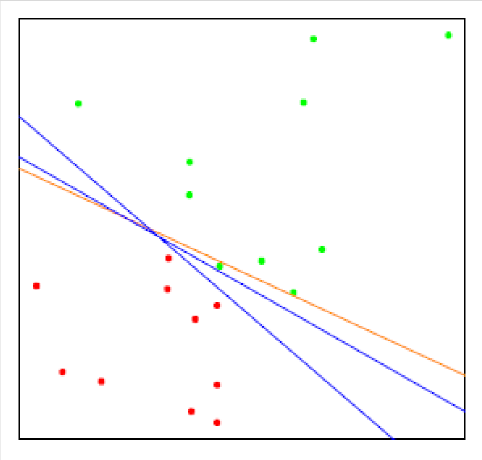


Supercomputing and Big Data

Parallel and Scalable Machine Learning Algorithms

Prof. Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

LECTURE 1

HPC Introduction & Parallel and Scalable Clustering using DBSCAN

July 26th, 2018, NextGen@Helmholtz Conference
GFZ German Research Centre for Geosciences Potsdam, Germany

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

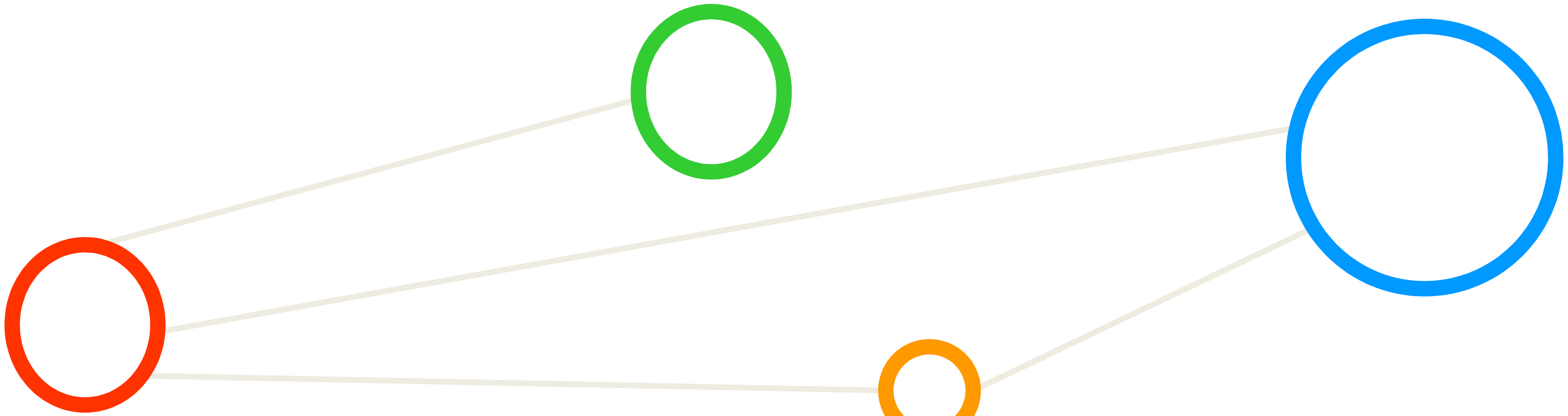
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

NextGen @HELMHOLTZ

DEEP Projects

Outline



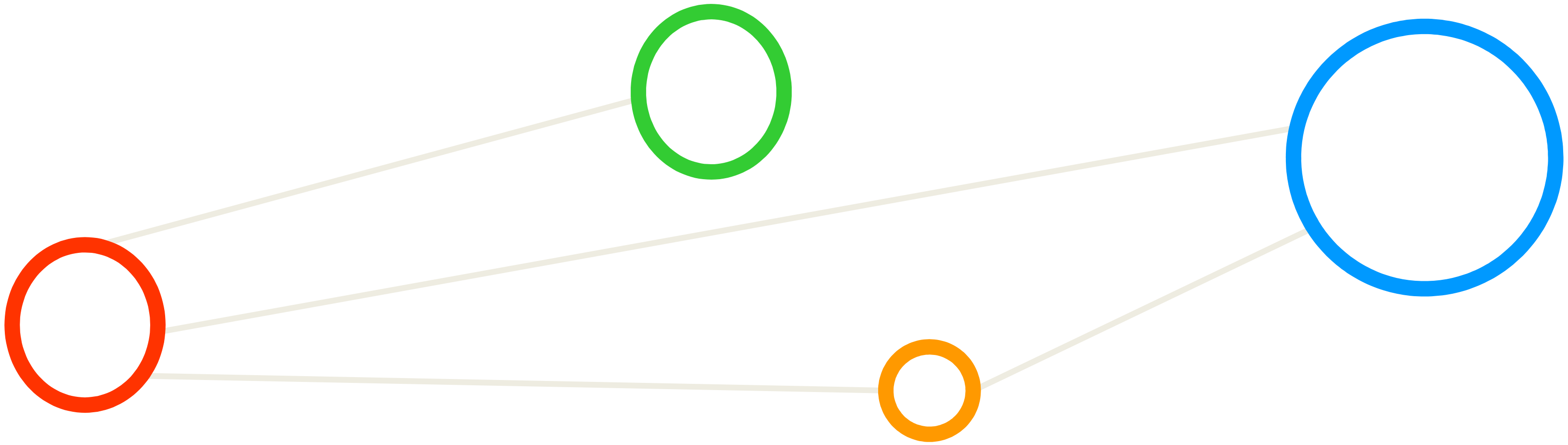
Outline of the Course

1. HPC Introduction & Parallel and Scalable Clustering using DBSCAN
2. Parallel and Scalable Classification using SVMs with Applications
3. Deep Learning using CNNs driven by HPC & GPUs
4. Deep Learning using LSTMs driven by HPC & GPUs

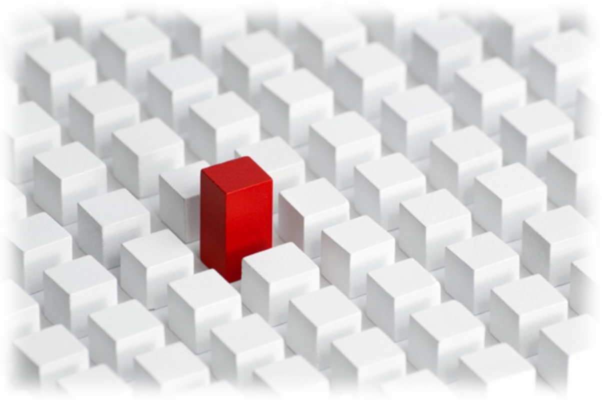
Outline

- High Performance Computing (HPC)
  - Supercomputing and Big Data for Machine Learning
  - PRACE & DEEP Projects shaped JSC Dual Concept
  - Modular Supercomputing Architecture (MSA)
  - Scheduling Principles with Job Scripts & Reservations
  - JURECA HPC System for Practicals
- Parallel & Scalable Clustering using DBSCAN
  - Clustering Methods and Approaches
  - DBSCAN Clustering Algorithm
  - Hierarchical Data Format (HDF) Basics
  - Parallel HPDBSCAN & HDF5 Data Format
  - Point Cloud Application Example Bremen

HPC Introduction



Supercomputing and Big Data for Machine Learning



- Rapid advances in 'big data' collection and storage technologies in the last decade
  - Extracting useful information is a challenge considering ever increasing massive datasets
  - Traditional data analysis techniques cannot be used in growing cases (e.g. memory limits)

- **Machine learning / Data Mining is a technology that blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data**
- **Machine Learning / Data Mining is the process of automatically discovering useful information in large data repositories ideally following a systematic process**

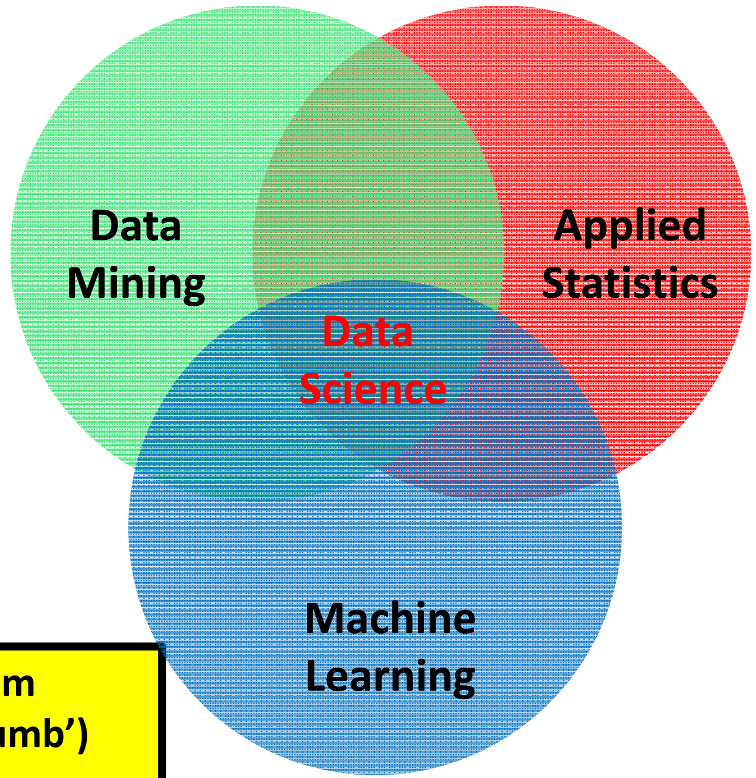
modified from [29] Introduction to Data Mining

- Machine Learning & Statistical Data Mining of Big Data
  - Traditional statistical approaches are still very useful to consider
  - E.g. in order to reduce large quantities of data to most expressive datasets

Machine Learning Prerequisites

1. Some pattern exists
2. No exact mathematical formula
3. Data exists

- Idea 'Learning from Data' shared with a wide variety of other disciplines
  - E.g. signal processing, data mining, etc.
- Challenge: Data is often complex

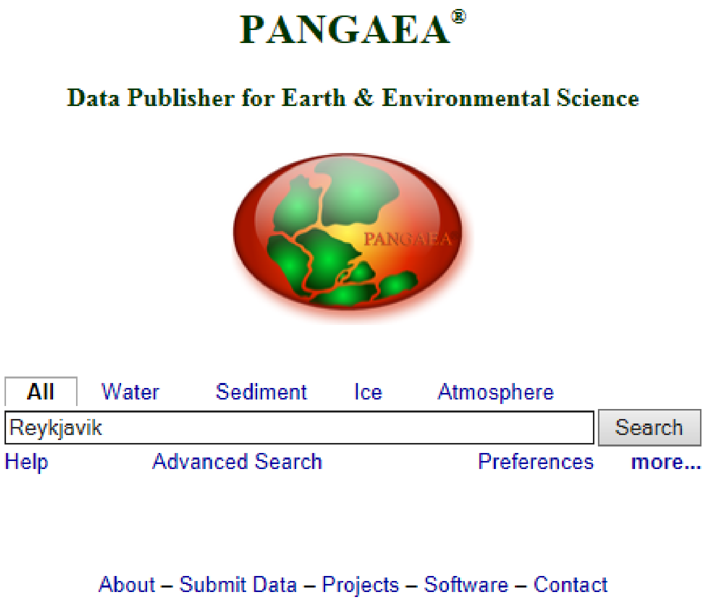


- **Machine learning is a very broad subject and goes from very abstract theory to extreme practice ('rules of thumb')**
- **Machine learning algorithms can leverage parallel computing**

Examples of Real Data Collections

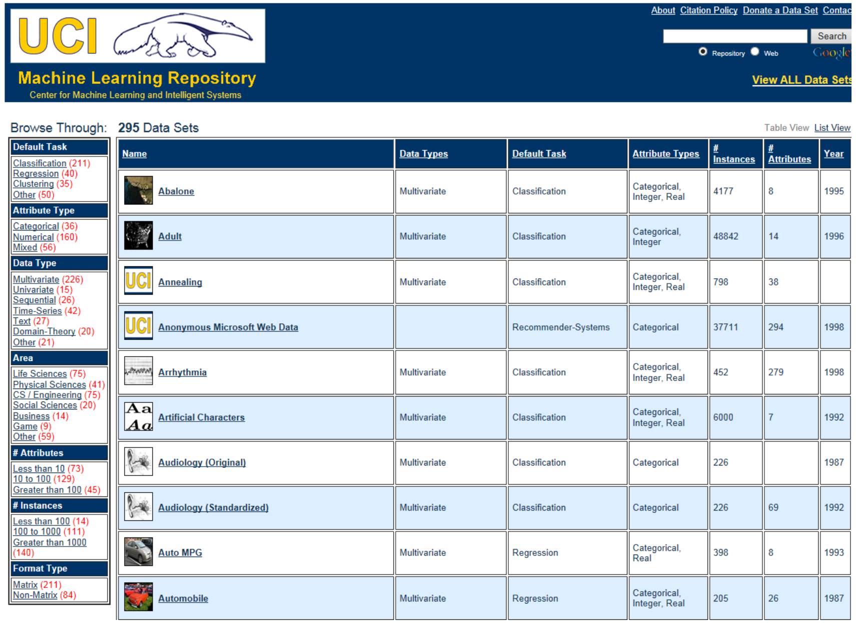
- Data collection of the earth and environmental science domain
  - Different from the known 'UCI machine learning repository examples'

(real science datasets examples)



[30] PANGAEA data collection

(examples for learning & comparisons)



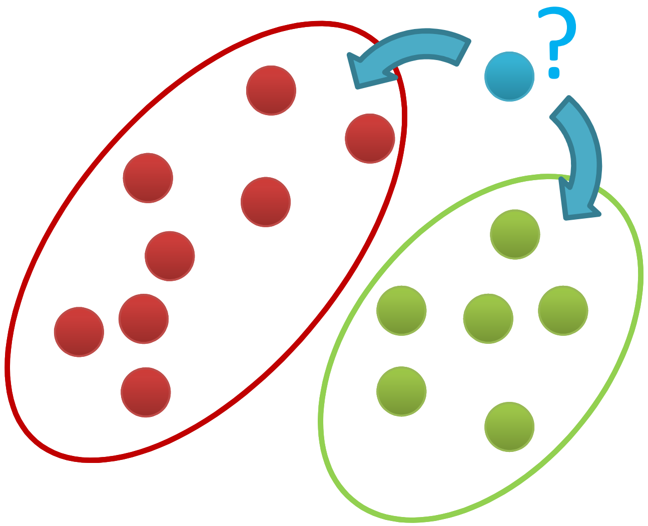
Name	Data Types	Default Task	Attribute Types	# Instances	# Attributes	Year
Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8	1995
Adult	Multivariate	Classification	Categorical, Integer	48842	14	1996
Annealing	Multivariate	Classification	Categorical, Integer, Real	798	38	
Anonymous Microsoft Web Data		Recommender-Systems	Categorical	37711	294	1998
Arrhythmia	Multivariate	Classification	Categorical, Integer, Real	452	279	1998
Artificial Characters	Multivariate	Classification	Categorical, Integer, Real	6000	7	1992
Audiology (Original)	Multivariate	Classification	Categorical	226		1987
Audiology (Standardized)	Multivariate	Classification	Categorical	226	69	1992
Auto MPG	Multivariate	Regression	Categorical, Real	398	8	1993
Automobile	Multivariate	Regression	Categorical, Integer, Real	205	26	1987

[31] UCI Machine Learning Repository

Methods Overview – Need for Parallel Algorithms

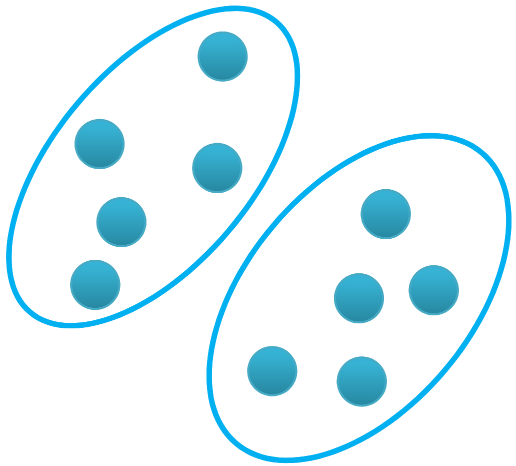
- **Machine learning methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction**

Classification



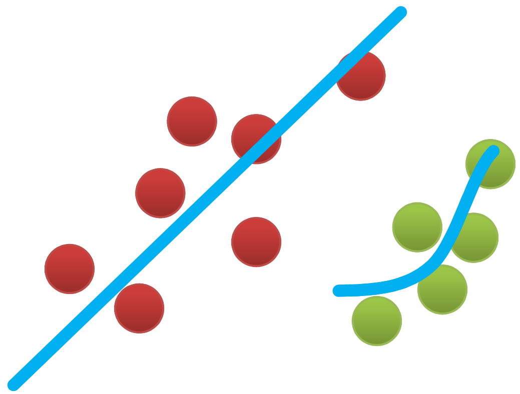
- Groups of data exist
- New data classified to existing groups

Clustering



- No groups of data exist
- Create groups from data close to each other

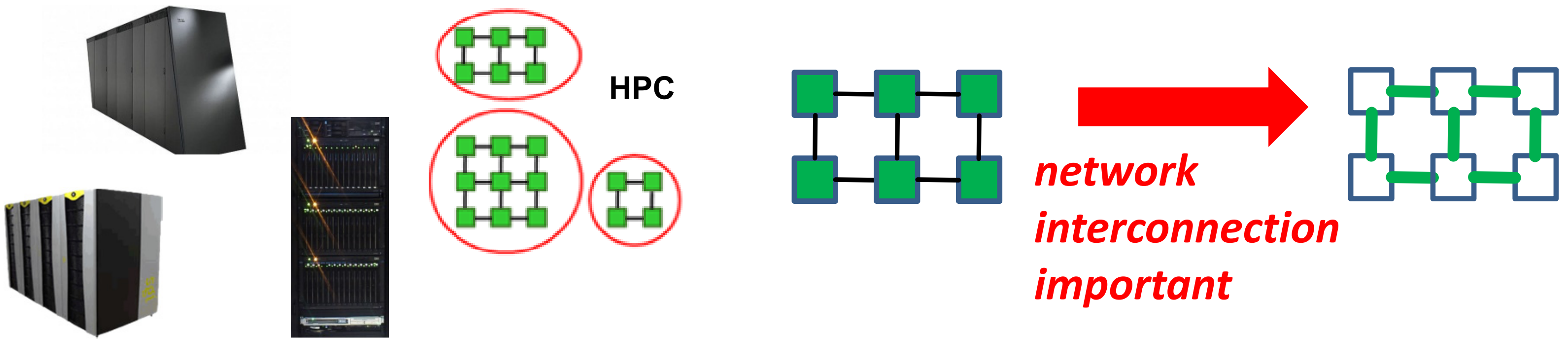
Regression



- Identify a line with a certain slope describing the data

Understanding High Performance Computing

- **High Performance Computing (HPC) is based on computing resources that enable the efficient use of parallel computing techniques through specific support with dedicated hardware such as high performance cpu/core interconnections.**

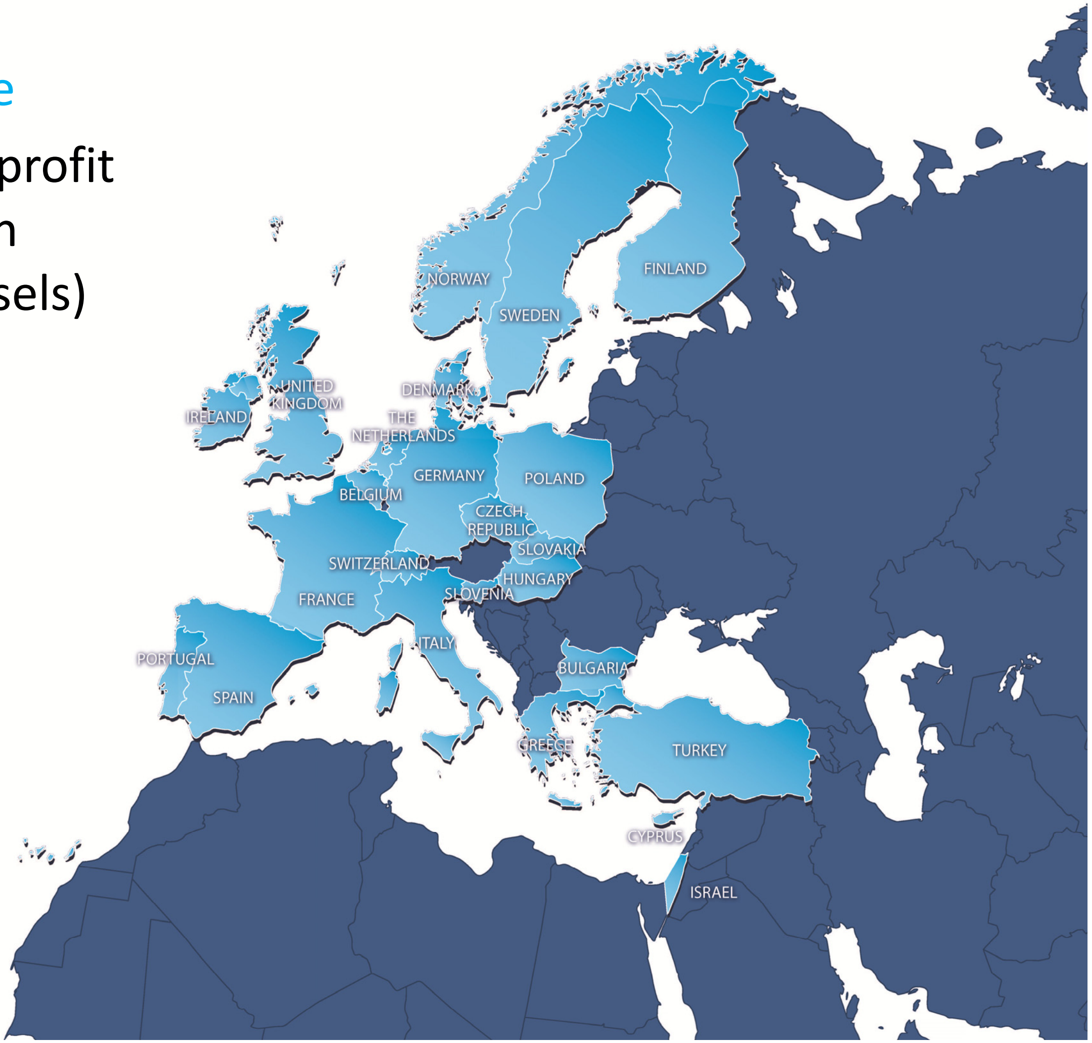


focus in the next slides

- **High Throughput Computing (HTC) is based on commonly available computing resources such as commodity PCs and small clusters that enable the execution of 'farming jobs' without providing a high performance interconnection between the cpu/cores.**

Partnership for Advanced Computing in Europe (PRACE)

- Basic Facts
  - HPC-driven infrastructure
  - An international not-for-profit association under Belgien law (with its seat in Brussels)
  - Has 25 members and 2 observers
  - Governed by the PRACE Council in which each member has a seat
  - Daily management of the association is delegated to the Board of Directors



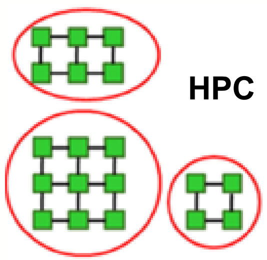
[19] PRACE

PRACE as Persistent pan-European HPC Infrastructure

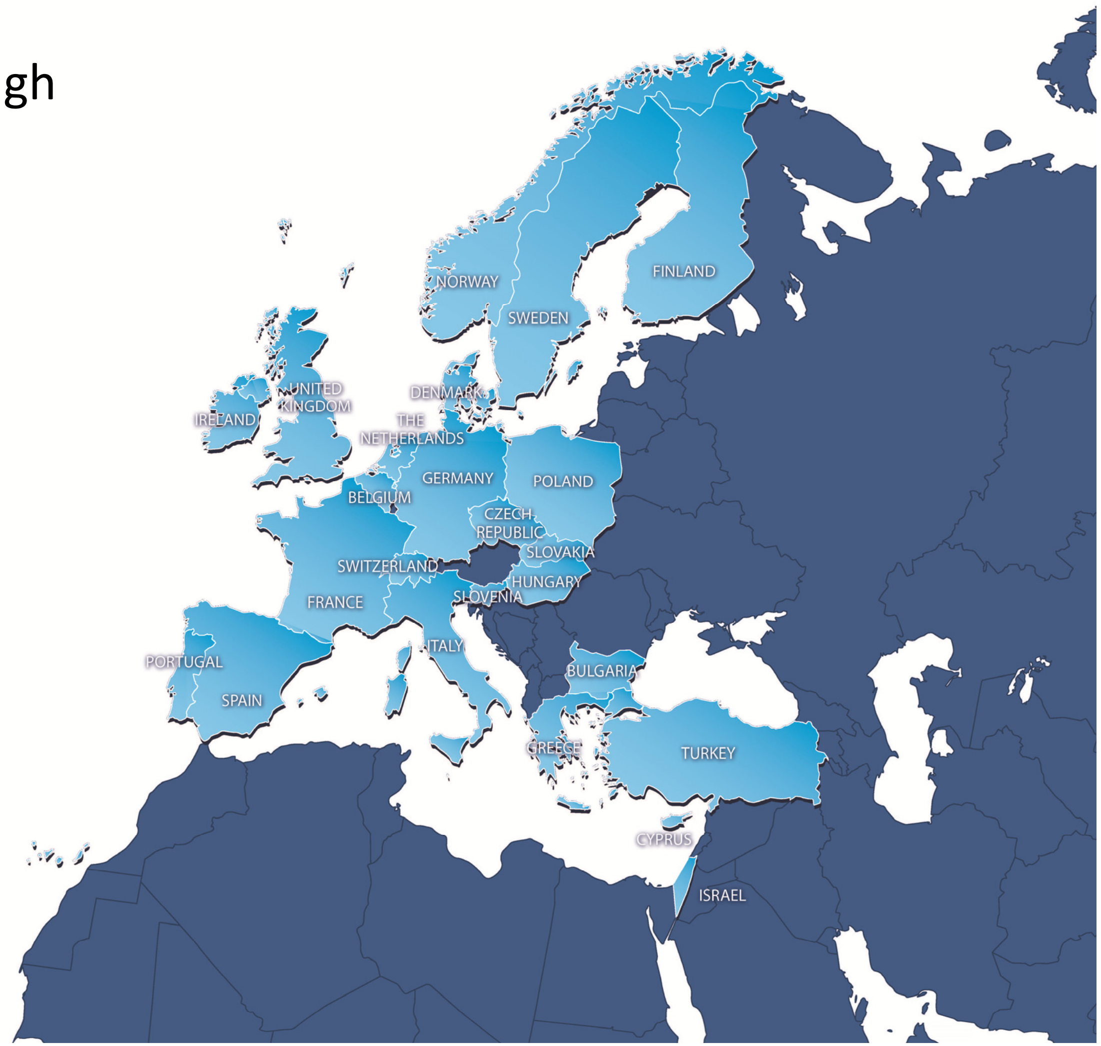
Mission:
enabling world-class science through large scale simulations

Offering:
HPC resources on leading edge capability systems

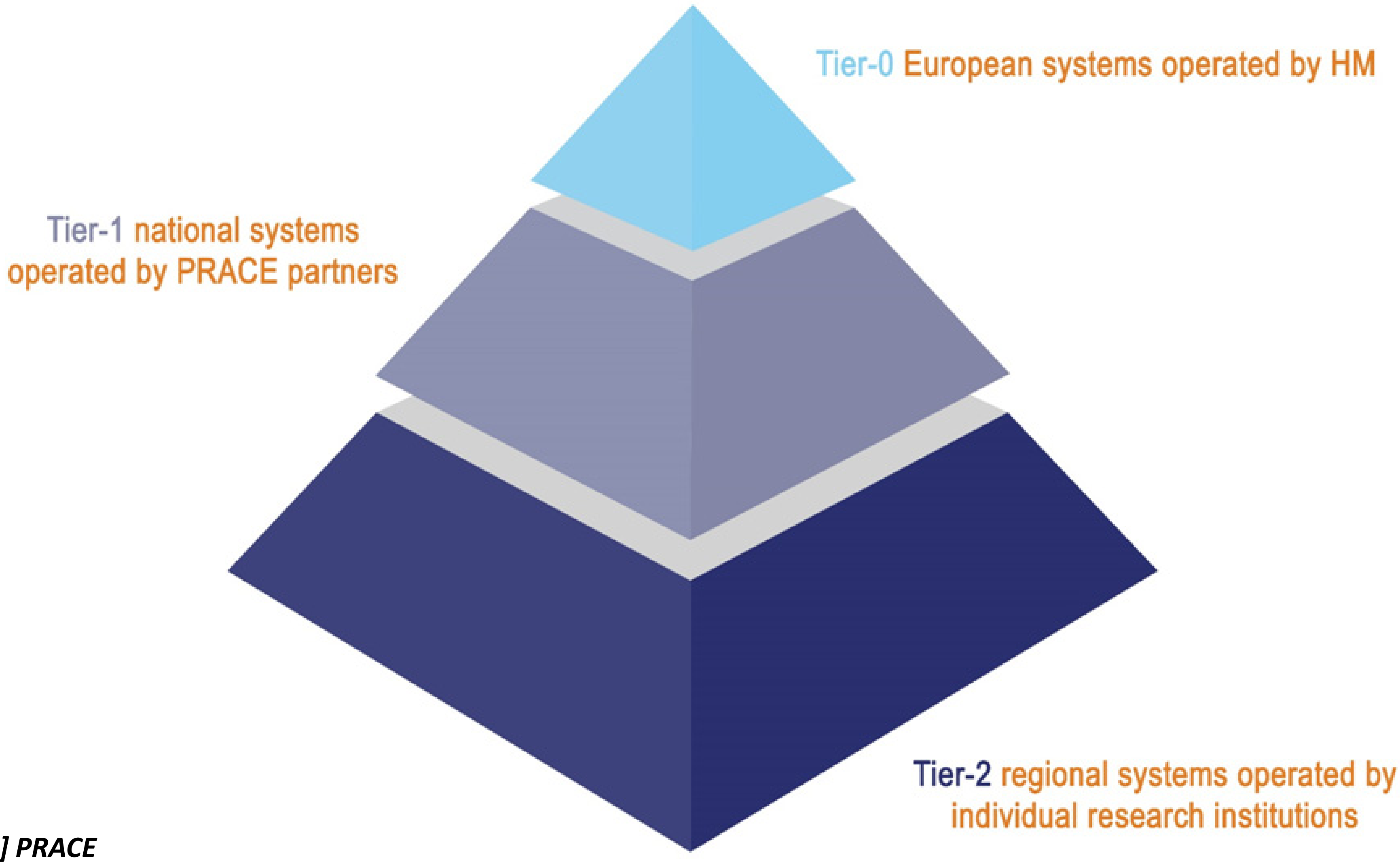
Resource award:
through a single and fair pan-European peer review process for open research



[19] PRACE



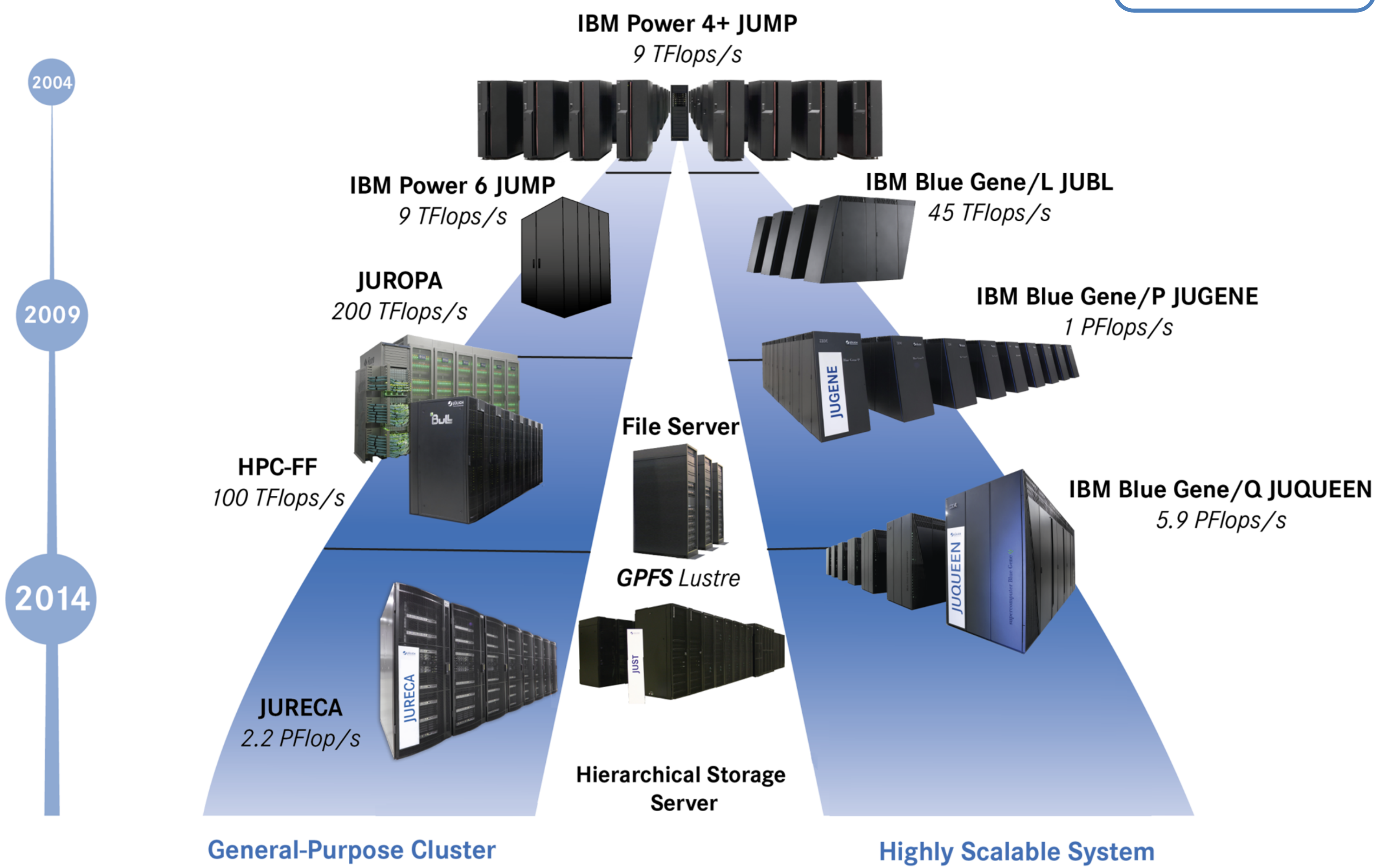
PRACE European vs. Regional Systems



[19] PRACE

[Video] JUQUEEN – Supercomputer Upgrade Example

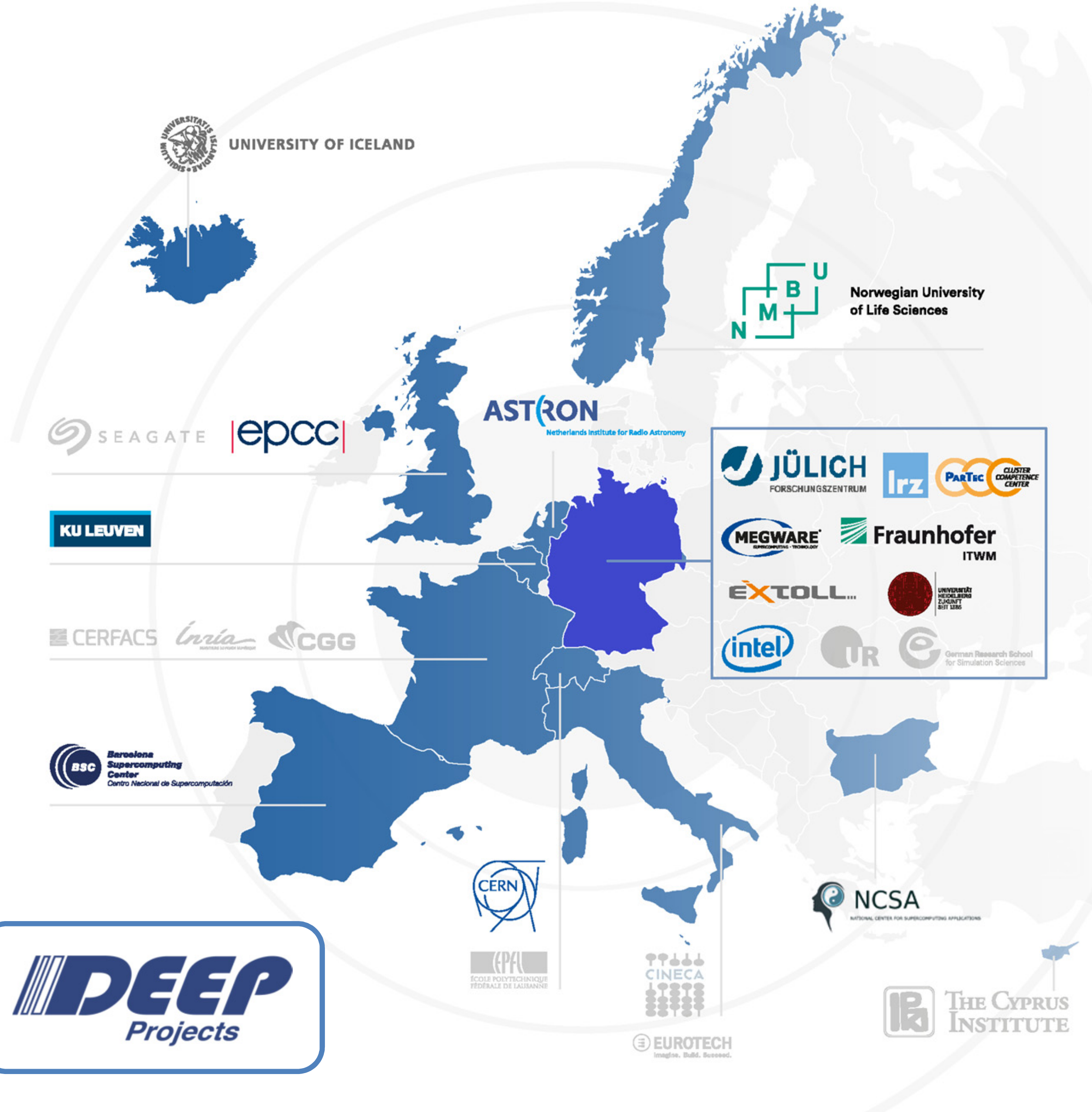
JSC Dual Approach



DEEP Projects & Partners

- DEEP
  - Dynamic Exascale Entry Platform
- 3 EU Exascale projects
  DEEP
  DEEP-ER
  DEEP-EST
- 27 partners
  Coordinated by JSC
- EU-funding: 30 M€
  JSC-part > 5,3 M€
- Nov 2011 – Jun 2020

[20] DEEP-EST EU Project



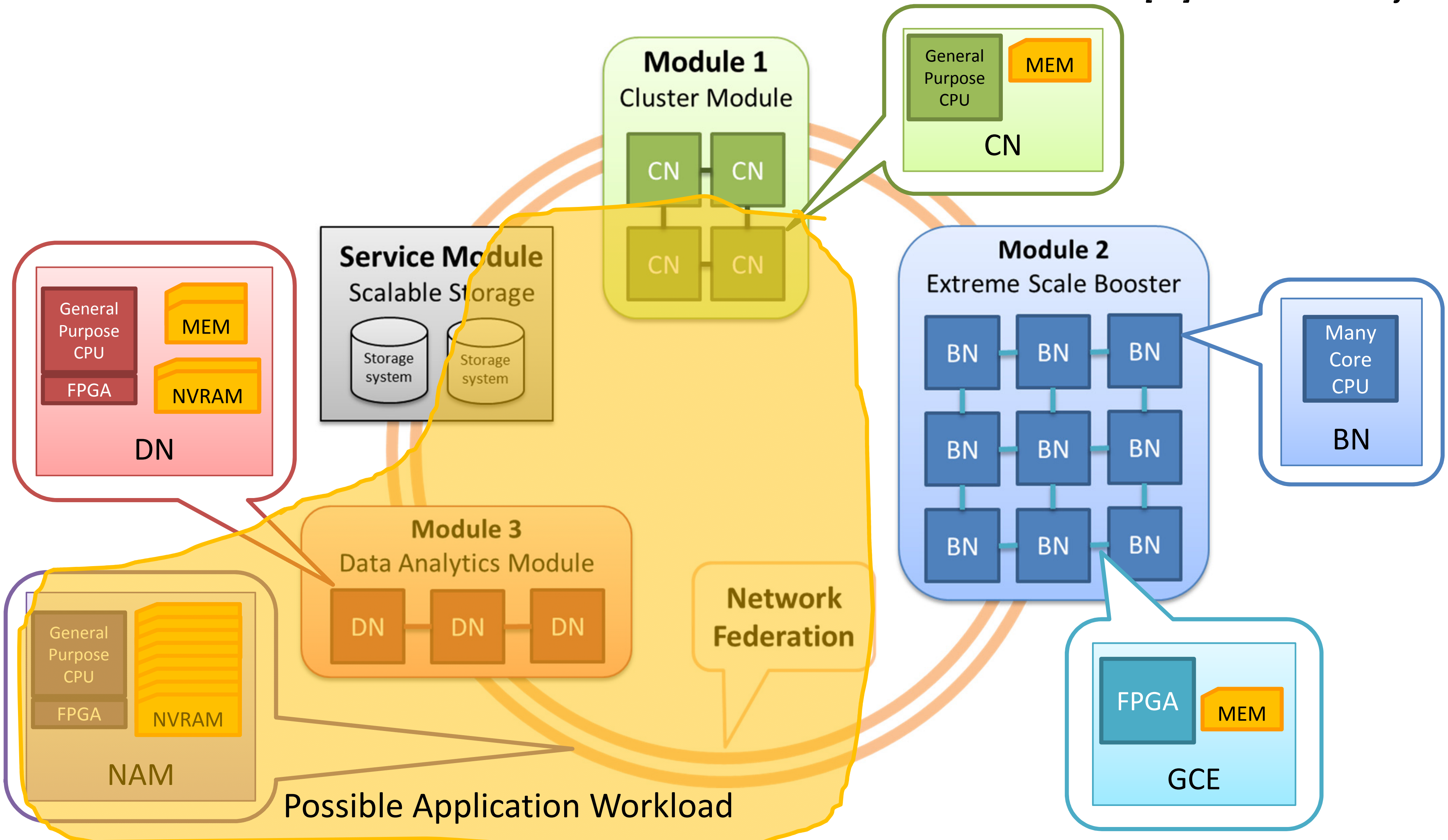
DEEP Projects

Deep Projects – Application Co-Design → Heterogenity

[20] DEEP-EST EU Project

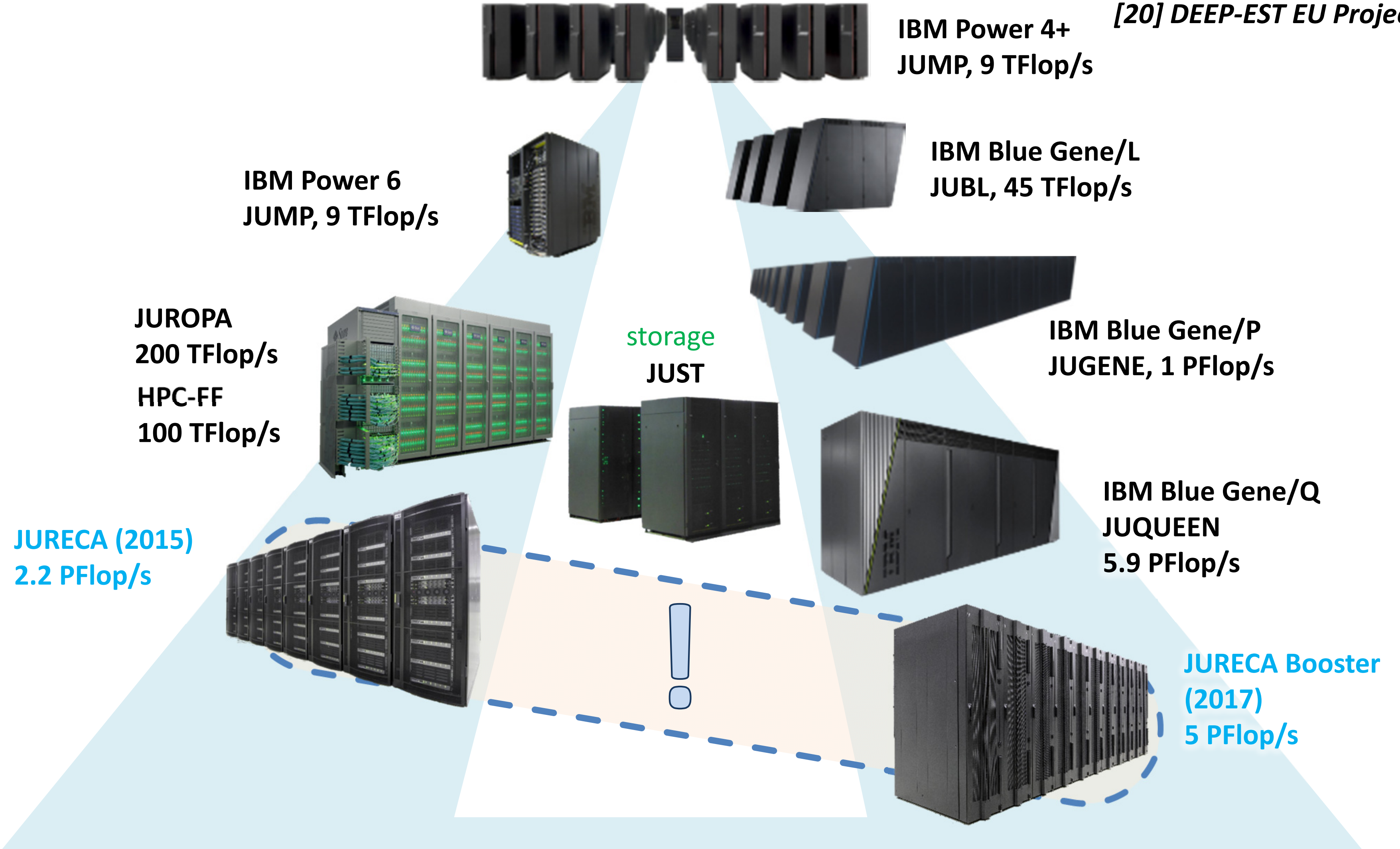
Emerging HPC Architecture

[20] DEEP-EST EU Project



Modular Supercomputing @ JSC

[20] DEEP-EST EU Project



JURECA HPC System

[20] DEEP-EST EU Project

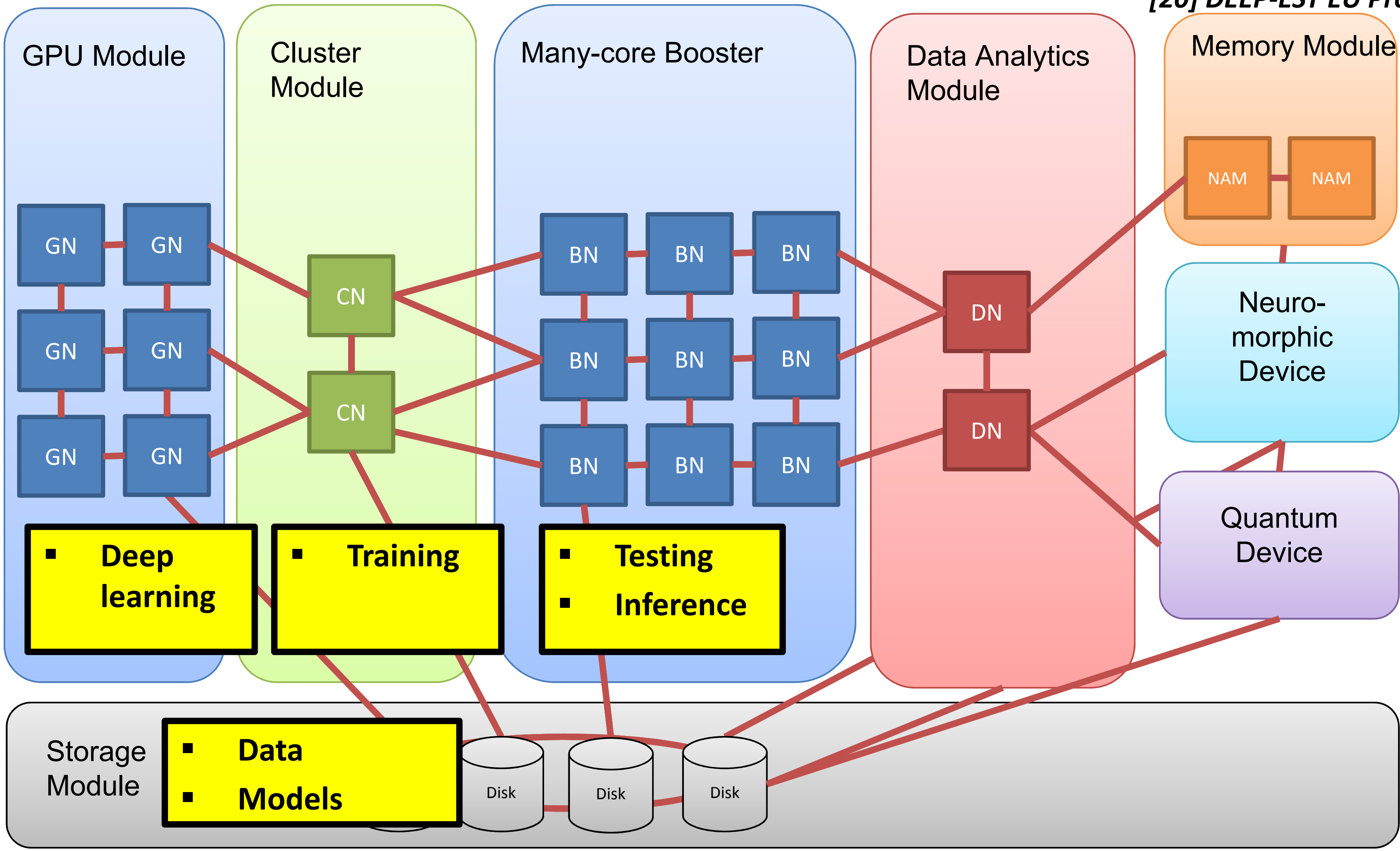
JURECA

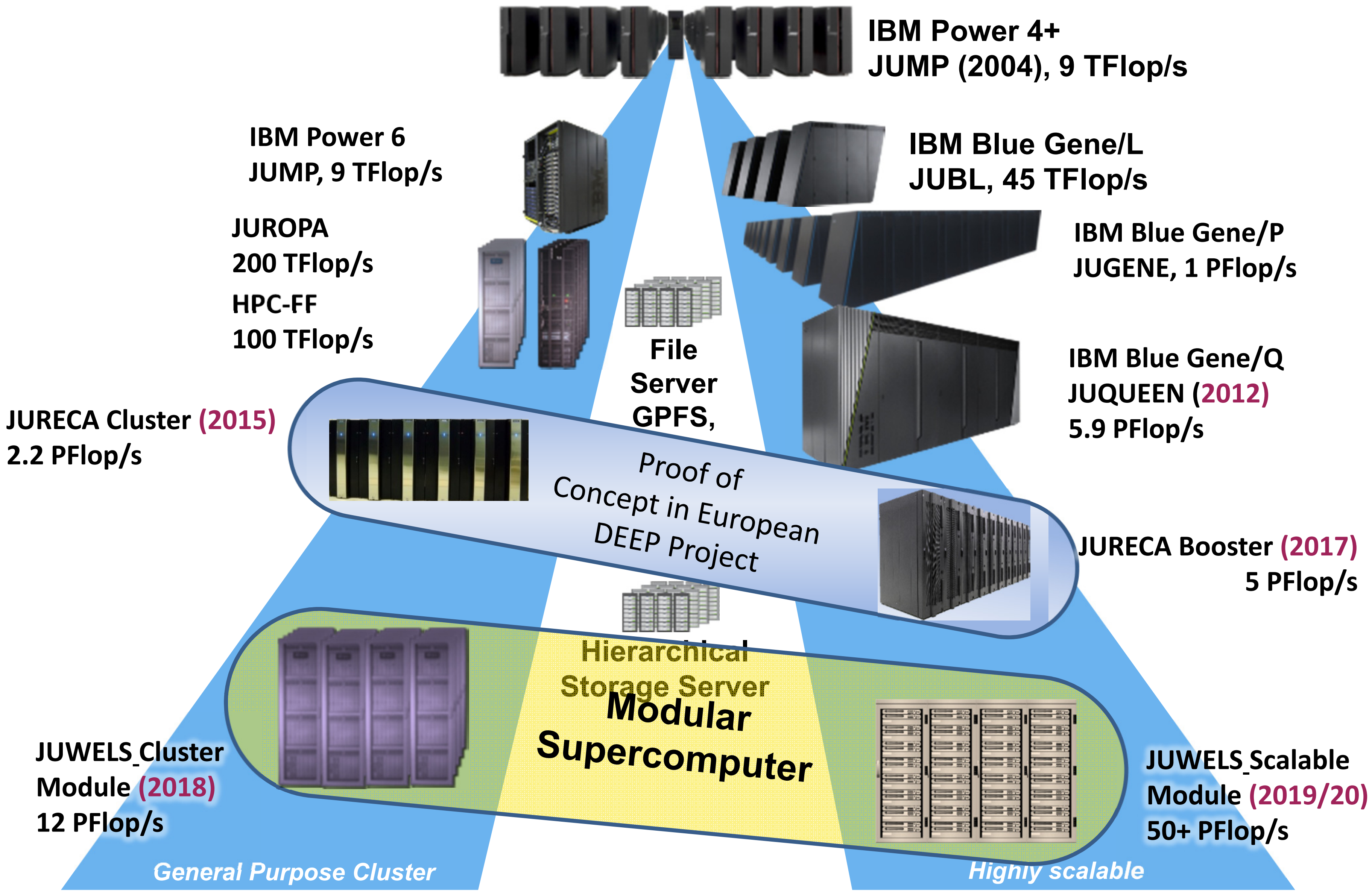
- T-Platforms V210 blade server solution
  - Dual-socket Intel Xeon Haswell CPUs
- Mellanox InfiniBand EDR network
- Peak: 1.8 PF (CPUs) + 0.4 PF (GPUs)
- 281 TiB main memory
- 100 GBps storage bandwidth

- Dell PowerEdge C6320P solution
  - Intel Xeon Phi “Knights Landing” 7250-F
- Intel OPA network
- Peak: 5 PF
- 157 TiB main memory + 26 TiB MCDRAM
- 200 GBps storage BW

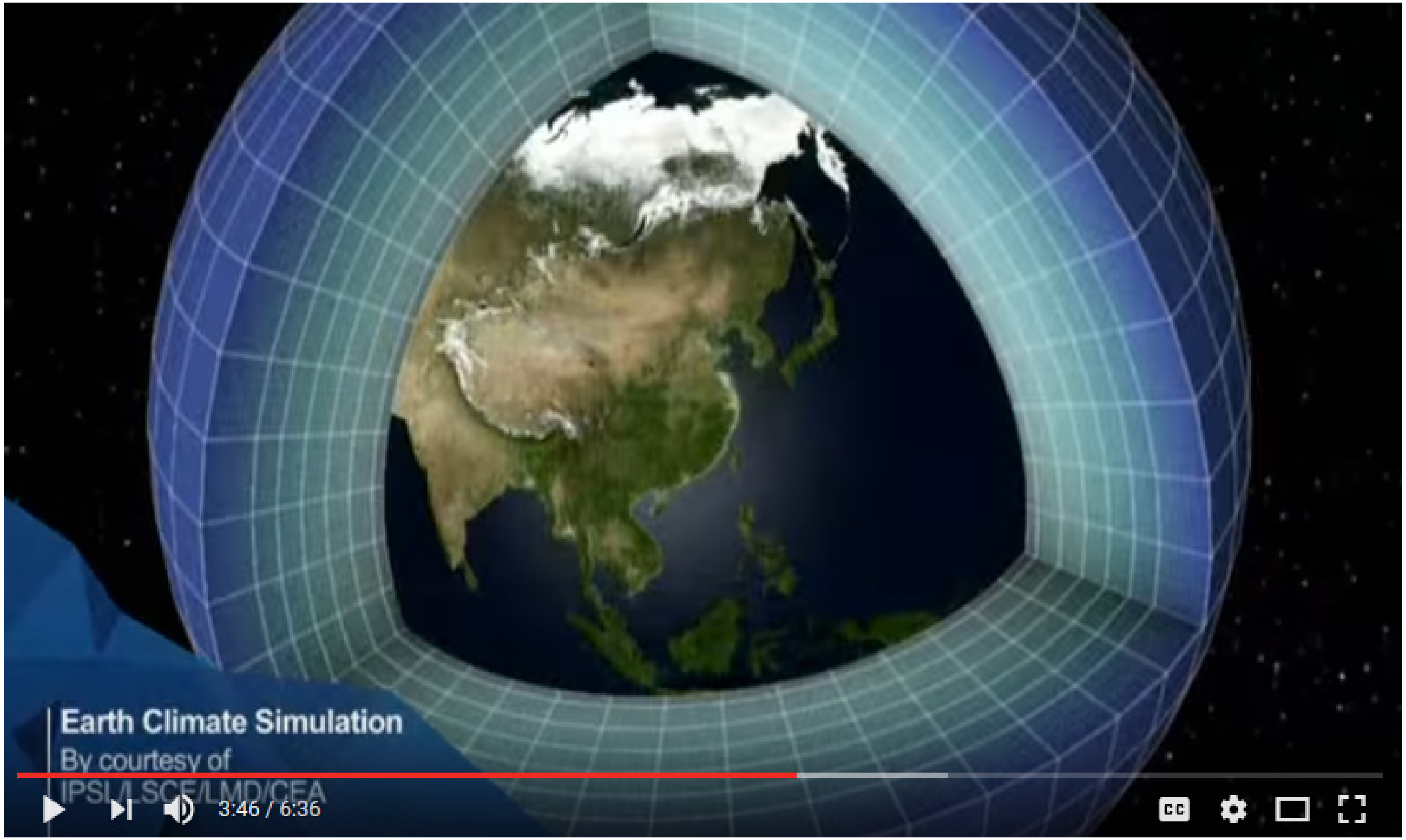
Roadmap & Machine Learning

[20] DEEP-EST EU Project




IBM Power 4+
JUMP (2004), 9 TFlop/s
IBM Power 6
JUMP, 9 TFlop/s
IBM Blue Gene/L
JUBL, 45 TFlop/s
JUROPA
200 TFlop/s
HPC-FF
100 TFlop/s
IBM Blue Gene/P
JUGENE, 1 PFlop/s
File
Server
GPFS,
IBM Blue Gene/Q
JUQUEEN (2012)
5.9 PFlop/s
JURECA Cluster (2015)
2.2 PFlop/s
Proof of
Concept in European
DEEP Project
JURECA Booster (2017)
5 PFlop/s
Hierarchical
Storage Server
Modular
Supercomputer
JUWELS Cluster
Module (2018)
12 PFlop/s
JUWELS Scalable
Module (2019/20)
50+ PFlop/s
General Purpose Cluster
Highly scalable

[Video] PRACE – Introduction to Supercomputing



[21] YouTube, 'PRACE – Introduction to Supercomputing'

Parallel Computing

- All modern supercomputers depend heavily on parallelism

> - **We speak of parallel computing whenever a number of 'compute elements' (e.g. cores) solve a problem in a cooperative way**

[22] Introduction to High Performance Computing for Scientists and Engineers

- Often known as 'parallel processing' of some problem space
  - Tackle problems in parallel to enable the 'best performance' possible

- 'The measure of speed' in High Performance Computing matters
  - Common measure for parallel computers established by TOP500 list
  - Based on benchmark for ranking the best 500 computers worldwide

[23] TOP 500 supercomputing sites

network interconnection important

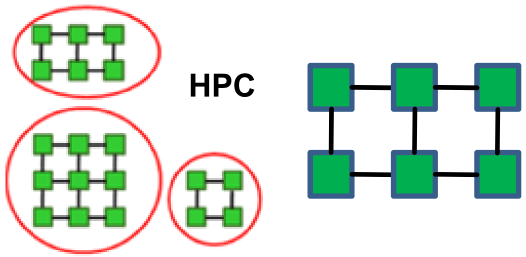
JURECA CLUSTER & BOOSTER Module @ JSC

Tutorial Machine: JSC JURECA System – CLUSTER Module

- Characteristics
  - Login nodes with 256 GB memory per node
  - 45,216 CPU cores
  - 1.8 (CPU) + 0.44 (GPU) Petaflop/s peak performance
  - Two Intel Xeon E5-2680 v3 Haswell CPUs per node: 2 x 12 cores, 2.5 GhZ
  - 75 compute nodes equipped with two NVIDIA K80 GPUs (2 x 4992 CUDA cores)
- Architecture & Network
  - Based on T-Platforms V-class server architecture
  - Mellanox EDR InfiniBand high-speed network with non-blocking fat tree topology
  - 100 GiB per second storage connection to JUST

[24] JURECA HPC System

- **Use our ssh keys to get an access and use reservation**
- **Put the private key into your ./ssh directory (UNIX)**
- **Use the private key with your putty tool (Windows)**



Exercise – Login to JURECA with TrainXYZ Accounts

- train003 – train020 are training accounts that are used in the tutorial for JURECA using SSH Keys
- Every participant needs to pick one trainXYZ account from the list
- Download keys: https://fz-juelich.sciebo.de/s/IIFC5QojZZfIcCC
- Password: request to m.riedel@fz-juelich.de
- UNIX: chmod 600 for changing the rights to the key for the ssh client, ssh-add (not standard name)

JURECA System – SSH Login

- Use your account train004 - train050
- Windows: use putty or MobaXterm (better with x-server)
- UNIX: ssh trainXYZ@jureca.fz-juelich.de
- Example

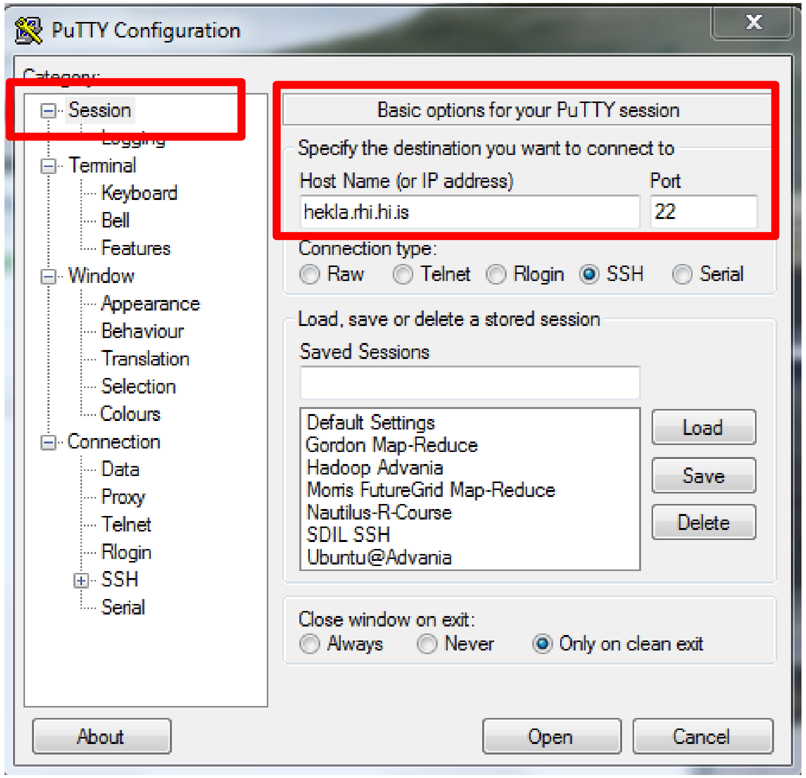
```
adminuser@linux-8djg:~> ssh train001@jureca.fz-juelich.de
Warning: the ECDSA host key for 'jureca.fz-juelich.de' differs from the key for the IP address '134.94.33.
9'
Offending key for IP in /home/adminuser/.ssh/known_hosts:12
Matching host key in /home/adminuser/.ssh/known_hosts:19
Are you sure you want to continue connecting (yes/no)? yes
]Last login: Mon Aug 21 14:29:03 2017 from zam2036.zam.kfa-juelich.de
*******************************************************************************
*                             Welcome to JURECA                               *
*                                                                             *
*  Information about the system, latest changes, user documentation and FAQs: *
*                  http://www.fz-juelich.de/ias/jsc/jureca                    *
*******************************************************************************
*                            ### Known Issues ###                             *
*                                                                             *
* An up-to-date list of known issues on the system is maintained at           *
*               http://www.fz-juelich.de/ias/jsc/jureca-known-issues          *
* Open issues:                                                                *
*   - Intel compiler error with std::valarray and                             *
*     optimized headers, added 2016-03-20                                     *
```

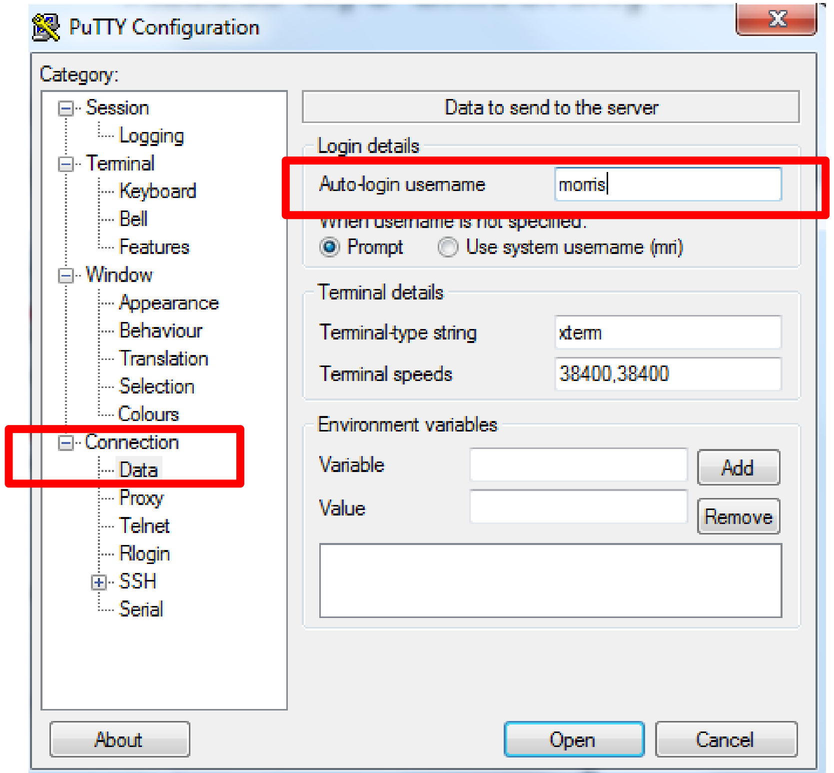
➢ **Remember to use your own trainXYZ account in order to login to the JURECA system**

Using SSH Clients for Windows

[25] PUTTY tool

- Example: using the Putty SSH client
  (other SSH tools exist, e.g. could be MoabXTerm, etc.)





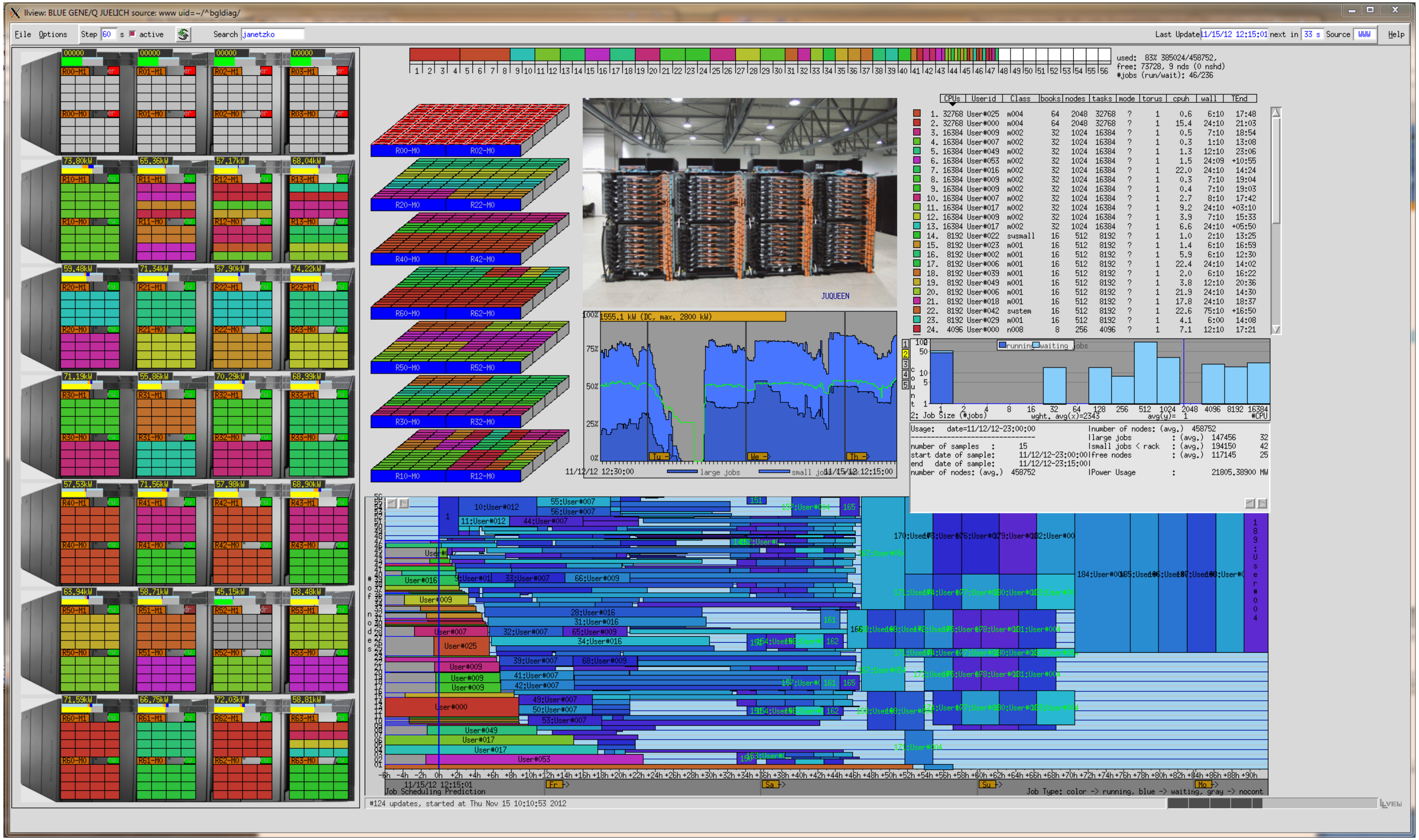
➢ **Configure Keys under SSH, change username to trainXYZ, and use hostname for JURECA**

Scheduling Principles – SLURM Scheduler in Tutorial

- HPC Systems are typically not used in an interactive fashion
  - Program application starts 'processes' on processors ('do a job for a user')
  - Users of HPC systems send 'job scripts' to schedulers to start programs
  - Scheduling enables the sharing of the HPC system with other users
  - Closely related to Operating Systems with a wide varity of algorithms
- E.g. First Come First Serve (FCFS)
  - Queues processes in the order that they arrive in the ready queue.
- E.g. Backfilling
  - Enables to maximize cluster utilization and throughput
  - Scheduler searches to find jobs that can fill gaps in the schedule
  - Smaller jobs farther back in the queue run ahead of a job waiting at the front of the queue (but this job should not be delayed by backfilling!)

- **Scheduling is the method by which user processes are given access to processor time (shared)**

Example: Supercomputer BlueGene/Q



[26] LLView Tool

JURECA Example – Tutorial Reservations

ReservationName=bigdata-cpu StartTime=2018-07-26T09:45:00

EndTime=2018-07-26T14:15:00 Duration=04:30:00

Nodes=jrc[0056-0076] NodeCnt=21 CoreCnt=504 Features=thin

PartitionName=batch Flags=

TRES=cpu=1008

Users=s.luehrs,train001,train003,train004,train005,train006,train007,train008,train009,train010,train011,train012,train013,train014,train015,train016,train017,train018,train019,train020,train021,train022

Accounts=(null) Licenses=(null) State=INACTIVE BurstBuffer=(null) Watts=n/a

ReservationName=bigdata-gpu StartTime=2018-07-26T13:45:00

EndTime=2018-07-26T18:15:00 Duration=04:30:00

Nodes=jrc[0006-0010,0013-0028] NodeCnt=21 CoreCnt=504

Features=(null) PartitionName=gpus Flags=

TRES=cpu=1008

Users=s.luehrs,train001,train003,train004,train005,train006,train007,train008,train009,train010,train011,train012,train013,train014,train015,train016,train017,train018,train019,train020,train021,train022

Accounts=(null) Licenses=(null) State=INACTIVE BurstBuffer=(null) Watts=n/a

Jobscript JURECA Example & Reservation

```
#!/bin/bash -x
#SBATCH--nodes=2
#SBATCH--ntasks=48
#SBATCH--ntasks-per-node=24
#SBATCH--output=mpi-out.%j
#SBATCH--error=mpi-err.%j
#SBATCH--time=06:00:00
#SBATCH--partition=batch
#SBATCH--mail-user=m.riedel@fz-juelich.de
#SBATCH--mail-type=ALL
#SBATCH--job-name=train-indianpines-2-48-24
#SBATCH--reservation=ml-hpc-1

### location executable
PISVM=/homea/hpclab/train001/tools/pisvm-1.2.1/pisvm-train
#PISVM=/homeb/zam/mriedel/tools/pisvm-1.2.1jurecanew/pisvm-train

### location data
TRAINDATA=/homea/hpclab/train001/data/indianpines/indian_processed_training.el
#TRAINDATA=/homeb/zam/mriedel/bigdata/172-indianpinesrawproc/indian_processed_training.el

### submit
srun $PISVM -D -o 1024 -q 512 -c 100 -g 8 -t 2 -m 1024 -s 0 $TRAINDATA
```

- **Every day the reservation string is changed on the HPC systems (below)**
- **Change the number of nodes and tasks to use more or less CPUs for jobs**
- **Use the command sbatch <jobsript> in order to submit parallel jobs to the supercomputer and remember your <job id> returned**
- **Use the command squeue –u <userid> in order to check the status of your parallel job**

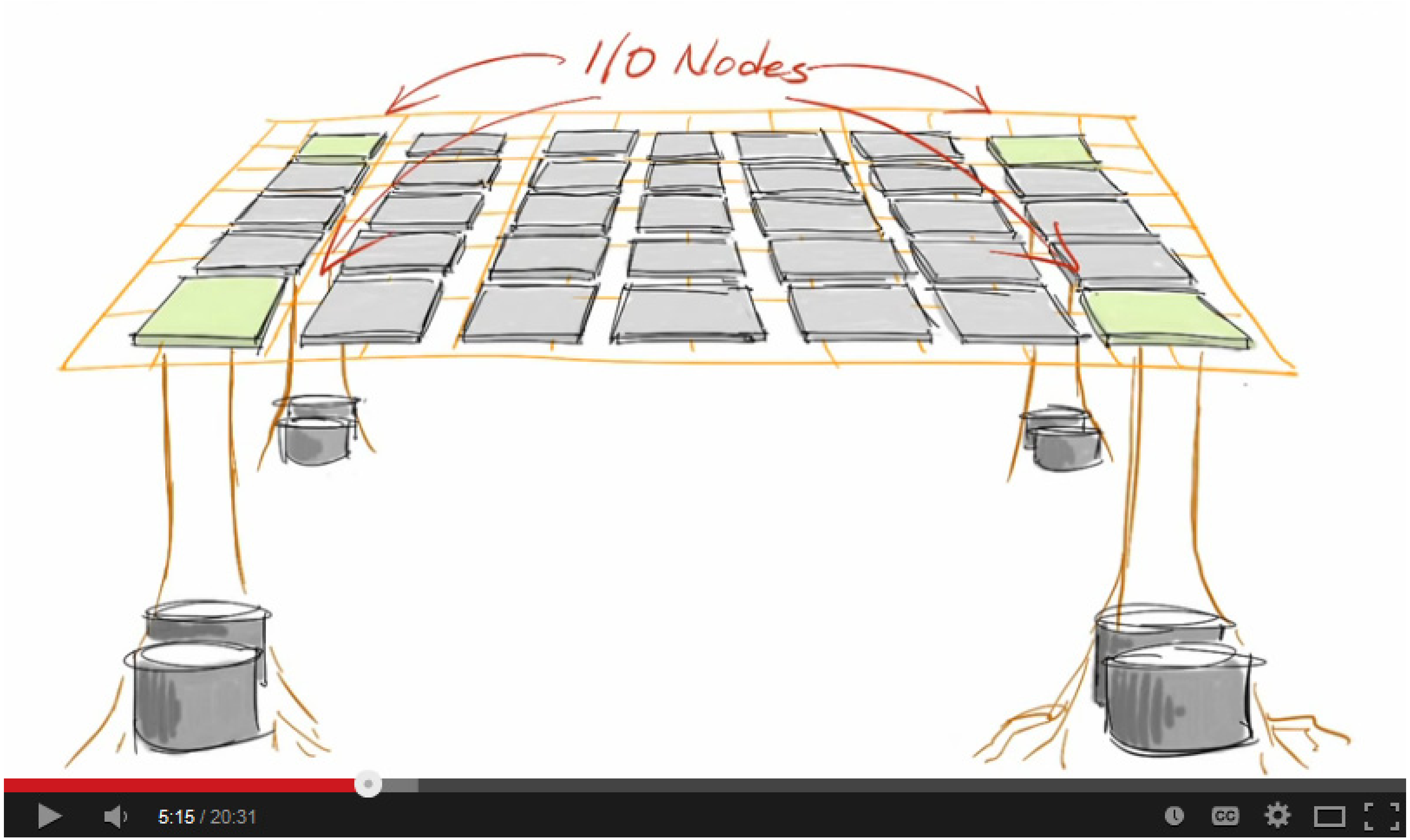
> **JURECA HPC System – Reservation Wedneday morning → bigdata-cpu**

HPC Environment – Modules

- Module environment tool
  - Avoids to manually setup environment information for every application
  - Simplifies shell initialization and lets users easily modify their environment
- Module avail
  - Lists all available modules on the HPC system (e.g. compilers, MPI, etc.)
- Module spider
  - Find modules in the installed set of modules and more information
- Module load
  - Loads particular modules into the current work environment, E.g.:

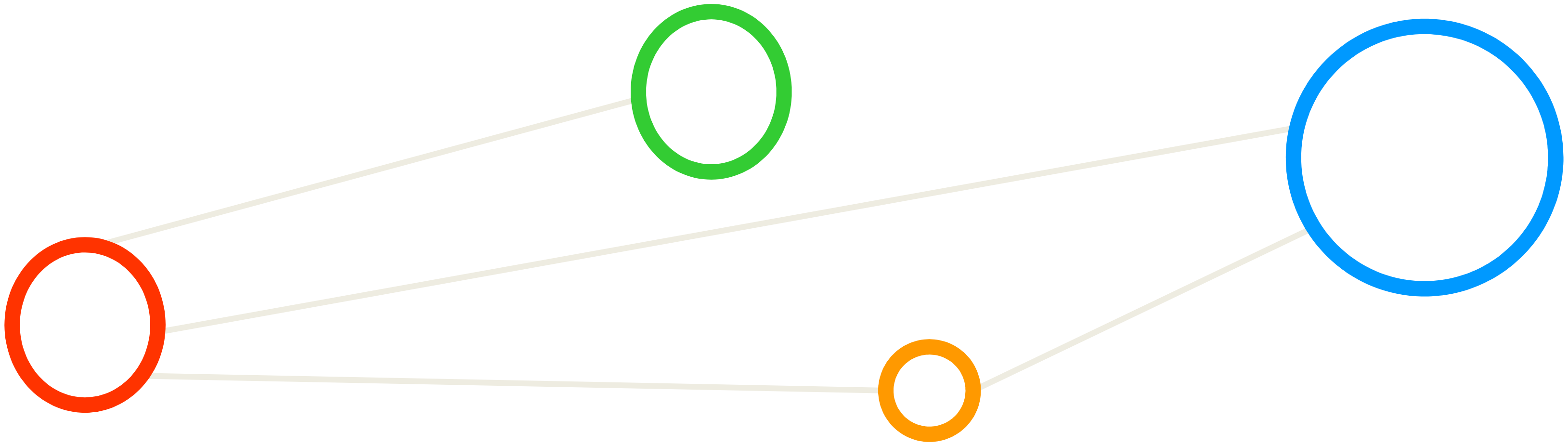
```
[train001@jrl12 ~]$ module load GCC

Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29

The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0

[train001@jrl12 ~]$ module load ParaStationMPI/5.2.0-1
[train001@jrl12 ~]$ module load HDF5/1.8.19
```

[Video] Parallel I/O with I/O Nodes



[27] Simplifying HPC Architectures, YouTube Video

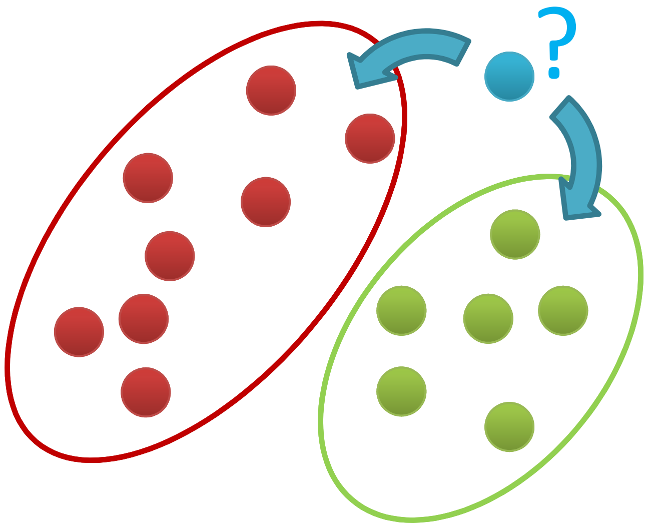
Unsupervised Clustering



Methods Overview

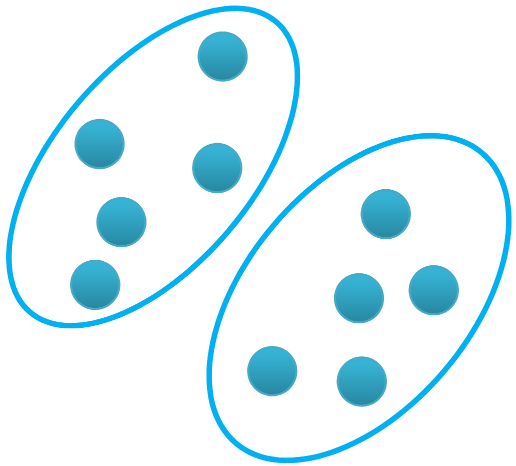
- **Machine learning methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction**

Classification



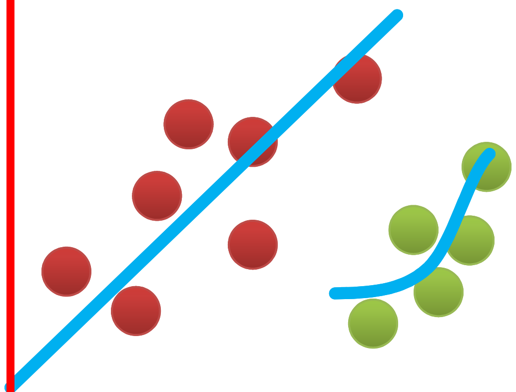
- Groups of data exist
- New data classified to existing groups

Clustering



- No groups of data exist
- Create groups from data close to each other

Regression



- Identify a line with a certain slope describing the data

What means Learning?

- **The basic meaning of learning is 'to use a set of observations to uncover an underlying process'**
- **The three different learning approaches are supervised, unsupervised, and reinforcement learning**

- Supervised Learning
  - Majority of methods follow this approach in this course
  - Example: credit card approval based on previous customer applications
- Unsupervised Learning
  - Often applied before other learning → higher level data representation
  - Example: Coin recognition in vending machine based on weight and size
- Reinforcement Learning
  - Typical 'human way' of learning
  - Example: Toddler tries to touch a hot cup of tea (again and again)

Learning Approaches – Unsupervised Learning – Revisited

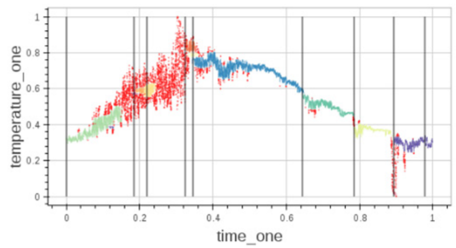
- Each observation of the predictor measurement(s) has no associated response measurement:
  - Input $\mathbf{x} = x_1, \ldots, x_d$
  - No output
  - Data $(\mathbf{x}_1), \ldots, (\mathbf{x}_N)$
- Goal: Seek to understand relationships between the observations
  - Clustering analysis: check whether the observations fall into distinct groups
- Challenges
  - No response/output that could supervise our data analysis
  - Clustering groups that overlap might be hardly recognized as distinct group

- **Unsupervised learning approaches seek to understand relationships between the observations**
- **Unsupervised learning approaches are used in clustering algorithms such as k-means, etc.**
- **Unupervised learning works with data = [input, ---]**

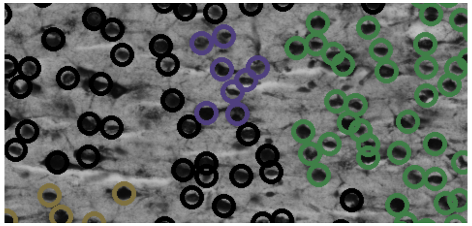
[2] An Introduction to Statistical Learning

Learning Approaches – Unsupervised Learning Use Cases

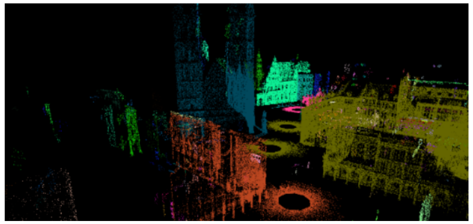
- Earth Science Data (PANGAEA, cf. Lecture 1)
  - Automatic quality control and event detection
  - Collaboration with the University of Gothenburg
  - Koljoefjords Sweden – Detect water mixing events



- Human Brain Data
  - Analyse human brain images as brain slices
  - Segment cell nuclei in brain slice images
  - Step in detecting layers of the cerebral cortex



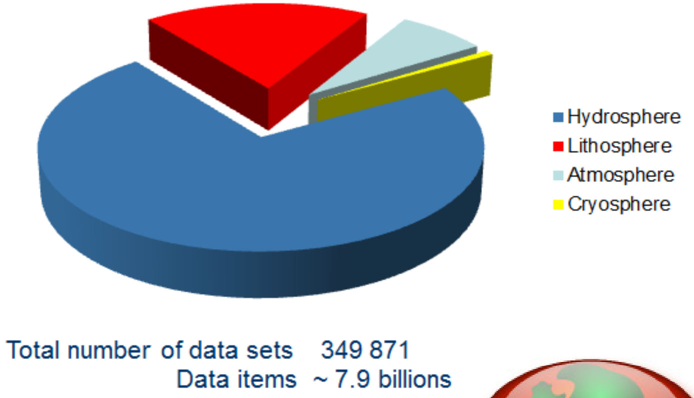
- Point Cloud Data
  - Analysis of point cloud datasets of various sizes
  - 3D/4D LIDAR scans of territories (cities, ruins, etc.)
  - Filter noise and reconstruct objects

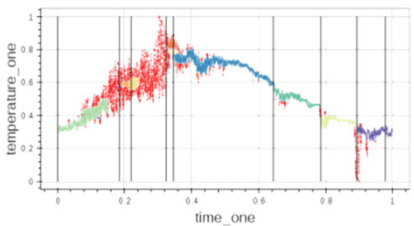


> ➢ **This clustering lecture uses a point cloud dataset of the city of Bremen as one concrete example**

Unsupervised Learning – Earth Science Data Example

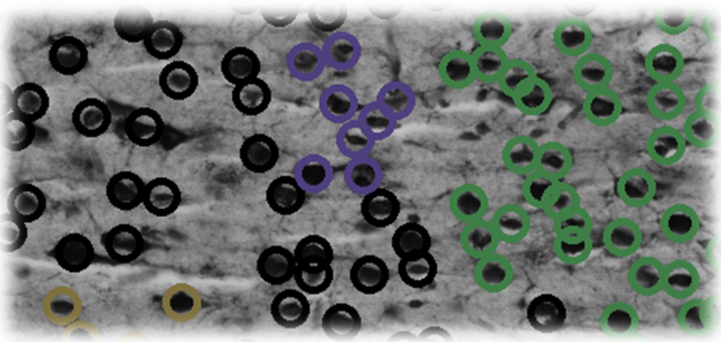
- Earth Science Data Repository
  - Time series measurements (e.g. salinity)
  - Millions to billions of data items/locations
  - Less capacity of experts to analyse data
- Selected Scientific Case
  - Data from Koljöfjords in Sweden (Skagerrak)
  - Each measurement small data, but whole sets are ‘big data‘
  - Automated water mixing event detection & quality control (e.g. biofouling)
  - Verification through domain experts

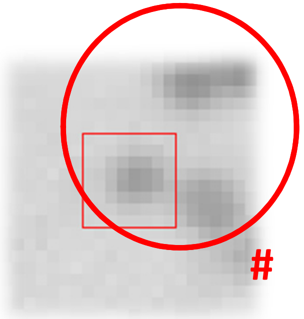


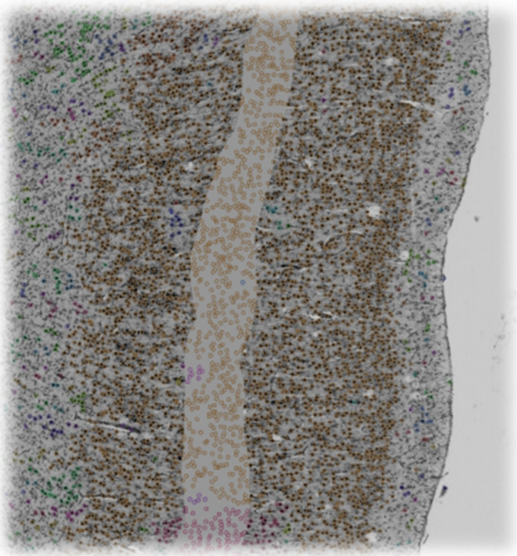


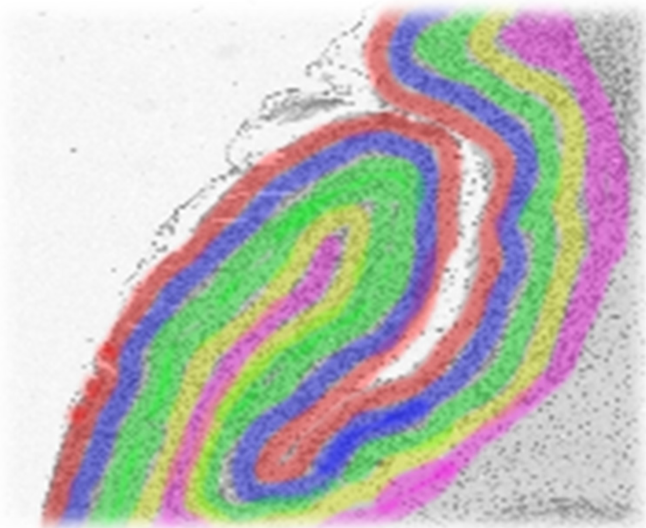
[3] PANGAEA data collection

Unsupervised Learning – Human Brain Data Example





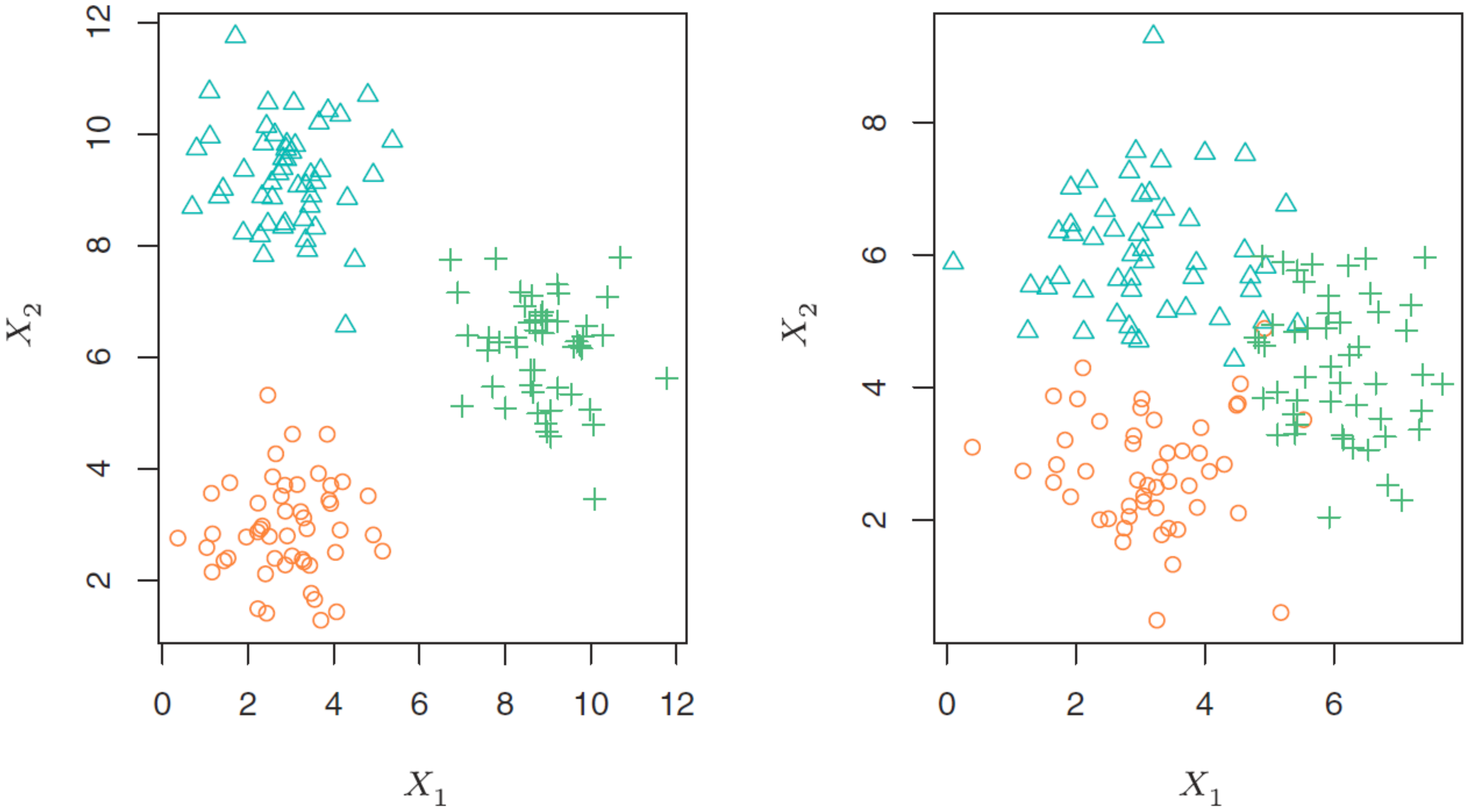




➢ **Research activities jointly with T. Dickscheid et al. (Juelich Institute of Neuroscience & Medicine)**

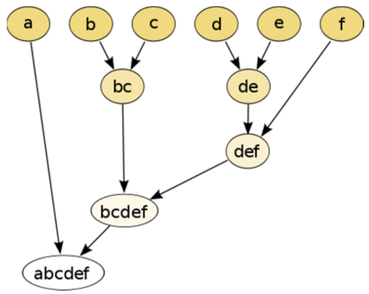
Learning Approaches – Unsupervised Learning Challenges

- Practice: The number of clusters can be ambiguities

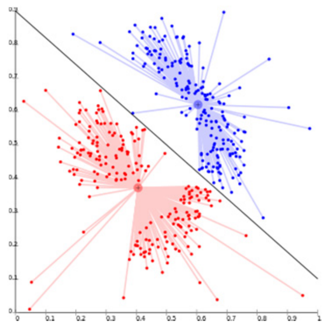


[2] An Introduction to Statistical Learning

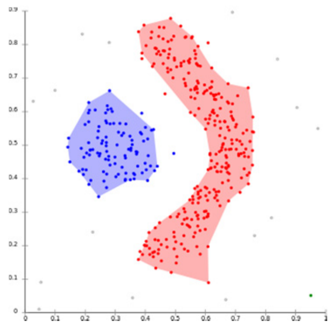
Unsupervised Learning – Different Clustering Approaches



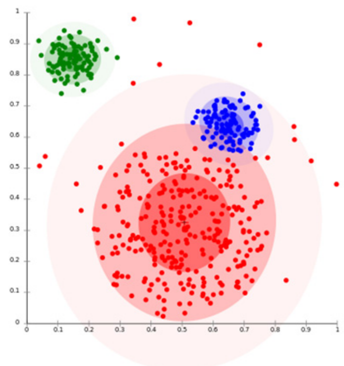
(hierarchical)



(centroid)



(density)



(distribution)

- **Clustering approaches can be categorized into four different approaches: (1) hierarchical, (2) centroid, (3) density, (4) distribution**

Unsupervised Learning – Clustering Methods

- Characterization of clustering tasks
  - No prediction as there is no associated response Y to given inputs X
  - Discovering interesting facts & relationships about the inputs X
  - Partitioning of data in subgroups (i.e. 'clusters') previously unknown
  - Being more subjective (and more challenging) than supervised learning

- Considered often as part of 'exploratory data analysis'
  - Assessing the results is hard, because no real validation mechanism exists
  - Simplifies data via a 'small number of summaries' good for interpretation

- **Clustering are a broad class of methods for discovering previously unknown subgroups in data**

Selected Clustering Methods

- K-Means Clustering – Centroid based clustering
  - Partitions a data set into K distinct clusters (centroids can be artificial)
- K-Medoids Clustering – Centroid based clustering (variation)
  - Partitions a data set into K distinct clusters (centroids are actual points)
- Sequential Agglomerative hierarchic nonoverlapping (SAHN)
  - Hiearchical Clustering (create tree-like data structure → 'dendrogram')
- Clustering Using Representatives (CURE)
  - Select representative points / cluster – as far from one another as possible
- Density-based spatial clustering of applications + noise (DBSCAN)
  - Assumes clusters of similar density or areas of higher density in dataset

Clustering Methods – Similiarity Measures

- How to partition data into distinct groups?
  - Data in same (homogenous) groups are somehow 'similiar' to each other
  - Data not in same sub-groups are somehow 'different' from each other
  - Concrete definitions of 'similiarity' or 'difference' often domain-specific
- Wide variety of similiarity measures exist, e.g. distance measures
  - Jaccard Distance, Cosine Distance, Edit Distance, Hamming Distance, …

- **A distance measure in some space is a function d(x,y) that takes two points in the space as arguments and produces a real number**

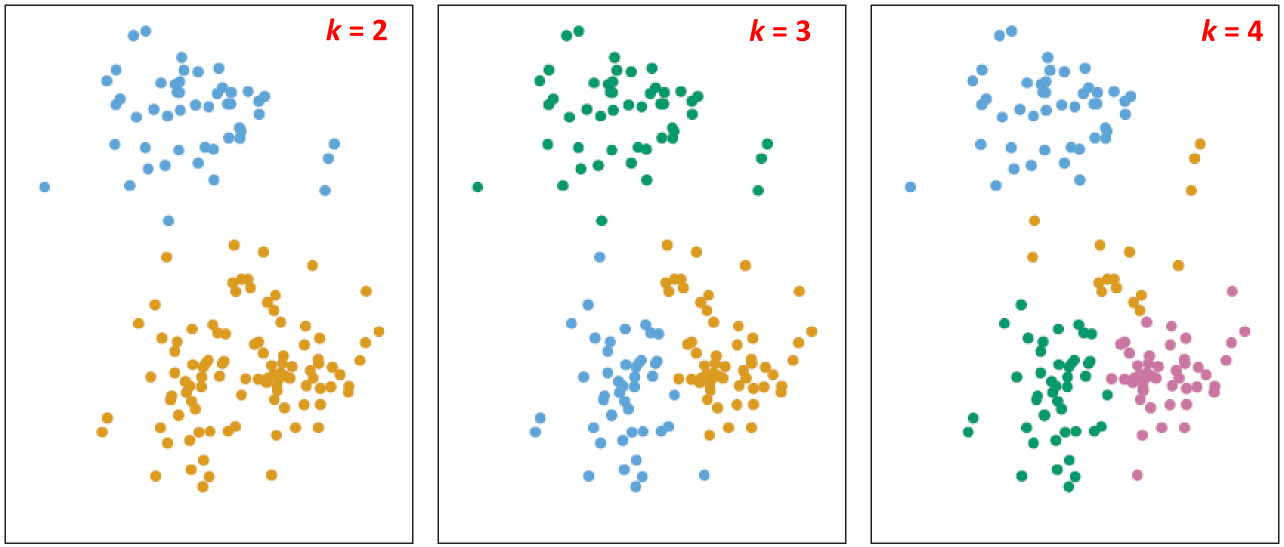
- Often used 'similiarity measure' example
  - Distance-based: Euclidean distance
  - n-dimensional Euclidean space:
    A space where points are vectors of n real numbers

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

(ruler distance)

Clustering Methods – K-Means Approach

- Approach Overview
  - Partitions a data set into *K* distinct (i.e. non-overlapping) clusters
  - Requires the definition of the desired number of clusters *K* in advance
  - Assigns each observation / data element to exactly one of the *K* clusters
  - Example: 150 observations; 2 dimensions; 3 different values of *K*



[2] An Introduction to Statistical Learning

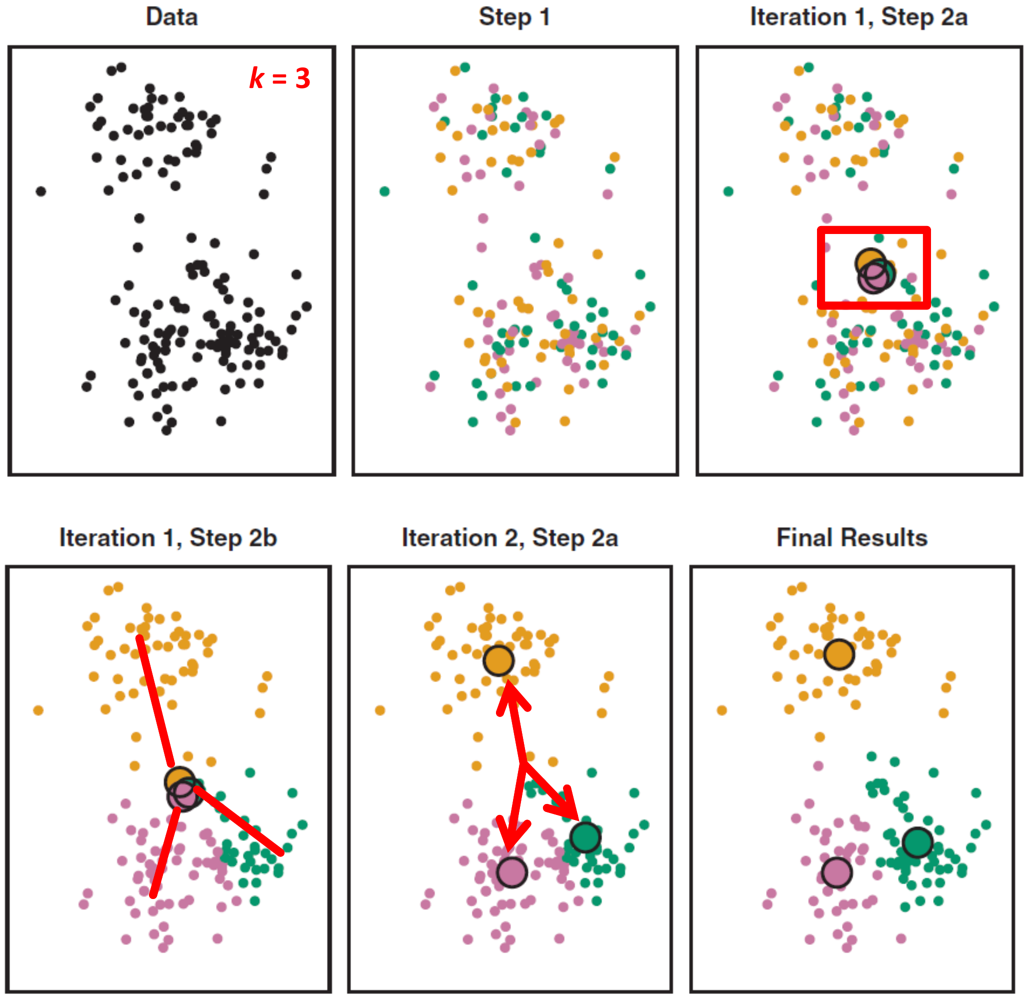
Clustering Methods – K-Means Algorithm

0. Set the desired number of clusters *K*
   - Picking the right number k is not simple (→ later)

1. Randomly assign a number from 1 to K to each observation
   - Initializes cluster assignments for the observations
   - Requires algorithm execution multiple times
     (results depend on random assignment, e.g. pick 'best' after 6 runs)

2. Iterate until the cluster assignments stop changing
   a. For each of the K clusters: compute the cluster centroid
      - The kth cluster centroid is the vector of the p feature means
        for all the observations in the kth cluster
   b. Assign each observation to the cluster K whose centroid is closest
      - The definition of 'closest' is the Euclidean distance

Clustering Methods – K-Means Algorithm Example



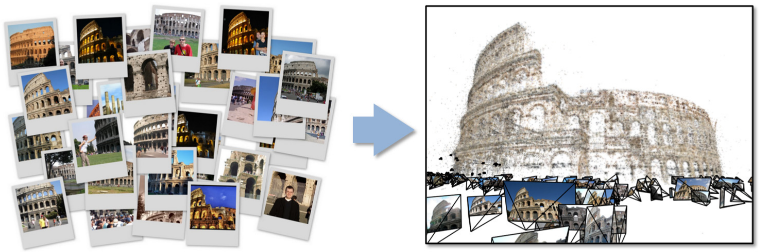
[2] An Introduction to Statistical Learning

1. Randomly assign a number from 1 to K to each observation
2. Iterate until the cluster assignments stop changing
   a. For each of the K clusters: compute the cluster centroid [centroids appear and move]
   b. Assign each observation to the cluster K whose centroid is closest

      [Euclidean distance]

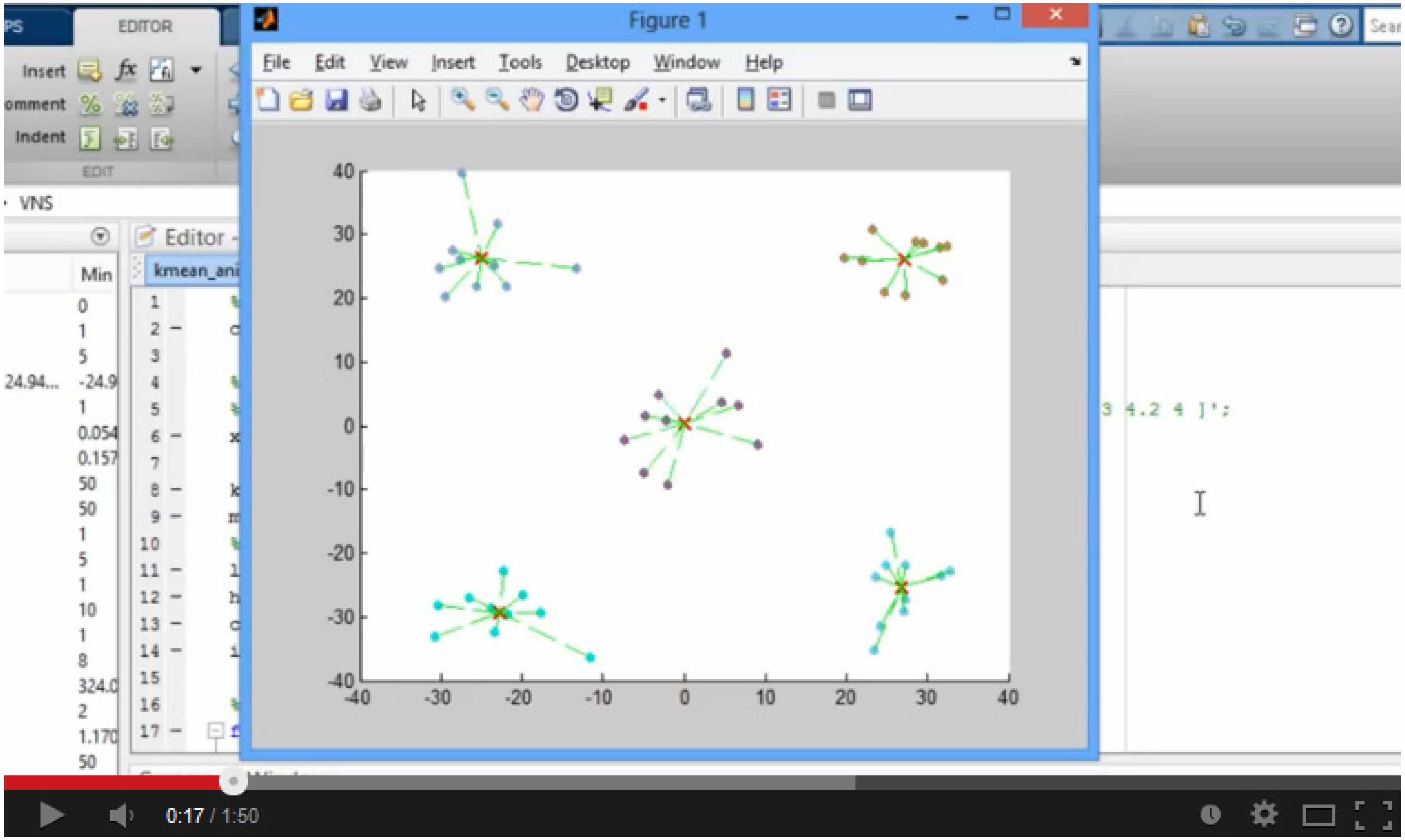
Clustering Methods – K-Means Usage

- Advantages
  - Handles large datasets (larger than hierarchical cluster approaches)
  - Move of observations / data elements between clusters (often improves the overall solution)
- Disadvantages
  - Use of 'means' implies that all variables must be continous
  - Severaly affected by datasets with outliers (→ means)
  - Perform poorly in cases with non-convex (e.g. U-shaped) clusters
- 'Big Data' Application Example
  - Image processing: 7 million images
  - 512 features/attributes per image;
  - 1 million clusters
  - 10000 Map tasks; 64GB broadcasting;
  - 20 TB intermediate data in shuffling;



[4] Judy Qiu, 'Collective communication on Hadoop', 2014

[Video] K-Means Clustering



[5] Animation of the k-means clustering algorithm, YouTube Video

Serial Tool: Statistical Computing with R

- The tool R is a free software environment
  - Many functions/algorithms used for statistical computing and graphics
  - It is a command-line tool with many libraries to download ‘instantly‘
  - Despite of command-line, there are sophisticated graphics possible
- Usage
  - R uses *functions* to perform operations, use `?funcname` for help
  - Call a function with arguments/inputs: `funcname(input1, input2)`
- Selected Hints
  - Hitting [up] n times for previous commands (e.g. slightly modify)
  - [strg+l] clears console

```
[train001@jrl06 tools]$ module load GCC/7.2.0

Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29

The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0

[train001@jrl06 tools]$ module load ParaStationMPI/5.2.0-1
[train001@jrl06 tools]$ module load R/3.4.2
```

[6] Statistical Computing with R

Exercises – Use K-Means Algorithm in R changing K values

Serial Tool: Statistical Computing with R – Startup

- Remember to load modules!

```
[train001@jrl06 tools]$ R

R version 3.4.2 (2017-09-28) -- "Short Summer"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> █
```

Clustering Methods – K-Means with R

```
> ?kmeans
starting httpd help server ... done
```

- Function `kmeans()`
  - `kmeans (x, c, iter.max, nstart, alg, trace)`

Parameters	Description
x	Numeric matrix of data
c	Centers: number of k clusters or set of initial (distinct) cluster centres
iter.max	maximum number of iterations
nstart	If centers number k: amount of random sets
alg	Different types of Algorithms (default:
trace	true/false: trace information on the algorithm progress

Clustering Methods – K-Means with R Example

- Prepare artificial dataset
  - Input x: 50 observations; two dimensional data
  - `set.seed()` function to guarantee reproducible results

```
> set.seed(2)
> x = matrix(rnorm(50*2), ncol=2)
```

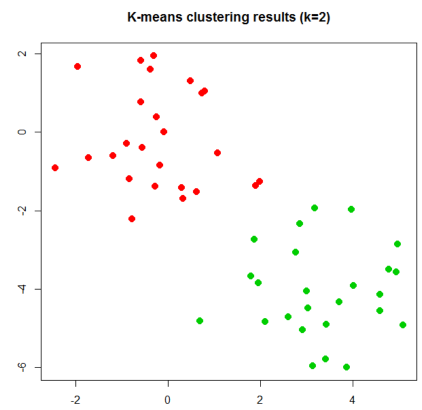
```
> x[1:25,1]=x[1:25,1]+3
> x[1:25,2]=x[1:25,2]-4
```

- Call function `kmeans()`
  - K = 2
  - Output placeholder `km.out`

```
> km.out=kmeans(x,2,nstart=20)
```

- Retrieve cluster assignments `km.out$cluster`
  - Visualize clusters better with `plot()`

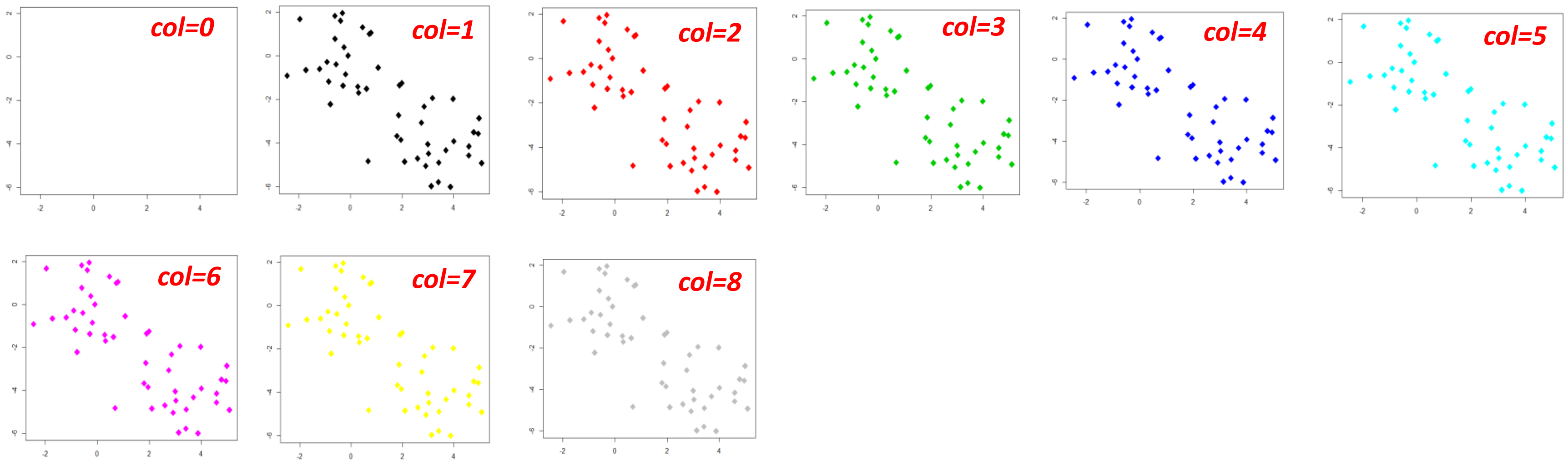
```
> km.out$cluster
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1
[28] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

> plot(x, col=(km.out$cluster+1),main="K-means clustering results (k=2)",
+ xlab="",ylab="",pch=20,cex=2)
```



[R Tool] Data Visualization – Different Colors

- Example: Visualizte data points with `plot()`
  - Using different colors for data points

```
> ?plot
```



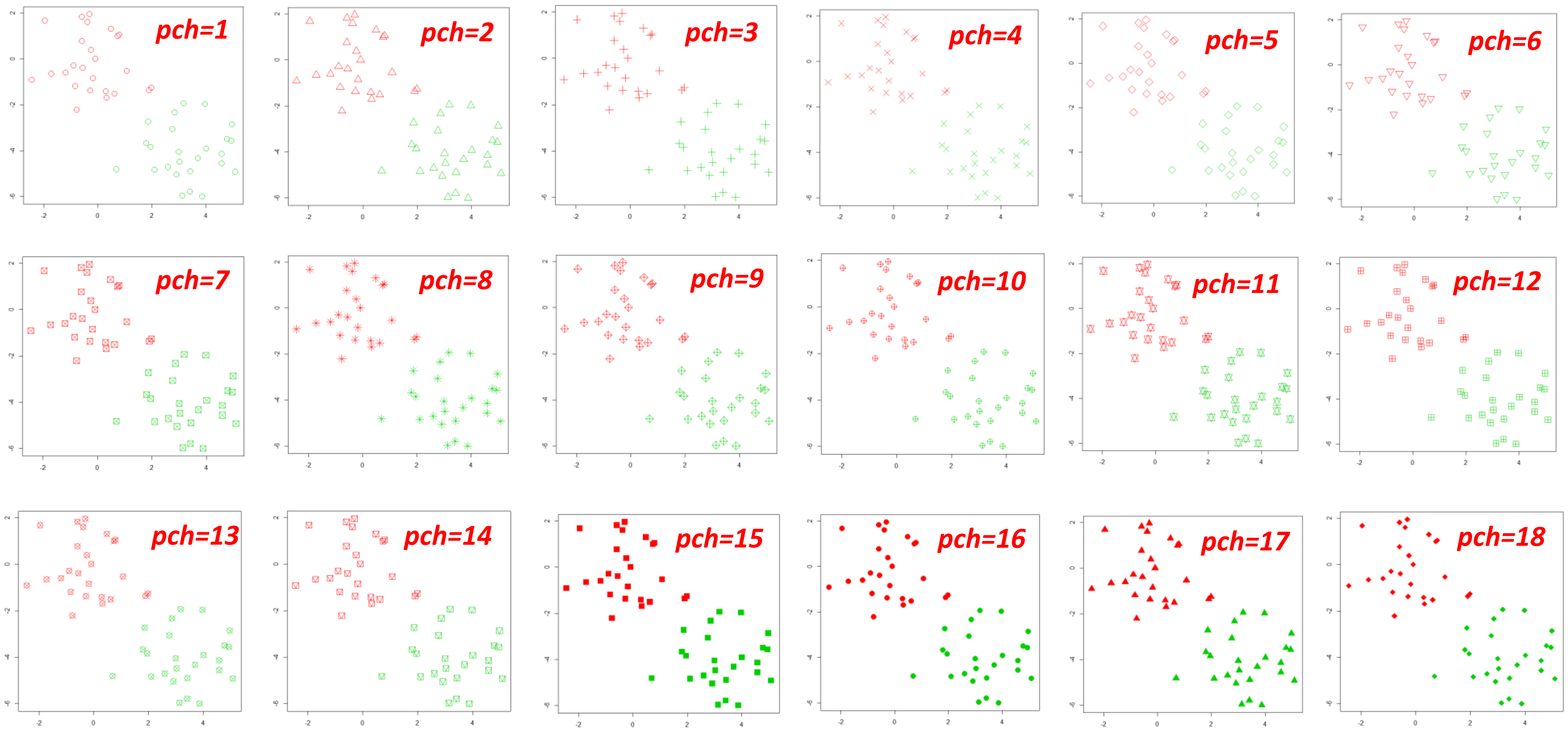

```
> plot(x, col=1,main="K-means clustering results (k=2)", xlab="", ylab="", pch=18,cex=2)
```

- For e.g. 'multi-class' problems above 8 colors
  - Use different 'data point types'

[R Tool] Data Visualization – Different Data Point Types

```
> ?plot
```

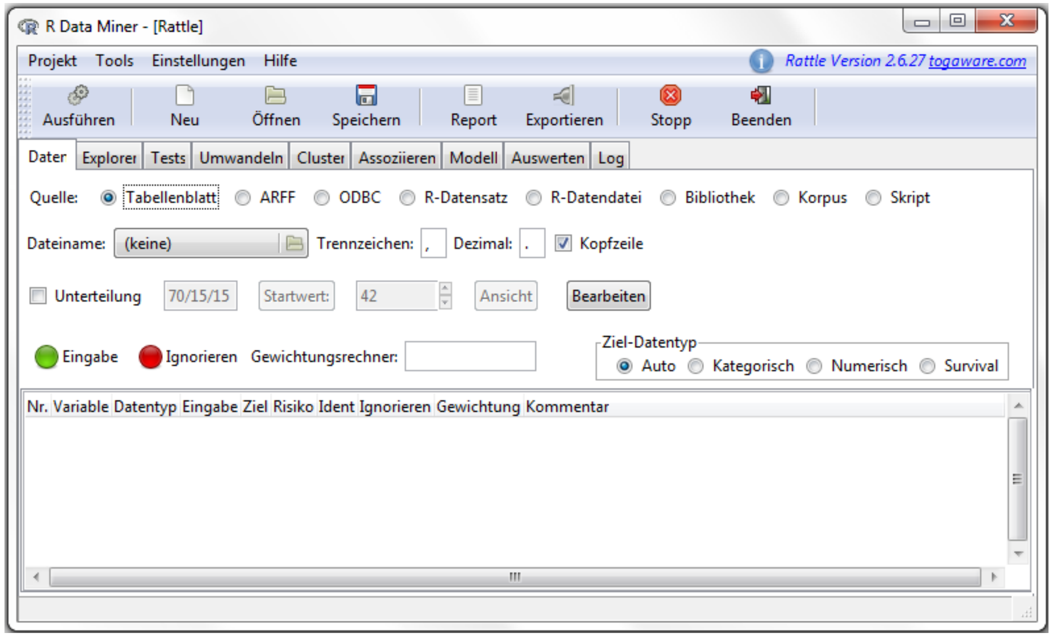
- Example: Clustering output with `plot()`
  - Using different types for data points, cex = [1,2,..] magnifies




```
> plot(x, col=(km.out$cluster+1),main="K-means clustering results (k=2)", xlab="", ylab="", pch=1,cex=1)
```

Working with Rattle: Load and Startup

- Rattle GUI for R
  - Uses a workspace
  - Use mouse instead of commands
  - Loaded with `library()`
  - Start with `rattle()`

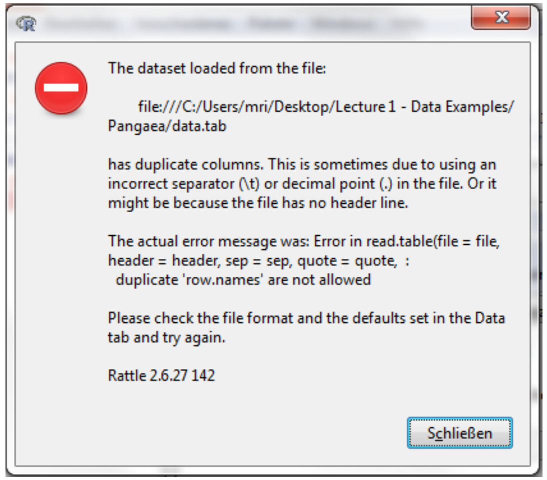



```
> library(rattle)
Rattle: Ein kostenloses grafisches Interface für Data Mining mit R.
Version 2.6.27 r142 Copyright (c) 2006-2013 Togaware Pty Ltd.
Geben Sie 'rattle()' ein, um Ihre Daten mischen.
> rattle()
> |
```

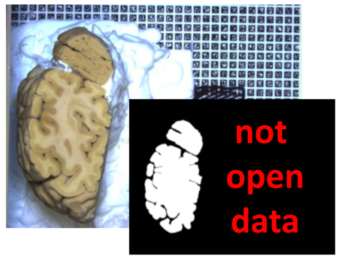
[7] Rattle brochure

Working with Rattle: Load different files – Examples

- Scientific measurement data – Koljoefjords
  - data.tab → original dataset
  - data_ok.tab → header removed
  - data_reform.tab → header reformatted

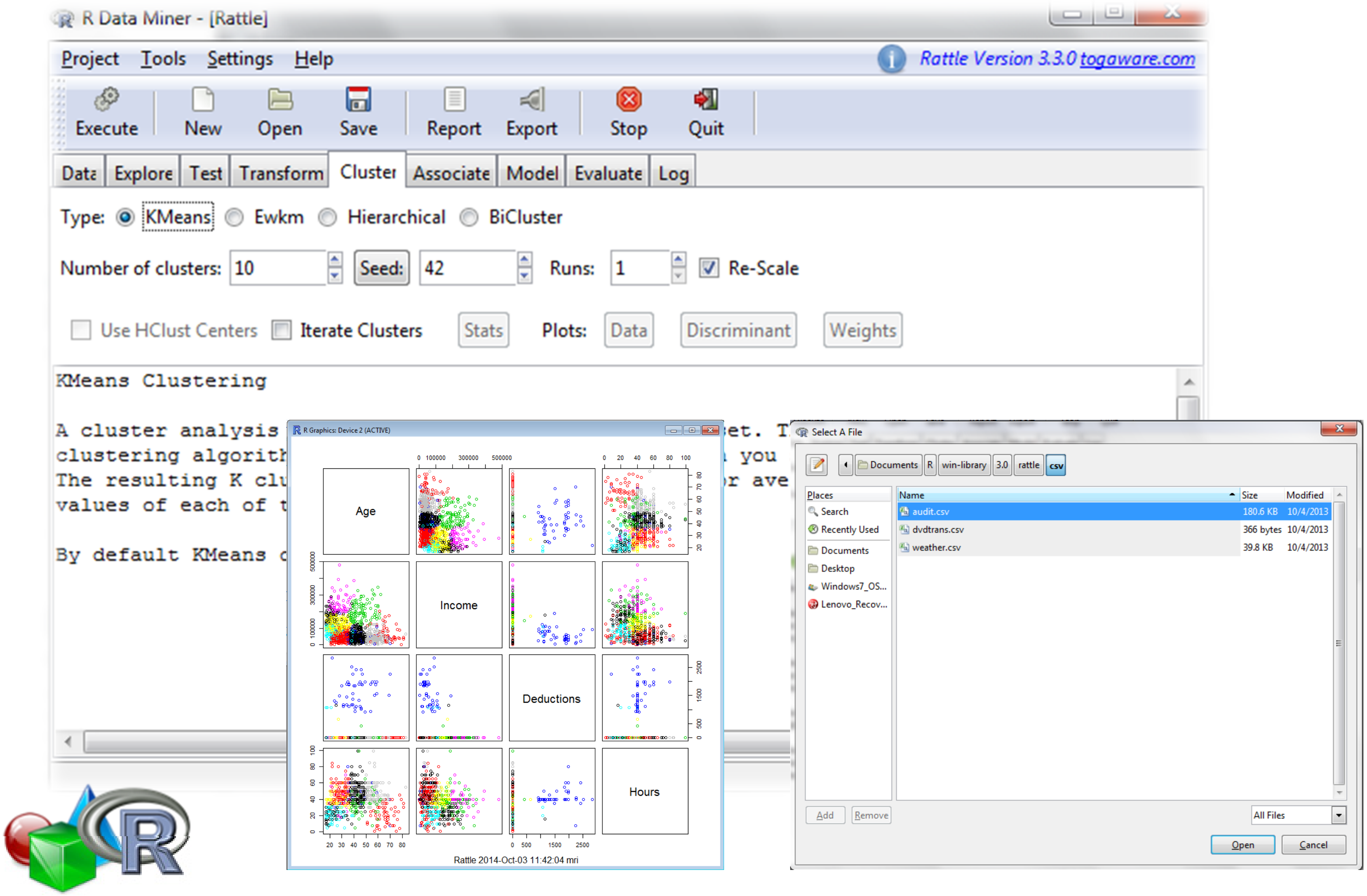


- Shop data – Reykjavik area
  - Shop.csv → shop data
  - Challenges: different languages / encoding
- Scientific big data example – brain images
  - ~700 images: ~40 GB, ~14 MB/image RGB, ~8MB/image mask)
  - Challenges: Data representation, e.g. brain slice images
  - Challenges: Memory limits in serial programs
  - Possible solutions: smart sampling and/or parallelization



- **Serial tools like R, Matlab, scikit-learn show limitations when working on large datasets (big data)**

Clustering Methods – K-Means with Rattle

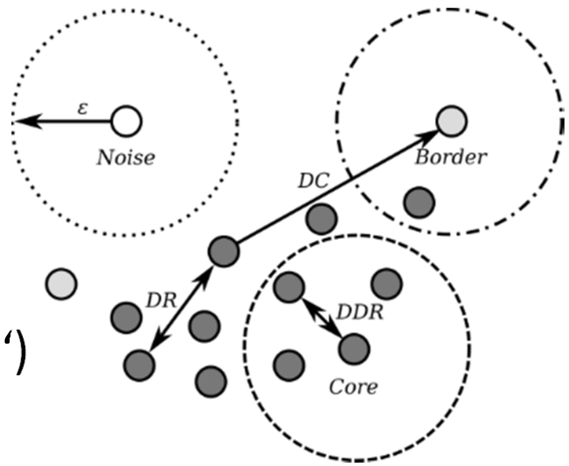


Selected Clustering Methods

- K-Means Clustering – Centroid based clustering
  - Partitions a data set into K distinct clusters (centroids can be artificial)
- K-Medoids Clustering – Centroid based clustering (variation)
  - Partitions a data set into K distinct clusters (centroids are actual points)
- Sequential Agglomerative hierarchic nonoverlapping (SAHN)
  - Hiearchical Clustering (create tree-like data structure → 'dendrogram')
- Clustering Using Representatives (CURE)
  - Select representative points / cluster – as far from one another as possible
- Density-based spatial clustering of applications + noise (DBSCAN)
  - Assumes clusters of similar density or areas of higher density in dataset

DBSCAN Algorithm

- DBSCAN Algorithm *[8] Ester et al.*
  - Introduced 1996 and most cited clustering algorithm
  - Groups number of similar points into clusters of data
  - Similarity is defined by a distance measure (e.g. *euclidean distance*)



(MinPoints = 4)

- Distinct Algorithm Features
  - Clusters a variable number of clusters
  - Forms arbitrarily shaped clusters (except 'bow ties')
  - Identifies inherently also outliers/noise

- Understanding Parameters
  - Looks for a similar points within a given search radius → Parameter *epsilon*
  - A cluster consist of a given minimum number of points → Parameter *minPoints*

(DR = Density Reachable)

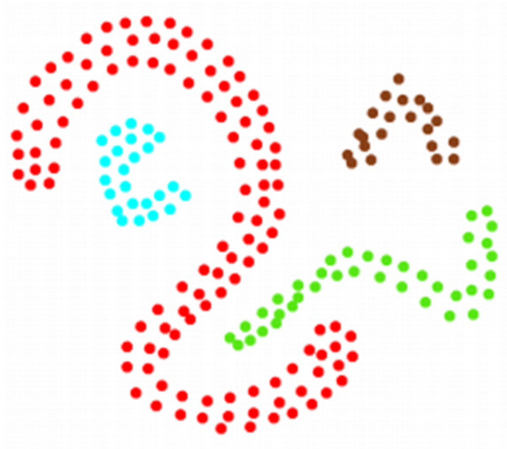
(DDR = Directly Density Reachable)

(DC = Density Connected)

DBSCAN Algorithm – Non-Trivial Example

- Compare K-Means vs. DBSCAN – How would K-Means work?

Unclustered
Data

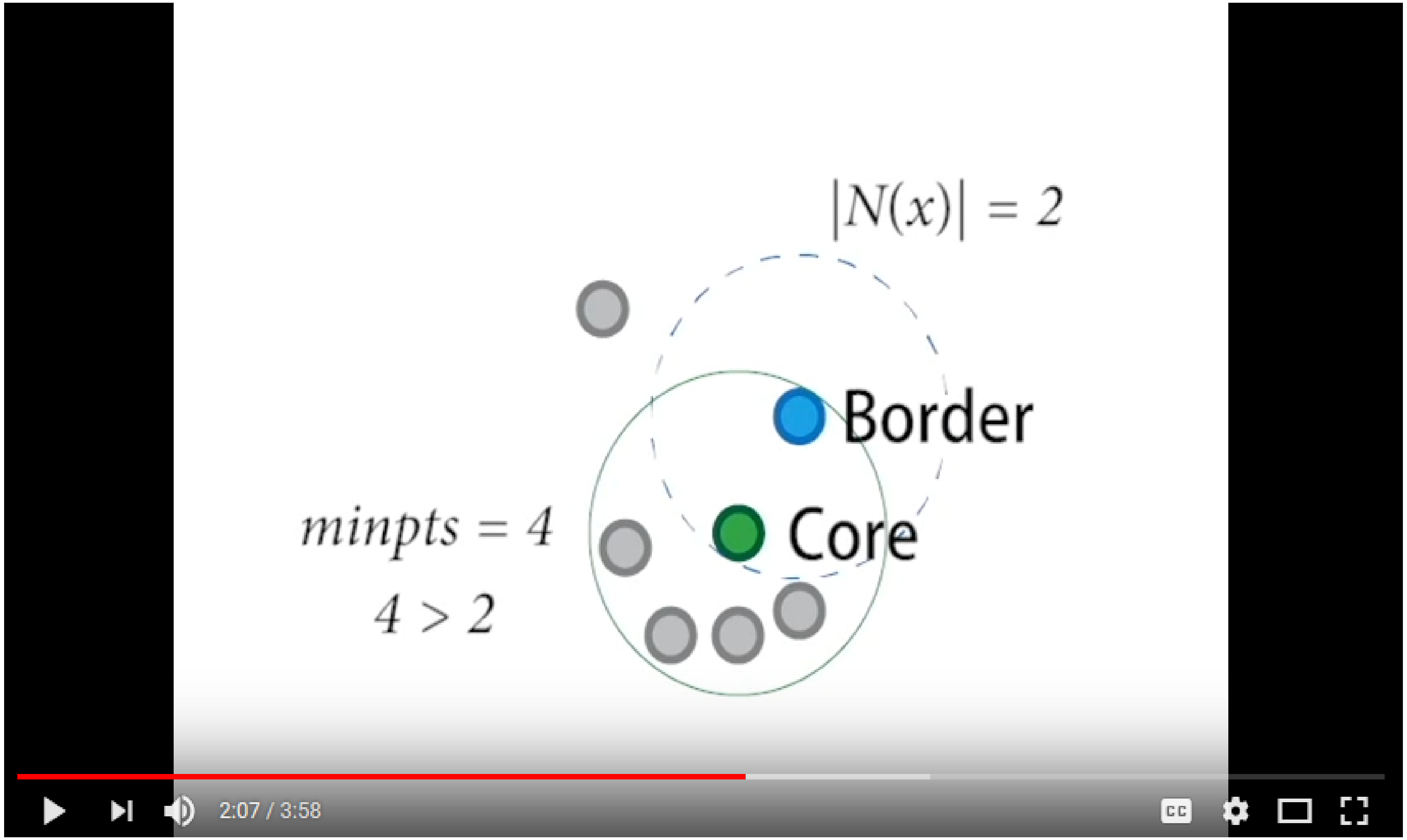


Clustered
Data

- **DBSCAN forms arbitrarily shaped clusters (except 'bow ties') where other clustering algorithms fail**

Exercises – Use DBSCAN Algorithm in R

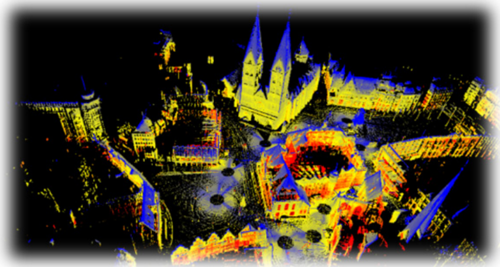
[Video] DBSCAN Clustering

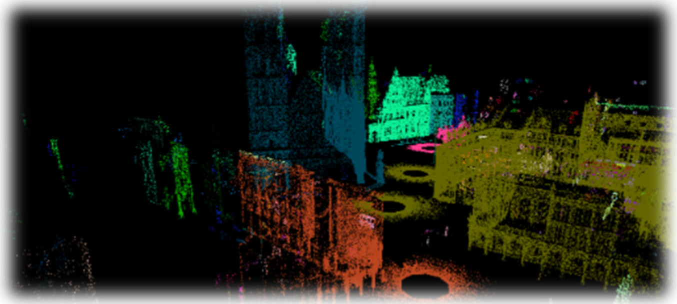


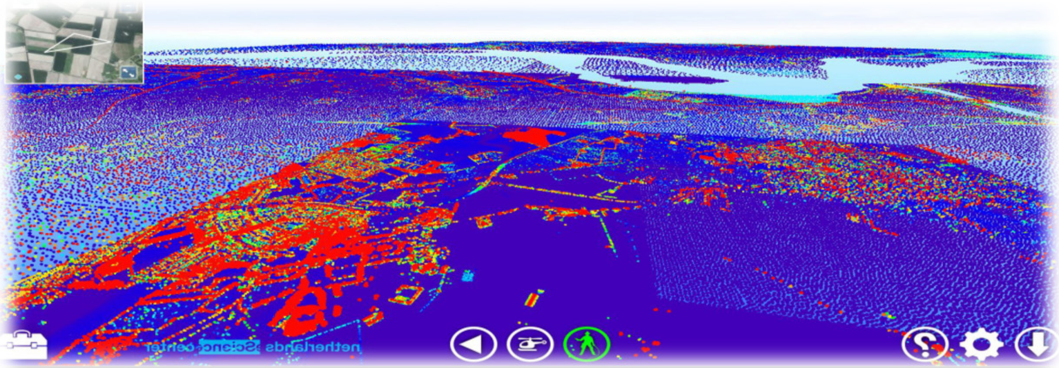
[9] DBSCAN, YouTube Video

Point Cloud Applications

- ‘Big Data‘: 3D/4D laser scans
  - Captured by robots or drones
  - Millions to billion entries
  - Inner cities (e.g. Bremen inner city)
  - Whole countries (e.g. Netherlands)
- Selected Scientific Cases
  - Filter noise to better represent real data
  - Grouping of objects (e.g. buildings)
  - Study level of continous details







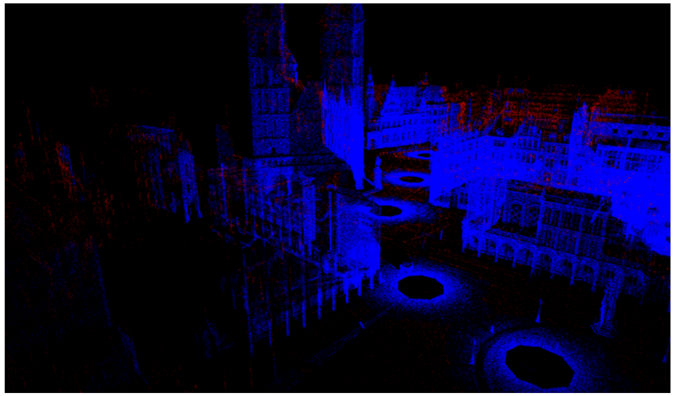
Point Cloud Application Example – Within Buildings

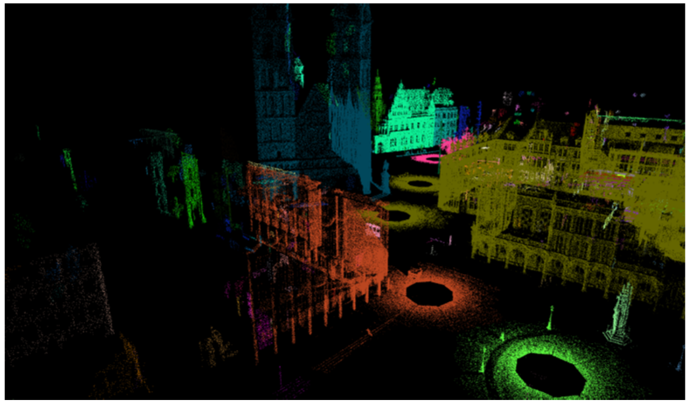
- Point based rendering example
  - Aachen Cathedral based on 3D laser scans and photos
  - Points are rendered as textured and blended splats
  - Visualisation can run in real-time on a desktop PC showing 6 million splats based of a 120 million point laser scan

[10] Aachen Cathedral Point Cloud Rendering, YouTube Video

Bremen Dataset & Locations – Attention: Your Own Copy!

- Different clusterings of the inner city of Bremen
  - Using smart visualizations of the point cloud library (PCL)
  - Big Bremen (81 mio points) & sub sampled Small Bremen (3 mio points)





- **The Bremen Dataset is encoded in the HDF5 format (binary)**
- **You need your own copy of the file in your home directory to cluster!**

```
[train001@jrl07 bremen]$ pwd
/homea/hpclab/train001/data/bremen
[train001@jrl07 bremen]$ ls -al
total 1342208
drwxr-xr-x 2 train001 hpclab        512 Jan 14 09:58 .
drwxr-xr-x 4 train001 hpclab        512 Jan 14 08:38 ..
-rw-r--r-- 1 train001 hpclab 1302382632 Jan 14 09:56 bremen.h5
-rw-r--r-- 1 train001 hpclab   72002416 Jan 14 08:25 bremenSmall.h5
```

[11] Bremen Dataset

B2SHARE Store and Share Research Data

Exercises – Explore & Copy Bremen HDF5 Datasets (binary)

Exercises – Explore & Copy Bremen HDF5 Datasets (binary)

- Copy Bremen datasets to your own home directory (~)

```
[train001@jrl07 bremen]$ pwd
/homea/hpclab/train001/data/bremen
[train001@jrl07 bremen]$ cp * ~
```

- Check your home directory for the Bremen datasets

```
[train001@jrl07 bremen]$ cd ~
[train001@jrl07 ~]$ ls -al
total 1341824
drwxr-x---   13 train001 hpclab      32768 Jan 14 09:44 .
drwxr-xr-x  302 root     sys         32768 Mar 25  2013 ..
-rw-------    1 train001 hpclab       7547 Jan 14 08:28 .bash_history
-rw-r--r--    1 train001 hpclab         18 Jan  8 08:58 .bash_logout
-rw-r--r--    1 train001 hpclab        176 Jan  8 08:58 .bash_profile
-rw-r--r--    1 train001 hpclab        124 Jan  8 08:58 .bashrc
drwxr-xr-x    3 train001 hpclab        512 Jan 14 00:28 bin
-rw-r--r--    1 train001 hpclab 1302382632 Jan 14 09:59 bremen.h5
-rw-r--r--    1 train001 hpclab   72002416 Jan 14 09:59 bremenSmall.h5
```

- Notice binary content

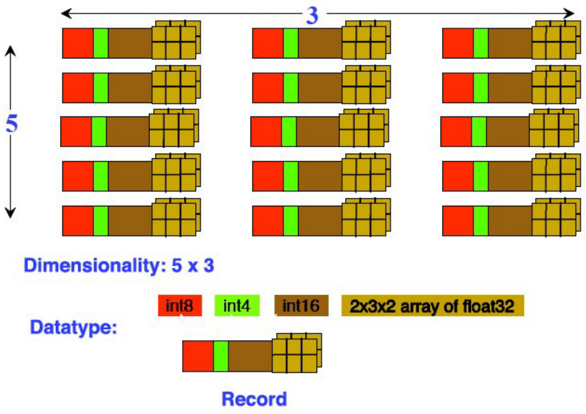
```
[train001@jrl07 ~]$ head bremen.h5
�HDF

��������(ηM���������`����TREE������������������0HEAPX��DBSCANCLOLORS0COLORS((
�
```

Hierarchical Data Format (HDF)

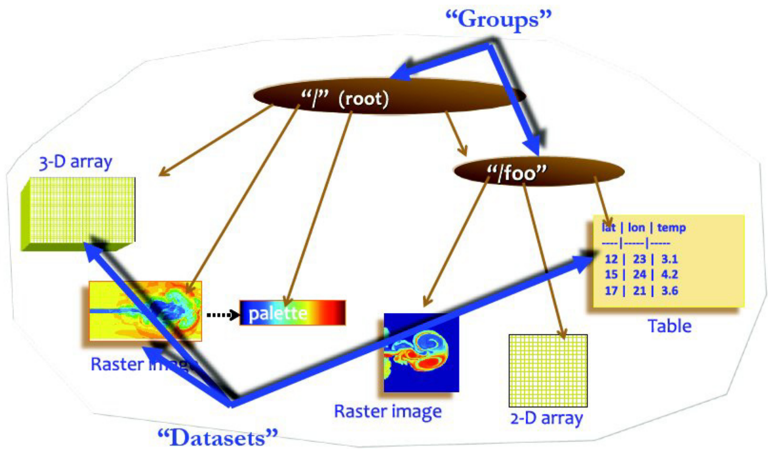
- **HDF is a technology suite that enables the work with extremely large and complex data collections**

[12] HDF@ I/O workshop

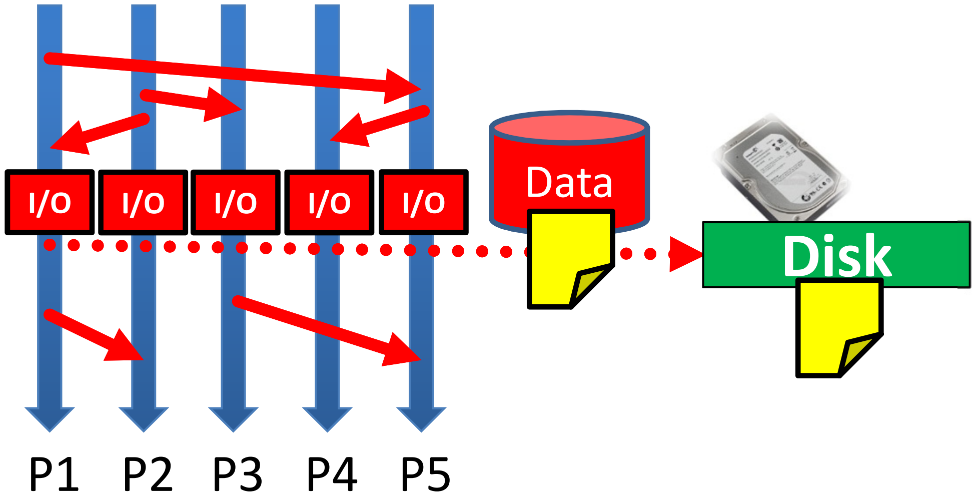
- Simple 'compound type' example:
  - Array of data records with some descriptive information (5x3 dimension)
  - HDF5 data structure type with int(8); int(4); int(16); 2x3x2 array (float32)



'HDF5 file is a container' to organize data objects



HDF5 – Parallel I/O: Shared file



- Each process performs I/O to a single file
  - The file access is 'shared' across all processors involved
  - E.g. MPI/IO functions represent 'collective operations'
- Scalability and Performance
  - 'Data layout' within the shared file is crucial to the performance
  - High number of processors can still create 'contention' for file systems

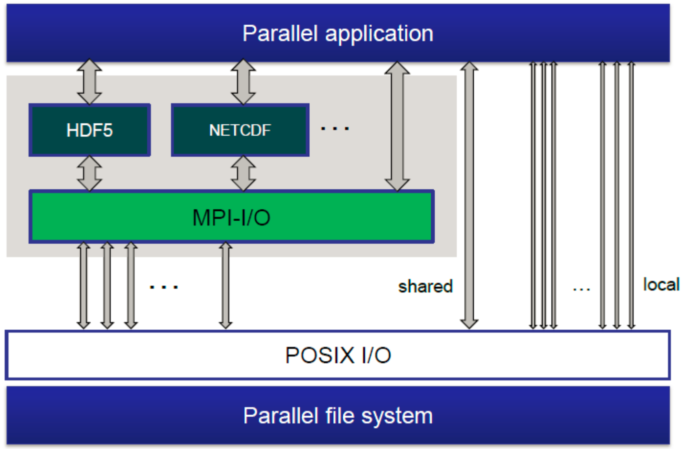
- **Parallel I/O: shared file means that processes can access their 'own portion' of a single file**
- **Parallel I/O with a shared file like MPI/IO is a scalable and even standardized solution**

HDF5 – Parallel I/O & File Systems

- **Hierarchical Data Format (HDF) is designed to store & organize large amounts of numerical data**
- **Parallel Network Common Data Form (NETCDF) is designed to store & organize array-oriented data**

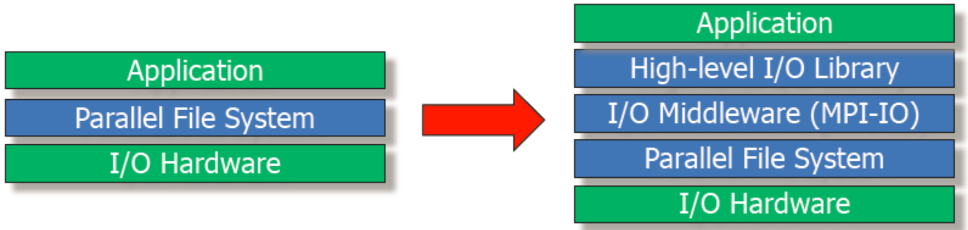
[13] HDF Group *[14] Parallel NETCDF*

- Portable Operating System Interface for UNIX (POSIX) I/O
  - Family of standards to maintain OS compatibility, including I/O interfaces
  - E.g. read(), write(), open(), close(), …(very old interface, some say 'too old')
- 'Higher level I/O libraries' HDF5 & NETCDF
  - Integrated into a parallel application
  - Built on top of MPI I/O for portability
  - Offers machine-independent data access and data formats



I/O with Multiple Layers and Distinct Roles

- **Parallel I/O is supported by multiple software layers with distinct roles that are high-level I/O libraries, I/O middleware, and parallel file systems**



[15] R. Thakur, PRACE Training, Parallel I/O and MPI I/O

- High-Level I/O Library
  - Maps application abstractions to a structured portable file format
  - E.g. HDF-5, Parallel NetCDF
- I/O Middleware
  - E.g. MPI I/O
  - Deals with organizing access by many processes
- Parallel Filesystem
  - Maintains logical space and provides efficient access to data
  - E.g. GPFS, Lustre, PVFS

Exercises – Bremen HDF5 Viewer

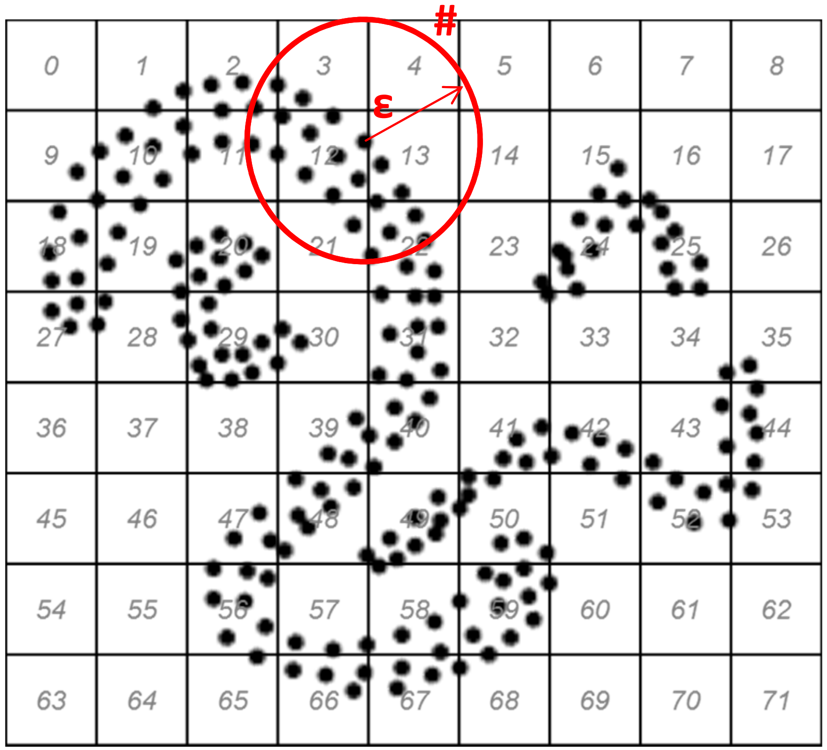
Review of Parallel DBSCAN Implementations

Technology	Platform Approach	Analysis
HPDBSCAN (authors implementation)	C; MPI; OpenMP	Parallel, hybrid, DBSCAN
Apache Mahout	Java; Hadoop	K-means variants, spectral, no DBSCAN
Apache Spark/MLlib	Java; Spark	Only k-means clustering, No DBSCAN
scikit-learn	Python	No parallelization strategy for DBSCAN
Northwestern University PDSDBSCAN-D	C++; MPI; OpenMP	Parallel DBSCAN

[16] M. Goetz, M. Riedel et al., 'On Parallel and Scalable Classification and Clustering Techniques for Earth Science Datasets', *6th Workshop on Data Mining in Earth System Science, International Conference of Computational Science (ICCS)*

HDBSCAN Algorithm Details

- Parallelization Strategy
  - Smart 'Big Data' Preprocessing into Spatial Cells ('indexed')
  - OpenMP and HDF5 parallel I/O
  - MPI (+ optional OpenMP hybrid)
- Preprocessing Step
  - Spatial indexing and redistribution according to the point localities
  - Data density based chunking of computations
- Computational Optimizations
  - Caching of point neighborhood searches
  - Cluster merging based on comparisons instead of zone reclustering



[17] M.Goetz, M. Riedel et al., 'HPDBSCAN – Highly Parallel DBSCAN', MLHPC Workshop at Supercomputing 2015

Exercises – Bremen Small HPDBSCAN Runs

HPC Environment – Modules Revisited

- Module environment tool
  - Avoids to manually setup environment information for every application
  - Simplifies shell initialization and lets users easily modify their environment
- Module avail
  - Lists all available modules on the HPC system (e.g. compilers, MPI, etc.)
- Module spider
  - Find modules in the installed set of modules and more information
- Module load → needed before HPDBSCAN run
  - Loads particular modules into the current work environment, E.g.:

```
[train001@jrl12 ~]$ module load GCC

Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29

The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0

[train001@jrl12 ~]$ module load ParaStationMPI/5.2.0-1
[train001@jrl12 ~]$ module load HDF5/1.8.19
```

JURECA HPC System – HPDBSCAN Job Script Example

```
#!/bin/bash
#SBATCH --job-name=HPDBSCAN
#SBATCH -o HPDBSCAN-%j.out
#SBATCH -e HPDBSCAN-%j.err
#SBATCH --nodes=2
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=4
#SBATCH --time=00:20:00
#SBATCH --cpus-per-task=4
#SBATCH --reservation=ml-hpc-1

export OMP_NUM_THREADS=4

# location executable
HPDBSCAN=/homea/hpclab/train001/tools/hpdbscan/dbscan

# your own copy of bremen small
BREMENSMALLDATA=/homea/hpclab/train001/bremenSmall.h5

# your own copy of bremen big
BREMENBIGDATA=/homea/hpclab/train001/bremen.h5

srun $HPDBSCAN -m 100 -e 300 -t 12 $BREMENSMALLDATA
```

- **Job submit using command: sbatch <jobscript>**
- **Remember your <jobid> that is returned from the sbatch command**
- **Show status of the job then with: squeue –u <your-user-id>**

(parameters of DBSCAN and file to be clustered)

- **Note the tutorial reservation with –reservation= bigdata-cpu just valid for Thursday morning**

JURECA HPC System – HPDBSCAN Job Submit

- Load module environment (once after login)

```
[train001@jrl07 jsc_mpi]$ module load GCC

Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29

The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0

[train001@jrl07 jsc_mpi]$ module load ParaStationMPI/5.2.0-1
[train001@jrl07 jsc_mpi]$ module load HDF5/1.8.19
```

- Submit job via jobscript

```
[train001@jrl07 jsc_mpi]$ sbatch submit-clustering-bremen.sh
Submitted batch job 4629728
```

- Check job status (and cancel if needed)

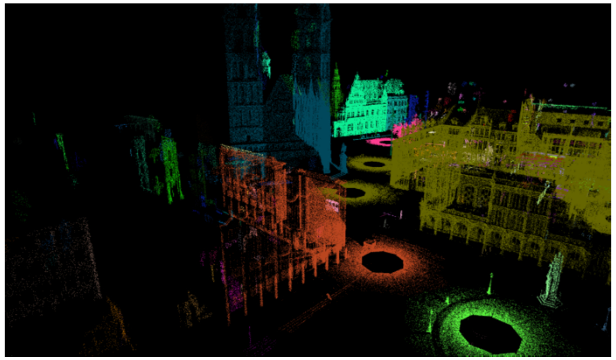
(scancel might take a second or two to take effect)

```
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4629867     batch HPDBSCAN train001  R       2:20      2 jrc[0672-0673]
[train001@jrl07 hpdbscan]$ scancel 4629867
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4629867     batch HPDBSCAN train001 CG       2:34      2 jrc[0672-0673]
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

JURECA HPC System – HPDBSCAN Check Outcome

```
[train001@jrl07 jsc_mpi]$ more HPDBSCAN-4629640.out
Calculating Cell Space...
        Computing Dimensions... [OK] in 0.001657
        Computing Cells...      [OK] in 0.029877
        Sorting Points...       [OK] in 0.174414
        Distributing Points...  [OK] in 0.113745
DBSCAN...
        Local Scan...           [OK] in 58.095238
        Merging Neighbors...    [OK] in 0.005433
        Adjust Labels ...       [OK] in 0.004473
        Rec. Init. Order ...    [OK] in 0.559311
        Writing File ...        [OK] in 0.008467
Result...
        65      Clusters
        2973821 Cluster Points
        26179   Noise Points
        2953129 Core Points
Took: 59.111594s
```

- The outcome of the clustering process is written directly into the HDF5 file using cluster IDs and noise IDs

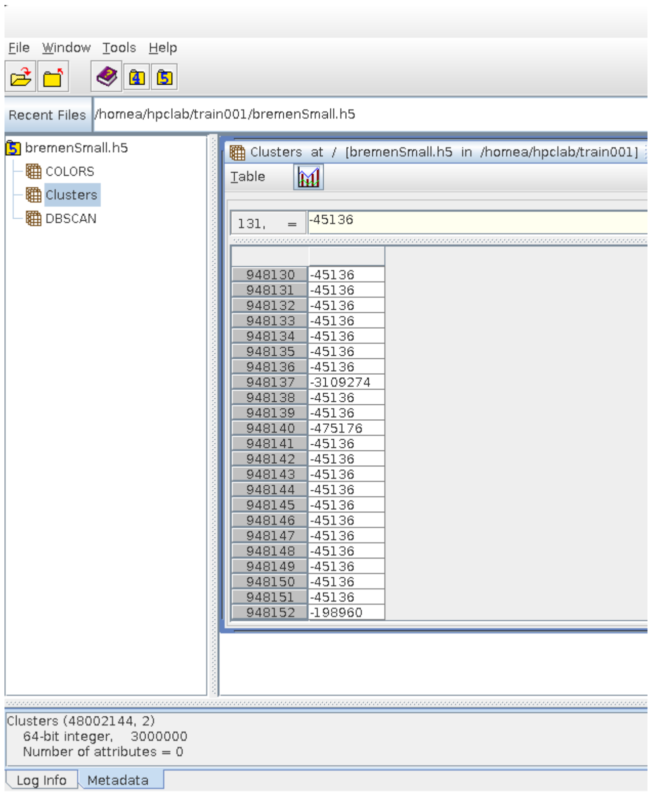


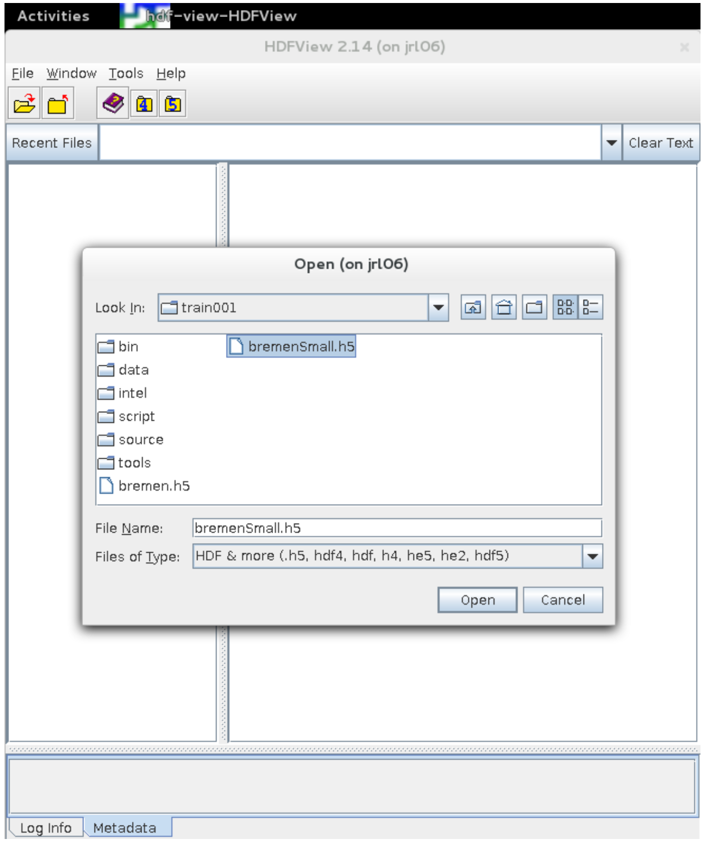
```
[train001@jrl07 ~]$ ls -al
total 1124800
drwxr-x---  13 train001 hpclab      32768 Jan 14 08:47 .
drwxr-xr-x 302 root     sys         32768 Mar 25  2013 ..
-rw-------   1 train001 hpclab       7547 Jan 14 08:28 .bash_history
-rw-r--r--   1 train001 hpclab         18 Jan  8 08:58 .bash_logout
-rw-r--r--   1 train001 hpclab        176 Jan  8 08:58 .bash_profile
-rw-r--r--   1 train001 hpclab        124 Jan  8 08:58 .bashrc
drwxr-xr-x   3 train001 hpclab        512 Jan 14 00:28 bin
-rw-r--r--   1 train001 hpclab 1079412312 Jan 14 08:39 bremen.h5.h5
-rw-r--r--   1 train001 hpclab   72002416 Jan 14 08:47 bremenSmall.h5.h5
```

HDFView Example – Bremen Output

- HDFView is a visual tool for browsing and editing HDF files
  - Tools is using a GUI thus needs ssh –X when log into JURECA

```
[train001@jrl06 ~]$ module load HDFView/2.14-Java-1.8.0_144
[train001@jrl06 ~]$ hdfview.sh
```





Point Cloud Viewer Example – Bremen Output (1)

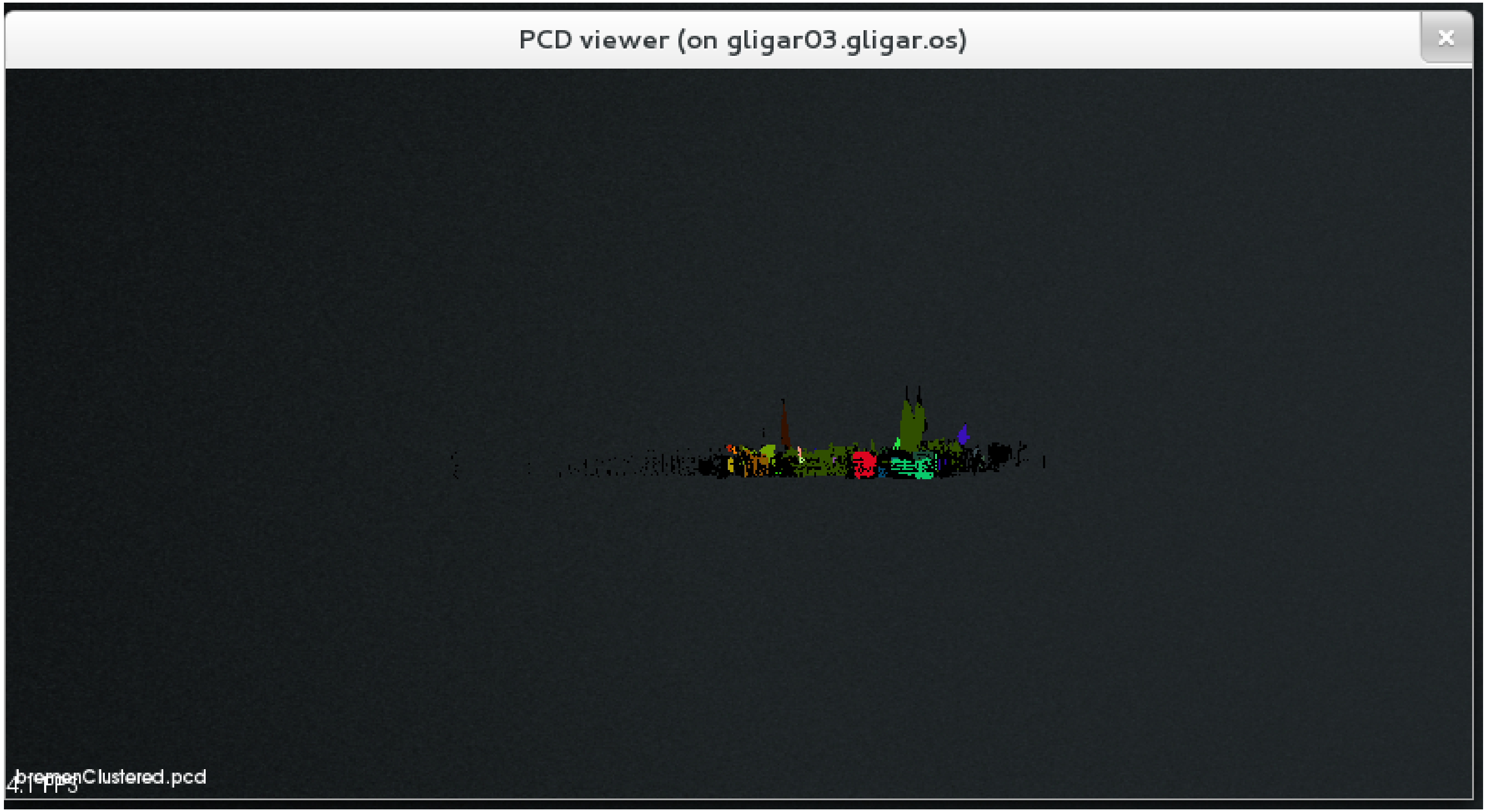
- Data formats
  - Small python script to change data formats from HDF5 to PCD

```
[train001@jrl09 ~]$ pwd
/homea/hpclab/train001
[train001@jrl09 ~]$ ls H5toPCD.py
H5toPCD.py
```

- Module load PCL
  - The PCL viewer application requires an SSH –X connection

```
[vsc42544@gligar03 Bremen]$ module load PCL/1.8.1-intel-2017b-Python-2.7.14
[vsc42544@gligar03 Bremen]$ pwd
/apps/gent/tutorials/machine_learning/clustering/Bremen
[vsc42544@gligar03 Bremen]$ ls -al
total 3431616
drwxr-xr-x 2 vsc40003 vsc40003       4096 Nov 22 22:39 .
drwxr-xr-x 5 vsc40003 vsc40003       4096 Nov 22 15:44 ..
-rw-r--r-- 1 vsc40003 vsc40003  382559971 Nov 22 22:39 bremenClustered.pcd
-rw-r--r-- 1 vsc40003 vsc40003 1302382632 Nov 22 14:07 bremen.h5.h5
-rw-r--r-- 1 vsc40003 vsc40003   72002416 Jan 13  2017 bremenSmall.h5.h5
[vsc42544@gligar03 Bremen]$ pcl_viewer bremenClustered.pcd
```

Point Cloud Viewer Example – Bremen Output (2)

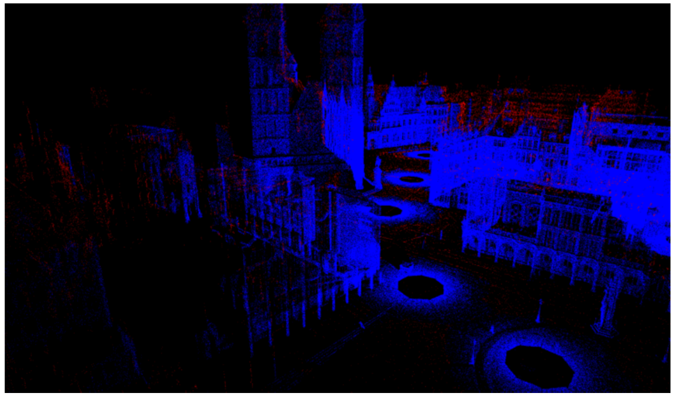


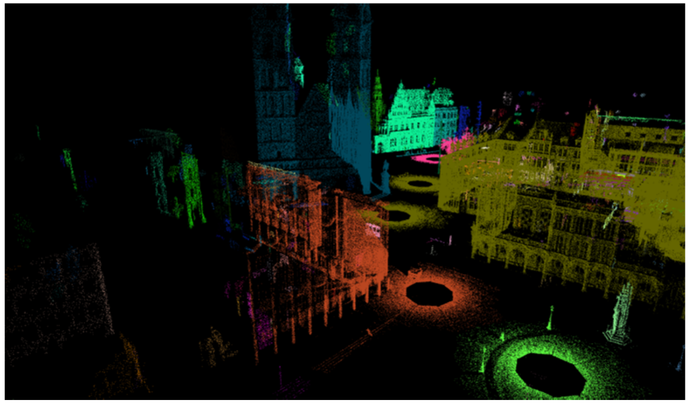
- **Use Strg and Mouse Wheel to Zoom and use numbers of keyboard for different visualizations**

Exercises – Bremen HPDBSCAN Check Outputs

Bremen Dataset & Locations – Revisited

- Different clusterings of the inner city of Bremen
  - Using smart visualizations of the point cloud library (PCL)
  - Big Bremen (81 mio points) & sub sampled Small Bremen (3 mio points)





- **The Bremen Dataset is encoded in the HDF5 format (binary)**
- **You need your own copy of the file in your home directory to cluster!**

```
[train001@jrl07 bremen]$ pwd
/homea/hpclab/train001/data/bremen
[train001@jrl07 bremen]$ ls -al
total 1342208
drwxr-xr-x 2 train001 hpclab        512 Jan 14 09:58 .
drwxr-xr-x 4 train001 hpclab        512 Jan 14 08:38 ..
-rw-r--r-- 1 train001 hpclab 1302382632 Jan 14 09:56 bremen.h5
-rw-r--r-- 1 train001 hpclab   72002416 Jan 14 08:25 bremenSmall.h5
```

[11] Bremen Dataset

B2SHARE Store and Share Research Data

Exercises – Bremen Big HPDBSCAN Runs

HPC Environment – Modules Revisited

- Module environment tool
  - Avoids to manually setup environment information for every application
  - Simplifies shell initialization and lets users easily modify their environment
- Module avail
  - Lists all available modules on the HPC system (e.g. compilers, MPI, etc.)
- Module spider
  - Find modules in the installed set of modules and more information
- Module load → needed before HPDBSCAN run
  - Loads particular modules into the current work environment, E.g.:

```
[train001@jrl12 ~]$ module load GCC

Due to MODULEPATH changes, the following have been reloaded:
  1) binutils/.2.29

The following have been reloaded with a version change:
  1) GCCcore/.5.4.0 => GCCcore/.7.2.0

[train001@jrl12 ~]$ module load ParaStationMPI/5.2.0-1
[train001@jrl12 ~]$ module load HDF5/1.8.19
```

JURECA HPC System – HPDBSCAN Job Script

```
#!/bin/bash
#SBATCH --job-name=HPDBSCAN
#SBATCH -o HPDBSCAN-%j.out
#SBATCH -e HPDBSCAN-%j.err
#SBATCH --nodes=2
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=4
#SBATCH --time=00:20:00
#SBATCH --cpus-per-task=4
#SBATCH --reservation=ml-hpc-1

export OMP_NUM_THREADS=4

# location executable
HPDBSCAN=/homea/hpclab/train001/tools/hpdbscan/dbscan

# your own copy of bremen small
BREMENSMALLDATA=/homea/hpclab/train001/bremenSmall.h5

# your own copy of bremen big
BREMENBIGDATA=/homea/hpclab/train001/bremen.h5

srun $HPDBSCAN -m 100 -e 300 -t 12 $BREMENSMALLDATA
```

- **Job submit using command:**
  sbatch <jobscript>
- **Remember your <jobid> that is returned from the sbatch command**
- **Show status of the job then with:**
  squeue –u <your-user-id>

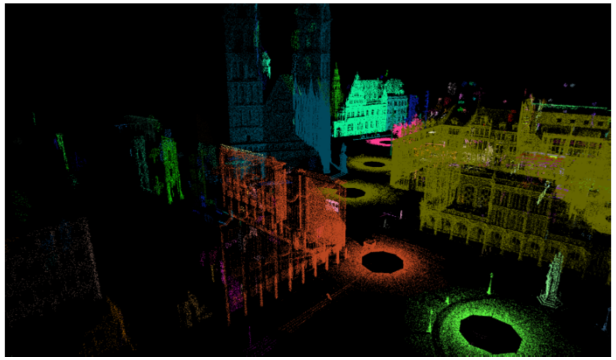
(parameters of DBSCAN and file to be clustered)

- **Note the tutorial reservation with –reservation= bigdata-cpu just valid for Thursday morning**

JURECA HPC System – HPDBSCAN Check Outcome

```
[train001@jrl07 jsc_mpi]$ more HPDBSCAN-4629640.out
Calculating Cell Space...
        Computing Dimensions... [OK] in 0.001657
        Computing Cells...      [OK] in 0.029877
        Sorting Points...       [OK] in 0.174414
        Distributing Points...  [OK] in 0.113745
DBSCAN...
        Local Scan...           [OK] in 58.095238
        Merging Neighbors...    [OK] in 0.005433
        Adjust Labels ...       [OK] in 0.004473
        Rec. Init. Order ...    [OK] in 0.559311
        Writing File ...        [OK] in 0.008467
Result...
        65      Clusters
        2973821 Cluster Points
        26179   Noise Points
        2953129 Core Points
Took: 59.111594s
```

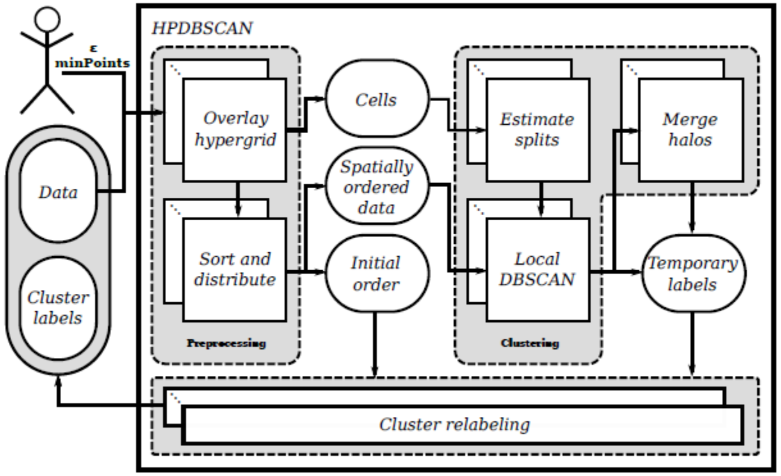
- The outcome of the clustering process is written directly into the HDF5 file using cluster IDs and noise IDs



```
[train001@jrl07 ~]$ ls -al
total 1124800
drwxr-x---  13 train001 hpclab      32768 Jan 14 08:47 .
drwxr-xr-x 302 root     sys         32768 Mar 25  2013 ..
-rw-------   1 train001 hpclab       7547 Jan 14 08:28 .bash_history
-rw-r--r--   1 train001 hpclab         18 Jan  8 08:58 .bash_logout
-rw-r--r--   1 train001 hpclab        176 Jan  8 08:58 .bash_profile
-rw-r--r--   1 train001 hpclab        124 Jan  8 08:58 .bashrc
drwxr-xr-x   3 train001 hpclab        512 Jan 14 00:28 bin
-rw-r--r--   1 train001 hpclab 1079412312 Jan 14 08:39 bremen.h5.h5
-rw-r--r--   1 train001 hpclab   72002416 Jan 14 08:47 bremenSmall.h5.h5
```

HPDBSCAN – Smart Domain Decomposition Example

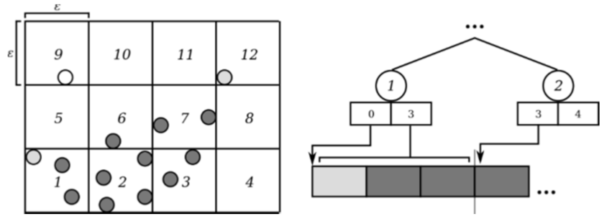
- Parallelization Strategy
  - Chunk data space equally
  - Overlay with hypergrid
  - Apply cost heuristic
  - Redistribute points (data locality)
  - Execute DBSCAN locally
  - Merge clusters at chunk edges
  - Restore initial order
- Data organization
  - Use of HDF5
  - Cluster Id / noise ID
    stored in HDF5 file

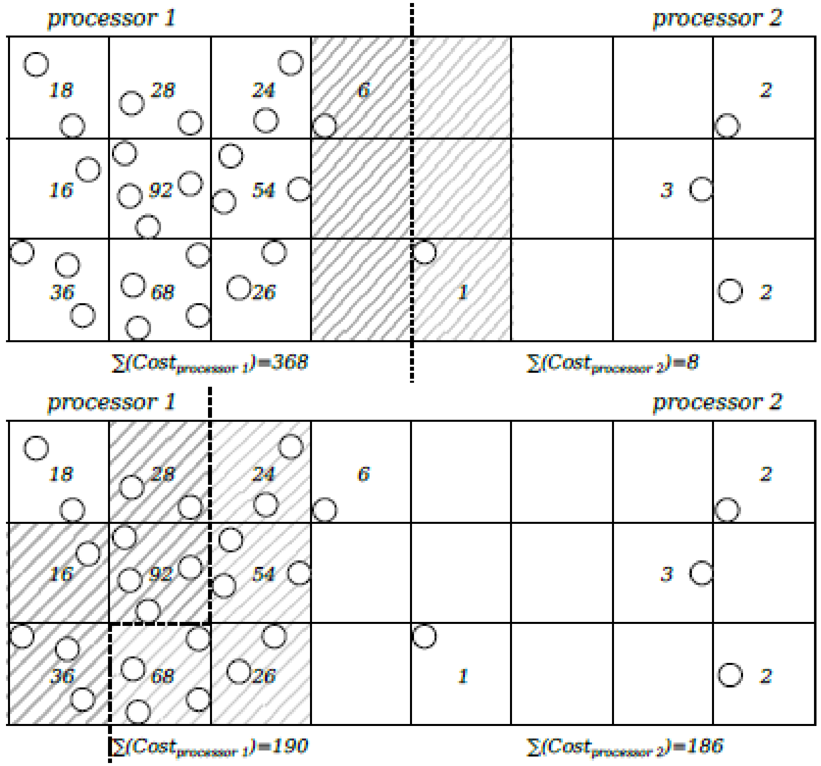


[17] M.Goetz, M. Riedel et al.,
'HPDBSCAN – Highly Parallel DBSCAN',
MLHPC Workshop at Supercomputing 2015

HPDBSCAN – Domain Decomposition

- Parallelization Strategy
  - Chunk data space equally
  - Overlay with hypergrid
  - Apply cost heuristic
  - Redistribute points (data locality)
  - Execute DBSCAN locally
  - Merge clusters at chunk edges
  - Restore initial order
- Data organization
  - Use of HDF5
  - Cluster Id / noise ID stored in HDF5 file

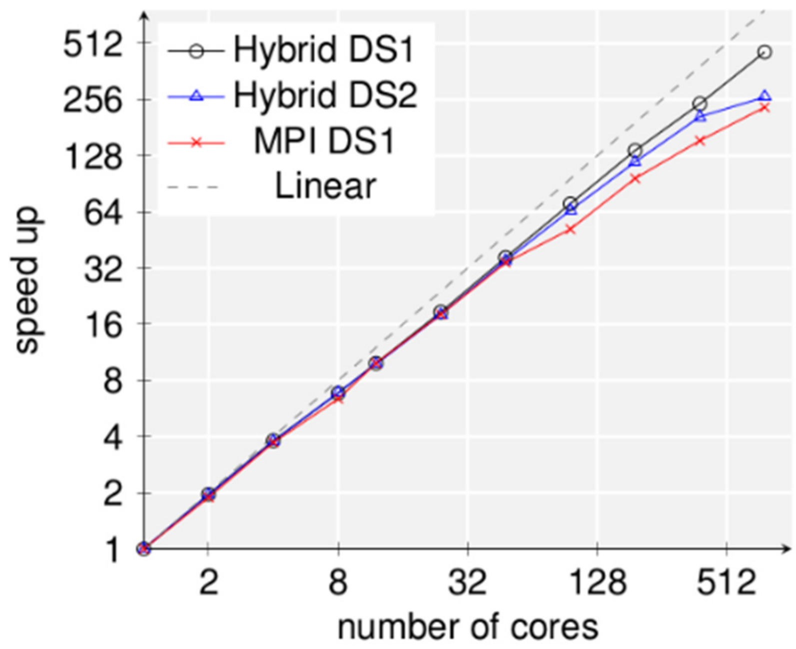




[17] M.Goetz, M. Riedel et al., 'HPDBSCAN – Highly Parallel DBSCAN', MLHPC Workshop at Supercomputing 2015

HPDBSCAN – Scaling

- Parallelization Strategy
  - Chunk data space equally
  - Overlay with hypergrid
  - Apply cost heuristic
  - Redistribute points (data locality)
  - Execute DBSCAN locally
  - Merge clusters at chunk edges
  - Restore initial order
- Data organization
  - Use of HDF5
  - Cluster Id / noise ID
    stored in HDF5 file



(DS1 = Bremen; DS2 = Twitter)

[17] M.Goetz, M. Riedel et al.,
'HPDBSCAN – Highly Parallel DBSCAN',
MLHPC Workshop at Supercomputing 2015

JURECA HPC System – HPDBSCAN Check Outcome

```
[train001@jrl04 hpdbscan]$ more HPDBSCAN-4632910.out
Calculating Cell Space...
        Computing Dimensions... [OK] in 0.002393
        Computing Cells...      [OK] in 0.498816
        Sorting Points...       [OK] in 0.891462
        Distributing Points...  [OK] in 2.576206
DBSCAN...
        Local Scan...           [OK] in 1.375779
        Merging Neighbors...    [OK] in 0.000586
        Adjust Labels ...       [OK] in 0.013686
        Rec. Init. Order ...    [OK] in 0.640681
        Writing File ...        [OK] in 0.006232
Result...
        8976    Clusters
        906807  Cluster Points
        2797544 Noise Points
        757369  Core Points
Took: 6.189666s
```

- **The outcome of the clustering process is written directly into the HDF5 file using cluster IDs and noise IDs**

Bremen Big Dataset – 'Running against the Wall' (1)

- Configured walltime
  - 1:00 hour in jobscript; 2 nodes (4 tasks per node)
- Check job status (shortly before the hour)

```
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4629896     batch HPDBSCAN train001  R      59:36      2 jrc[1250-1251]
```

- Job gets automatically cancelled by scheduler

```
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4629896     batch HPDBSCAN train001 CG    1:00:27      2 jrc[1250-1251]
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[train001@jrl07 hpdbscan]$ squeue -u train001
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

- **In parallel & scalable machine learning one needs to adjust the walltimes of jobs to the complexity in processing time and/or size of the dataset (cf. Bremen small vs. Bremen big)**
- **Determining the right amount of walltime is not easy and mostly be best obtained by test runs**
- **The required walltime depends on the number of used nodes (and tasks) and is directly linked**

Bremen Big Dataset – 'Running against the Wall' (2)

- Check outcome of the job

```
[train001@jrl07 hpdbscan]$ more HPDBSCAN-4629896.out
Calculating Cell Space...
        Computing Dimensions... [OK] in 0.043040
        Computing Cells...      [OK] in 0.157041
        Sorting Points...       [OK] in 1.041985
        Distributing Points...  [OK] in 2.126353
DBSCAN...
        Local Scan...
```

- Check error report of the job

```
[train001@jrl07 hpdbscan]$ more HPDBSCAN-4629896.err
HDF5-DIAG: Error detected in HDF5 (1.8.19) MPI-process 0:
  #000: H5F.c line 772 in H5Fclose(): not a file ID
    major: Invalid arguments to routine
    minor: Inappropriate type
error: *** step 4629896 CANCELLED DUE TO TIME LIMIT ***
srun: Job step aborted: Waiting up to 6 seconds for job step to finish.
srun: error: jrc1250: tasks 0-1: Terminated
```

- **The partial result of clustering when terminated is not useful and should be not used anymore**
- **In case of termination by scheduler even HDF problems might occur that corrupt the file**

Exercises – Increasing Number of Nodes

JURECA HPC System – HPDBSCAN Job Script

```
#!/bin/bash
#SBATCH --job-name=HPDBSCAN
#SBATCH -o HPDBSCAN-%j.out
#SBATCH -e HPDBSCAN-%j.err
#SBATCH --nodes=2
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=4
#SBATCH --time=00:20:00
#SBATCH --cpus-per-task=4
#SBATCH --reservation=ml-hpc-1

export OMP_NUM_THREADS=4

# location executable
HPDBSCAN=/homea/hpclab/train001/tools/hpdbscan/dbscan

# your own copy of bremen small
BREMENSMALLDATA=/homea/hpclab/train001/bremenSmall.h5

# your own copy of bremen big
BREMENBIGDATA=/homea/hpclab/train001/bremen.h5

srun $HPDBSCAN -m 100 -e 300 -t 12 $BREMENSMALLDATA
```

(parameters of DBSCAN and file to be clustered)

- **Job submit using command: sbatch <jobscript>**
- **Remember your <jobid> that is returned from the sbatch command**
- **Show status of the job then with: squeue –u <your-user-id>**

- **Note the tutorial reservation with –reservation= bigdata-cpu just valid for Thursday morning**

Exercises – Changing Epsilon & MinPoints Parameters

JURECA HPC System – HPDBSCAN Job Script

```
#!/bin/bash
#SBATCH --job-name=HPDBSCAN
#SBATCH -o HPDBSCAN-%j.out
#SBATCH -e HPDBSCAN-%j.err
#SBATCH --nodes=2
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=4
#SBATCH --time=00:20:00
#SBATCH --cpus-per-task=4
#SBATCH --reservation=ml-hpc-1

export OMP_NUM_THREADS=4

# location executable
HPDBSCAN=/homea/hpclab/train001/tools/hpdbscan/dbscan

# your own copy of bremen small
BREMENSMALLDATA=/homea/hpclab/train001/bremenSmall.h5

# your own copy of bremen big
BREMENBIGDATA=/homea/hpclab/train001/bremen.h5

srun $HPDBSCAN -m 100 -e 300 -t 12 $BREMENSMALLDATA
```

- **Job submit using command: sbatch <jobscript>**
- **Remember your <jobid> that is returned from the sbatch command**
- **Show status of the job then with: squeue –u <your-user-id>**

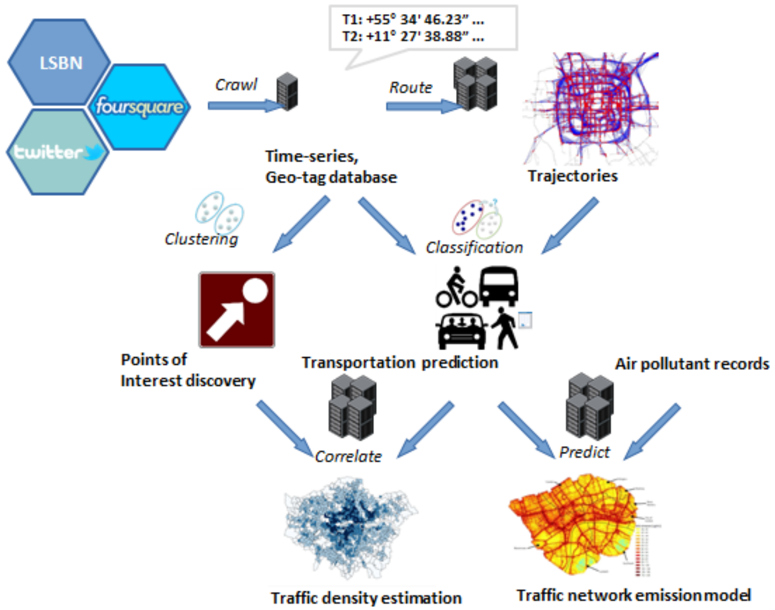
(parameters of DBSCAN and file to be clustered)

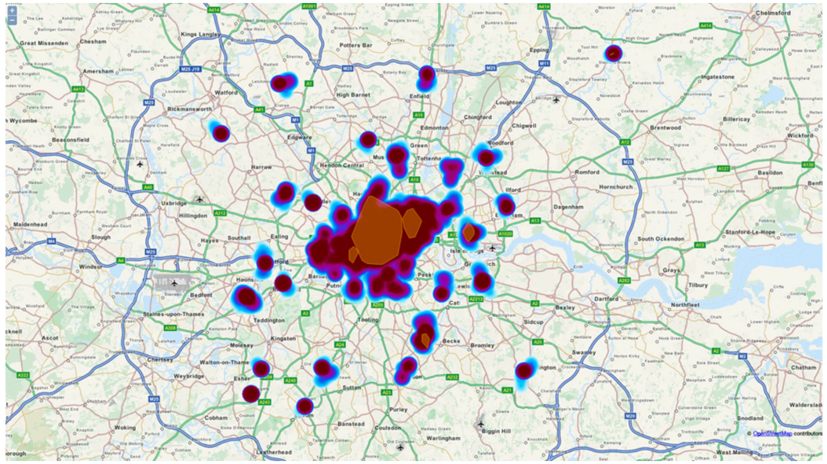
- **Note the tutorial reservation with –reservation = bigdata-cpu just valid for Thursday morning**

Exercises – Twitter Dataset

Twitter Dataset & Locations – Revisited

- Twitter streaming API data
  - Containts 1% of all geo-tagged of the UK in June 2014 (e.g. London)





- **The Twitter Dataset is encoded in the HDF5 format (binary)**
- **You need your own copy of the file in your home directory to cluster!**

```
[train001@jrl04 twitter]$ pwd
/homea/hpclab/train001/data/twitter
[train001@jrl04 twitter]$ ls -al
total 317312
drwxr-xr-x 2 train001 hpclab       512 Jan 14 23:00 .
drwxr-xr-x 8 train001 hpclab       512 Jan 14 22:06 ..
-rw-r--r-- 1 train001 hpclab 265636608 Jan 13  2017 twitter.h5
-rw-r--r-- 1 train001 hpclab  59272032 Jan 13  2017 twitterSmall.h5
```

[28] Twitter Dataset

B2SHARE
Store and Share Research Data

JURECA HPC System – HPDBSCAN Job Script

```
#!/bin/bash
#SBATCH --job-name=HPDBSCAN
#SBATCH -o HPDBSCAN-%j.out
#SBATCH -e HPDBSCAN-%j.err
#SBATCH --nodes=4
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=4
#SBATCH --time=01:00:00
#SBATCH --cpus-per-task=4
#SBATCH --reservation=ml-hpc-1

export OMP_NUM_THREADS=4

# location executable
HPDBSCAN=/homea/hpclab/train001/tools/hpdbscan/dbscan

# your own copy of bremen small
TWITTERSMALLDATA=/homea/hpclab/train001/twitterSmall.h5

# your own copy of bremen big
TWITTERBIGDATA=/homea/hpclab/train001/twitter.h5

srun $HPDBSCAN -m 40 -e 0.0001 -t 12 $TWITTERSMALLDATA
```

(parameters of DBSCAN and file to be clustered)

- **Job submit using command: sbatch <jobscript>**
- **Remember your <jobid> that is returned from the sbatch command**
- **Show status of the job then with: squeue –u <your-user-id>**

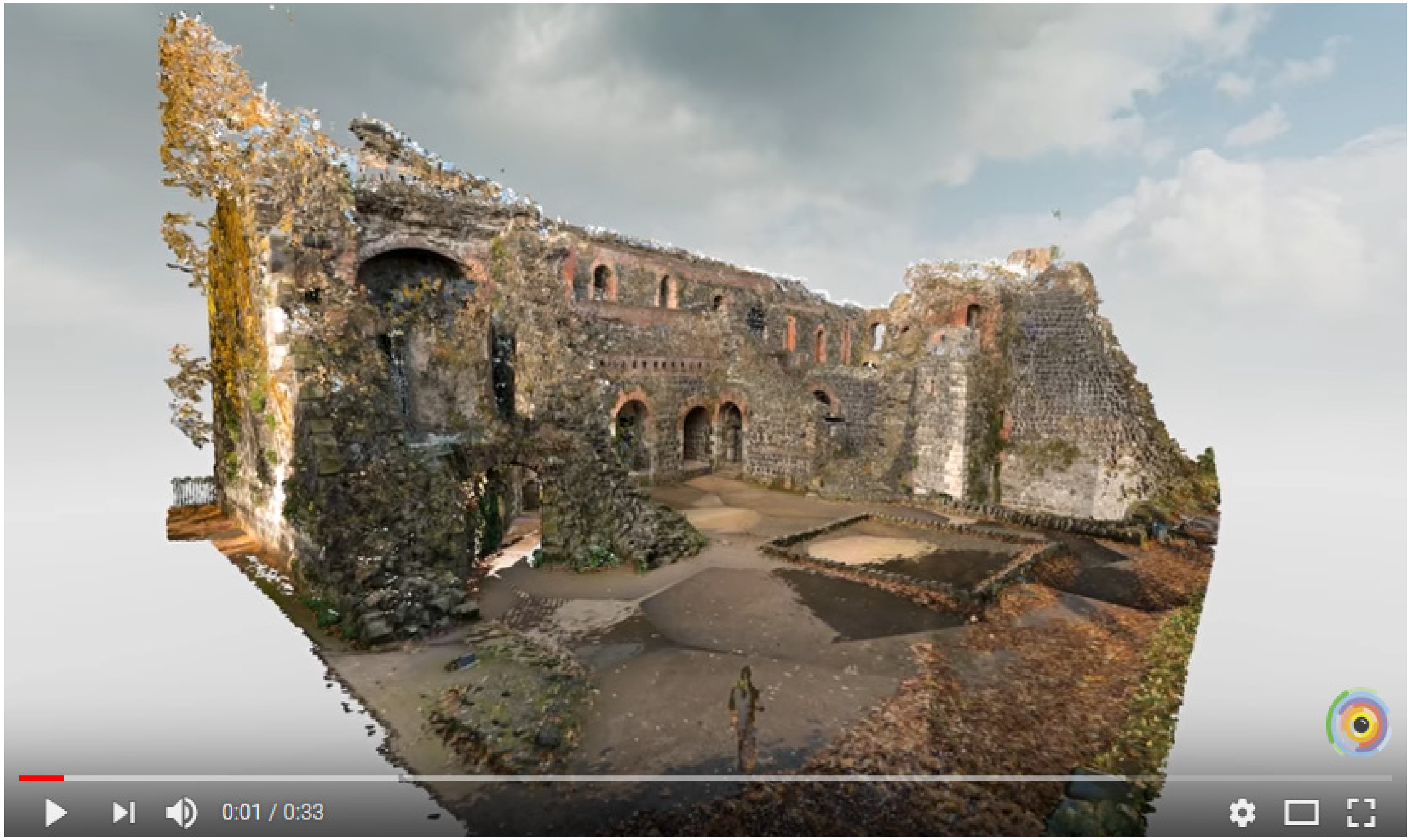
- **Note the tutorial reservation with –reservation= bigdata-cpu just valid for Thursday morning**

JURECA HPC System – HPDBSCAN Check Outcome

```
[train001@jrl04 hpdbscan]$ more HPDBSCAN-4632910.out
Calculating Cell Space...
        Computing Dimensions... [OK] in 0.002393
        Computing Cells...      [OK] in 0.498816
        Sorting Points...       [OK] in 0.891462
        Distributing Points...  [OK] in 2.576206
DBSCAN...
        Local Scan...           [OK] in 1.375779
        Merging Neighbors...    [OK] in 0.000586
        Adjust Labels ...       [OK] in 0.013686
        Rec. Init. Order ...    [OK] in 0.640681
        Writing File ...        [OK] in 0.006232
Result...
        8976    Clusters
        906807  Cluster Points
        2797544 Noise Points
        757369  Core Points
Took: 6.189666s
```

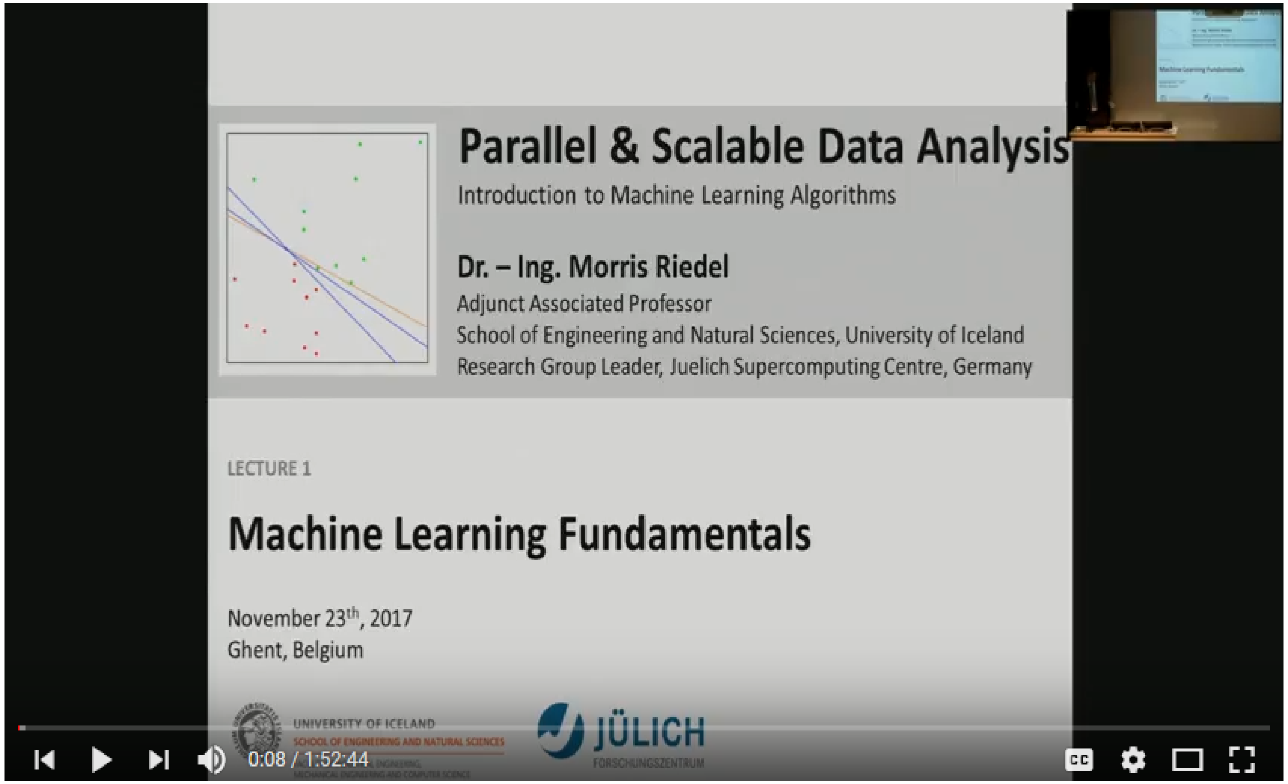
- The outcome of the clustering process
  is written directly into the HDF5 file using cluster IDs and noise IDs

[Video] Point Clouds



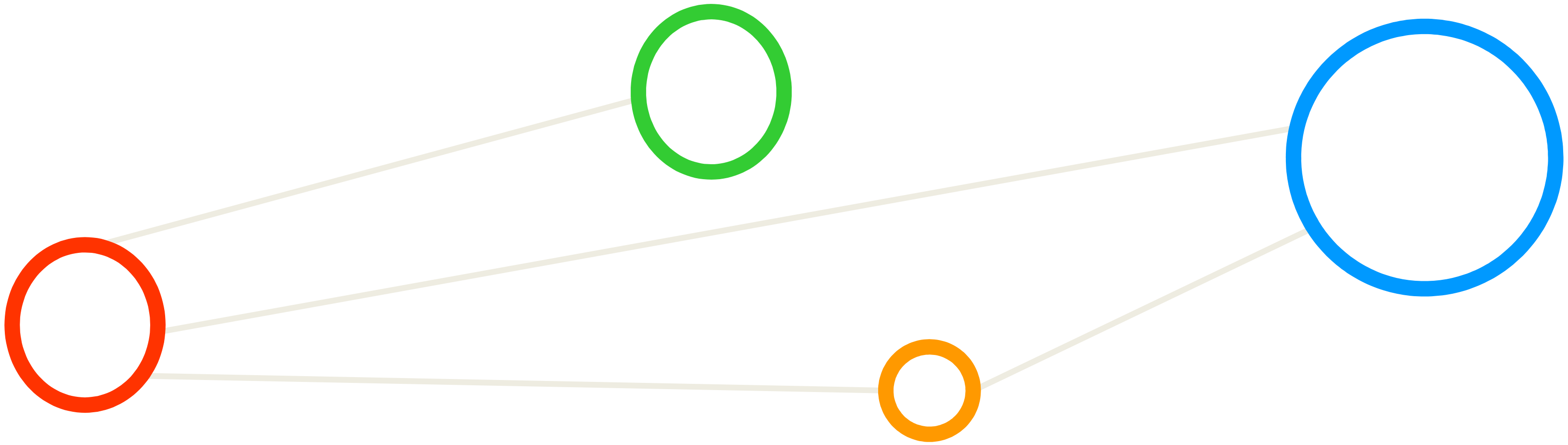
[18] Point Based Rendering of the Kaiserpfalz in Kaiserswerth, YouTube Video

[YouTube Lectures] More about parallel DBSCANs & HPC



[32] Morris Riedel, 'Introduction to Machine Learning Algorithms', Invited YouTube Lecture, six lectures, University of Ghent, 2017

Appendix



Working with Vectors

- E.g. creating a vector using the concatenate function `c()`
  - Assigning values can be done using `<-` and `=`
  - Be careful with overwriting (see below, x was overwritten)

```
> x <- c(1,3,2,5)
> x
[1] 1 3 2 5
> x = c(1,6,2)
> x
[1] 1 6 2
> 
```

- E.g. number of elements using `length()`

```
> length (x)
[1] 3
> 
```

Useful Working Commands: List and Remove Objects

- List all already defined objects (data and functions) with `ls()`

```
> y = c(1,4,3)
> z = c(0,0,7)
> ls()
[1] "ds_140518213752" "x"                 "y"                 "z"
> |
```

- Remove objects with `rm()`

```
> ls()
[1] "ds_140518213752" "x"                 "y"                 "z"
> rm (ds_140518213752)
> ls()
[1] "x" "y" "z"
> |
```

Working with Matrices

- E.g. creating a matrix using the function `matrix()`
  - Different versions exist, here we use the function with three parameters
  - It takes a number of inputs: matrix data, # roms, and # columns
  - You may specify parameter names: e.g. `nrow=2`
  - Default: filling columns, filling rows use parameter `byrow=TRUE`

```
> x = matrix(data=c(1,2,3,4), nrow=2, ncol=2)
> x
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> x = matrix(c(1,2,3,4),2,2)
> x
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> 
```

```
> x = matrix(c(1,2,3,4),2,2,byrow=TRUE)
> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> 
```

Working with sqrt and power

- E.g. applying each element of a matrix with sqrt()
  - Remember: you not changing x

```
> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> sqrt(x)
         [,1]     [,2]
[1,] 1.000000 1.414214
[2,] 1.732051 2.000000
> 
```

- E.g. applying each element of a matrix with power ^
  - Raises each element of x to the power 2
  - Remember: you not changing x

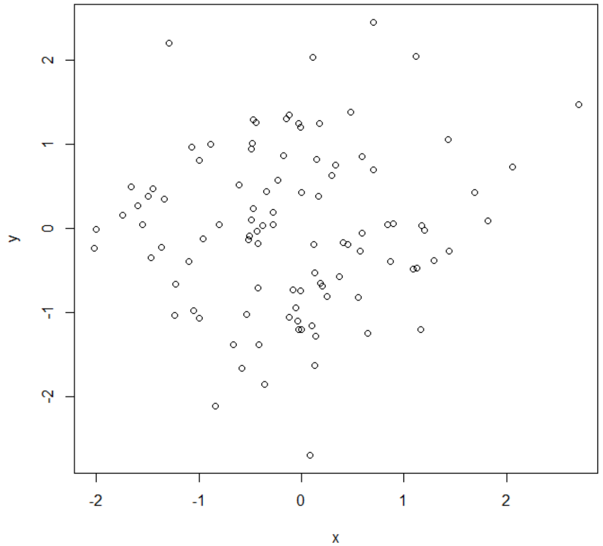
```
> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> x^2
     [,1] [,2]
[1,]    1    4
[2,]    9   16
```

Working with Random Variables

- E.g. creating 100 random normal variables with `rnorm()`

```
> x = rnorm(100)
> y = rnorm(100)
> plot(x,y)
>
```

- E.g. visualizing random variables in a simple diagram with `plot()`



Working with Datasets – Weather Patterns Example

- `library(rattle)`
  - Loads the Rattle package and the associated datasets into the memory
- `weatherAUS`
  - Loads in the weatherAUS dataset
- `names(weatherAUS)`
  - Shows the variables Names
- `nrow(weatherAUS)`
  - Displays the number of rows (observations on the longest variable)
- `ncol(weatherAUS)`
  - Displays the number of columns (variables)
- `head(weatherAUS)`
  - First six records of the dataset.
- `tail(weatherAUS)`
  - The last six rows of the dataset.
- `sample(weatherAUS)`
  - A snapshot of some of the data

Working with data: weatherAUS

- Load the already available Dataset
  - **`weatherAUS`**
  - Loads in the weatherAUS dataset
  - Dataset will be listed (lots of rows)

4151	13.9	No	1.2	Yes
4152	18.4	Yes	0.0	No
4153	22.1	No	0.0	No
4154	25.5	No	0.0	No
4155	26.4	No	0.0	No
4156	30.3	No	0.0	No
4157	30.6	No	0.0	No
4158	30.1	No	0.0	No
4159	31.1	No	0.0	No
4160	20.8	No	0.0	No
4161	22.6	No	0.0	No
4162	27.4	No	0.0	No
4163	27.4	No	0.0	No
4164	25.4	No	0.0	No
4165	19.8	No	0.0	No
4166	23.3	No	0.0	No

Weather Patterns – Get a Feel for a Dataset (1)

- Look at the names of the variables
  - `names(weatherAUS)`
  - Shows the variables Names

```
> names(weatherAUS)
 [1] "Date"          "Location"      "MinTemp"       "MaxTemp"
 [5] "Rainfall"      "Evaporation"   "Sunshine"      "WindGustDir"
 [9] "WindGustSpeed" "WindDir9am"    "WindDir3pm"    "WindSpeed9am"
[13] "WindSpeed3pm"  "Humidity9am"   "Humidity3pm"   "Pressure9am"
[17] "Pressure3pm"   "Cloud9am"      "Cloud3pm"      "Temp9am"
[21] "Temp3pm"       "RainToday"     "RISK_MM"       "RainTomorrow"
```

Weather Patterns – Get a Feel for the Dataset (2)

- Look at the number of variables and the number of observations
  - `nrow(weatherAUS)`
  - Displays the number of rows (observations on the longest variable)

```
> nrow(weatherAUS)
[1] 75136
```

  - `ncol(weatherAUS)`
  - Displays the number of columns (variables)

```
> ncol(weatherAUS)
[1] 24
```

Weather Patterns – Get Knowledge about the DataSet (1)

- Look at the Head
  - `head(weatherAUS)`
  - Displays first six records of the dataset

```
> head(weatherAUS)
        Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir WindGustSpeed
1 2008-12-01   Albury    13.4    22.9      0.6          NA       NA           W            44
2 2008-12-02   Albury     7.4    25.1      0.0          NA       NA         WNW            44
3 2008-12-03   Albury    12.9    25.7      0.0          NA       NA         WSW            46
4 2008-12-04   Albury     9.2    28.0      0.0          NA       NA          NE            24
5 2008-12-05   Albury    17.5    32.3      1.0          NA       NA           W            41
6 2008-12-06   Albury    14.6    29.7      0.2          NA       NA         WNW            56
  WindDir9am WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am Humidity3pm Pressure9am Pressure3pm
1          W        WNW           20           24          71          22      1007.7      1007.1
2        NNW        WSW            4           22          44          25      1010.6      1007.8
3          W        WSW           19           26          38          30      1007.6      1008.7
4         SE          E           11            9          45          16      1017.6      1012.8
5        ENE         NW            7           20          82          33      1010.8      1006.0
6          W          W           19           24          55          23      1009.2      1005.4
  Cloud9am Cloud3pm Temp9am Temp3pm RainToday RISK_MM RainTomorrow
1        8       NA    16.9    21.8        No     0.0           No
2       NA       NA    17.2    24.3        No     0.0           No
3       NA        2    21.0    23.2        No     0.0           No
4       NA       NA    18.1    26.5        No     1.0           No
5        7        8    17.8    29.7        No     0.2           No
6       NA       NA    20.6    28.9        No     0.0           No
```

Weather Patterns – Get Knowledge about the DataSet (2)

- Look at the Tail
  - `tail(weatherAUS)`
  - Displays last six records of the dataset

```
> tail(weatherAUS)
            Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir WindGustSpeed
75131 2013-07-25   Darwin    17.0    30.3        0         9.0     11.2         ESE            50
75132 2013-07-26   Darwin    17.3    30.6        0        10.2     11.2         ESE            35
75133 2013-07-27   Darwin    17.6    31.0        0         7.4     10.4          NW            31
75134 2013-07-28   Darwin    18.9    30.5        0         5.8      7.8         NNW            24
75135 2013-07-29   Darwin    17.6    32.3        0         4.4     10.1         ESE            37
75136 2013-07-30   Darwin    18.5    33.0        0         4.4     10.9         ENE            50
      WindDir9am WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am Humidity3pm Pressure9am
75131         SE        SSE           24           22          24          15      1016.7
75132         SE        WNW           19           13          30          22      1015.9
75133        ESE         NW           11           19          41          45      1015.5
75134        SSE         NW           11           17          52          42      1014.6
75135         NE        NNW           13           20          68          44      1014.7
75136        ESE        ENE           11           26          59          19      1013.1
      Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday RISK_MM RainTomorrow
75131      1012.6        1        1    20.8    29.2        No       0           No
75132      1012.2        0        1    20.1    29.5        No       0           No
75133      1011.8        4        1    23.3    28.7        No       0           No
75134      1011.2        7        7    23.1    29.4        No       0           No
75135      1010.2        7        7    24.4    29.6        No       0           No
75136      1009.1        6        7    24.1    32.0        No       0           No
```

Weather Patterns – Get Knowledge about the DataSet (2)

3. Look at the Sample
   - `sample(weatherAUS)`
   - A snapshot of some of the data

4145	22	1	No	No	1025.6	3	41
4146	17	5	No	No	1019.4	2	65
4147	13	1	No	No	1020.6	2	30
4148	20	8	No	Yes	1019.1	7	35
4149	24	7	Yes	No	1016.9	4	56
4150	22	0	No	No	1021.8	1	46
4151	17	7	No	Yes	1020.2	2	54
4152	9	0	Yes	No	1026.0	0	24
4153	6	1	No	No	1028.7	0	24
4154	13	1	No	No	1026.2	0	39
4155	6	1	No	No	1024.5	NA	67
4156	17	NA	No	No	1020.9	NA	43
4157	9	NA	No	No	1017.9	NA	48
4158	7	7	No	No	1018.0	6	41
4159	17	0	No	No	1015.4	0	67
4160	28	1	No	No	1014.8	4	52
4161	7	0	No	No	1023.6	0	24
4162	13	0	No	No	1022.6	0	39
4163	11	NA	No	No	1019.7	NA	33
4164	9	NA	No	No	1019.2	NA	37
4165	15	NA	No	No	1019.1	NA	41
4166	24	1	No	No	1020.9	6	48

Read tab data file (1)

- Read tab separated data from a real science project
  - Often different than UCI machine learning repository datasets
  - Example: measurement data with descriptive information first

```
> data <- read.table("data.tab", sep="\t")
Fehler in scan(file, what, nmax, sep, dec, quote, skip, nlines, na.strings,  :
  Zeile 24 hatte keine 2 Elemente
> data <- read.table("data.tab", sep="\t")
```

- Addressing the error:
  - Check if data.tab file may have descriptive information/comments
  - Descriptive information is sometimes put in front of the real data set (e.g. metadata = explaining where data was measured, by whom, etc.)
  - Removing metadata works if file is properly tab separated
  - What other surprised we encounter when we load the data?

Read tab data file (2)

- Look at the names of the variables/attributes

```
> names(data)
 [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12" "V13" "V14" "V15" "V16"
```

- Look at one variable/attribute value only

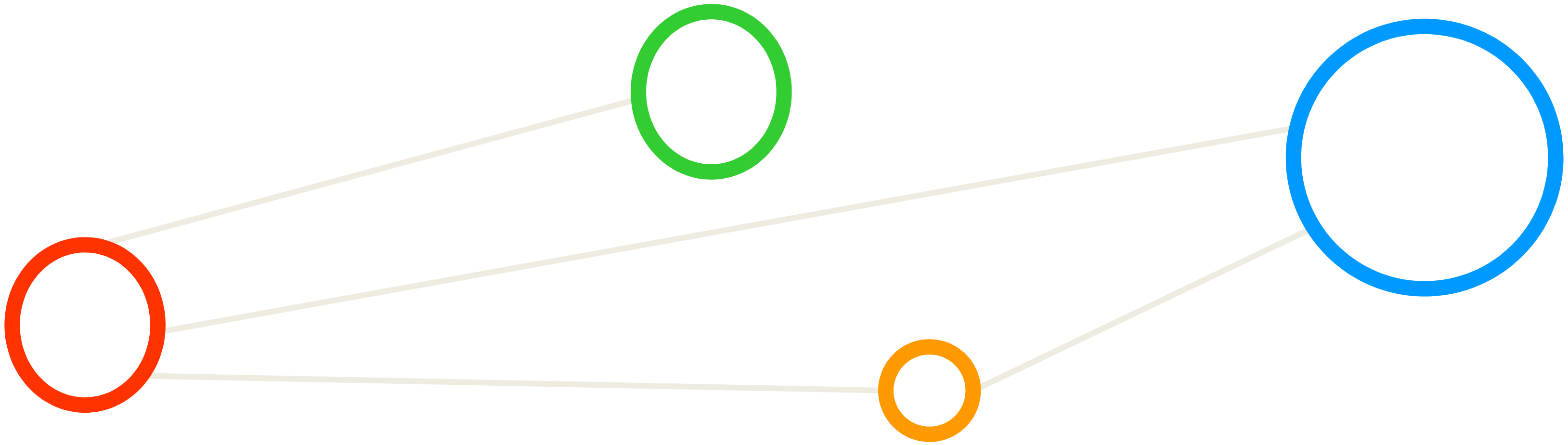
```
> data$V8
```

```
/* DATA DESCRIPTION:
Citation:  Hall, Per (2012): Koljoefjord cabled observatory
RDCP data, Sweden (2012-03). Department of Chemistry,
University of Gothenburg, Unpublished dataset #779644
Project(s):     In situ monitoring of oxygen depletion in
hypoxic ecosystems of coastal and open seas and land-locked
water bodies (HYPOX) (URI: http://www.hypox.net/)
Coverage:  LATITUDE: 58.228250 * LONGITUDE: 11.574000
    DATE/TIME START: 2012-03-01T00:13:16 * DATE/TIME END:
2012-03-31T23:43:17
    MINIMUM DEPTH, water: 5.0 m * MAXIMUM DEPTH, water: 35.0
...

Size: 168428 data points
*/
Date/Time  Pitch [deg]    Roll [deg] Head [deg] Temp [°C]
    Cond [mS/cm]    Press [dbar]    O2 [µmol/l]     Sal
    Vbat [V]   Depth water [m] CV hor [cm/s]   Direction
[deg] CV vert [cm/s]  Sig str [dB]    Std dev [±]
2012-03-01T00:13:16
           5.000 9.311 339.974    1.791 -41.973    5.921
2012-03-01T00:13:16
           6.000 9.090 345.673    1.976 -42.196    6.497
```

Display Head of the data

```
> head(data, n=100)
                    V1          V2          V3          V4        V5            V6            V7          V8     V9       V10              V11            V12              V13             V14            V15
1            Date/Time Pitch [deg] Roll [deg] Head [deg] Temp [°C] Cond [mS/cm] Press [dbar] O2 [µmol/l]    Sal Vbat [V] Depth water [m] CV hor [cm/s] Direction [deg] CV vert [cm/s] Sig str [dB]
2  2012-03-01T00:13:16                                                                                                           5.000          9.311          339.974           1.791        -41.973
3  2012-03-01T00:13:16                                                                                                           6.000          9.090          345.673           1.976        -42.196
4  2012-03-01T00:13:16                                                                                                           7.000          8.618          332.505           1.557        -41.963
5  2012-03-01T00:13:16                                                                                                           8.000          8.628          345.900           2.235        -41.499
6  2012-03-01T00:13:16                                                                                                           9.000         12.633          351.931           1.902        -41.294
7  2012-03-01T00:13:16                                                                                                          10.000         13.068          341.203           2.096        -41.215
8  2012-03-01T00:13:16                                                                                                          11.000          9.369          345.225           1.350        -40.645
9  2012-03-01T00:13:16                                                                                                          12.000          9.007          338.305           1.412        -40.513
10 2012-03-01T00:13:16                                                                                                          13.000          9.387          335.719           1.538        -41.481
11 2012-03-01T00:13:16                                                                                                          14.000          9.629          343.446           1.899        -38.749
12 2012-03-01T00:13:16                                                                                                          15.000          9.431          356.062           1.635        -33.261
13 2012-03-01T00:13:16                                                                                                          16.000          6.344          359.442           0.454        -31.690
14 2012-03-01T00:13:16                                                                                                          17.000          4.576            2.839           0.896        -33.841
15 2012-03-01T00:13:16                                                                                                          18.000          4.899          354.866           1.323        -40.046
16 2012-03-01T00:13:16                                                                                                          19.000          5.208          343.540           0.856        -40.025
17 2012-03-01T00:13:16                                                                                                          20.000          6.063          323.617           0.491        -37.558
18 2012-03-01T00:13:16                                                                                                          21.000          5.740          335.278           0.302        -35.335
19 2012-03-01T00:13:16                                                                                                          22.000          4.397          311.251          -0.151        -34.859
20 2012-03-01T00:13:16                                                                                                          23.000          4.413          307.293          -0.103        -35.103
21 2012-03-01T00:13:16                                                                                                          24.000          5.033          318.156          -0.265        -34.160
22 2012-03-01T00:13:16                                                                                                          25.000          5.198          346.344          -0.341        -32.763
23 2012-03-01T00:13:16                                                                                                          26.000          4.979          327.433           0.355        -31.670
24 2012-03-01T00:13:16                                                                                                          27.000          4.880          330.004           0.615        -30.567
25 2012-03-01T00:13:16                                                                                                          28.000          3.614          343.413           0.327        -29.375
26 2012-03-01T00:13:16                                                                                                          29.000          3.030          351.250           0.263        -27.823
27 2012-03-01T00:13:16                                                                                                          30.000          2.733          353.598           0.138        -25.901
28 2012-03-01T00:13:16                                                                                                          31.000          3.025          343.426          -0.056        -23.790
29 2012-03-01T00:13:16                                                                                                          32.000          1.211          297.778           0.407        -21.957
30 2012-03-01T00:13:16                                                                                                          33.000          1.456          277.830           0.583        -20.461
31 2012-03-01T00:13:16                                                                                                          34.000          3.091          314.535           0.839        -17.164
32 2012-03-01T00:13:16                                                                                                          35.000          5.694          285.980           0.329        -10.920
33 2012-03-01T00:13:16       1.243       0.076     319.693     5.038         1.465        27.612      28.264 40.218   10.000
34 2012-03-01T00:43:16                                                                                                           5.000          9.617          346.366           1.405        -40.898
35 2012-03-01T00:43:16                                                                                                           6.000          8.341          331.550           1.667        -40.545
36 2012-03-01T00:43:16                                                                                                           7.000         10.992          326.578           1.433        -41.197
37 2012-03-01T00:43:16                                                                                                           8.000         12.012          342.534           1.913        -42.057
38 2012-03-01T00:43:16                                                                                                           9.000         11.059          338.113           0.930        -41.991
39 2012-03-01T00:43:16                                                                                                          10.000          8.675          326.847           0.425        -40.630
40 2012-03-01T00:43:16                                                                                                          11.000          8.518          337.461           0.897        -39.605
41 2012-03-01T00:43:16                                                                                                          12.000          3.779          346.858           0.909        -33.934
42 2012-03-01T00:43:16                                                                                                          13.000          3.844          328.829           1.325        -28.494
43 2012-03-01T00:43:16                                                                                                          14.000          5.673          329.306           1.086        -26.221
44 2012-03-01T00:43:16                                                                                                          15.000          7.886          339.201           0.484        -26.391
45 2012-03-01T00:43:16                                                                                                          16.000          7.675          339.114           0.154        -28.577
46 2012-03-01T00:43:16                                                                                                          17.000          8.393          343.118           1.371        -23.468
47 2012-03-01T00:43:16                                                                                                          18.000          6.809          347.856           1.752        -19.350
```

Lecture Bibliography



Lecture Bibliography (1)

- [1] LLView Tool,
  Online: http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/LLview/_node.html
- [2] An Introduction to Statistical Learning with Applications in R,
  Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [3] PANGAEA Data Collection, Data Publisher for Earth & Environmental Science,
  Online: http://www.pangaea.de/
- [4] Judy Qiu,'Harp: Collective Communication on Hadoop', 2014
- [5] Animation of the k-means algorithm using Matlab 2013, YouTube Video,
  Online: http://www.youtube.com/watch?v=5FmnJVv73fU
- [6] Statistical Computing with R Tool,
  Online: http://www.r-project.org/
- [7] Rattle brochure,
  Online: http://rattle.togaware.com/RattleBrochure.pdf
- [8] Ester, Martin, et al. "A density-based algorithm for discoveringclusters in large spatial databases with noise." Kdd. Vol. 96. 1996.
- [9] YouTube Video, 'CSCE 420 Communication Project – DBSCAN',
  Online: https://www.youtube.com/watch?v=5E097ZLE9Sg
- [10] YouTube Video ,'Point Based Rendering of the Aachen Cathedral'
  Online: https://www.youtube.com/watch?v=X_wyoroo4co
- [11] B2SHARE, 'HPDBSCAN Benchmark test files',
  Online: http://hdl.handle.net/11304/6eacaa76-c275-11e4-ac7e-860aa0063d1f

Lecture Bibliography (2)

- [12] Michael Stephan,'Portable Parallel IO - 'Handling large datasets in heterogeneous parallel environments',
  Online: http://www.fz-juelich.de/SharedDocs/Downloads/IAS/JSC/EN/slides/parallelio-2014/parallel-io-hdf5.pdf?__blob=publicationFile
- [13] HDF Group,
  Online: http://www.hdfgroup.org/
- [14] Parallel NETCDF,
  Online: http://trac.mcs.anl.gov/projects/parallel-netcdf
- [15] E. Hartnett, 2010-09: NetCDF and HDF5 - HDF5 Workshop 2010
- [16] M. Goetz, M. Riedel et al.,' On Parallel and Scalable Classification and Clustering Techniques for Earth Science Datasets' 6th Workshop on Data Mining in Earth System Science, Proceedings of the International Conference of Computational Science (ICCS), Reykjavik,
  Online: http://www.proceedings.com/26605.html
- [17] M.Goetz, M. Riedel et al.,'HPDBSCAN – Highly Parallel DBSCAN', Proceedings of MLHPC Workshop at Supercomputing 2015, Online: http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=46948
- [18] YouTube Video, 'Point Based Rendering of the Kaiserpfalz in Kaiserswerth',
  Online: https://www.youtube.com/watch?v=KvDb58YvlvQ
- [19] Partnership for Advanced Computing in Europe (PRACE),
  Online: http://www.prace-ri.eu/
- [20] DEEP-EST EU Project,
  Online: http://www.deep-projects.eu/
- [21] PRACE – Introduction to Supercomputing,
  Online: https://www.youtube.com/watch?v=D94FJx9vxFA

Lecture Bibliography (3)

- [22] Introduction to High Performance Computing for Scientists and Engineers, Georg Hager & Gerhard Wellein, Chapman & Hall/CRC Computational Science, ISBN 143981192X, English, ~330 pages, 2010, Online: http://www.amazon.de/Introduction-Performance-Computing-Scientists-Computational/dp/143981192X
- [23] TOP500 Supercomputing Sites,
  Online: http://www.top500.org/
- [24] JURECA HPC System @ JSC,
  Online: http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JURECA/JURECA_node.html
- [25] Putty Tool,
  Online: http://www.putty.org/
- [26] LLView Tool,
  Online: http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/LLview/_node.html
- [27] YouTube Video, 'Big Ideas: Simplifying High Performance Computing Architectures',
  Online: https://www.youtube.com/watch?v=ISS_OGVamBk
- [28] B2SHARE, 'HPDBSCAN Benchmark test files', contains Twitter dataset
  Online: http://hdl.handle.net/11304/6eacaa76-c275-11e4-ac7e-860aa0063d1f
- [29] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison Wesley, ISBN 0321321367, English, ~769 pages, 2005
- [30] PANGAEA Data Collection, Data Publisher for Earth & Environmental Science,
  Online: http://www.pangaea.de/
- [31] UCI Machine Learning Repository,
  Online: http://archive.ics.uci.edu/ml/datasets.html

Lecture Bibliography (4)

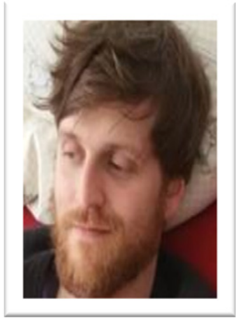
- [32] Morris Riedel, ‘Introduction to Machine Learning Algorithms‘, Invited YouTube Lecture, six lectures University of Ghent, 2017
  Online: https://www.youtube.com/watch?v=KgiuUZ3WeP8&list=PLrmNhuZo9sgbcWtMGN0i6G9HEvh08JG0J

Acknowledgements – Membery of my HPDP Research Group

PD Dr.
G. Cavallaro

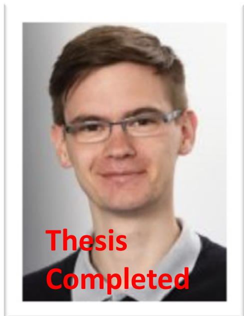
Senior PhD
Student A.S. Memon

Senior PhD
Student M.S. Memon



PhD Student
E. Erlingsson

PhD Student
C. Bakarat

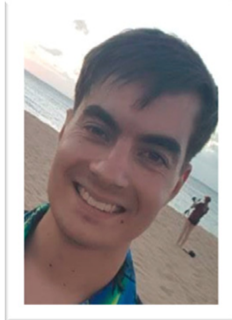


Dr. M. Goetz
(now KIT)

MSc M.
Richerzhagen,
now other group

MSc
P. Glock
(now INM-1)

MSc
C. Bodenstein
(now Soccerwatch.tv)



MSc Student
G.S. Guðmundsson
(Landsverkjun)

Slides Available at http://www.morrisriedel.de/talks