University of BRISTOL

Advanced Data Analytics: Analysis Case Studies

Ian Nabney

ian.nabney@bristol.ac.uk

bristol.ac.uk

University of BRISTOL

Overview

- Reading: Chapter 15 of Munzner
- Understand how to analyse a visualisation system
- Build a wider portfolio of possible solutions to visualisation challenges

University of BRISTOL

Rationale

- The ability to concisely describe existing systems gives you a firm foundation for considering the full array of possibilities when you generate new systems.
- These case studies illustrate how to use the analysis framework in the course to decompose a vis approach, however complex, into pieces that you can systematically think about and compare with other approaches.
- Now that all design choices have been introduced (apart from colour – which will be covered later in Introduction to Data Analytics), each example has a complete analysis.

University of BRISTOL

Overview

- At the **abstraction** level, these analyses include the types and semantics of the data abstraction and the targeted task abstraction.
- At the **idiom** level, the choices are decomposed into the design choices of how to encode data, facet data between multiple views, and reduce the data shown within a view.
- The analyses also include a discussion of scalability and based on screen space of one million pixels (1000 x 1000).

University of BRISTOL

Overview 2

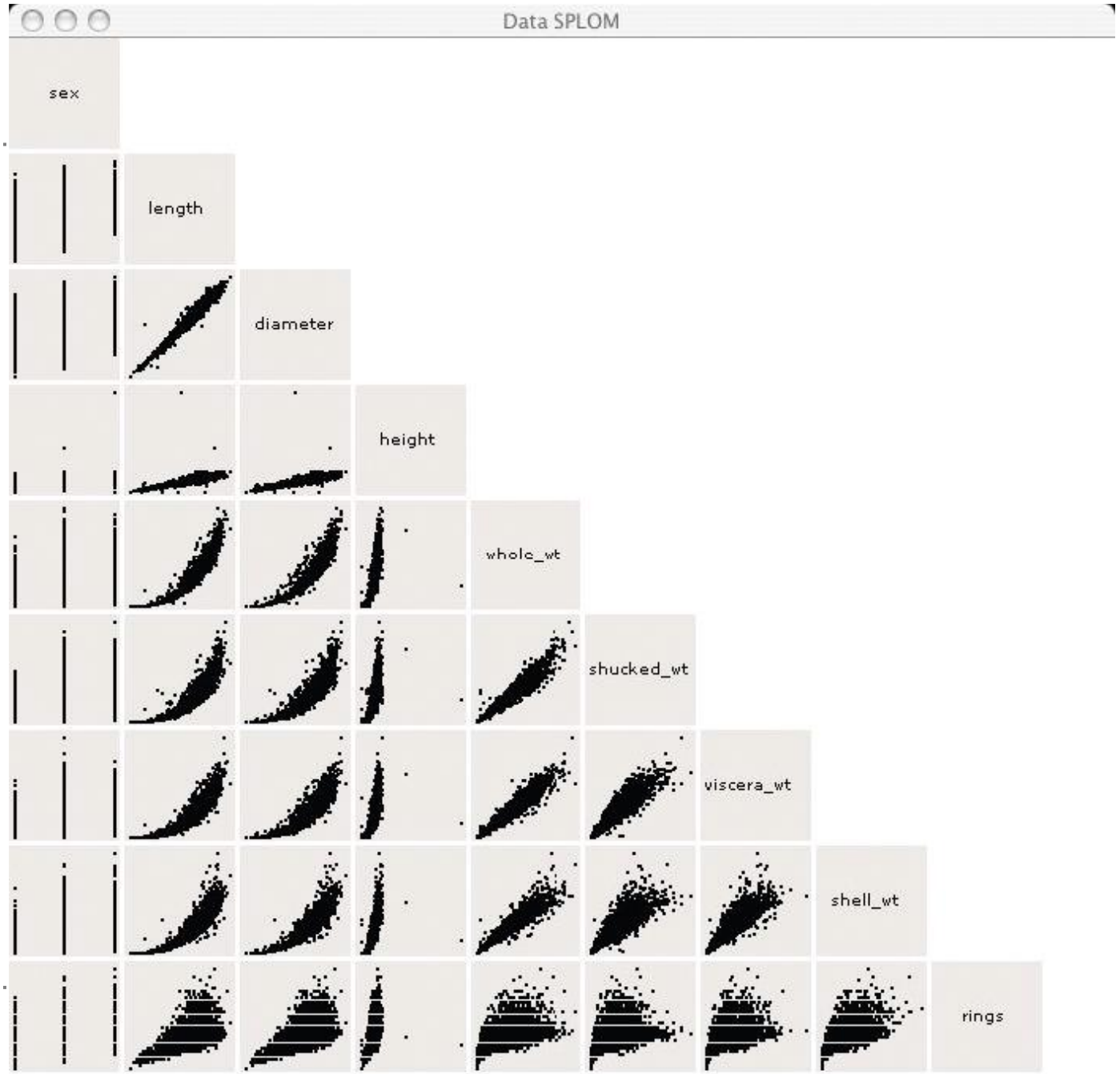
- A few of these systems have simple data abstractions and can be unambiguously classified as handling a particular simple dataset type: tables, or networks, or spatial data.
- Most of them have more complex data abstractions, which is the common case in real-world problems. Many of these systems carry out significant transformations of the original data to create derived data and handle combinations of multiple basic types.
- Often the system is designed to support exploration of interesting structure at multiple levels.

University of BRISTOL

Case Study 1: Scagnostics

- A **single scatterplot** supports direct comparison between two attributes by plotting their values along two spatial axes.
- A **scatterplot matrix (SPLOM)** is the systematic way to compare all possible pairs of attributes, with the attributes ordered along both the rows and the columns and one scatterplot at each cell of the matrix.
- The **scalability** challenge of a SPLOM is that the size of the matrix grows quadratically. Each individual plot requires enough screen space to distinguish the points within it, so this idiom does not scale well past a few dozen attributes.

University of BRISTOL



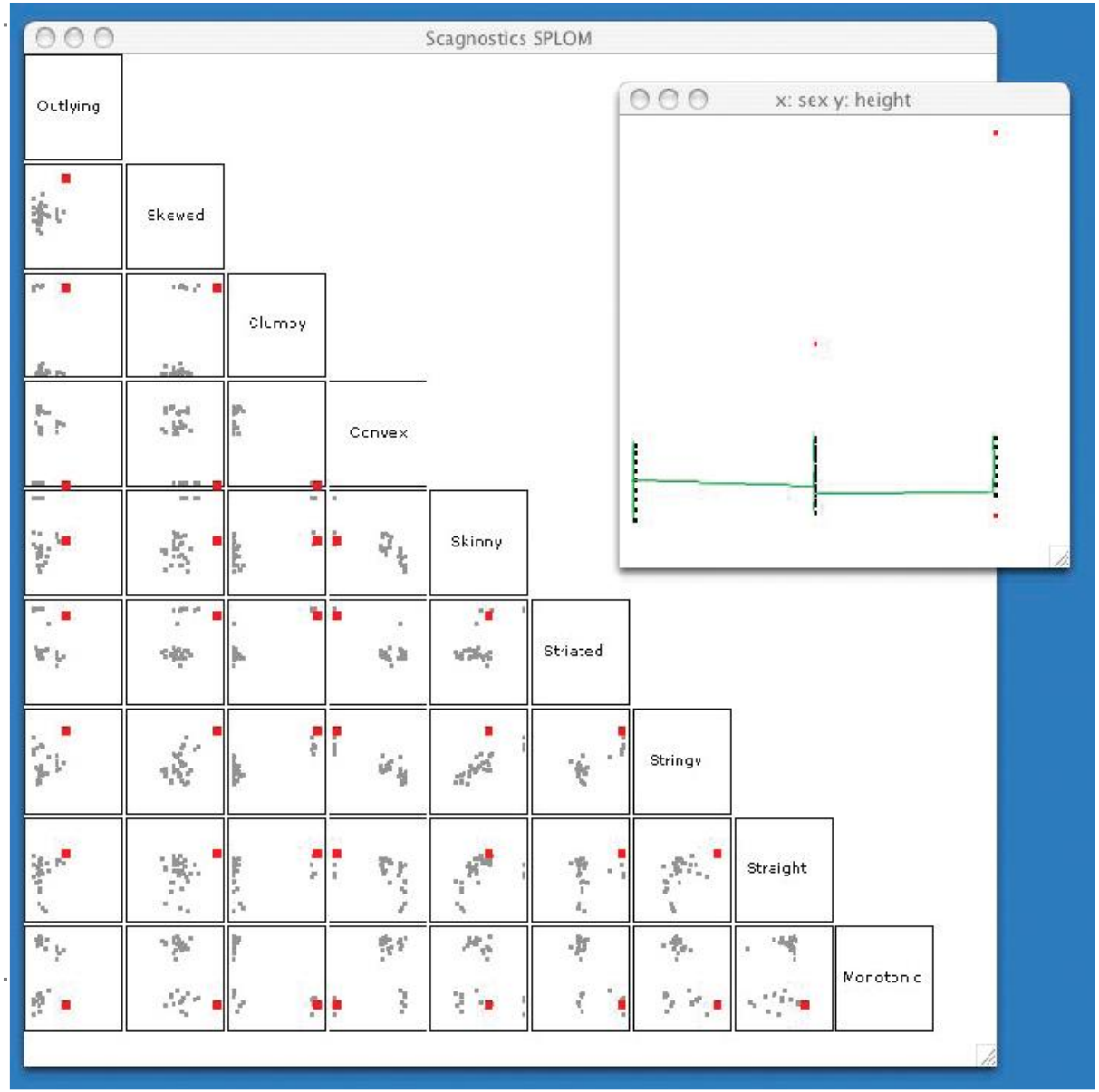
University of BRISTOL

Case Study 1: Scagnostics

- The idea of **scagnostics**, short for scatterplot computer-guided diagnostics, is to identify a small number of measurements that nicely categorize the shape of the point distributions within each scatterplot.
- The nine measures are outlying for outlier detection; skewed, clumpy, sparse, and striated for point distribution and density; convex, skinny, and stringy for shape, and monotonic for the association.
- These measurements are then shown in a new **scagnostics SPLOM** that is a scatterplot of scatterplots. That is, each point in the scagnostics SPLOM represents an entire scatterplot in the original SPLOM, which is shown when the point is selected.
- As with standard SPLOMs, there is linked highlighting between views, and selecting a point also triggers a popup detail view showing the full scatterplot.
- The idea is that the distribution of points in the scagnostics SPLOM should provide a fast overview of the most important characteristics of the original SPLOM. The outliers guide the user to the unusually shaped, and thus potentially interesting, scatterplots in the original SPLOM.

Case Study 1: Scagnostics

- Scagnostics SPLOM for the abalone dataset, where each point represents an entire scatterplot in the original matrix.
- The selected point is highlighted in red in each view, and the scatterplot corresponding to it is shown in a popup detail view.



University of BRISTOL

Case Study 1: Scagnostics

System	Scagnostics
What: Data	Table.
What: Derived	Nine quantitative attributes per scatterplot (pairwise combination of original attributes).
Why: Tasks	Identify, compare, and summarize; distributions and correlation.
How: Encode	Scatterplot, scatterplot matrix.
How: Manipulate	Select.
How: Facet	Juxtaposed small-multiple views coordinated with linked highlighting, popup detail view.
Scale	Original attributes: dozens.

University of BRISTOL

Case Study 2: VisDB

- The VisDB system for database vis treats an entire database as a very large table of data.
- The system shows that table with respect to a specific query that matches some subset of the items in it.
- VisDB computes a set of derived attributes that measure the relevance of the query with respect to the original attributes and items. Each item is given a relevance score for the query for each original attribute.
- An overall relevance score is computed that combines these individual scores, adding an additional derived attribute column to the original table.
- VisDB supports two different layout idioms that are dense, spacefilling, and use square color-coded area marks.

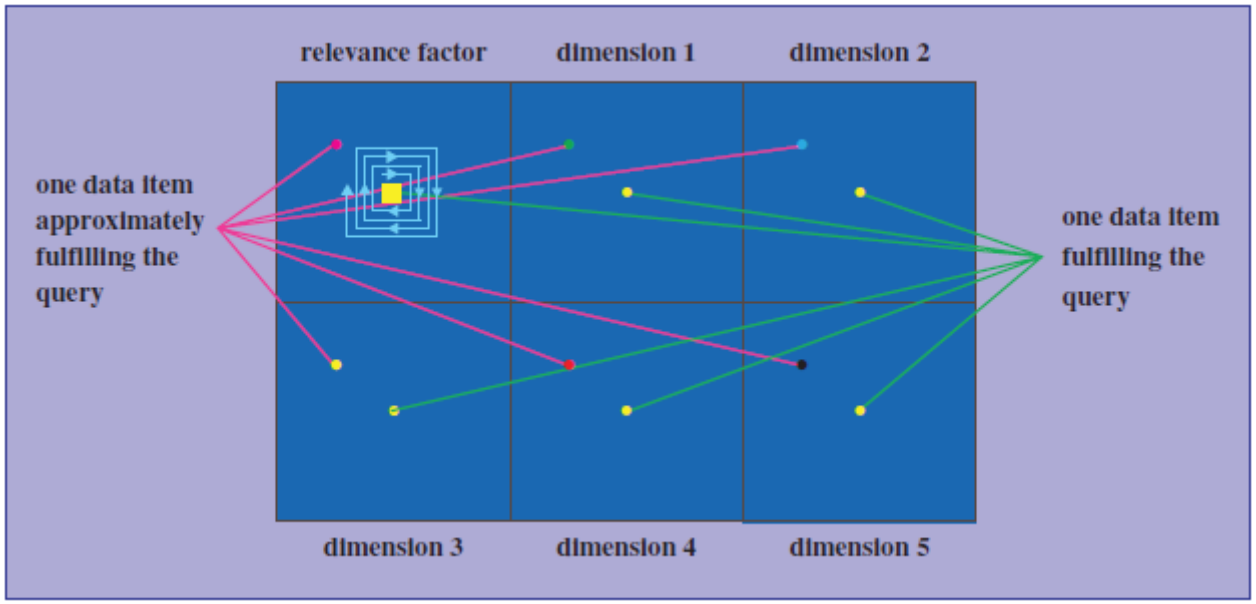
Case Study 2: VisDB

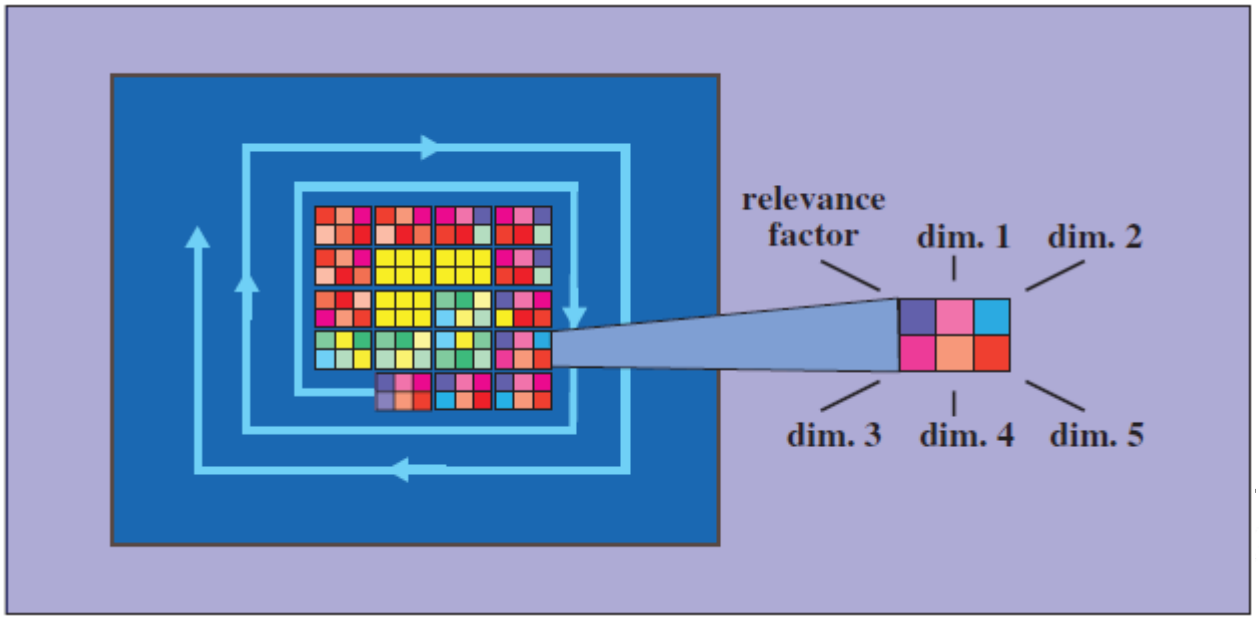
- VisDB layouts schematically, for a dataset with five attributes.

a) Each attribute is shown in a separate small-multiple view.

b) Each item is shown with a glyph with per-attribute sections in a single combined view.

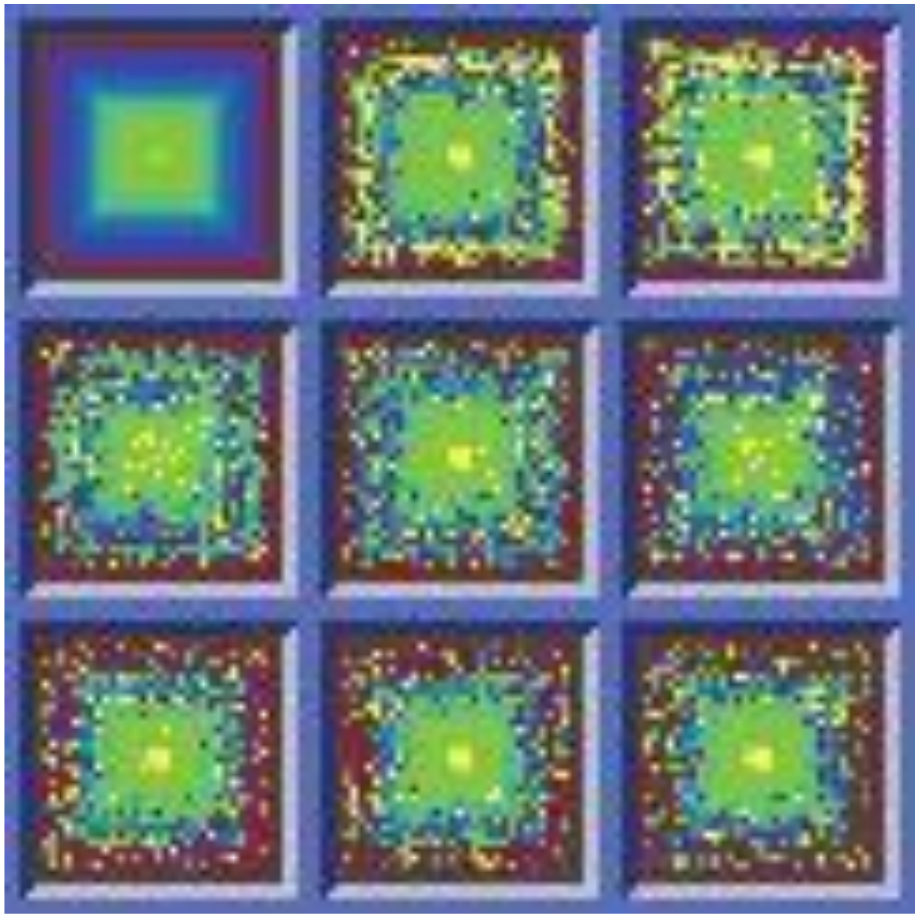
- Next slide shows VisDB screenshots with a dataset of eight attributes and 1000 items.

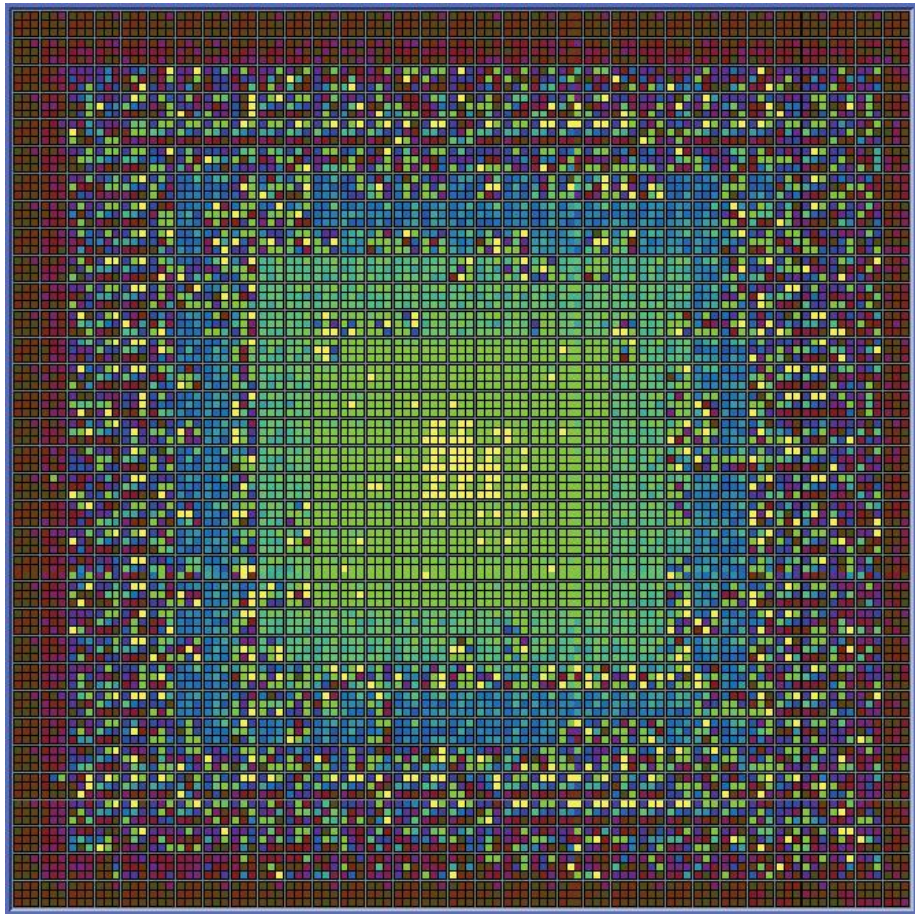




University of BRISTOL

Case Study 2: VisDB





University of BRISTOL

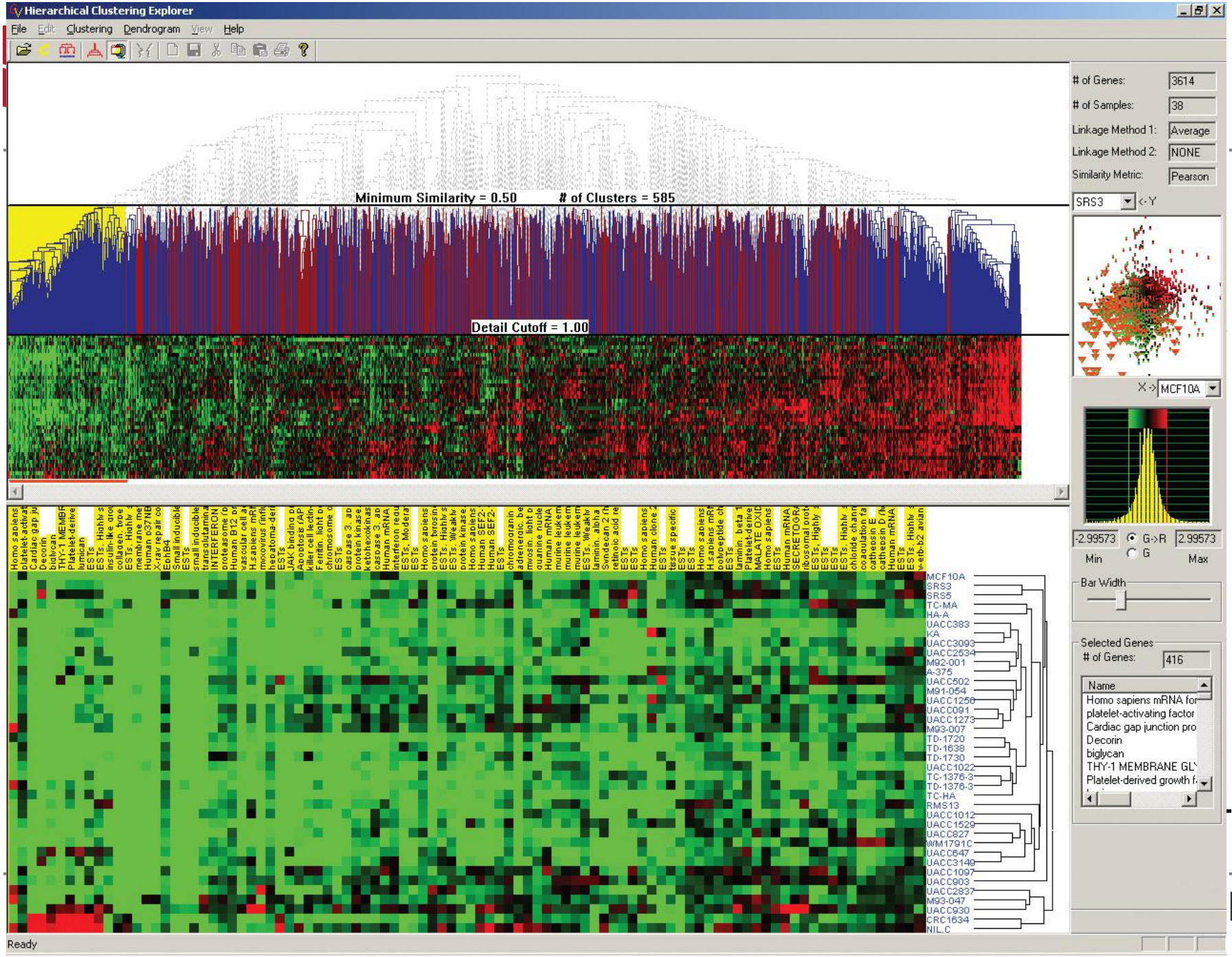
Case Study 2: VisDB Analysis

- Both of these layouts use filtering for data reduction. When the number of items is greater than the available room for the view, items are filtered according to the relevance values.
- The total table size handled by the system thus can be up to several million, even though only one million items can be shown at once in a single screen of one million pixels.
- The small-multiples layout is effective for up to 10–12 attributes, where each per-attribute view shows around 100,000 items, and one million items are shown across all the views.
- In contrast, the layout idiom using a single large glyph-based view only can handle up to around 100,000 visible items, an order of magnitude fewer.
- The elements within the glyph need to be boxes larger than a single pixel in order to be salient, and glyphs need borders around them in order to be distinguished from neighbouring ones.

University of BRISTOL

Case Study 3: Hierarchical Clustering Explorer

- The Hierarchical Clustering Explorer (HCE) system supports systematic exploration of a multidimensional table with an associated hierarchical clustering
- Originally designed for the genomics domain where the table represents microarray measurements of gene expression: a multidimensional table with two key attributes, genes and experimental conditions, and a single quantitative value attribute, the activity of each gene under each experimental condition.
- Achieves scalability through carefully designed combination of visual encoding and interaction idioms. The scalability target of HCE is between 100 and 20,000 gene attributes and between 2 and 80 experimental condition attributes.
- Achieved with the reduction design choices of interactively controlled aggregation, filtering, and navigation, and with the view coordination design choices of overview–detail, small multiple, and multiform side-by-side views with linked highlighting.


Hierarchical Clustering Explorer
File Edit Clustering Dendrogram View Help
Minimum Similarity = 0.50
# of Clusters = 585
Detail Cutoff = 1.00
# of Genes: 3614
# of Samples: 38
Linkage Method 1: Average
Linkage Method 2: NONE
Similarity Metric: Pearson
SRS3
<-Y
X->
MCF10A
-2.99573
G->R
G
2.99573
Min
Max
Bar Width
Selected Genes
# of Genes: 416
Name
Homo sapiens mRNA for
platelet-activating factor
Cardiac gap junction pro
Decorin
biglycan
THY-1 MEMBRANE GL
Platelet-derived growth f
MCF10A
SRS3
SRS5
TC-MA
HA-A
UACC383
KA
UACC3093
UACC2534
M92-001
A-375
UACC502
M91-054
UACC1256
UACC091
UACC1273
M93-007
TD-1720
TD-1638
TD-1730
UACC1022
TC-1376-3
TD-1376-3
TC-HA
RMS13
UACC1012
UACC1529
UACC827
WM1791C
UACC647
UACC3149
UACC1097
UACC903
UACC2837
M93-047
UACC930
CRC1634
NIL.C
Ready

Case Study 3: HCE on Census Data

The rank-by-feature idiom combines the design choice of changing the order and the reduction design choice of aggregation to guide exploration and achieve scalability. Data abstraction is augmented with many new derived attributes. Orderings for each original attribute and pairwise combination of attributes are computed for several choices of ordering criteria, and the user can select which of them to use.