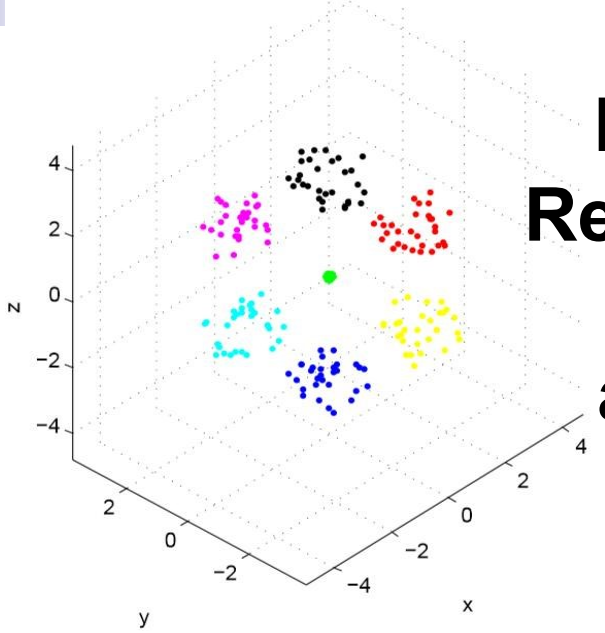
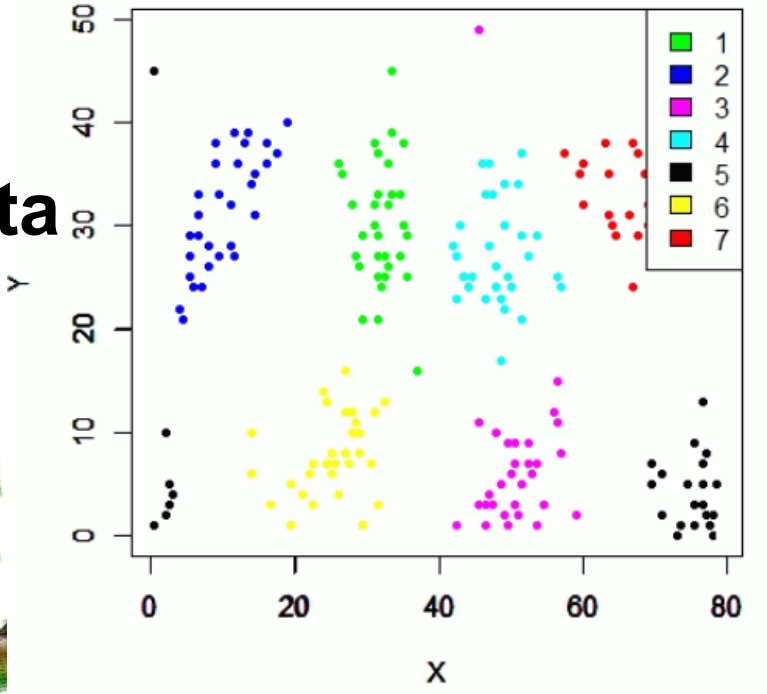
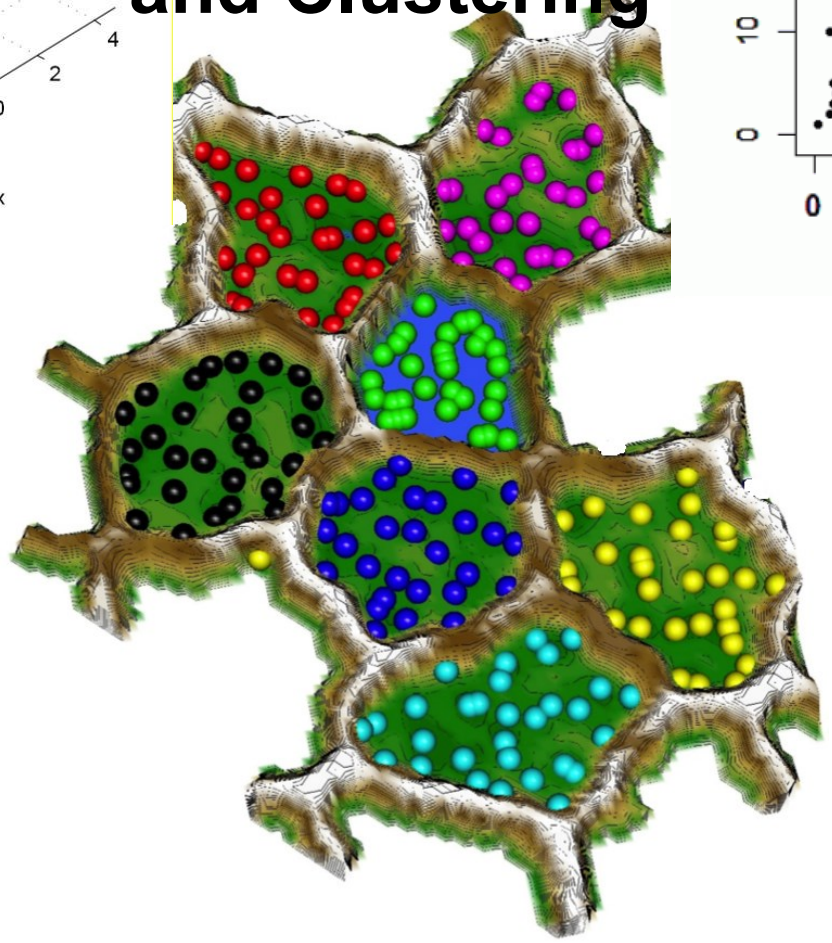
M. Thrun, AG Datenbionik

# DataBionicSwarm (DBS)

# Application: Dimensionality Reduction for Data Visualization and Clustering



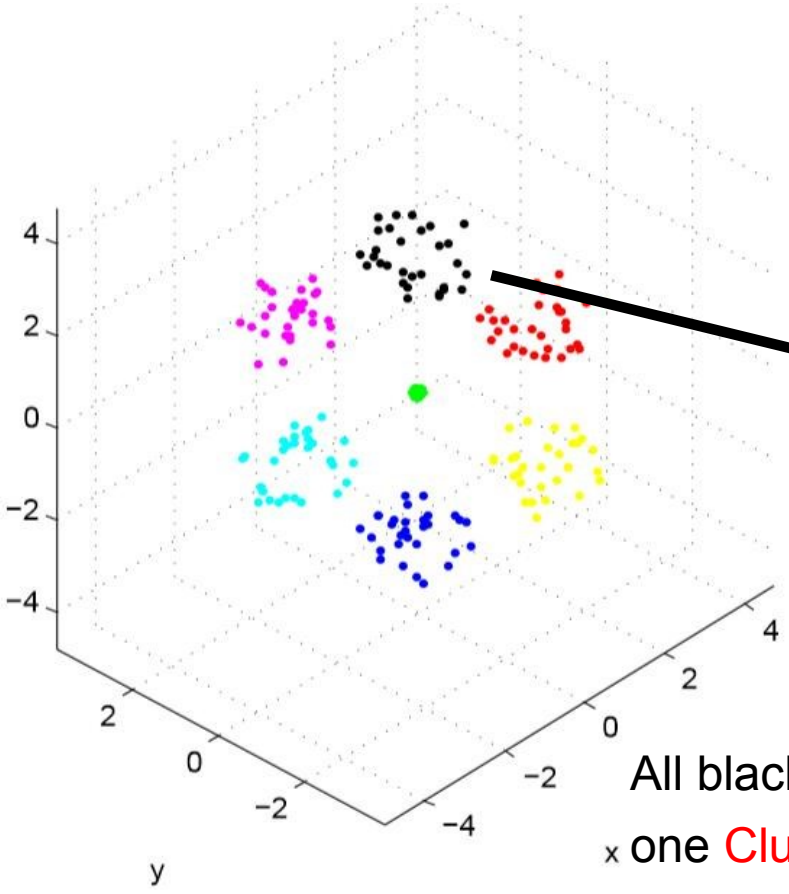



# Problem

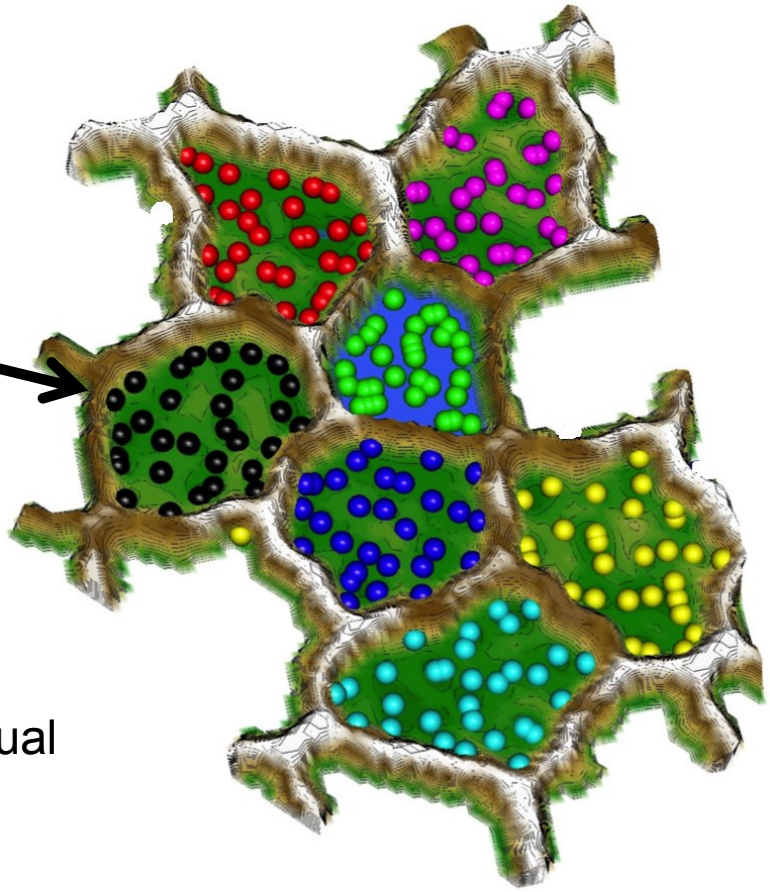
Problem: separate data into similar groups -> **Clustering**

Solution: Visualize data in two dimensions as landscape -> **Projection**

Goal: detect and visualize meaningful cluster structures




All black points equal one Cluster



# Common Concepts of Swarms

1. Swarm intelligence

[Beni 1989], [Cao et al., 1997], [Grosan et al., 2006]

"the **collective behavior** of **many** simple entities called agents"

1. Self-Organization
   - Swarms: [Bonabeau/Dorigo et al., 1999]
   - SOM: [Ultsch 1992]

"spontaneous pattern formation by a system itself, without responsibility of any determinate inside agent"

2. Bionics

[Deneubourg 1991, Reynolds, 1987]

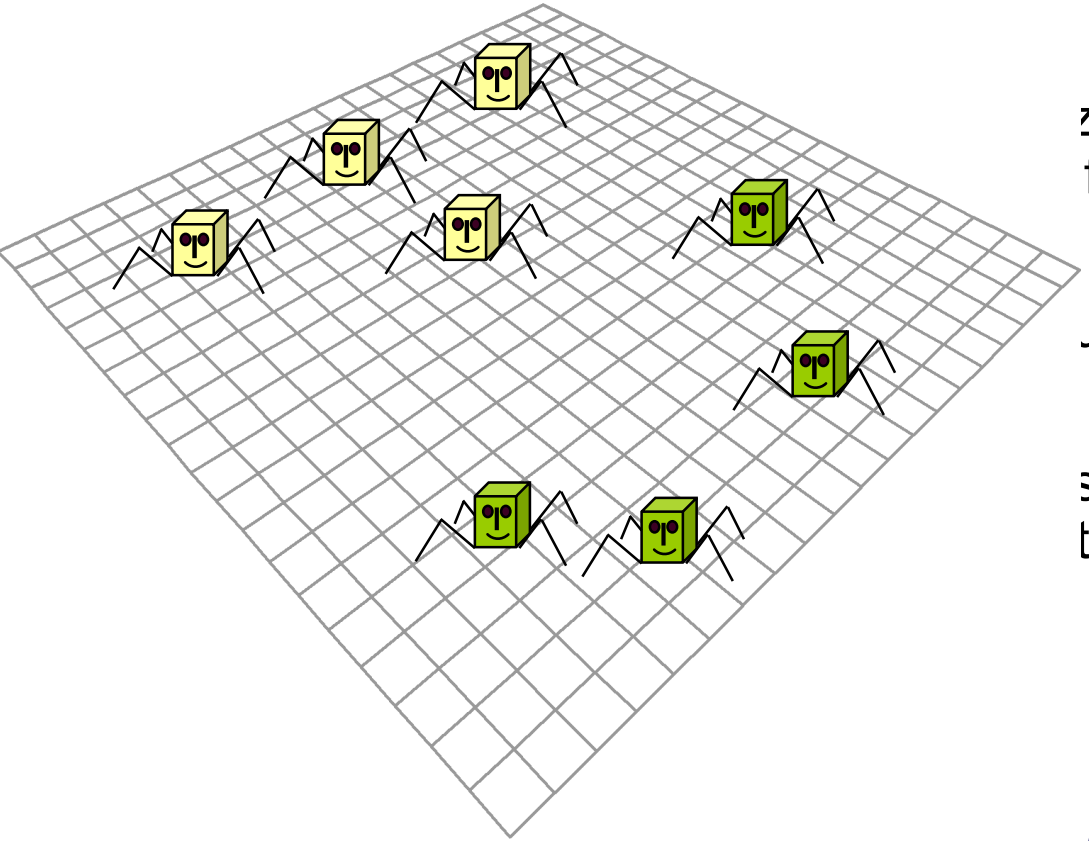
"the application of biological methods and systems found in nature"

# My Ideas

- Combine these three concepts
- Make use of Game theory
- Make use of Emergence

the arising of novel and coherent properties during the process of self-organization

# Prior: DataBots [Ultsch 1999]



jeder DataBot repräsentiert **genau einen** Datensatz

zt auf einer Gitterposition
fänglich zufällig)

urteilt seine Umgebung

sitzt Bewegungsstrategien für jede
ter Richtung

Bewegt sich in Richtung
Verbesserung

# Introduction into Game theory

- decision-making situations are modeled in which several parties influence each other
- Derive rational decision-making behavior in social conflict situations by using mathematical mpodels
- Used in economics (Operations Research), political science, and psychology, biology and poker
- 8 Nobel Memorial Prices
- Important representatives: John von Neumann and Oskar Morgenstern (“Theory of Games and Economic Behavior”), **John Forbes Nash Jr**, Thomas Schelling, Daniel Kahneman
  - RAND Corporation (offer research and analysis to the United States Armed Forces -> space race, nuclear deterrence in Cold war)

- Swarm Intelligence <-> Game theory [Thrun, 2016]

# Game theory

- A game is defined by i=1…n players, where each player makes a choice P
- The choice of each player determines the outcome for each player
  - in general the outcome will be different for different players

=> In a game, the payoff $\lambda_i$ of each player i depends not only on his own choice, but also on the choice of all other players

=> models situations in which multiple players interact or affect each other's outcomes

# Game theory

- the choices $P_i, i = 1, \ldots, n$ are defined as a set of mixed strategies $b_i$ for each player i
- A strategy $b_i' \in P_i$ is the equilibrium, if no deviation in strategy by any single player is more profitable for that player $i$
- For **non-cooperative** games [J. Nash, 1951] proved the existence of at least one equilibrium point:

$$\forall i, \qquad b_i \in P_i : \lambda_i(b_i') \geq \lambda_i(b_i)$$

# Examples

⇒ *a solution concept always exists for **egoistic agents** in a **non cooperative** game*

***What does this mean?***

-> watering garden during drought

What would be an quilibirum for a game?

-> Prisoner's dilemma

# watering garden during drought

- Should the garden be watered?
  - **self-interest (payoff: living garden) <-> restraint (conserving water for community)**
- No matter what the other persons do, a person is always better off watering his garden:
  - unnecessary for one person to exercise restraint if the most other persons are restraining as well
  - Even if the rest of the community doesn't exercise restraint, it is futile for just one person to do so since one person does not have that big of an impact on the whole water supply
- Cooperative game: e.g. fine for watering the garden

# Prisoner's dilemma - requirements

- Two persons A and B imprisoned for a crime.
- Each prisoner is in solitary confinement with no means of communicating with the other.
- The prosecutors lack sufficient evidence to convict the pair on the principal charge
- Decision of each person does not affect future reward/punishment

# Prisoner's dilemma - choices

1. If A and B each betray the other, each of them serves 2 years in prison
2. If A betrays B but B remains silent, A will be set free and B will serve 3 years in prison (and vice versa)
3. If A and B both remain silent, both of them will only serve 1 year in prison (on the lesser charge)

**Rational solution: (3)**

**Nash equilibirium: (1)**

- Details see: ../Subversion\WGR\Hausarbeiten\16Spieltheorie

# Concepts for DataBionicSwarm (DBS)

- Swarm intelligence
- Self-Organization and Emergence
- Bionics
  1. Communication: Scent
  2. Living in and moving on a flat surface
  3. “Ants” wear their colony’s dead -> *will wear data points*
  4. Preference: Smelling the surroundings of ones place
  5. “Ant” moves to a free positions, if it prefers the scent of the new position
- Application of Game Theory
  - Non-cooperative game -> Nash equilibrium

## *How to describe the preference for having neighbors of their own type?*

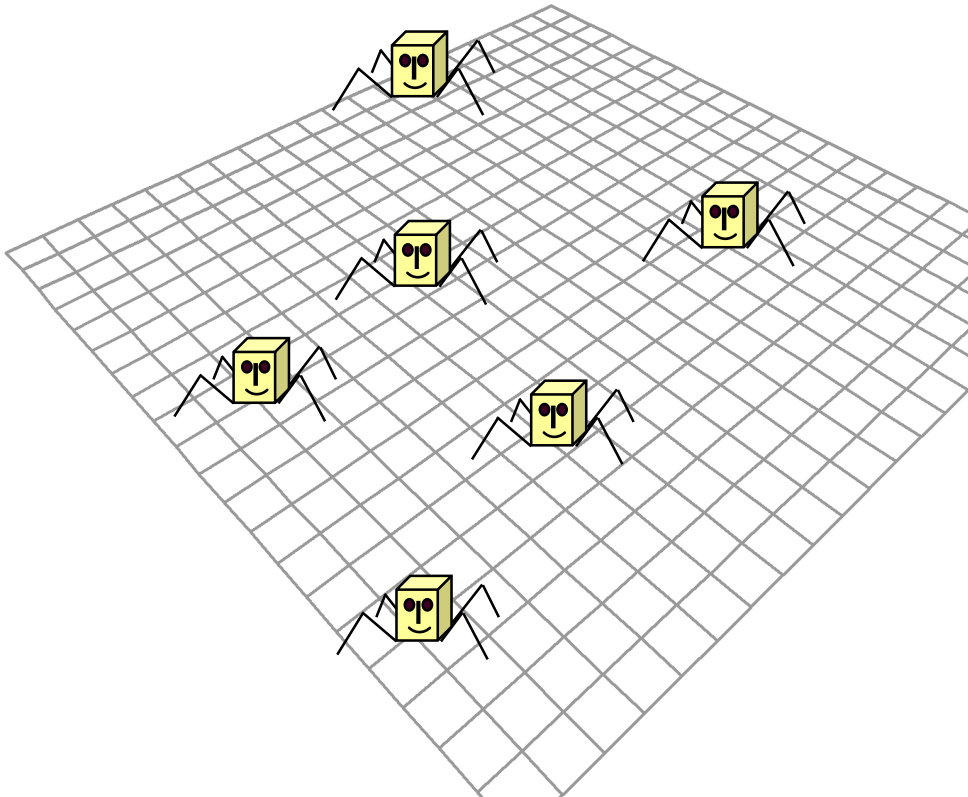
- Requirements for ***preference***
  - Topoligical neighborhood and distance on grid
  - Correct position determination
- Requirements for projection method
  - Focussing scheme with proper reduction of neighborhood -> Nash equlibrium

*physics:*

- Given problem -> Search first for symmetry
- Solution results in parameter reduction
  - Rmin, Rmax, eppochs, …
  - Neighborfunction
  - Grid form and grid distance

# Databionic Swarm Principle

- A projection based clustering method based on emergent, self-organizing, egoistic, artificial life forms
- The collective is called DataBionic Swarm (DBS)
  - Every entity is defined as a DataBot
- Every DataBot is represented by ex one data point
- DataBots communicate through stigmergy:
  1. live on a two-dimensional toroid gı an environment
  2. are able to smell a position and er scent



# DBS has three interchangeable modules

I. Polar swarm (Pswarm)
   - symmetry considerations
     - Polar Coordinates on a Hexadiagonal toroidal grid
     - Adept definition of distance in two dimensions
   - Application of game theory: Scent -> payoff
   - Annealing scheme based on game theory
     - Searches for Nash equilibirum in each eppoch
   - Emergence: No Objective function or parameters (Except $R^n$ distance)

II. Visualization: topographic map with hypsometric tints
   - Simplified ESOM-Algorithmus for Interpolation, parameter- free
   - Automated U*matrix computation
   - 3D landscape generation

III. Semi-automated Clustering

# Scent of DataBots == payoff in game theory

**Hermann 2009 (SOP):**

Let D(l,j) be the distance of $x_l, x_j \in I$, let d(l,j) be the distance in the Output space O, let $F_R$: $R \rightarrow [0,1]$ be an arbitrary but continuous and monotone decreasing function, then the scent $\lambda(b_j, R)$: $\mathbb{R}_0^+ \times O \rightarrow \mathbb{R}_0^+$ is defined with

**Prefernce/Scent:** $\lambda(b_j, R) = \frac{\sum_{l \in I} F_R(d(j,l)) * D(j,l)}{\sum_{l \in I} F_R(d(j,l))}$

**DataBionicSwarm**

**Payoff:** $\lambda(b_j, R, S_0) = \begin{cases} S_0 - \frac{\sum_{l \in I} H_R(r(j,l)) * D(j,l)}{\sum_{l \in I} H_R(r(j,l))}, iff \sum_{l \in W} H_R(r(j,l)) > 0 \\ S_0, \ else \end{cases}$

# Scent of DataBots == payoff in game theory

**Hermann 2009 (SOP):**

Let D(l,j) be the distance of $x_l, x_j \in I$, let d(l,j) be the distance in the Output space O, let $F_R$: $R \rightarrow [0,1]$ be an arbitrary but continuous and monotone decreasing function, then the scent $\lambda(b_j, R)$: $\mathbb{R}_0^+ \times O \rightarrow \mathbb{R}_0^+$ is defined with

**Maximizes structure preservation**

**Prefernce/Scent:** $\lambda(b_j, R) = \frac{\sum_{l \in I} F_R(d(j,l)) * D(j,l)}{\sum_{l \in I} F_R(d(j,l))}$

**DataBionicSwarm**

**Payoff:** $\lambda(b_j, R, S_0) = \begin{cases} S_0 - \frac{\sum_{l \in I} H_R(r(j,l)) * D(j,l)}{\sum_{l \in I} H_R(r(j,l))}, iff \sum_{l \in W} H_R(r(j,l)) > 0 \\ S_0, \ else \end{cases}$

# Scent of DataBots == payoff in game theory

**Hermann 2009 (SOP):**

Let D(l,j) be the distance of $x_l, x_j \in I$, let d(l,j) be the distance in the Output space O, let $F_R: R \rightarrow [0,1]$ be an arbitrary but continuous and monotone decreasing function, then the scent $\lambda(b_j, R): \mathbb{R}_0^+ \times O \rightarrow \mathbb{R}_0^+$ is defined with

**Prefernce/Scent:** $$\lambda(b_j, R) = \frac{\sum_{l \in I} F_R(d(j,l)) * D(j,l)}{\sum_{l \in I} F_R(d(j,l))}$$

**DataBionicSwarm**

**Payoff:** $$\lambda(b_j, R, S_0) = \begin{cases} S_0 - \frac{\sum_{l \in I} H_R(r(j,l)) * D(j,l)}{\sum_{l \in I} H_R(r(j,l))}, iff \sum_{l \in W} H_R(r(j,l)) > 0 \\ S_0, \ else \end{cases}$$

# Algorithm

*function Positions O=pswarm(matrix D(l,j))*

*for all* $z_i \epsilon$ *I: assign an initial random polar position* $i_\phi(r) \epsilon$ *O on the grid*

*to generate DataBots* $b_i \in B$

*for R={Rmax=Lines/2,…,Rmin} do*

*calculate chance c(R)*

*Repeat for each iteration*

$c = sample(c(R), B)$

$m_{\tilde{\imath}}(c) = uniform(1, Rmax)$, $\tilde{\imath}$=1,..,$\alpha$

$l(c) = argmax_j(\lambda(b_j, R))$ *with* $j = \{i, m_{\tilde{\imath}}(c)\} \epsilon O$

$l(!\,c) = i$

$S = \sum_l \lambda_l(b_l, R)$

*until* $\frac{\partial S(e,\lambda(R))}{\partial e} = 0$

*return O in cartesian coordinates*

*end function pswarm*

# Algorithm

*function Positions O=pswarm(matrix D(l,j))*

*for all* $z_i \epsilon$ *I: assign an initial random polar position* $i_\phi(r) \epsilon$ *O on the grid*

*to generate DataBots* $b_i \in B$

*for R={Rmax=Lines/2,…,Rmin} do*

*calculate chance c(R)*

*Repeat for each iteration*

$c = sample(c(R), B)$

$m_{\tilde{\imath}}(c) = uniform(1, Rmax)$, $\tilde{\imath}$=1,..,$\alpha$

$l(c) = argmax_j(\lambda(b_j, R))$ *with* $j = \{i, m_{\tilde{\imath}}(c)\} \epsilon O$

$l(!\,c) = i$

$S = \sum_l \lambda_l(b_l, R)$

*until* $\frac{\partial S(e,\lambda(R))}{\partial e} = 0$

*return O in cartesian coordinates*

*end function pswarm*

*Grid size automatically chosen, contrary to ESOM*

# Details: Define Gridsize

Let the size of the grid defined by Lines L and Columns C, N number of DataBots, $\alpha$ be the number of possible jumping positions, $\beta \in (0.5,1]$, $p_{01}(D)$ robust minum, $p_{99}(D)$ robust maximum

**I.:** $\frac{\sqrt{C^2+L^2}}{1} \geq \frac{\mathrm{p}_{01}(D)}{\mathrm{p}_{99}(D)} =: A$ **shortest/longest distances assignable to grid units**

**II.:** $L * C \geq \alpha * N$ **enough free positions for a DataBot to jump**

**III.:** $\frac{L}{C} = \frac{\beta}{1}$ **rectangular grid**

Resulting in a bi-quadratic equation of $C^4 - A^2 * C^2 + \alpha^2 * N^2 = 0$

$$z_{1/2} = A^2 \pm \frac{1}{2}\sqrt{A^4 - \frac{\alpha^2}{4}N^2}$$

$$\textbf{=>}\ C = \begin{cases} \frac{1}{\sqrt{2}}\sqrt{A^2 + \sqrt{A^4 - \frac{\alpha^2}{4}N^2}}, \text{If } A^4 > \frac{\alpha^2}{4}N^2 \\ approximation, else \end{cases}$$

# Algorithm

*function Positions O=pswarm(matrix D(I,j))*

*for all* $z_i \epsilon$ *I: assign an initial random polar position* $i_\phi(r) \epsilon$ *O on the grid*

*to generate DataBots* $b_i \in B$

*for R={Rmax=Lines/2,…,Rmin} do*

*calculate chance c(R)*

*Repeat for each iteration*

$c = sample(c(R), B)$

$m_{\tilde{\imath}}(c) = uniform(1, Rmax)$, $\tilde{\imath}$=1,..,$\alpha$

$l(c) = argmax_j(\lambda(b_j, R))$ *with* $j = \{i, m_{\tilde{\imath}}(c)\} \epsilon O$

$l(!\,c) = i$

$S = \sum_l \lambda_l(b_l, R)$

*until* $\frac{\partial\, S(e,\lambda(R))}{\partial e} = 0$

*return O in cartesian coordinates*

*end function pswarm*

*Parameter free annealing process*

# Details: *annealing process*

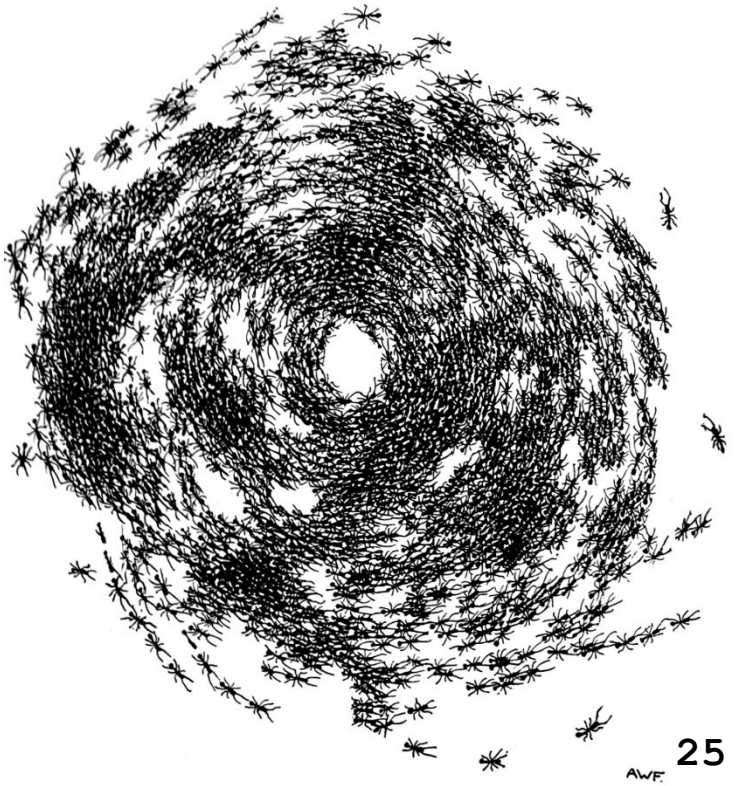
- Precise neighborhood function (polar **(r,phi)** coordinates)

Symmetry considerations (physics): $d(j,l)$ -> radius between l and j and

$$F_R = H_R = \begin{cases} 1 - \dfrac{r(j,l)^2}{\pi R^2}, iff\ \dfrac{r(j,l)^2}{\pi R^2} < 1 \\ 0, else \end{cases}$$

- Rmax=Lines/2 else self interaction

- Rmin >>1 else „ant mill“ [Schneirla 1971]
  - Fixed
  - Determined by grid size,
  - Number of DataBots



# Algorithm

*function Positions O=pswarm(matrix D(I,j))*

*for all $z_i \epsilon$ I: assign an initial random polar position $i_\phi(r) \epsilon$ O on the grid*

*to generate DataBots $b_i \in B$*

*for R={Rmax=Lines/2,…,Rmin}*

*do calculate chance c(R)*

*Repeat for each iteration*

$c = sample(c(R), B)$

$m_{\tilde{\imath}}(c) = uniform(1, Rmax)$, $\tilde{\imath}$=1,..,$\alpha$

$l(c) = argmax_j(\lambda(b_j, R))$ with $j = \{i, m_{\tilde{\imath}}(c)\} \epsilon O$

$l(!\,c) = i$

$S = \sum_l \lambda_l(b_l, R)$

*until* $\frac{\partial S(e,\lambda(R))}{\partial e} = 0$

*TradeOff Batch vs Online*

*Searching for Nash equilibirum*

*return O in cartesian coordinates*

*end function pswarm*

# Algorithm

*function Positions O=pswarm(matrix D(l,j))*

*for all* $z_i \epsilon$ *I: assign an initial random polar position* $i_\phi(r) \epsilon$ *O on the grid*

*to generate DataBots* $b_i \in B$

*for R={Rmax=Lines/2,…,Rmin} do*

*calculate chance c(R)*

*Repeat for each iteration*

$c = sample(c(R), B)$

$m_{\tilde{\imath}}(c) = uniform(1, Rmax)$, $\tilde{\imath}$=1,..,$\alpha$

$l(c) = argmax_j(\lambda(b_j, R))$ *with* $j = \{i, m_{\tilde{\imath}}(c)\} \epsilon O$

$l(!\,c) = i$

$S = \sum_l \lambda_l(b_l, R)$

*until* $\frac{\partial S(e,\lambda(R))}{\partial e} = 0$

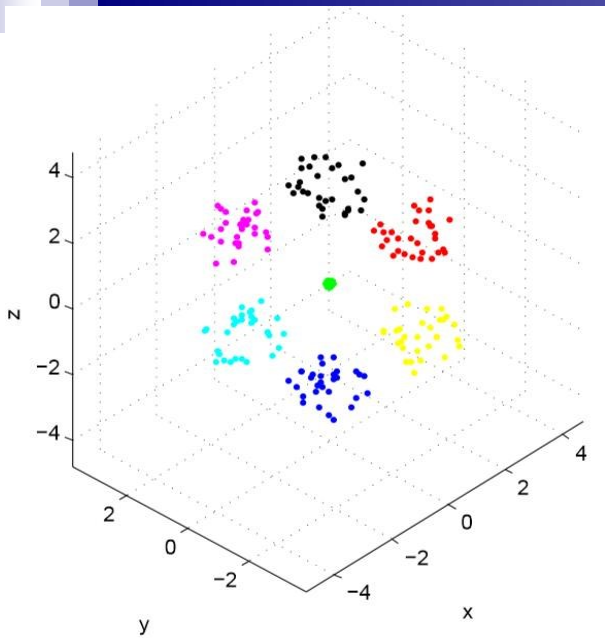
*return O in cartesian coordinates*

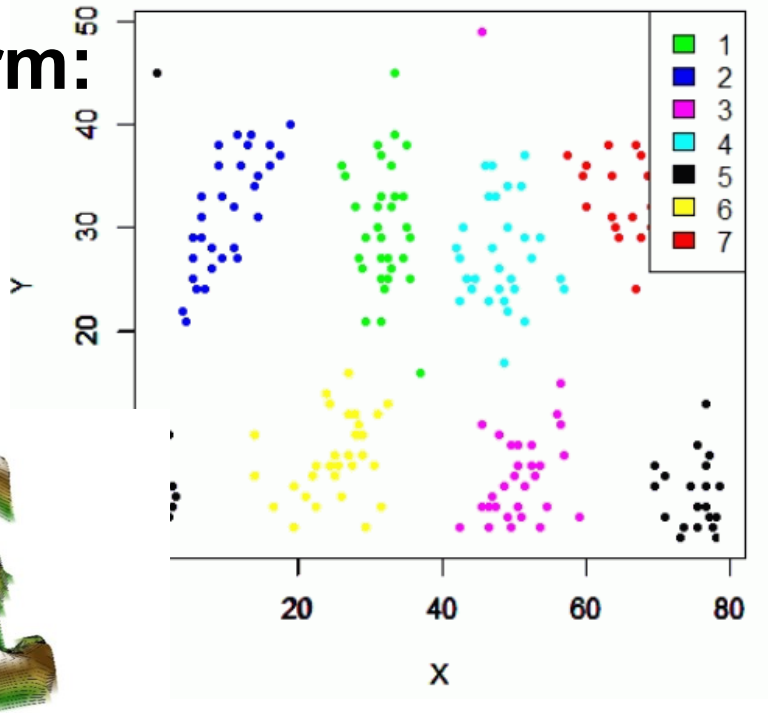
*end function pswarm*

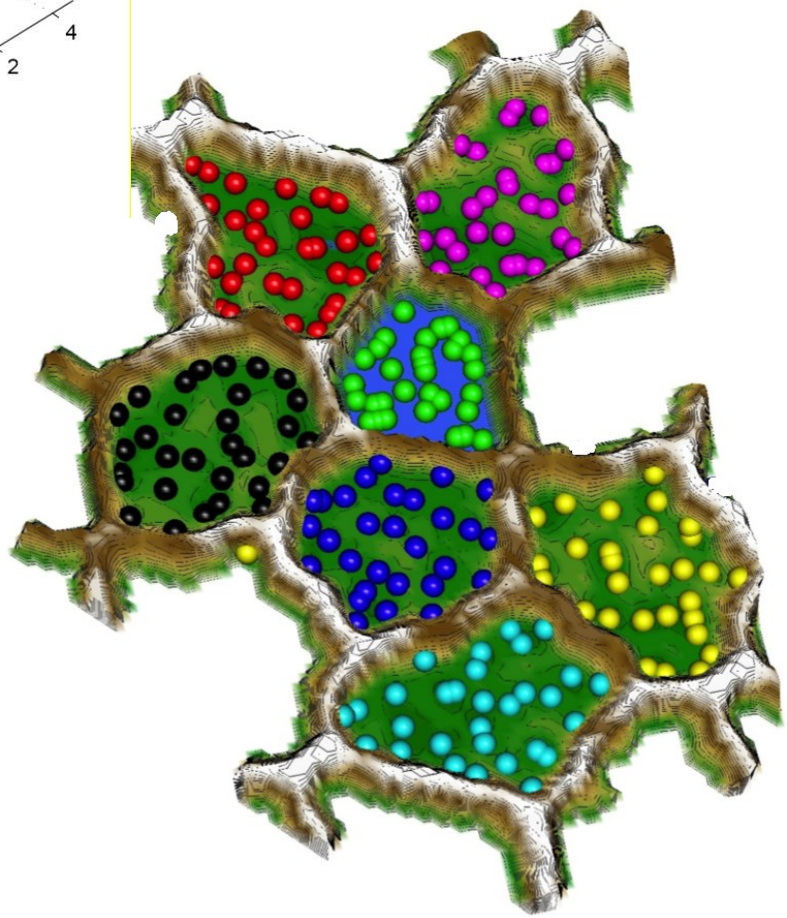
*Choice is* set of mixed strategies

*Egoistic, non-cooperative game*

# Result Pswarm:



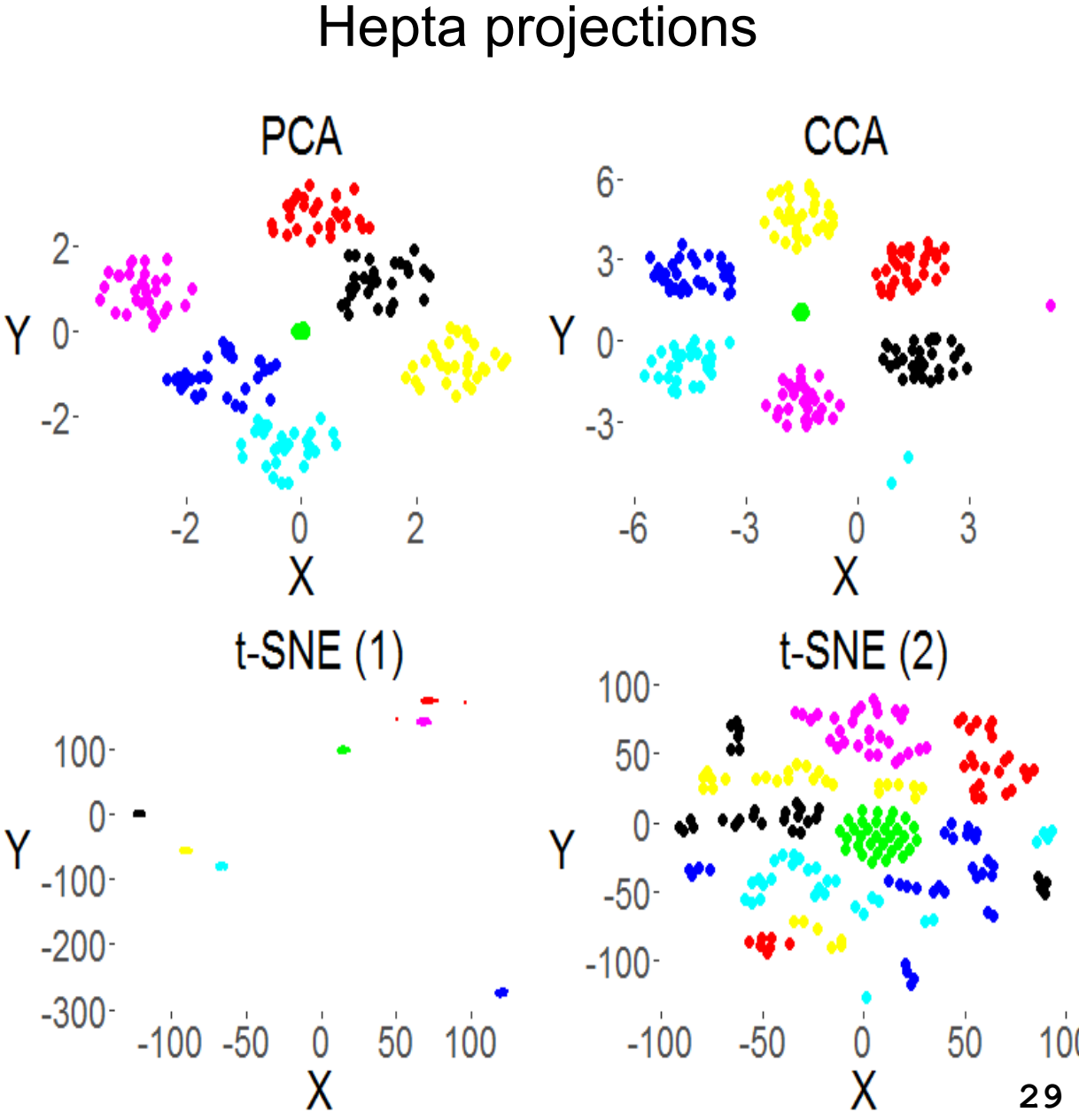



**Why do we want this?**

# Example:

- Choice of projection method
- Choice of parameters for the projection




**Structure preservation:**

**projection**

$R^N \rightarrow R^m$ **with** $N > M$ **cannot preserve all distances!**

# DataBionic Swarm II

- **Contains:**

I. Pswarm

II. Visualization: topographic map with hypsometric tints
   1. Transformation from hexagonal grid to BestMatches
   2. Simplified ESOM-Algorithmus for Interpolation
   3. Automated estimation for P-Matrix Radius [Thrun et al. 2016]
   4. U*matrix computation [Ultsch et al. 2016]
   5. 3D landscape generation [Thrun et al. 2016]
      1. Normalization of U*-Matrix + automatic rectangular Island generation

III. Semi-automated Clustering
   - Uses abstract U-distances [Lötsch/Ultsch, 2014]
   - Hierarchical Clustering [Ultsch et al. 2016]

# Visualization technique: topographic map with hypsometric tints

U*-Matrix Vizualization of Pswarm:

- **considers distance and density based structures of data**

Reason:

- take the structures seen on the U-matrix into account, s.

  Lötsch, J., Ultsch, A.: Exploiting the structures of the U-matrix, 2014

-> Clustering:

- Formalization of the structures through a U-matrix such that
  - distance calculations between best-matching units
  - the height structures of a U-matrix (U-cell distance).
- This enables
  - the assessment of structure preservation and the implementation of clustering algorithms.

# Simplified ESOM-Algorithmus for Interpolation

Two Problems:

1.)

ESOM algorithm is about a Best Matching Units (BMUs) search

But Pswarm results in given BMUs

2.)

In the normal ESOM algorithm following parameters have to be set

- Neighborfunction
- maxEppochs
- AnfangsLernrate
- EndLernrate
- CoolingMethod
- AnfangsRadius
- EndRadius
- Lines and Columns

# Detail1: ProjectedPoints to BestMatches

Let the Grid begin on the x-axis at 1 and end with a maximal number defined as Columns, similar for Lines, then

$$\frac{Lines - 1}{Columns - 1} \approx \frac{|\max(y) - \min(y)|}{|\max(x) - \min(x)|} = \frac{dy}{dy} = \Delta$$

- **describes, that the grid size should equal the size of the coordinate system as accurate as possible**

**The grid size should be around 4096:**

$$Lines * Columns \geq 4096$$

- The resulting equation to be solved is

$$Lines^2 + Lines(1 + \Delta) - 4096\Delta \geq 0$$

- Resulting in

$$Lines \geq -\frac{1 + \Delta}{2} + \sqrt{\left(\frac{1 + \Delta}{2}\right)^2 + 4096\Delta}$$

# sESOM: Special Case of ESOM

Idea: Let the bestmatches be fix [Mörchen/Utlsch 2005]

My idea: with good neighborhood function

the algorithmus can be simplified

The neighborhood function h is defined by

$$H_R: R \rightarrow [0,1]:$$

$$h = \begin{cases} 1 - \frac{d(j,l)^2}{\pi R^2}, iff \ \frac{d(j,l)^2}{\pi R^2} < 1 \\ 0, else \end{cases}$$

- maxEppochs=10
- Learning rate not required (==1)
- *anealingScheme*='linear' (and only for Radius)
  - CurrentRadius = max(AnfangsRadius-CurrentEpoch,1)
  - AnfangsRadius, EndRadius well and autimatically defined

# Simplified ESOM (sESOM)

Initialisierung

Setze An BMU Positionen der Karte die Gewichtsvektoren = Datensätze

Für jede Eppoche

Führe ESOM-Algorithmus mit oberen Parametern ohne Suche nach bestmatches aus

Ersetze an BMU Positionen der Karte die Gewichtsvektoren= Datensätze

U-Matrix: Standard über Moore-Nachbarschaft

# Details: P-Matrix Radius Abschätzung

- Schätze minimalen, korrekten Radius und maximal möglichen Radius über ABCanalyse[Ultsch/Lötsch 2014] der Distanzen ab.
- u.U. Korrigiere Interaktiv an iPmatrix() nach
- P=InterClusterdistanz/InnerClusterdistanz
- InterClusterdistanz=min(Distanz(GruppeA))
- InnerClusterdistanz=max(Distanz(GruppeC))

# U*matrix

- 3D landscape defined by **U*matrix** [Ultsch et al., 2016]
  - Combines Umatrix and Pmatrix
- Umatrix: folding of high dimensional space [Ultsch/Siemon, 1990]

  **(-)** In literature cited as grey-scaled 2D vizualization (e.g. [Kadim Tasdemir/Merényi, 2012 ] )

  -> Precise colored definition required
- Pmatrix: high dimensional density estimation technique [Ultsch, 2003]

  **(-)** Estimation for hypersphere of radius is trying

  -> use ABCanalsys [Ultsch/Lötsch, 2015]
- => **U*matrix represents distance and density based structures!**

# U*matrix -> 3D landscape

- use colors proposed in [Ultsch, 2003]

**=> hypsometric tints:** surface colors which depict ranges of elevation

- Calculate contour lines
  - Normalization of U*heights, define height intervalls, ...
- Combine specific color scale with contour lines
- Create island of toroid map (rectangular)

**=> topographic map with hypsometric tints**

# (DBS) DataBionic Swarm III

I. Pswarm

II. Visualization: topographic map with hypsometric tints

III. Semi-automated Clustering

- Uses abstract U-distances [Lötsch/Ultsch, 2014]
- My idea: Calculate Shortest Paths between these distances

  -> new distance matrix $\widehat{D}$

- Apply Hierarchical Clustering [Ultsch et al. 2016] to $\widehat{D}$
- My Idea: Only single linkage (SL) or ward relevant:
  - See Compact versus connected structures.

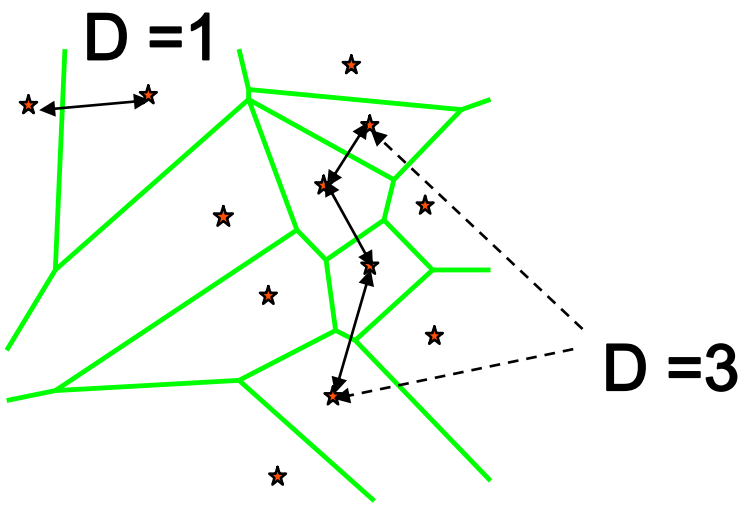
  -> Cluster analysis requires the choice of compact or connected as a parameter
- Second parameter: number of clusters
  - Estimate from dendrogram oder visualization

# Abstract U-distances

- Delaunay Graph: Graph of Voronoi cells
  - A region corresponds to each BestMatch consisting of all points closer to that BestMatch than to any other
- Delaunay Path: Number of edges between two BestMatches
- Djikstra Shortest Paths of Delaunay graph weighted with highdimensional Distances

Reason:

- U-matrix is approximation of Abstract Umatrix [Lötsch/Ultsch 2014] which is based on Voronoi cells
- Height of Voronoi-Borders=Euclidean High-dimensional distance of data

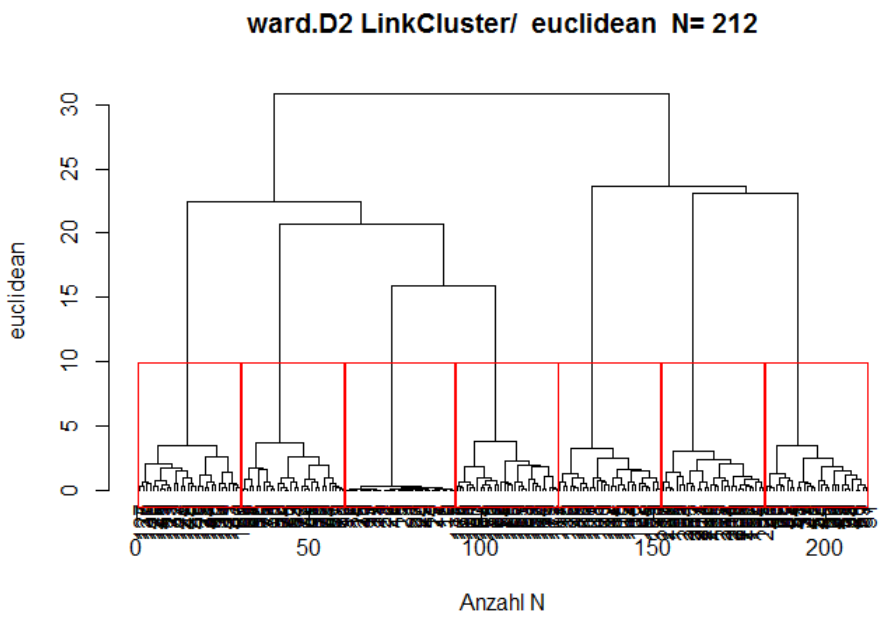





Übernommen von Prof. Ultsch

# Hierarchical Clustering

- **Clustering distance= shortest paths**
- **Aggloremative:**
  - **beginning: every point is one cluster**
    - Calculate „similarity“ between clusters
    - Unite the two most „similar“ clusters to new cluster
  - **Do this until only one cluster exists**
- **Similarity of clusters: depends on algorithm**
- **Evaluation of number of clusters:**
  - **Dendrogram -> „ultrametric“**



# Verification of DBS

- No general error for projections exists (Thesis)
- Statistical: Only through accuracy of given Classification possible
  - Natural datasets: may have different clusters depending on goal

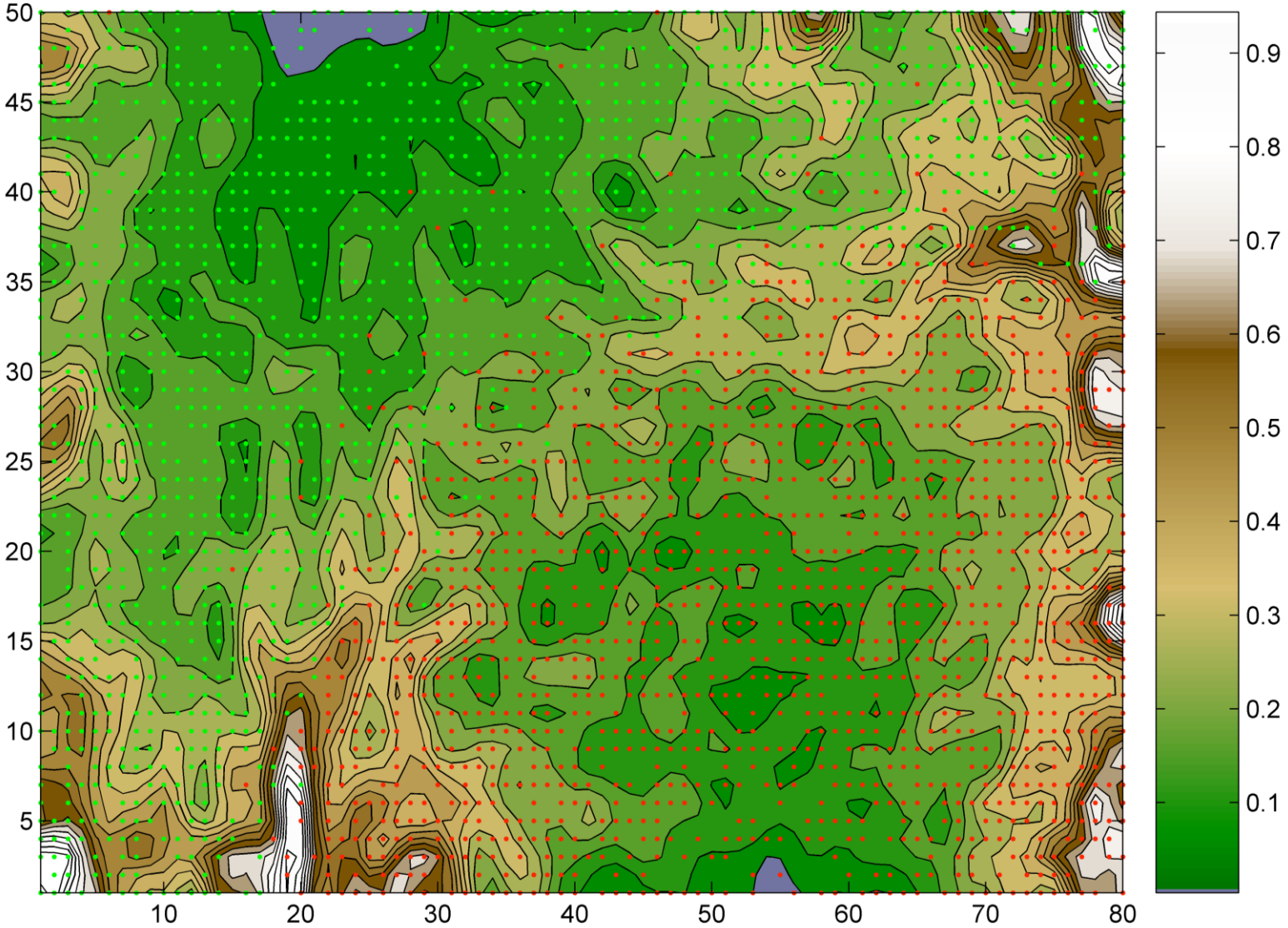
    =>Artifical data sets: Fundamental Clustering Problem Suite (FCPS)
- Visual: 3D Landscape of known datasets
- Application based: Searching for new Knowledge

# Visual Benchmarking on FCPS

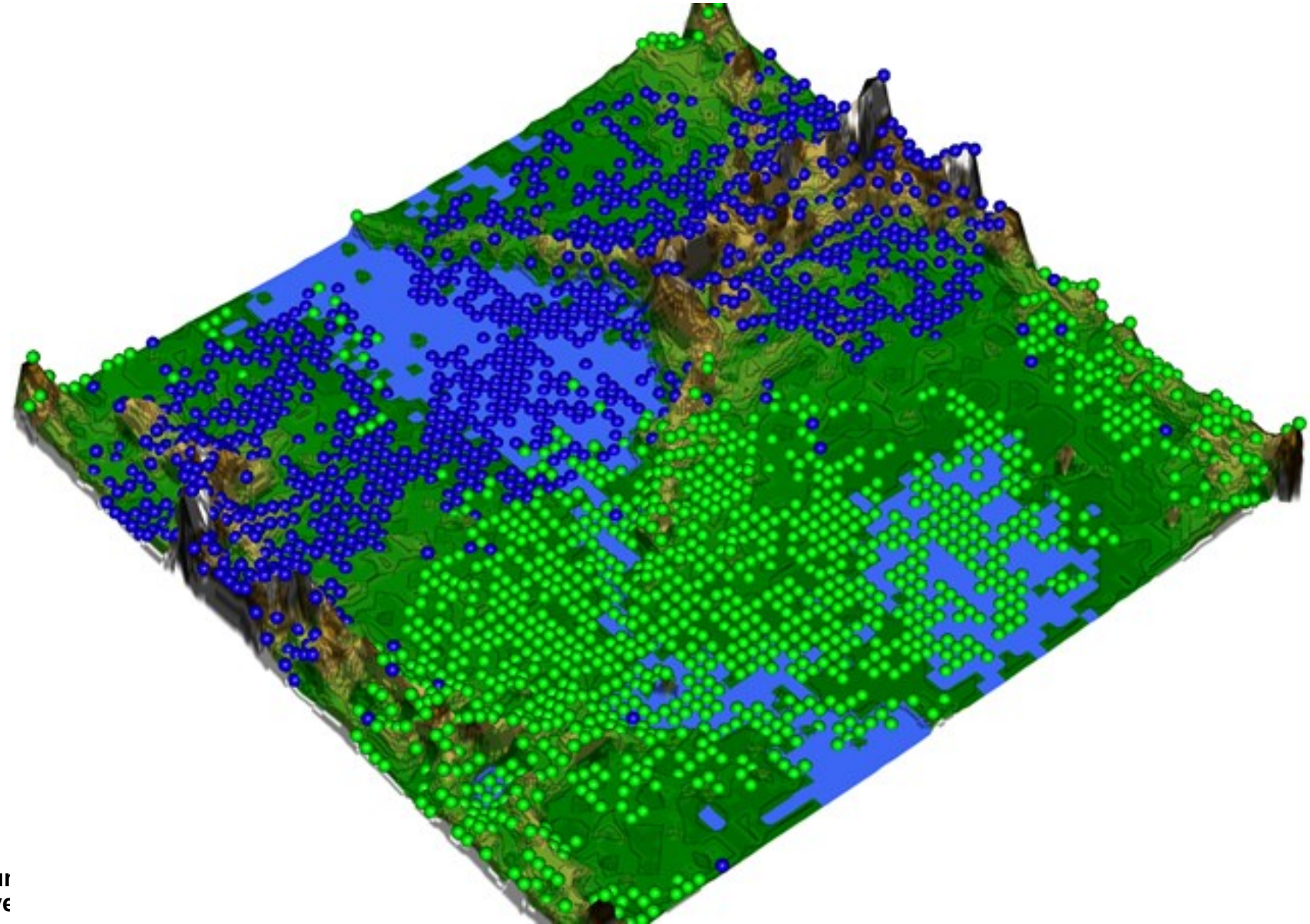
- Using 3d landscape: Topogrpahic map with hypsometric tints

| Data/Projection | ESOM | NerV | SOP | DBS |
|---|---|---|---|---|
| Atom (k=2) | Trial and par-ameter depended | OK | 3 Clusters | OK |
| Chainlink (k=2) | OK | OK | (OK) | OK |
| EngyTime (k=2) | Clusters invisible | PCA init | Intermixed Clusters | OK |
| Lsun (k=3) | (OK) | PCA init | Ok? | OK |
| Golfball (k=1) | OK | Clusters visible | OK | OK |
| WingNut | Too many substructures | PCA init | Ok? | OK |
| Other FCPS data sets | OK | OK | OK | OK |

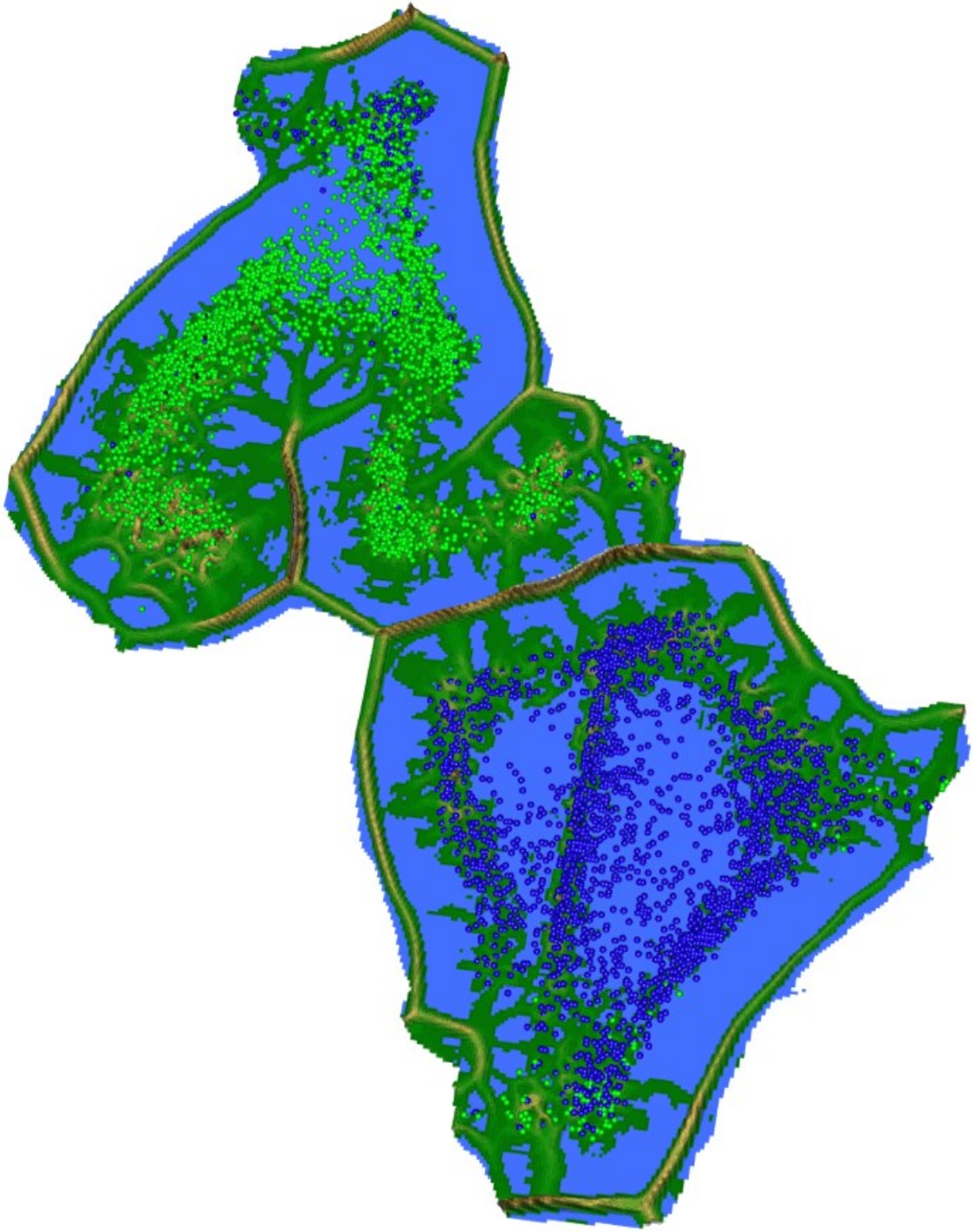
# Example: EngyTime ESOM



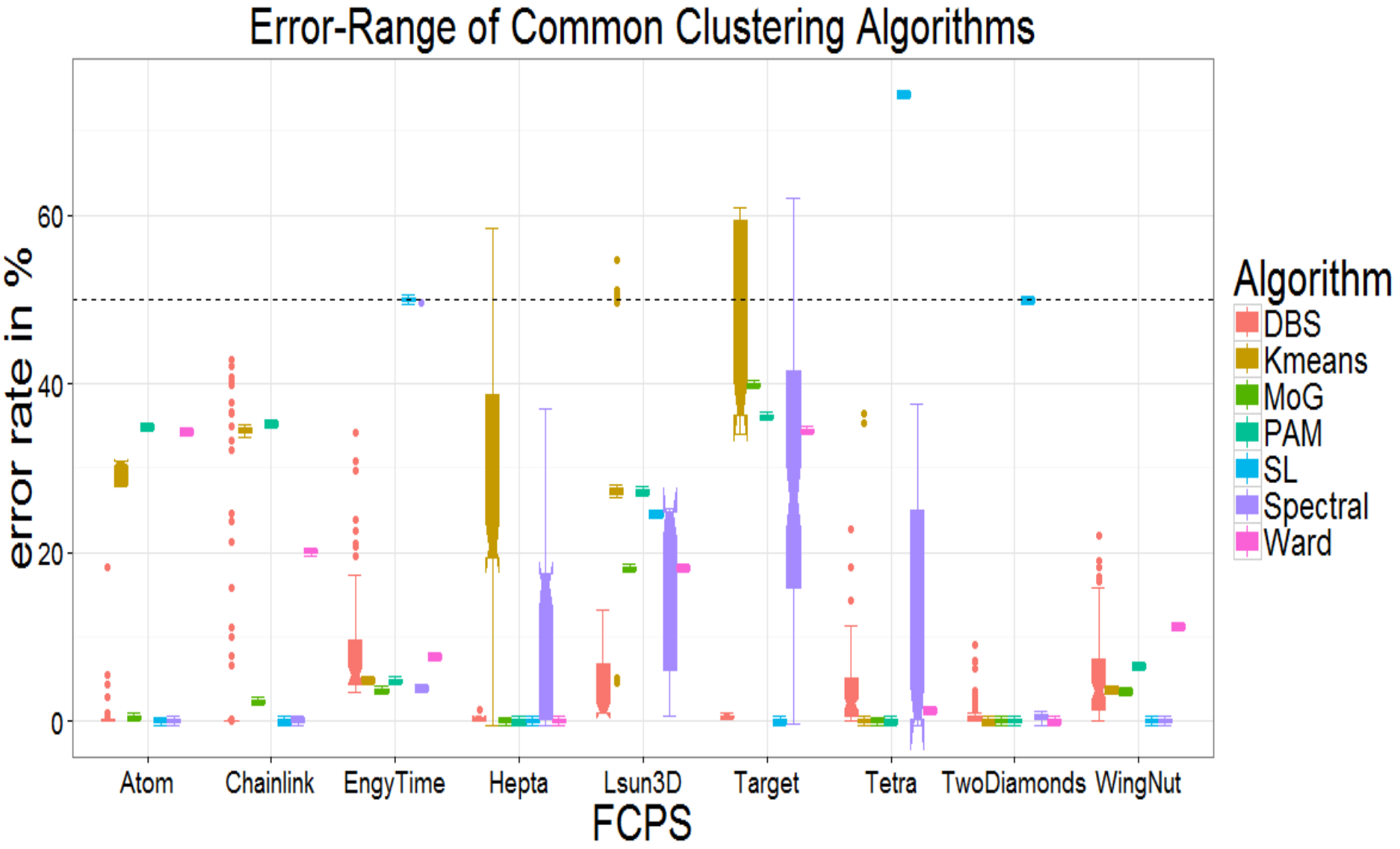
# Example: EngyTime SOP



# DBS EngyTime Projection



# Accuracy of DBS Clustering



# Application I: High-dimensional Data

- Leucemia
- 554 Patients with **7447 active Genes**
- Four diagnosis
- 6 Klassen reproduzieren

Die Vorklassifizierung mit einer Accuracy von 99.6%

- Two Outliers:

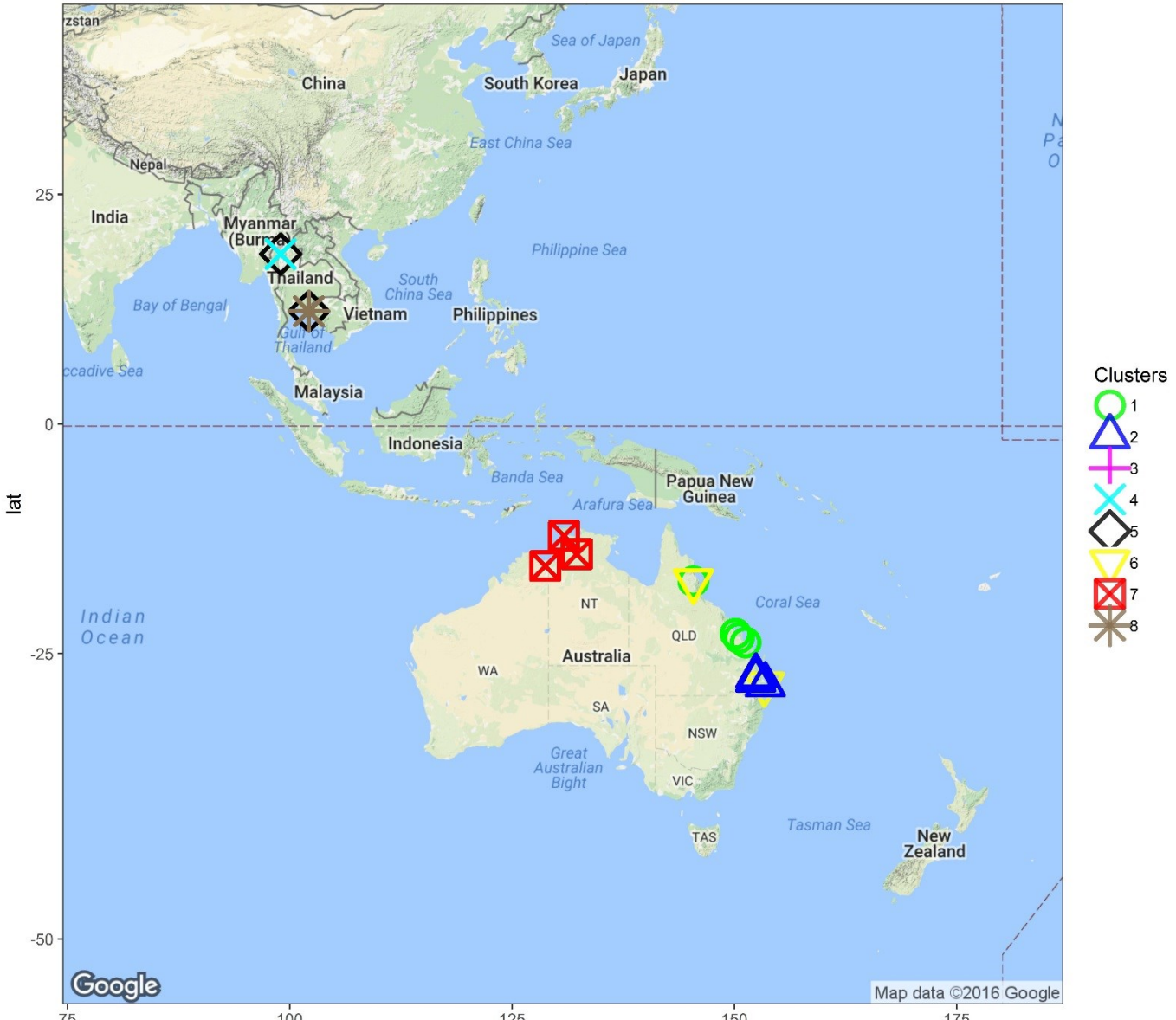
-> problem in diagnosis

-> see our future publication

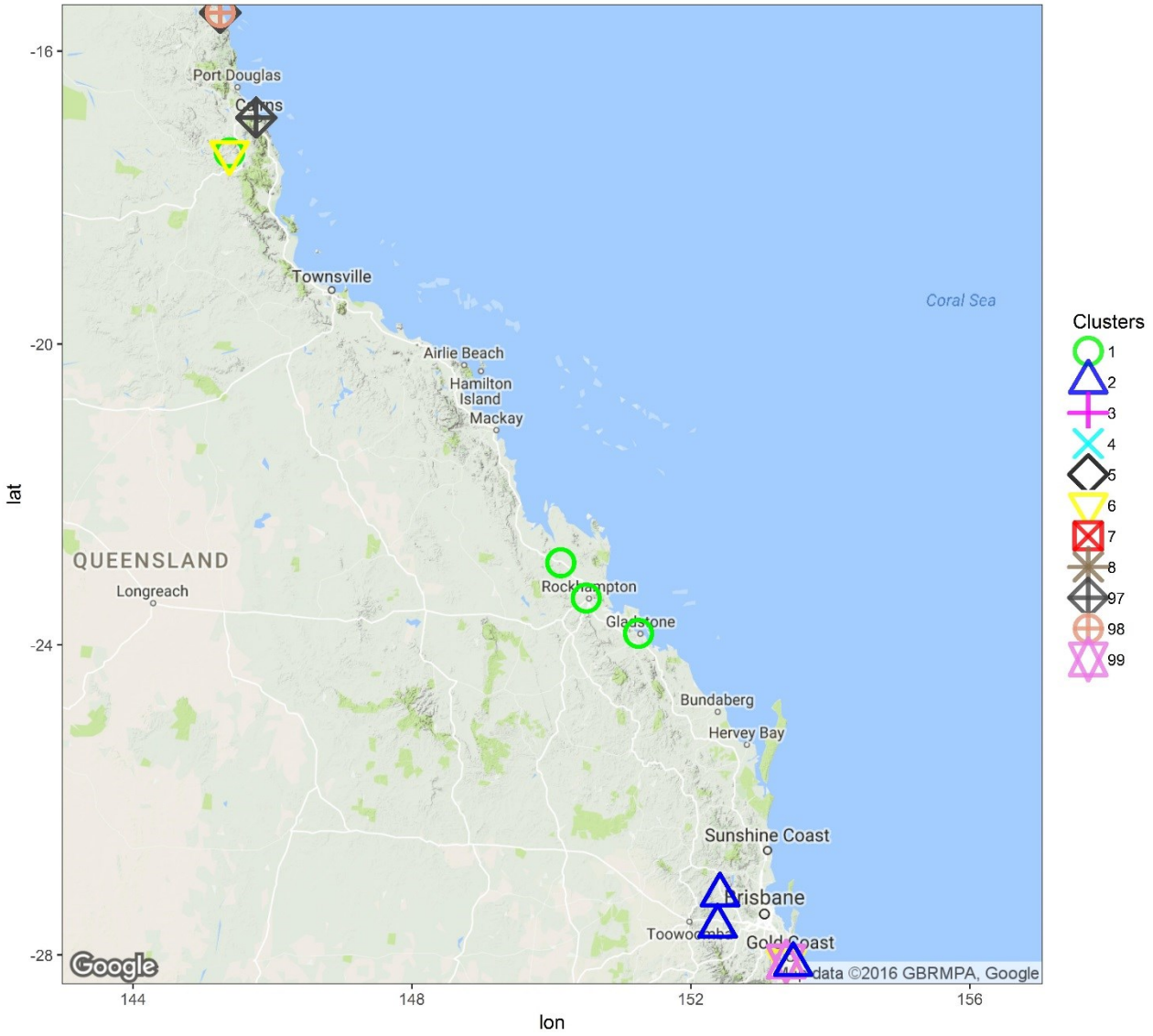
[Brendel, et al.,2017]

# Application 2: Genetic Data of Bees

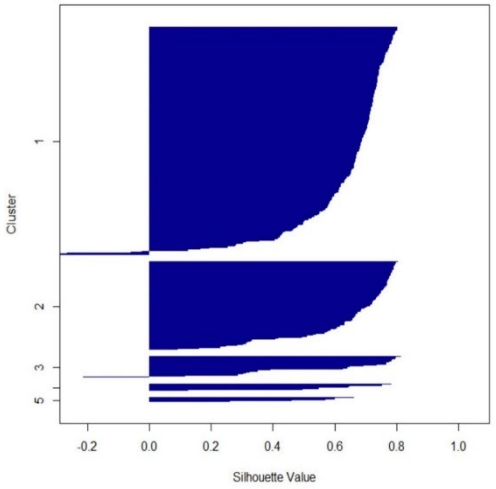
# Main Clusters depend on Location of Bees

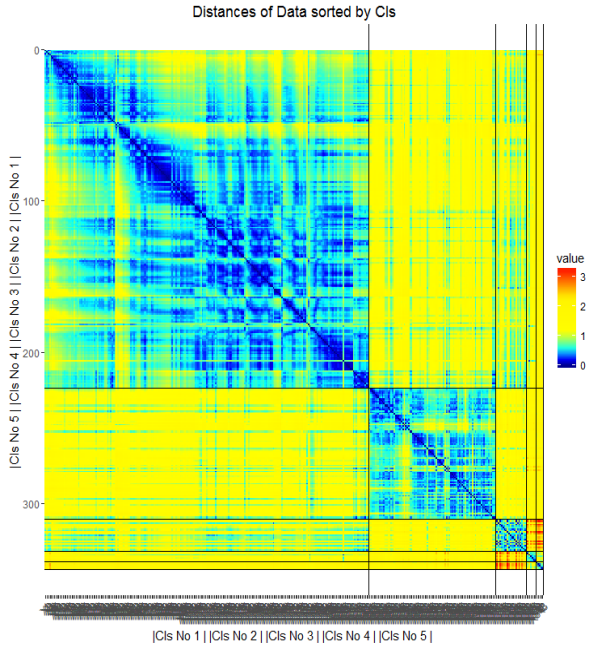


# Outliers are in Queensland

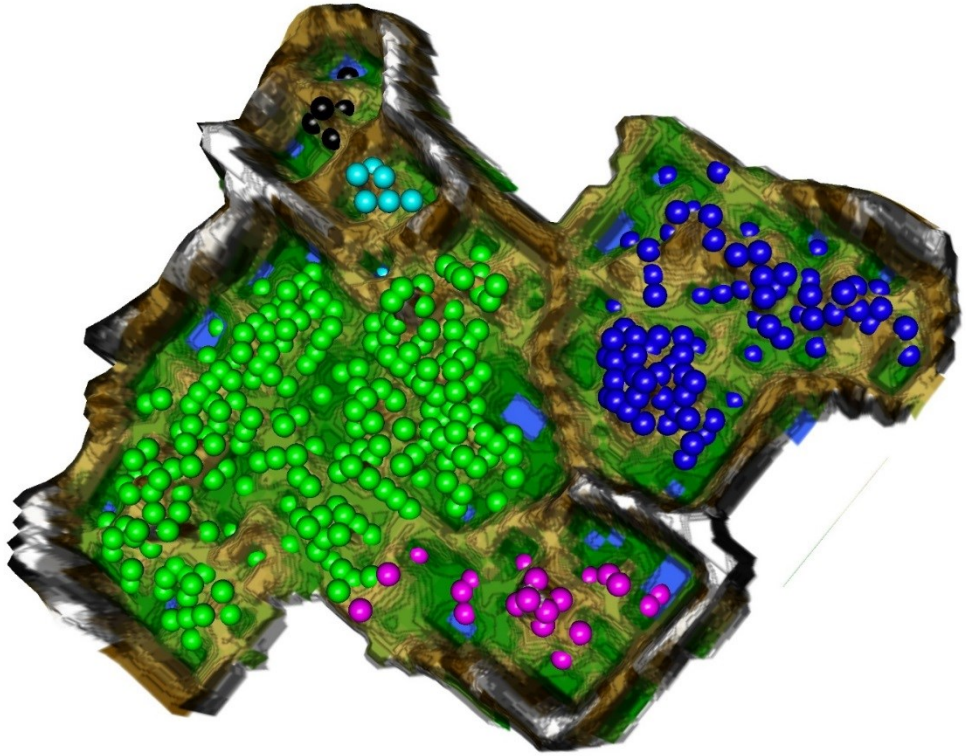


# Knowledge Discovery with DBS I



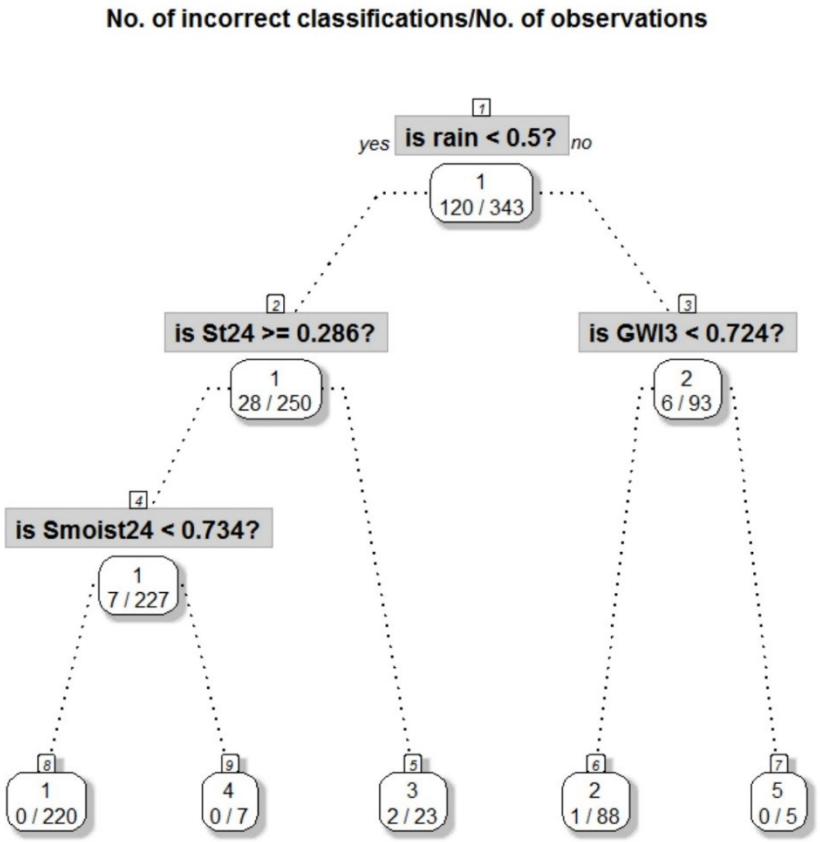




- Daily courses of 14 Variables over 2 years are analyzed
- Five clusters are shown in the visualization of DBS



# Clusters explained I

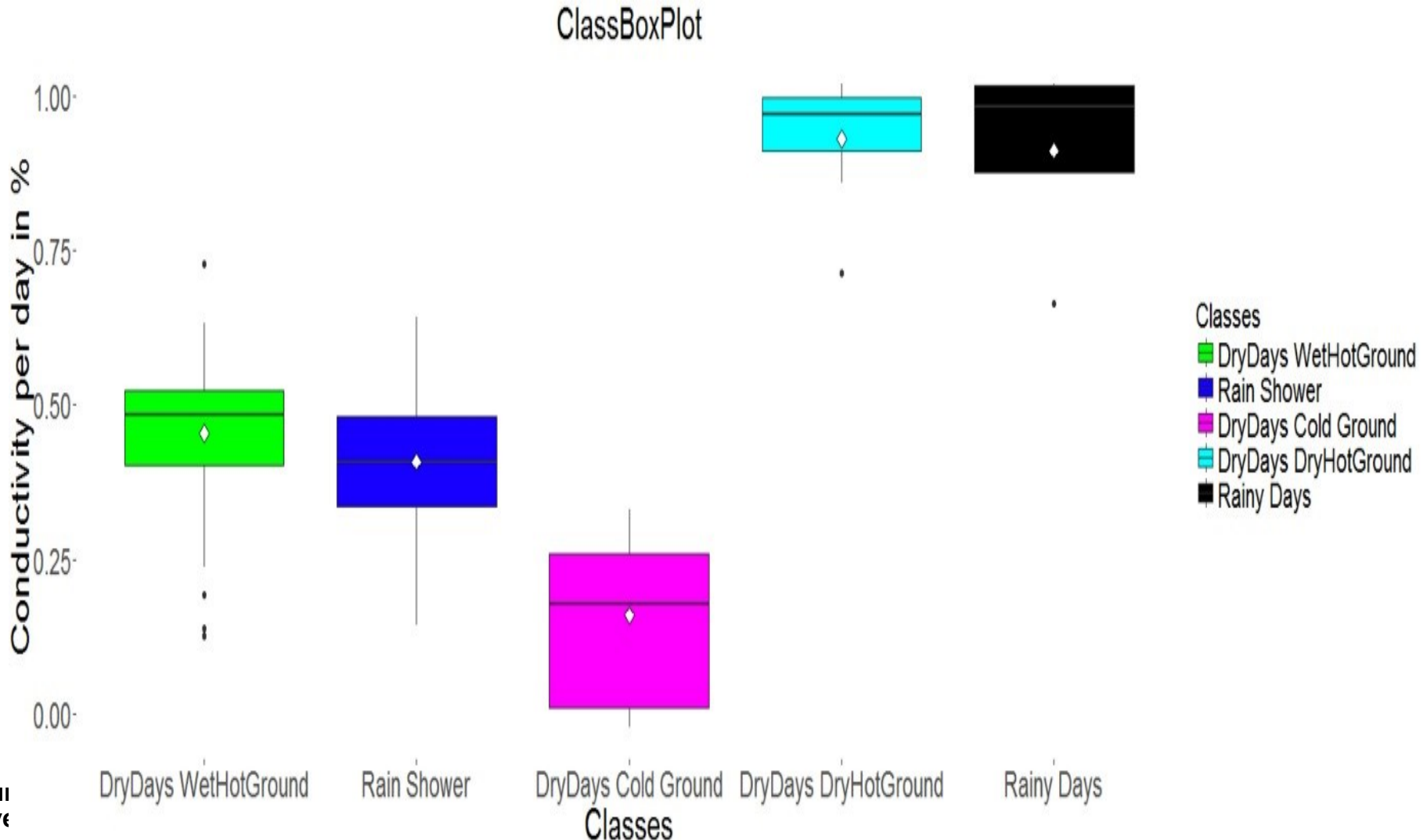
- Apply Cart-Tree to clusters: 99% of data correctly classified
- Knowledge aquisition from a Carrt Tree results in 5 classes



# Clusters explained II

- **Clusters can be explained by electric conductivity in the stream**

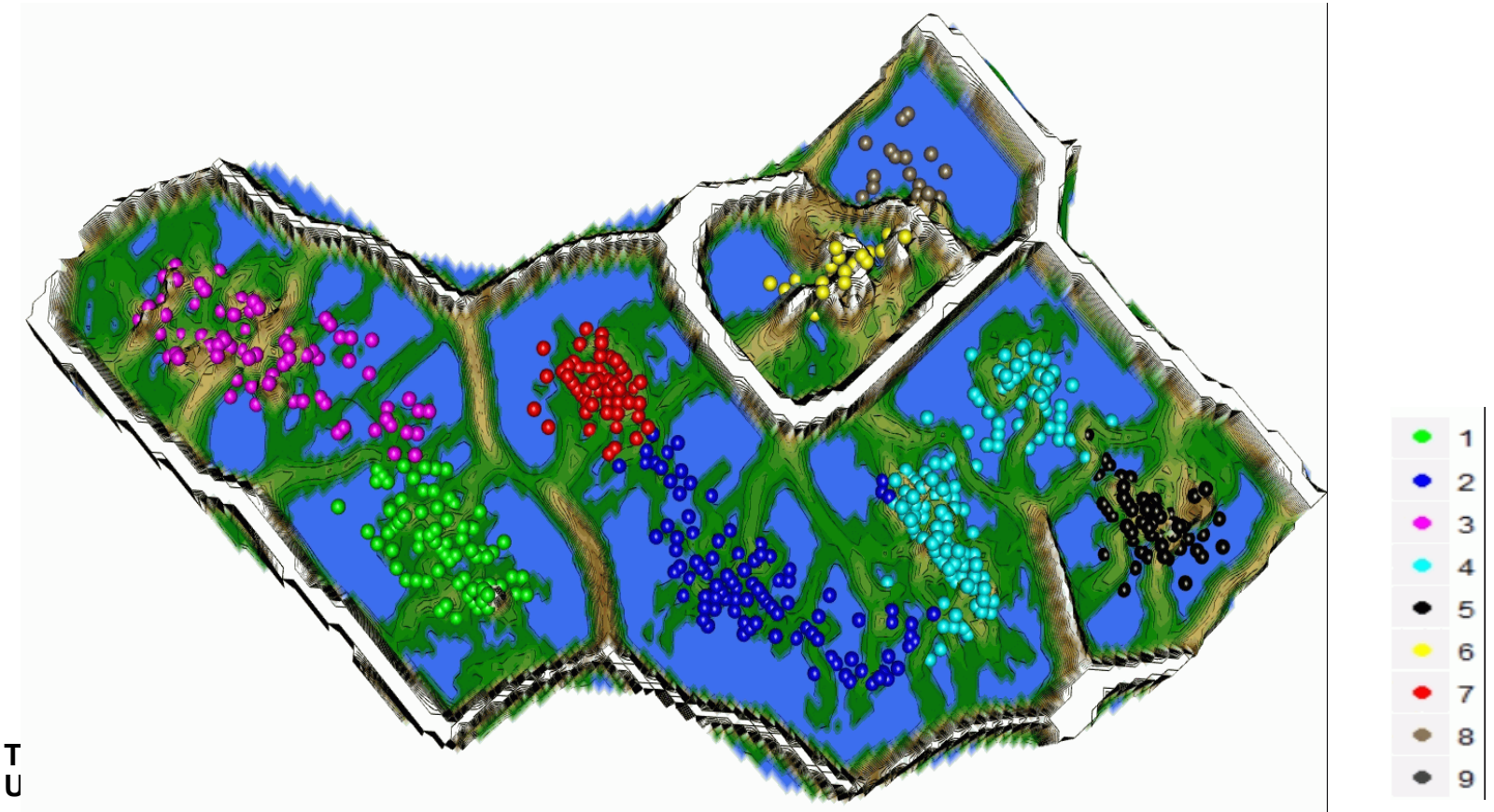
  **=> Define water quality => allow Prediciton**



# Knowledge Discovery with DBS II (a)

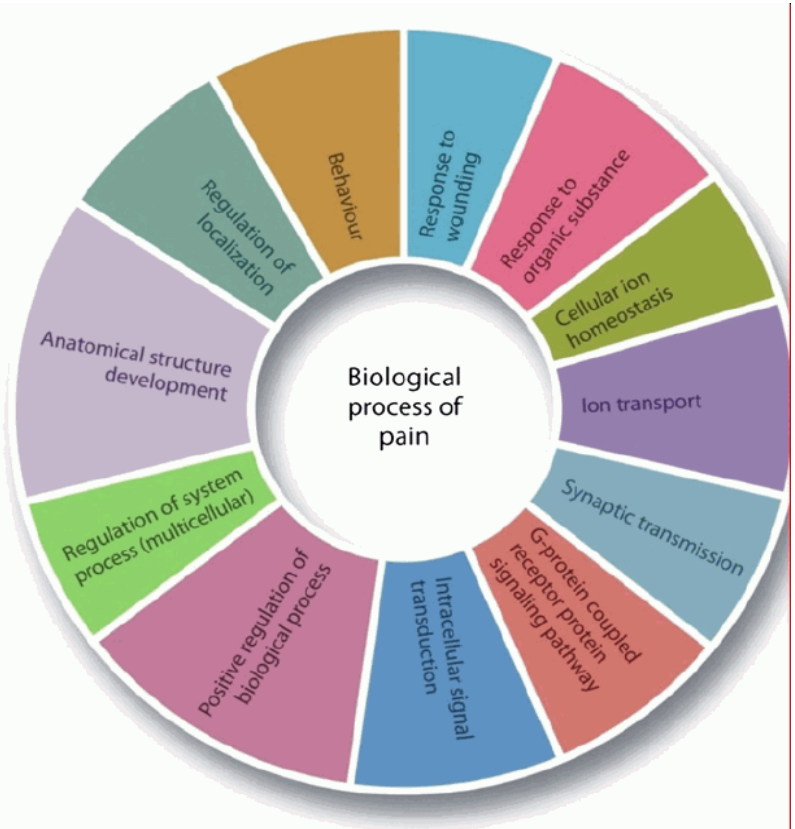
- 535 genes of Pain
- Additonal information from GeneOntology data base

  => Define tf-idf distances



# Knowledge Discovery with DBS II (b)

- Use in Geneset of each cluster of Overrepresentation anaylsis (ORA)
- Filter pValues with ABCanalysis
- Interpret the directed acyclic graph (DAG)
  - ⇒ Reproduces knowledge of [Lötsch et al., 2013
- Outliers from „noise“
- To new functional areas found:
  - *Creatine metabolic process*
  - *hematopoietic stem cell differentiation*



# Summary

- DataBionicSwarm (DBS) is an interactive Visualization and Clustering approach consting of 3 modules
  - Parameter free
  - Independ of number of variables
  - Self-organizing and emergent
- DBS uses projection and high-dimensional data for clustering
- Every module is interchangable with another state of the art application of each respective field, e.g.
  - Projection: NeRV instead of Pswarm
  - Clustering: Spectral clustering instead of DBSclustering

# Future Work

- find a strong Nash equilibrium
- apply, connections to solid-state physics
  - E.g Use Bravais lattice or apply a Fourier transformation to the reciprocal lattice (do calculations in the reciprocal space)
- Abtract U*-distances instead of abstract U-distances
  - How to integrate Pmatrix information?
- Add new data incrementely to given projection
  - Efficiency: Project more than 4000 points in less than 24h
- Better Knowledge Extraction from Cluster structures

# Thank you for listening