Impact of algorithmic data analysis

Heikki Mannila
Helsinki Institute for Information Technology (HIIT)
University of Helsinki and
Helsinki University of Technology
Heikki.Mannila@cs.helsinki.fi
October 2008

Contents

- Algorithmic data analysis – what is it?
- Examples
- Impact – on what?
- Examples, again
- Conclusions

Basic message

- There is great demand for algorithmic data analysis

- A lot of the impact comes from figuring what needs to be computed (with lots of interaction with the application people)
- … and then designing a nice and clean algorithm for it

Why data analysis?

- Our ability to measure things and to collect data has increased a lot
- Lots of data
- Heterogeneous data: lots of different tables, time series, trees, etc
- Lack of good methods
- Data management and data analysis

What is algorithmic data analysis?

- Summarizing or modeling [large or complex data sets]
- Algorithmic as opposed to just traditional statistical approaches (least squares methods etc.)
- Also different from classical scientific and statistical computing: PDEs

What is algorithmic data analysis?

- Not just any algorithms research
- More or less direct connection to data analysis application, and at least some real observational data

Examples of algorithmic data analysis

- Fast Fourier transform
- Dynamic programming for sequence segmentation
- Heavy hitters in data streams
- Biclustering in microarray data

Boundaries are not strict

- There is no clear boundary between algorithmic and other types of data analysis

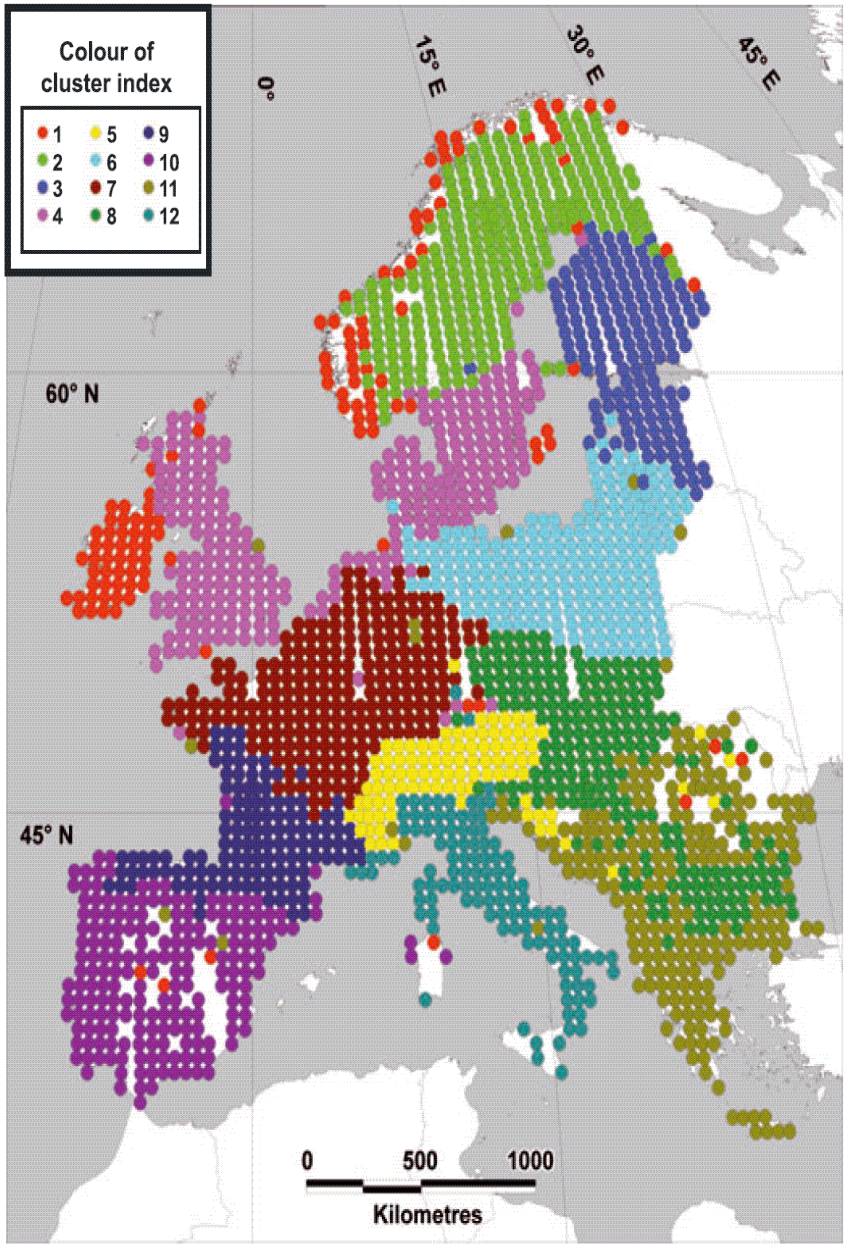
- E.g., variable selection in multivariate regression
  - A clearly algorithmic problem

Why algorithmic data analysis?

- A lot of the data is in nontraditional forms
  - E.g., trees, graphs, strings, not just matrices
- Traditional approaches are not sufficient
- No large body of analysis methods exists

Why algorithmic data analysis?

- Computational science on the rise
  - Not scientific computation, but something more general
- Bioinformatics is a nice example, but there are lots of other application areas out there
  - Environment, energy, …
- Algorithmic data analysis is also lots of fun: theory and practice, quick cycle time, good applications



imple result

H. Heikinheimo, M. Fortelius, J. Eronen and H. Mannila, Biogeography of European land mammals shows environmentally distinct and spatially coherent clusters. Journal of Biogeography (J. Biogeogr.) (2007) 34, 1053–1064

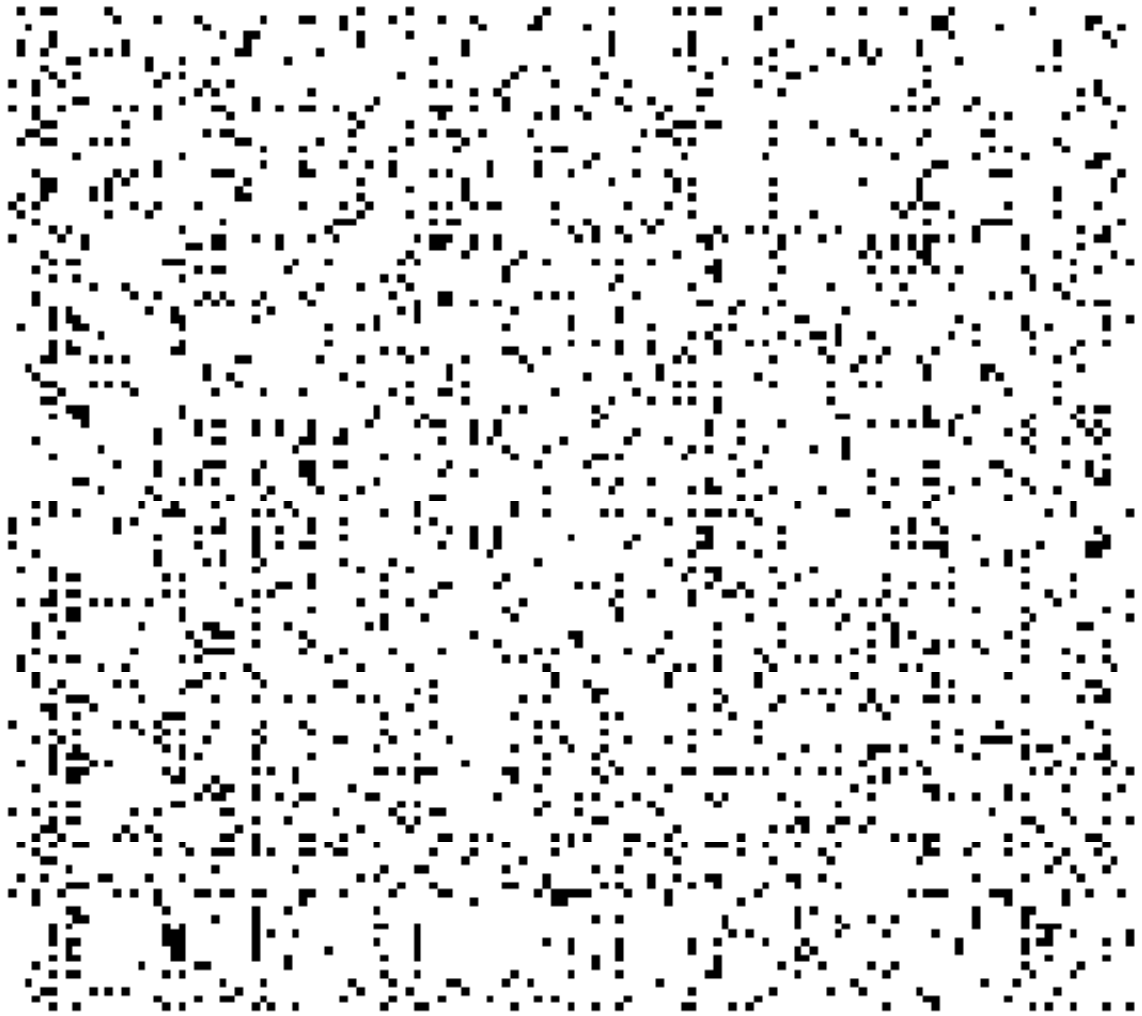
Examples of data analysis tasks

- Small and clean data, hard problem
  - Seriation in paleontological data
- Large but clean data, hard problem
  - Heavy hitters in data streams
- Large and relatively clean data, hard problem
  - Sequence assembly
- Large and noisy and heterogeneous data, hard problem
  - Gene regulation

Seriation in paleontological data

- Site-species matrix
- Rows correspond to fossil sites
- Columns correspond to species
- **Seriation** problem:
  - Find a good ordering for the sites
  - Approximating temporal order

Site-species –matrix: European mammal genera in the last 25 Ma



What is the criterion for a good order?

- Genera (or species) first are not there, then they appear, and then they go extinct
- A good ordering in time has the form

0
0
1
1
1
0
0

time

A simple computational problem

- Given a 0-1 matrix, find an ordering of the rows such that there as as few Lazarus events as possible
- I.e., there are as few 0s between 1s in the columns
- I.e., the matrix is as close to being a consecutive ones matrix as possible

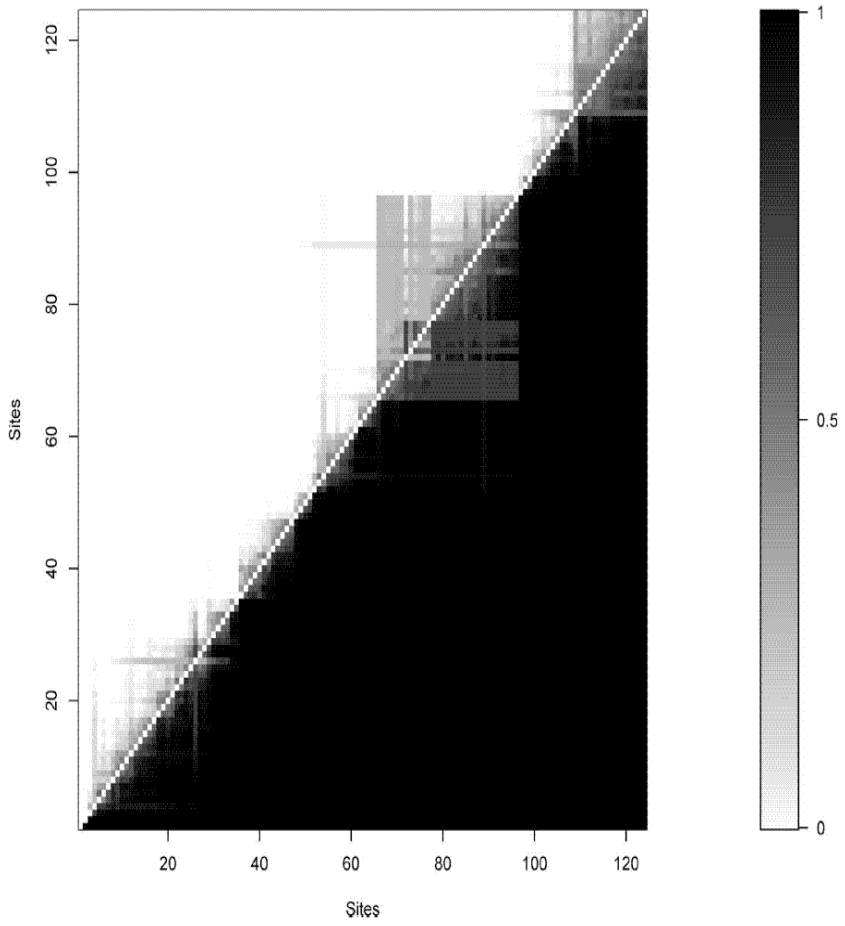
- Small and clean data

Properties

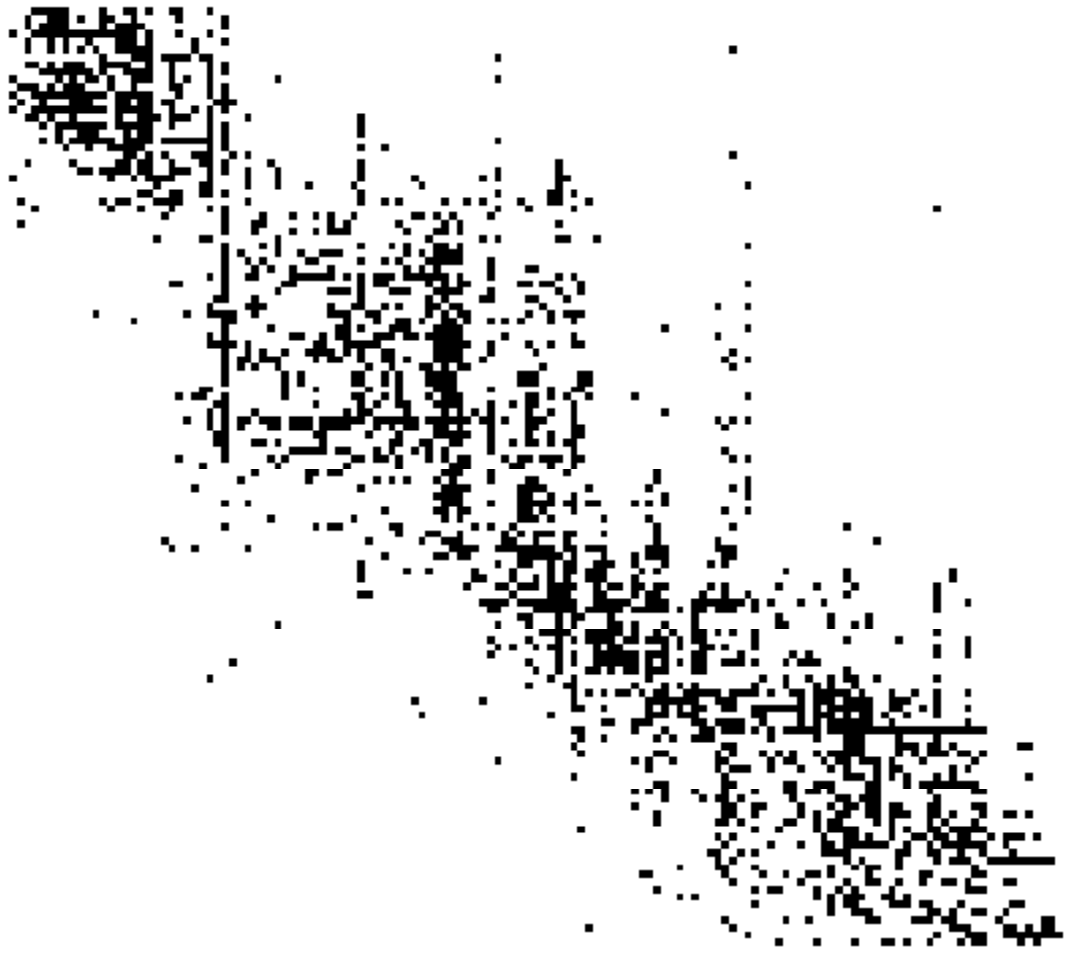
- Can be addressed by using
  - Eigenvalue methods (spectral techniques)
  - Probabilistic models and MCMC
  - Bucket orders: combinatorial algorithms

Site-genus -matrix

After probabilistic modeling



After spectral ordering

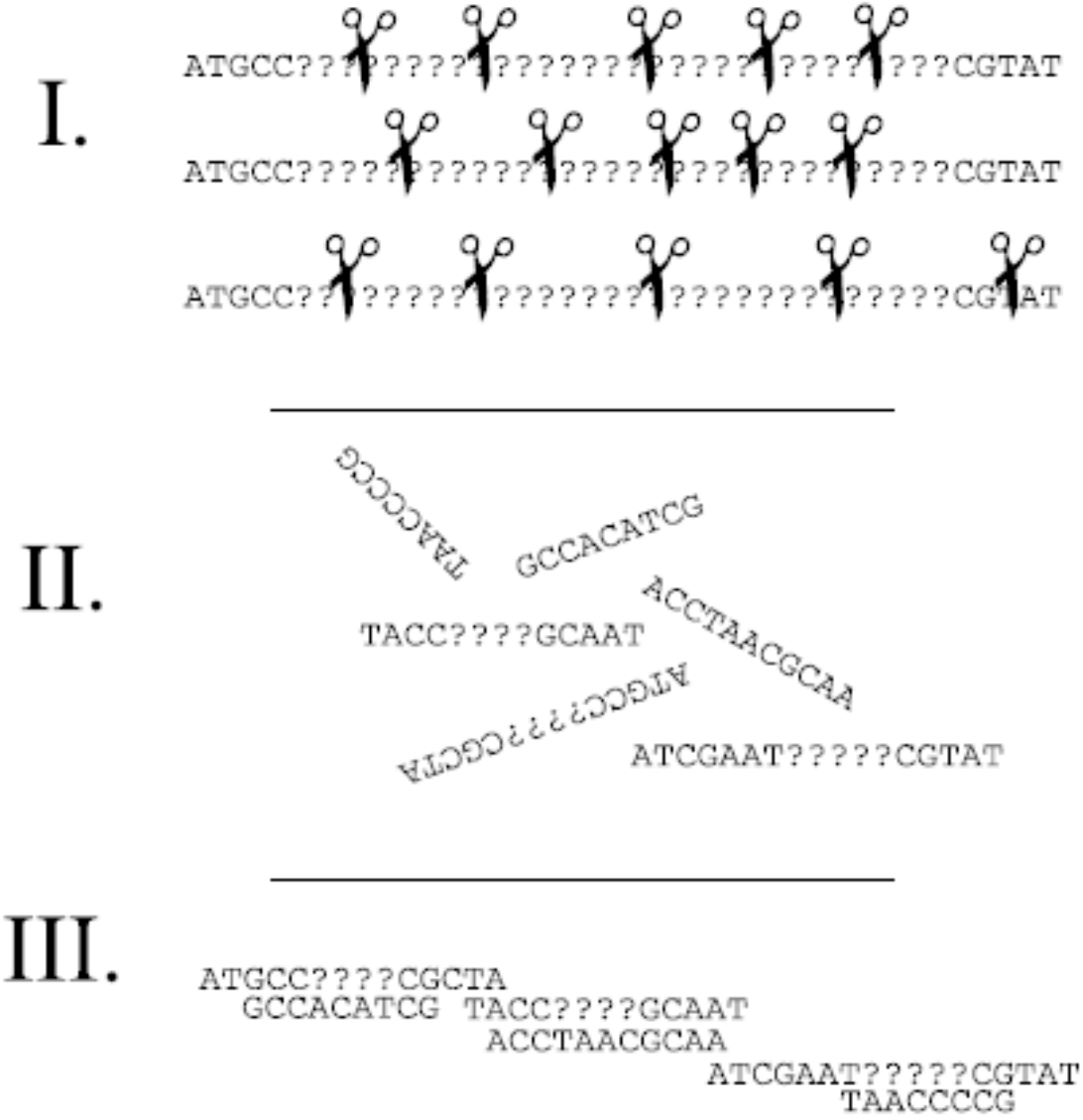


Finding heavy hitters in streams

- Nice clean problem (at least in the papers)
- Sequence of pairs *(u,c)*, where *u* is an identifier from a huge space and *c* is an integer
- Same *u* can occur many times
- Find the identifiers *u* such that the sum of the associated *c*'s is large
- Count-min data structure works
- Large and clean data

Sequence assembly

- Genome sequencing methods produce small parts of the sequence
- These have to be stiched together to a long sequence
- Can be formulated as a longest TSP problem
- Simple algorithms work quite well
- Large and sort of messy data

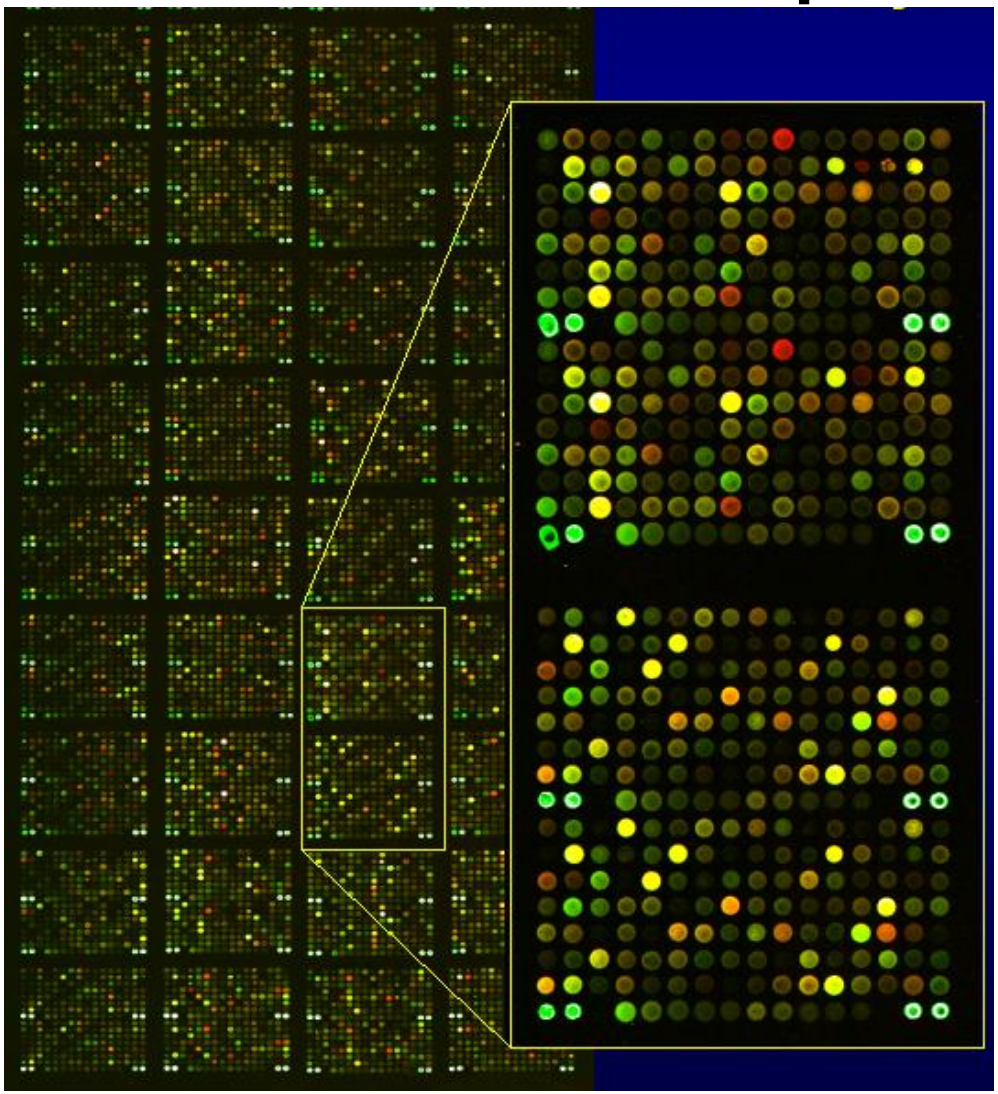


Huge volumes of sequence data now (2008→)

Gene regulation

- Understanding the mechanisms that cause genes to work or not
- Data: genome, gene expression data, lots of other types of data
- Genome sequence → motifs that might have something to do with the regulation
- Motifs → modules (combinations of motifs)
- Gene expression: which genes are actually working under certain conditions
- Large and messy data

Gene expression



Noisy,
noisy,
noisy data

How to make a controlled experiment?

- Difficult in higher organisms
- Possible in, say, yeast
- Knock out or silence each gene in turn, and see what changes
- Try to obtain some structure from this
- Very difficult problem

Where is the difficulty?

- In the data?
- In missing algorithms?
- In not knowing what to compute?

Impact on what?

- Applications in science
- Applications in industry
- Computer science itself
- (Education)

Impact on applications

- Some things have clearly had enormous effect on a wide range of applications
  - FFT
- Or a single application
  - PageRank
- Or as a general tool
  - Data compression algorithms

Example: sequence assembly

- The measurement technology produced some type of data (fragments)
- The algorithms were needed, and they were developed
- Original data was of good quality → useful results

Example: biclustering in gene expression data

- Find a group G of genes and a set C of conditions such that all genes of G behave in the same way in the conditions C
- Finding a clique in a bipartite graph
- Hard problem
- Many heuristic algorithms
- Impact so far?
- Bad algorithms or noisy data?

Impact on applications

- Not very useful to improve something which is already good enough
- This is often very hard to judge
- Rule of thumb: if the application people know what they want to compute, they are already doing fine

Example from data mining

- Association rules: initial concept by Agrawal, Imielinski, Swami 1993
- Second algorithm by Agrawal et al al. 1994
- Since then 300+ papers on the algorithms
- Few papers on applications
- Even the second algorithm was good enough

Example (cont.)

- The original concept was very important
- The follow-up algorithmic work not so

What determines impact?

- Deep algorithms?
- Good applications?
- Simple concepts?

Impact?

- Recipe for applicability:
  - Finding an important application or a set of applications where improvement is needed
  - And figuring out good concepts: what needs to be computed
  - Simplicity is good

Steps in algorithmic data analysis

- Deep and continuous interaction with the application experts
- Formulating computational concepts
- Analyzing the properties of the concepts
- Designing algorithms and analyzing their performance
- Implementing and experimenting with the algorithms
- Applying the results in practice

What is the role of theoretical analysis?

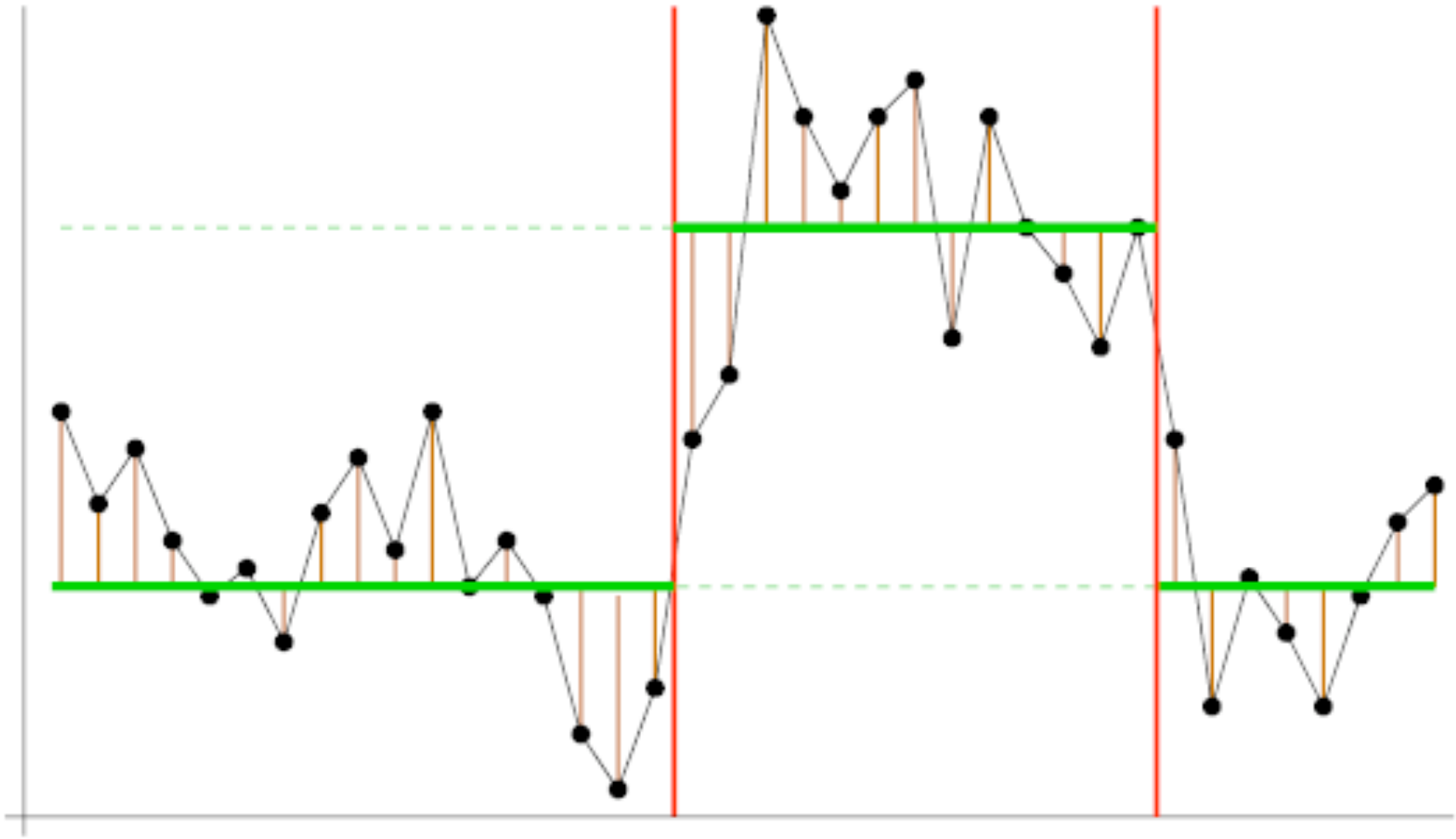
- Computer science background: like to be able to prove properties of the method
- Applications people sometimes ask: why?

Example: recurrent segmentation

- What are the criteria for a good algorithm?
- Good concept, simple algorithm, provable properties

Aesthetics and impact

- Example: (k,h)-segmentation of sequences: recurrent segments
- Give a piecewise constant representation of the data using k segments but only h different constants
- Minimizing the error of the representation



Results

- The problem NP-hard
- A simple but nonobvious algorithm has approximation ratio 3; nontrivial proof
- Experimentally the approximation ratios are about 1.01
- So who cares about the approximation ratio?
- I do, but on what grounds?

Results

- What does the approximation bound tell us?
- We won’t ever be awfully far away from the optimum
- Shows that the basic idea has wide validity

Another example

- ”k-means++: the advantages of careful seeding” (D. Arthur, S. Vassilvitskii)
- A beautiful paper showing that a simple change in k-means algorithm gives an approximation bound
- … and does not increase the running time
- … and improves the results in practice

Impact of algoritmic data analysis on computer science

- Computer science and algorithms research have changed
  - Look at what the algorithms in STOC or FOCS are about
  - Data analysis is a first-class subject
  - Internal applications within computer science are of less interest than some time ago

Why?

- Outward-looking trend, at least to some extent
- Good applications are there
- Increasing cooperation

- Large values of *n* (!)

Where is algorithmic data analysis going?

- Increasing need for it
- Increasing participation from computer scientists
- Good concepts, simplicity

- Looking at the whole chain of analysis is becoming more common

- (Data management issues)

Impact on education

- An excellent theme for research-oriented education
- Theory, practice, collaborations
- Relevant problems
- Easy to get into

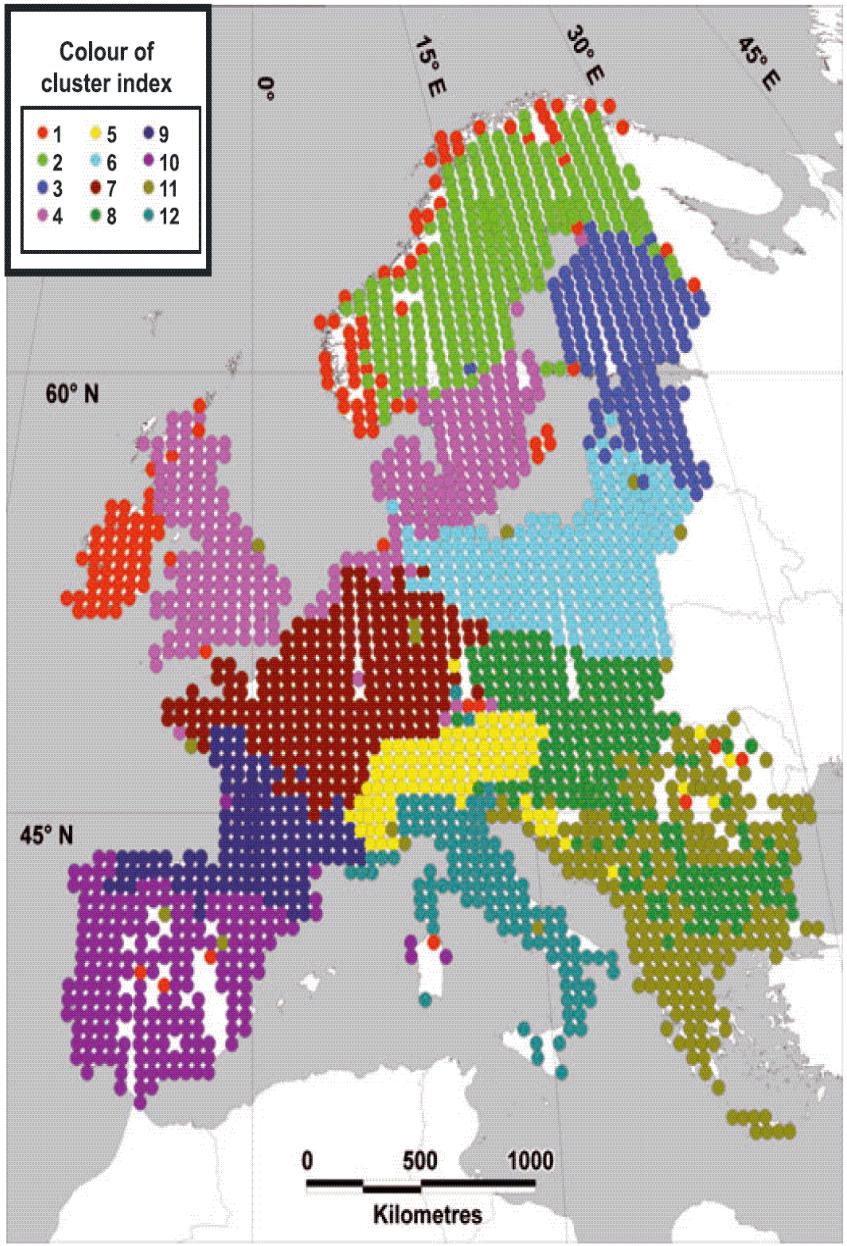
Steps in algorithmic data analysis

- Deep and continuous interaction with the application experts
- Formulating computational concepts
- Analyzing the properties of the concepts
- Designing algorithms and analyzing their performance
- Implementing and experimenting with the algorithms
- Applying the results in practice

Summary

- Good applications where improvement is needed
- Lots of data
- Figuring out what needs to be computed
- Need to find good concepts
- Simple methods have impact
- Theory and applications! Quick cycle time

Heikki.Mannila@cs.helsinki.fi



H. Heikinheimo, M. Fortelius, J. Eronen and H. Mannila, Biogeography of European land mammals shows environmentally distinct and spatially coherent clusters. Journal of Biogeography (J. Biogeogr.) (2007) 34, 1053–1064