DATABURST: INTERACTIVE ANALYSIS OF HIERARCHICAL DATA USING RADIAL SPACE-FILLING DIAGRAMS

by

Christopher Ravi Smith

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
September 2010

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "DATABURST: INTERACTIVE ANALYSIS OF HIERARCHICAL DATA USING RADIAL SPACE-FILLING DIAGRAMS" by Christopher Ravi Smith in partial fulfillment of the requirements for the degree of Master of Computer Science.

Dated: September 2, 2010

Supervisors: ______________________________
Dr. Stephen Brooks

____
Dr. Robert Beiko

Reader: ______________________________
Dr. Christian Blouin

DALHOUSIE UNIVERSITY

DATE: September 2, 2010

AUTHOR: Christopher Ravi Smith

TITLE: DATABURST: INTERACTIVE ANALYSIS OF HIERARCHICAL DATA USING RADIAL SPACE-FILLING DIAGRAMS

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: M.C.Sc. CONVOCATION: October YEAR: 2010

Signature of Author

Table of Contents

List of Figures

Abstract

Radial visualization methods have been used frequently in the field of information visualization due to their compact design and aesthetic layout of data items that can often aid in the communication of relationships between them. Published methods offer varying levels of effectiveness however, due to restrictions in design, data and interactivity. Their use is often restricted to particular data types, perspectives and interfaces that may not completely match the needs of the analysis, facilitate detailed data investigation or allow for knowledge discovery outside of what was originally intended when the visualization was developed. We therefore present DataBurst, an open, flexible data visualization that extends traditional concentric radial space-filling diagrams using two methods of Focus+Context (Hyperbolic Distortion and TearAway Subdiagrams) and uniquely leveraging three radial diagram characteristics. DataBurst's effectiveness with varying volumes and types of data is then shown with case studies in two active areas of research (network security and bioinformatics) and show that insight can be quickly gained by interactively viewing up to three attributes of data at once in the context of the hierarchy of the data.

Acknowledgements

Heartfelt gratitude is sincerely extended to my supervisors Dr. Stephen Brooks and Dr. Robert Beiko, both of whom offered useful advice and helpful input throughout the development of this body of work. While their areas of expertise are loosely linked, each offered unique insights that increased the functionality, interactivity and usefulness of the visualization, and often gave me new and engaging development goals to strive for.

Having worked with the research teams of both supervisors, I am also indebted to many other colleagues who fielded technical programming questions (including Teryl Taylor) and helped my understanding of the new and daunting field of bioinformatics (including Norman MacDonald and Donovan Parks).

And last but not least, to my family and friends. My loving wife Kate, who helped me through the busiest of times with a smile, and my mother, father and sister who cheered me onwards at all times.

Chapter 1

Introduction

Data visualization seeks to present large volumes of data in a manner that can be easily navigated and investigated, and can effectively offer a level of insight that is not immediately clear when analyzing the underlying data. Growing sources of data are natural occurrences in many fields due to the need to store historical data, larger disk or communication capacity, enhanced data collection techniques, or growth in an organization or customer base.

The first task often employed when dealing with large amounts of data is data summary, where we aim to reduce the volume and complexity of a large dataset such that we do not remove information relevant to the analysis at hand. While correct data aggregation can be difficult in some cases, it allows one to obtain a unique high-level view of large datasets, making their usage tractable while still conveying enough relevant information.

Reducing input data size and complexity has the additional benefit of producing simpler and less confusing visualizations that are then easier to work with as well. But visualizations that are attractive and easy to use are not by themselves useful. They should facilitate insight into the data and support decision making that leverages human logic and intuition, attributes that are not easily reproduced by computer code.

There are many instances where human knowledge proves useful, especially so when large quantities of noise data are present, as in the realm of computer networks security where a very small quantity of intrusive network activity may be present among a large amount of benign activity. Here we use the signal detection analogy of [23], with intrusive activity as the signal, and the benign activity as the noise which makes the former difficult to detect. It is also useful in cases where several very similar entities need to be differentiated, as in the realm of bioinformatics, where amino acids at particular positions of an influenza virus' protein sequence confer resistance to particular antiviral drugs.

Described in this thesis, is DataBurst - a novel extension of radial space-filling (RSF) diagrams that allows users to obtain a high-level view of hierarchical data, employ two focus + context techniques [1] to investigate interesting subsets of data in greater detail, and compare up to three numerical attributes simultaneously. We will show, using two case studies, that these novel techniques allow analysts to flexibly gain useful insight from their datasets.

This thesis is arranged as follows. A list of related visualizations and their shortcomings are surveyed and discussed in Chapter 2. As these cover a wide range of visualization techniques often seen in various other implementations and academic disciplines, we focus only on those that best represent the particular techniques and are most related to the chosen case studies i.e. those related to general computer science and computer network security.

Chapter 3 focuses on the diagram implementation, where we discuss the motivation for DataBurst's creation and design (due to the previously identified shortcomings), and present an example based on census data that showcases basic usage, input data, radial attribute assignment and the two focus + context techniques.

Chapters 4 and 5 discuss an evaluation of DataBurst in two case studies, where we discuss general aspects of analysts' workflows in the fields of network security and bioinformatics, and how insight can be gained from either sets of related data.

We end our discussions in Chapter 6 by summarizing all key contributions and observations, and highlight useful suggestions for future work.

[1] Focus + context refers to a visualization technique that focuses on an area of higher detail within the original diagram, much like viewing a document with a magnifying glass, which allows the viewer to retain overall context [58].

Chapter 2

Background

This chapter introduces the necessary concepts and background knowledge used in this thesis. We begin by describing the origins and general structural properties of the Radial Space-Filling diagram used in this visualization. We then review previous research efforts related to DataBurst and the network security case study performed, focusing not only on their contributions, but their shortcomings that we believe are addressed with DataBurst.

2.1 Radial Space-Filling Diagrams

Radial Space-Filling (RSF) diagrams are examples of space-filling diagrams in that they maximally make use of a pre-defined and usually constant area of screen space. Diagram components are scaled appropriately such that their relative sizes remain the same, in much the same way as treemaps [62]. RSF diagrams were developed from the need to visualize and interact with large data hierarchies, and the term was first formally described by Yang et al. in their presentation of InterRing [74], even though examples could be found in numerous research efforts that predate it.

Parent-child relationships among nodes may be expressed in the diagram - a parent contains the child in a treemap, and in the case of RSF diagrams the hierarchy of nodes is visualized using concentric rings. Each ring is divided into a number of segments each representing a node, where parents are closer to the diagram centre than their children, and the immediate children of an individual parent are adjacent to it and occupy the same sector as the parent.

The basic structure of an RSF diagram can be seen in a simple genealogical fan chart discussed in [21] and shown in Figure 2.1. The input data of this diagram is a simple binary tree where each parent has two children (both genealogically and literally) and the great grandparent or root node, John McKay Baker, has 4 grandchildren and 8 great grandchildren.

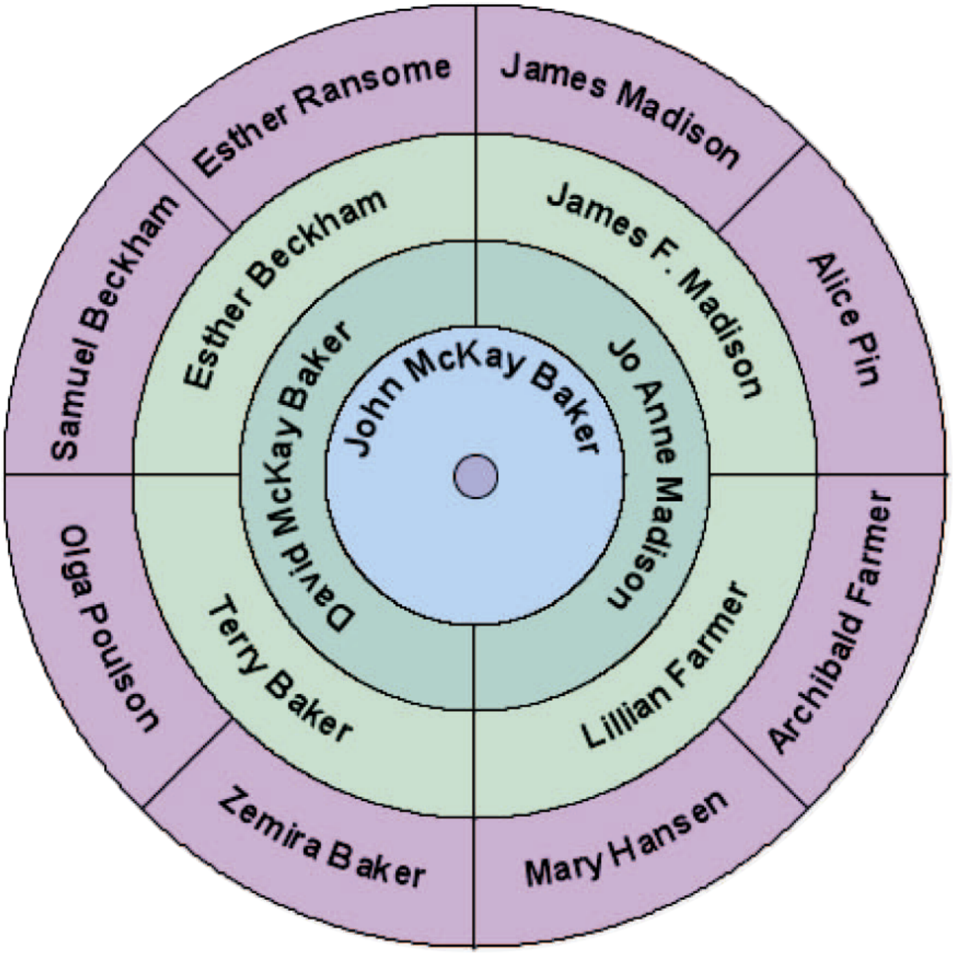


Figure 2.1: Sample RSF diagram - Genealogical fan chart, reproduced from [21].

With respect to the radial diagram, nodes in the higher levels of the data hierarchy are closer to the radial center, while those at the lowest level (the grandchildren or leaf nodes) are located at the outermost ring. Additionally, a parent node has a radial angle equal to the sum of those of its child nodes, and is positioned immediately next to their children in the inner adjacent ring. The children of a node at level i then would be located at level $i+1$, and would be in the outer adjacent ring.

The diagram structure in this way inherently describes the hierarchical relationships between nodes being rendered.

We formally use the term radial angle as an angular measurement around the diagram centre. We will later use the term radial size, which is a distance measurement from the diagram centre. In Figure 2.1 the node representing Samuel Beckham has a radial angle of 45 degrees, and Esther Beckham's radial size is roughly the same as all others.

While formal evaluations that highlight radial space-filling diagrams' effectiveness over other types of visualizations are difficult to find (one can argue, due to the many associated difficulties [57], that few such comprehensive evaluations exist for many well known visualizations), Stask empirically showed that a circular space-filling diagram is often preferred over a rectangular equivalent in particular search and analysis tasks [64]. When study participants were asked to identify or compare files or directories by the name, type or size (e.g. the largest file, deepest subdirectory, duplicates etc), using both visualizations and the same dataset, higher performance, both in correctness and time, was observed with the radial diagram.

Participants drew particular attention to the utility of having the structure and hierarchy available when investigating the individual nodes with the radial diagram, and having directories shown separately from their constituent files (in contrast to the rectangular approach that placed files within directories). We agree that it is this context and explicit separation of hierarchy levels that make radial space-filling diagrams useful when visualizing many types of data.

However, participants of that study also noted that with large volumes of data, leaf nodes became small enough that labels were difficult to see in the radial representation. It is important then that useful visualization and interactive techniques are present to assist users in identifying these nodes when necessary. Yang et al. also cited this need in [74].

2.2 Related Work

We present a historical overview of a subset of those visualizations that are related to DataBurst in varying degrees; listing their features and introducing necessary terminology and ideas.

Note that for more recent solutions for complex datasets, multiple linked visualizations may be presented to the user at once on the same screen. We focus in those cases on the subvisualizations that are most related.

2.2.1 RSF Diagrams

Early forms of radial space-filling diagrams have predated InterRing [74], and bore many similarities both in terms of presentation and interaction. These included the cascading, semi-circular discs and information slices of Andrews et al. [4] as seen in Figure 2.2, which presented one or more static half discs each showing 5 levels of a local hard disk directory structure.

Each node on the semi-circular disc was sized to give an indication of the storage space it consumes, and colour-coded based on file types present. Clicking on a sub-directory in the outermost ring of one diagram would retrieve the following 5 levels of directories (if present) in another diagram immediately to its right.

While this was a practical solution to the problem of large visualizations based on deep data hierarchies, overall context could be lost when progressively traveling to lower directory levels, and information of nodes in different discs could not be directly compared.

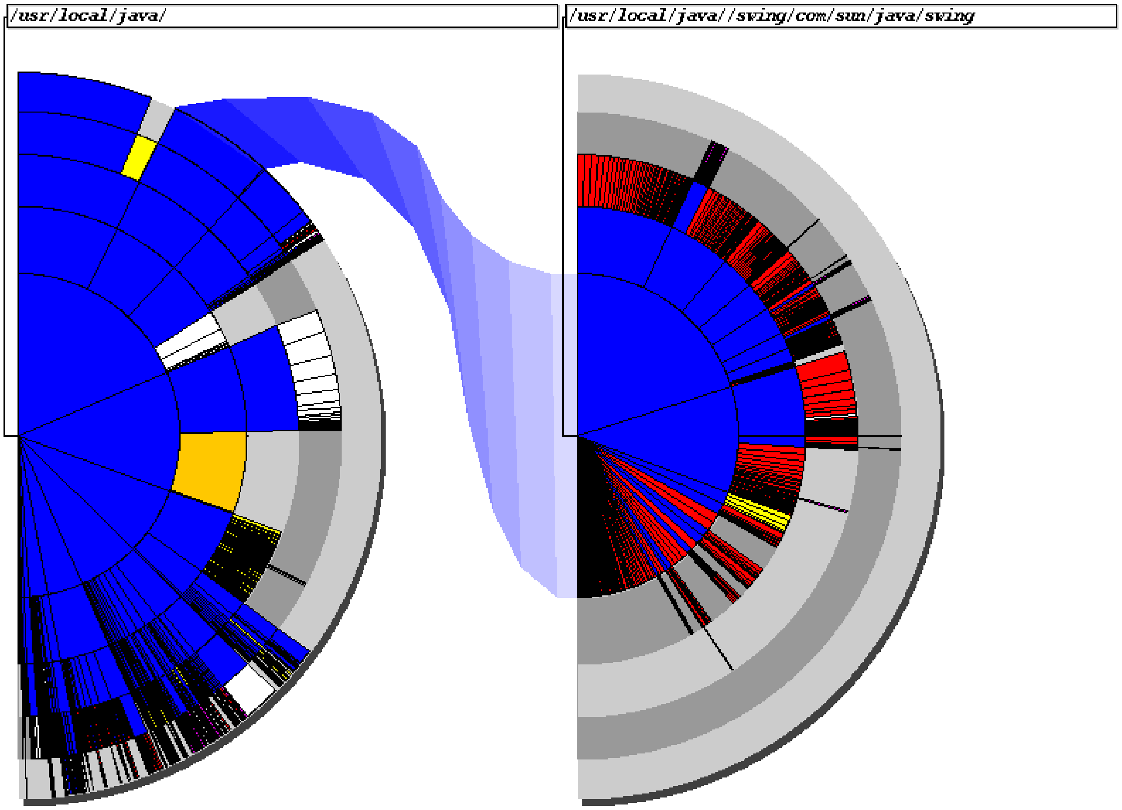
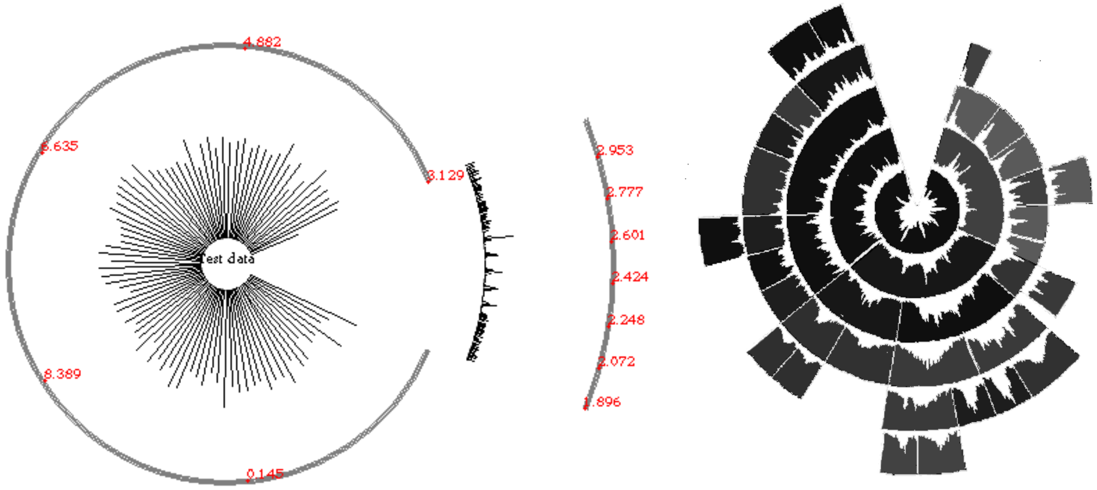


Figure 2.2: Cascading, semi-circular discs, showing disk contents, reproduced from [4].



(a) Basic SolarPlot with focus area.

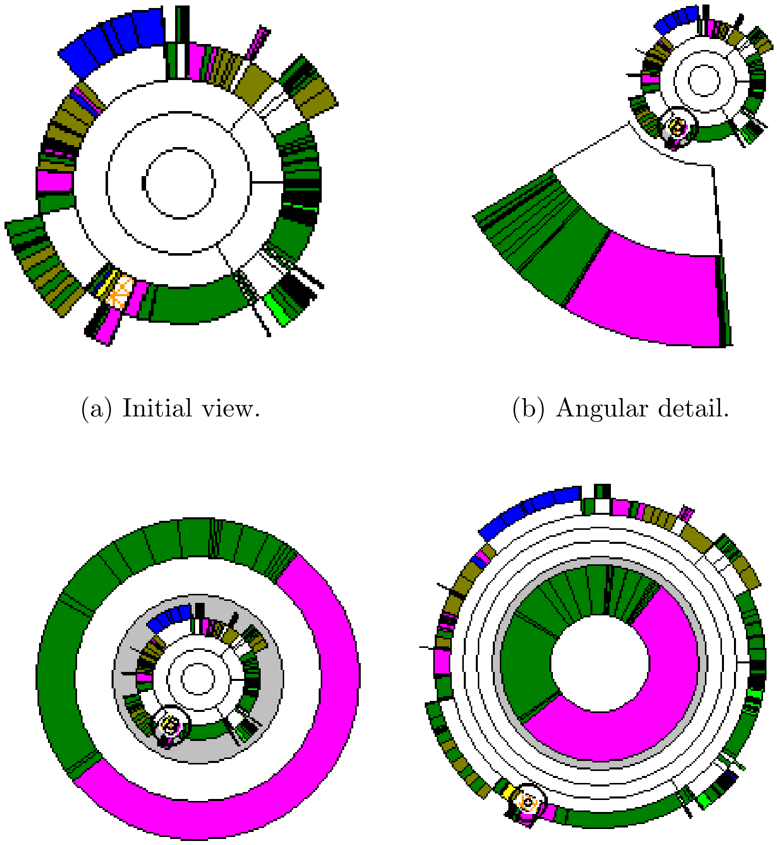
(b) SolarPlot and aggregate treemap.

Figure 2.3: SolarPlot showing sales data over time, reproduced from [13].

Chuah's SolarPlot [13] (Figure 2.3(a)) aggregated histogram data into bins that were radially aligned around a centre point and employed varying types of automatic and user-controlled aggregation techniques to reduce complexity. Focus + context was employed by allowing a region of interest to be displayed within a sector of the same diagram, while the remainder of the data display remains unchanged.

SolarPlots could be merged with aggregate treemaps to produce circular aggregate treemaps (Figure 2.3(b)), which share several properties with RSF diagrams. The author noted that some information loss is normally expected due to data aggregation, which we surmise may hinder data exploration, as data bin components are not listed. The goal with DataBurst is to ultimately show all input data, allowing the user to view any data element present.

Stasko and Zhang's Sunburst [65] (Figure 2.4(a)) also leveraged focus + context by allowing either the global view to be scaled down while rendering a detail view of select leaf nodes in a new ring surrounding it (called angular detail and detail outside, shown in Figures 2.4(b) and 2.4(c)), or reducing the thickness of the circles in the global view to render the detail view in the empty central area within it (referred to as detail inside, shown in Figure 2.4(d)). This was done using transitions which helped



(a) Initial view.

(b) Angular detail.

(c) Detail outside.

(d) Detail inside.

Figure 2.4: Sunburst showing disk contents and focus + context techniques, reproduced from [65].

the user track changes in focus, and kept the total amount of space used relatively constant. Sunburst's area of focus was however limited to the leaf level and could potentially make it difficult to relate emphasized leaf nodes with their non-emphasized parents and grandparents, leading to a loss of a valuable dimension of data.

InterRing's main contribution were two distortion techniques, which both involved the dragging of node edges. They were circular and radial distortion, which enabled the user to enlarge a node's radial angle and size respectively, by pinning one edge of the node (so that it remains stationary), and dragging the opposite edge to shrink or enlarge a node. Multiple selection, highlighting and hierarchy modification techniques were proposed and adapted from other non-radial implementations e.g. structure-aware brushing.

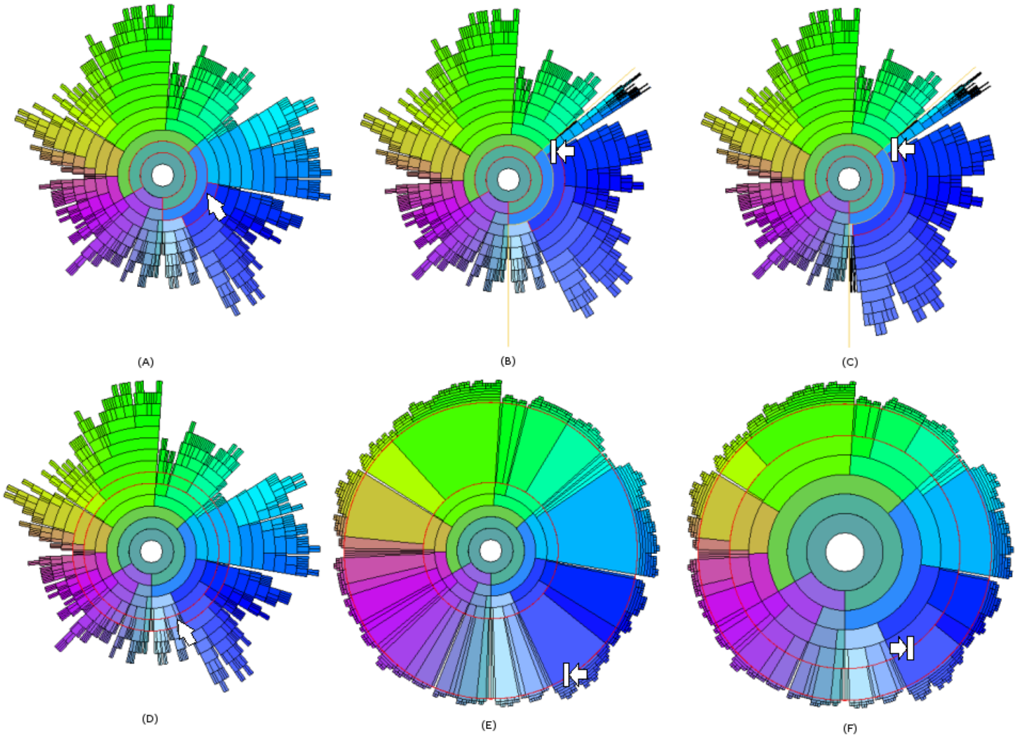
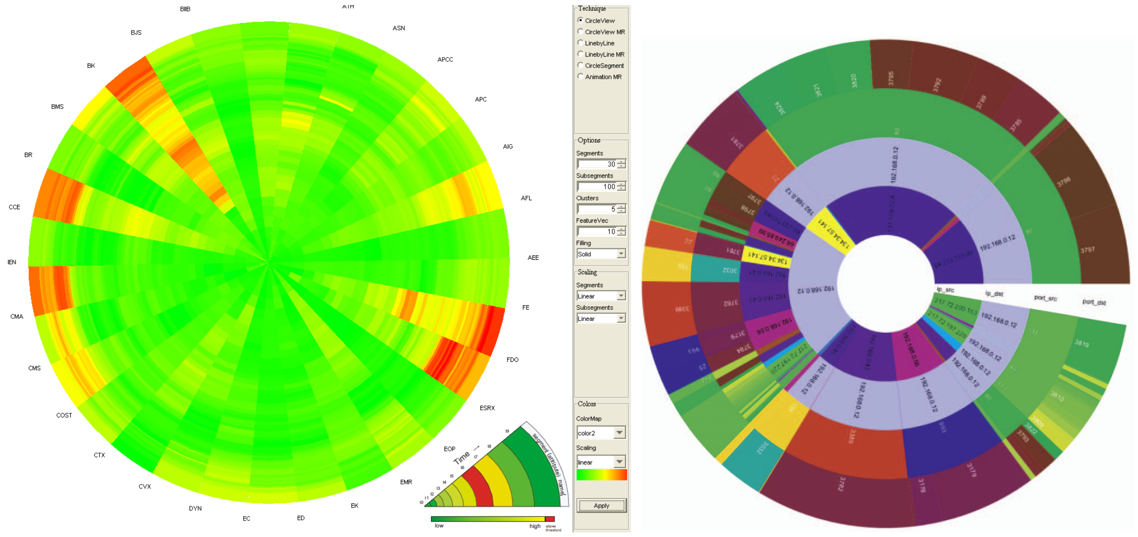


Figure 2.5: InterRing showing 8 variables related to cars (e.g. MPG, horsepower etc.) and circular (in a, b and c) and ring (in d, e, and f) distortions, reproduced from [74].

These two distortions are demonstrated in Figure 2.5, where a node with two siblings (highlighted in Figure 2.5(a)) is expanded into the space held by either sibling in Figures 2.5(b) and 2.5(c) using circular distortion, and a ring distortion is used to increase the radial sizes of different nodes in the final two subdiagrams.

While this technique can powerfully create multiple focus areas within a single diagram, we contend that interacting with individual diagram nodes can be difficult (especially so with tiny nodes with short edges), and manipulating a group of contiguous nodes individually may become tedious if their parent has many additional children. We argue that a simpler distortion technique would suffice.

Ankerst, Keim et al. [5] presented one of the earliest variants of RSF diagrams, Circle Segments, which was the first use of RSF diagrams to show evolution in time for multi-dimensional data. A circle was split into a number of equally sized sectors, which were each assigned to individual attributes. Each sector was subdivided equally into time bins where earlier times were either closer to the circle centre or to the outer edge, and node colours represented attribute values at that point in time.



Technique
● CircleView
○ CircleView MR
○ LinebyLine
○ LinebyLine MR
○ CircleSegment
○ Animation MR

Options
Segments: 30
Subsegments: 100
Clusters: 5
FeatureVec: 10
Filling: Solid

Scaling
Segments: Linear
Subsegments: Linear

Colors
ColorMap: color2
Scaling: linear

Apply

(a) CircleView and interface.

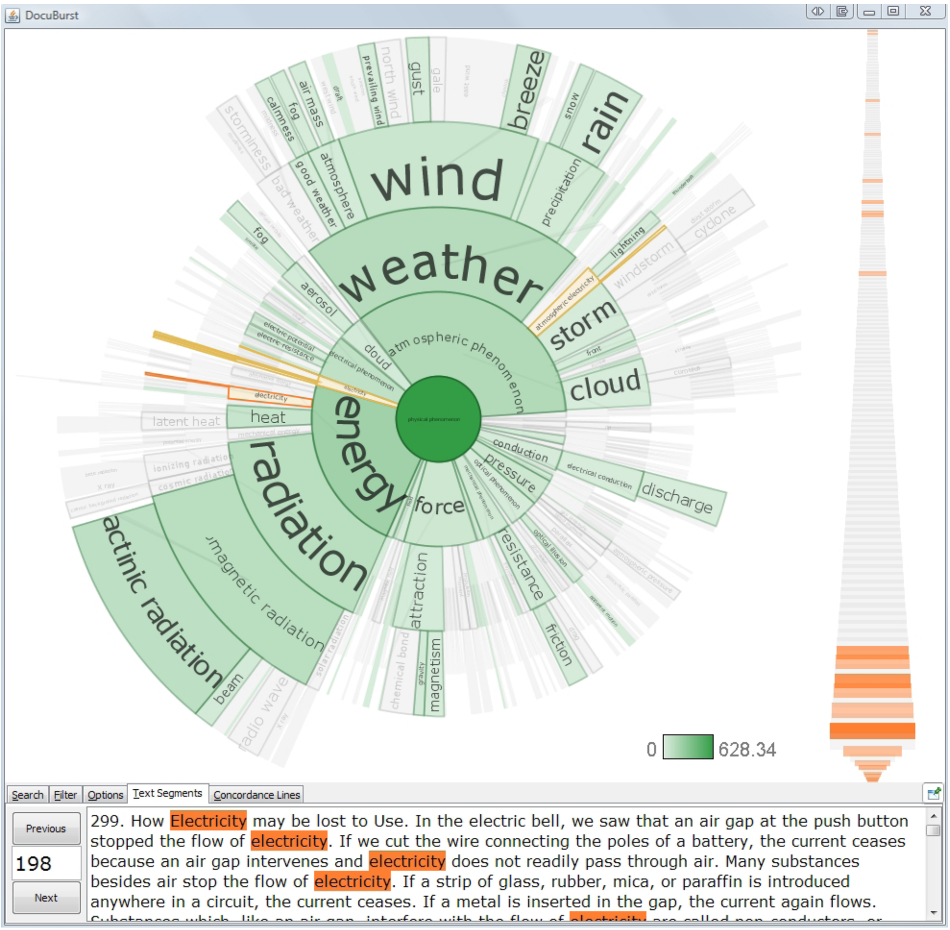
(b) Radial Traffic Analyzer.

Figure 2.6: Two RSF diagrams by Keim (showing stock prices for 30 stocks from the S&P 500 index and network communications) reproduced from [36] and [37].

Keim, whose primary work focuses on interactive visualization of large volumes of multi-dimensional data, later augmented this technique with CircleView [36], which enabled dynamic control of time ranges such that the user could examine different periods of time, as shown in Figure 2.6(a)).

While these two visualizations can show multiple attributes at once and continuous changes in their data streams over time, it would be difficult to do this for more than a handful of objects and would not facilitate the comparison of attributes among multiple objects. Keim addressed this problem by extending the concept in the Radial Traffic Analyzer [37], and used the attributes of network communications to define a data hierarchy which was then used to create an RSF with each attribute occupying a separate ring. However, this diagram could only express attribute values through the diagram structure, as seen in Figure 2.6(b) where source and destination IP addresses and ports involved in network communications are individually shown in the four concentric rings. No further information was available from this diagram e.g. communication time stamps, volume of data transferred.

Collins' DocuBurst [15], which serves as partial inspiration for this project, visualized document content using language structure, allowing users to glean the general



299. How Electricity may be lost to Use. In the electric bell, we saw that an air gap at the push button stopped the flow of electricity. If we cut the wire connecting the poles of a battery, the current ceases because an air gap intervenes and electricity does not readily pass through air. Many substances besides air stop the flow of electricity. If a strip of glass, rubber, mica, or paraffin is introduced anywhere in a circuit, the current ceases. If a metal is inserted in the gap, the current again flows. Substances which, like an air gap, interfere with the flow of electricity are called non-conductors, or

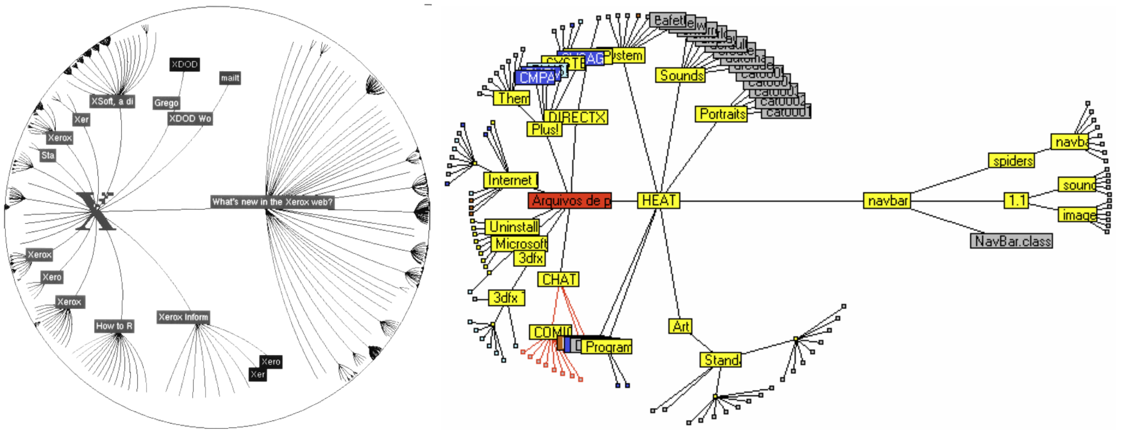
Figure 2.7: DocuBurst and interface summarizing the content of a science textbook, reproduced from [15].

idea of the content of a document by using word frequencies and lexical structures to visualize semantic content. This is shown in Figure 2.7, where all concepts in a document related to physical phenomena are shown, and searching for a particular word causes it to be highlighted in orange in the radial diagram, the text segment browser, and the fisheye distorted word browser to the right. This shows that while the word ‘electricity’ is not frequently mentioned directly, it is mentioned often toward the end of the document.

Collins argues that, in this context, frequently occurring stemmed words (the base or root word) are what most define a document’s semantic content and DocuBurst therefore communicates word counts with node colour, opacity and size. This may not be suitable in all cases, especially so for larger documents having rarely occurring yet highly meaningful terms that cannot be navigated to in this radial diagram.

2.2.2 Radial Diagrams

To an extent the hyperbolic browser of Lamping et al. [41] and bifocal tree of Cava et al. [10] are examples loosely related to RSF diagrams, but are both simple graphs (node-edge diagrams). The former visualization placed the nodes on a hyperbolic plane and transformed this to a Euclidean circle, allowing for a considerably large volume of data to be rendered while keeping the few nodes of interest at the centre of the diagram. It is however, possible to lose overall context with a large enough data hierarchy when the focus is near the leaf nodes causing the many higher nodes to become compressed into one portion of the diagram.



(a) The hyperbolic browser.

(b) The bifocal tree.

Figure 2.8: Two node-edge diagrams showing links between pages on Xerox's web server and disk contents respectively, reproduced from [41] and [10].

The bifocal tree bore some similarity to the hyperbolic browser, but rendered the single data hierarchy in two intersecting circular areas. The nodes at the point of interest and below were rendered in the rightmost circular sector, and the remaining nodes were rendered in the leftmost sector. This provided for clear and informative global and detail views, but could lead to a confusing loss of context when a distant node of interest was chosen in the global view due to the lack of transition animations. Heer et al. argued in [30] that transitions are necessary to help users track nodes and better understand the relationship between different views.

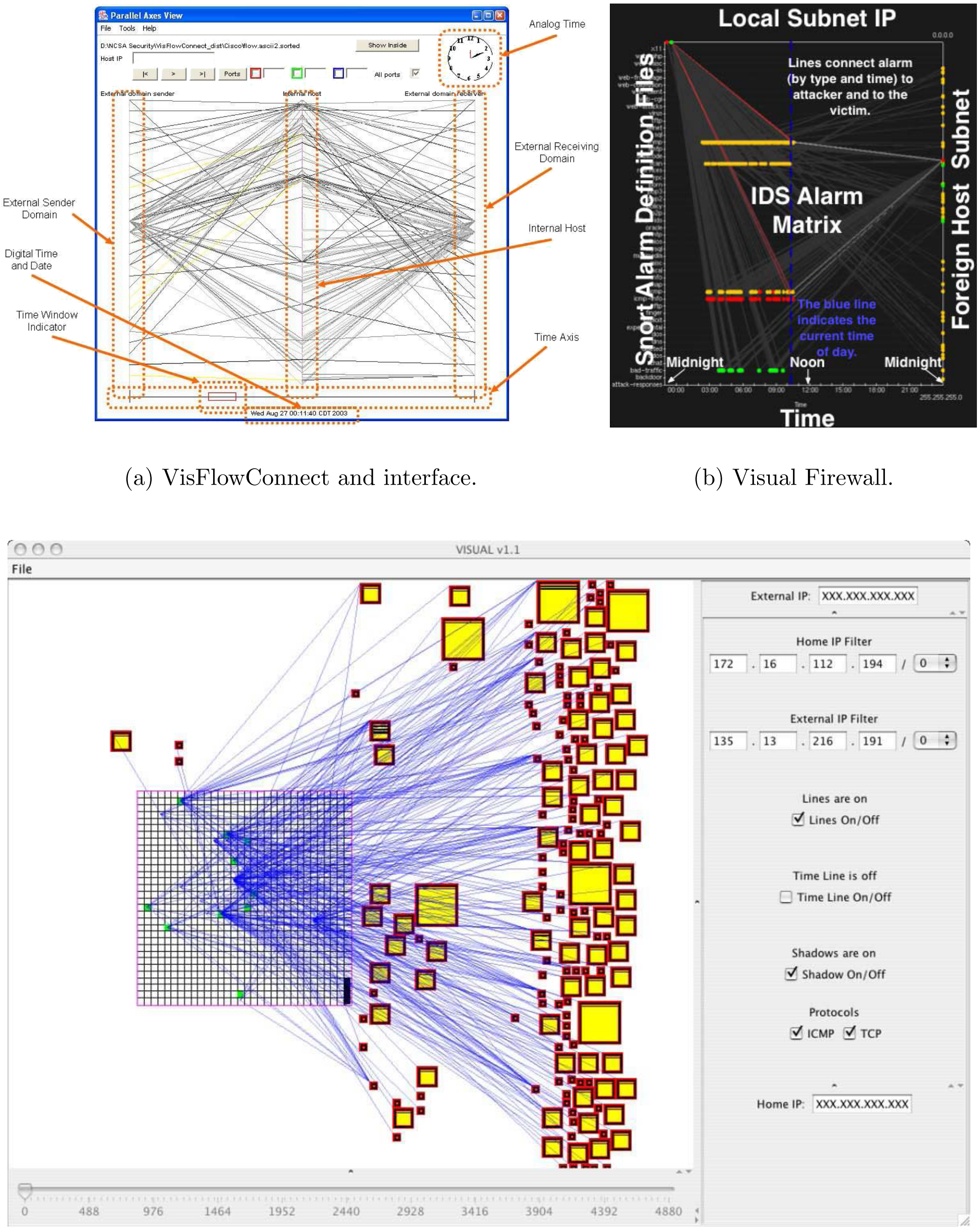
It should be noted that item layout, overlapping and occlusion are common issues with node-edge diagrams, as described by [19] and [35], especially with a large number of nodes with long labels. Radial space-filling methods do not suffer from these drawbacks and have the ability to show significantly more information.

2.2.3 Network Security Visualizations

When displaying a scalar attribute over time, a line graph would suffice. Generally speaking, most network security analyses investigate many attributes related to communications between interconnected computer hosts, which cannot easily be done with collections of line graphs, or graphs with a large number of overlapping lines. These attributes describing communications include IP addresses (that uniquely identify computers on a network e.g. 192.168.0.1), ports used (software constructs through which network services are provided e.g. FTP services are provided through port 21), timestamps and duration of communication, number of bytes and packets transferred (long bit streams are usually split into well-defined packets when transmitted), or may be application specific (e.g. those derived from packet content or country of origin across the World Wide Web). Many network security visualizations have investigated this high-dimension and volume problem and a subset are categorized below.

Parallel coordinate plot visualizations project each vector data point as a polyline passing through a set of parallel coordinates, one for each vector component which equates to the each attribute to be visualized. With VisFlowConnect [75], authors Yin et al. used source and destination IP addresses and ports as coordinate systems to show that many network attacks such as virus outbreaks and denials of service, can be easily seen when multiple lines intersect one or more points on particular axes. A slider facilitates playback of historical communications, as seen in Figure 2.9(a).

Lee et al.'s Visual Firewall [44] followed a similar approach by rendering a large box with edges representing coordinate systems, and showed data points having component lines that join pairs of adjacent box edges. An example of this is shown in Figure 2.9(b) where alarms produced by various intrusive activities are linked to foreign host subnets (subnets represent a group of computers e.g. the 192.168.0.0/16 subnet is made up of hosts 192.168.0.1 to 192.168.255.255) and victims (local subnet IP addresses) in the context of time. It is this aspect that can make visualizations of this type both useful and confusing, as a large dataset can produce a confusing diagram having indistinguishable sets of overlapping lines.



D:\NCSA Security\VisFlowConnect_dist\Cisco\flow.ascii2.sorted

Host IP:

All ports ☑

Wed Aug 27 00:11:40 CDT 2003

External IP: XXX.XXX.XXX.XXX

Home IP Filter: 172 . 16 . 112 . 194 / 0

External IP Filter: 135 . 13 . 216 . 191 / 0

Lines are on
☑ Lines On/Off

Time Line is off
☐ Time Line On/Off

Shadows are on
☑ Shadow On/Off

Protocols
☑ ICMP ☑ TCP

Home IP: XXX.XXX.XXX.XXX

(a) VisFlowConnect and interface.

(b) Visual Firewall.

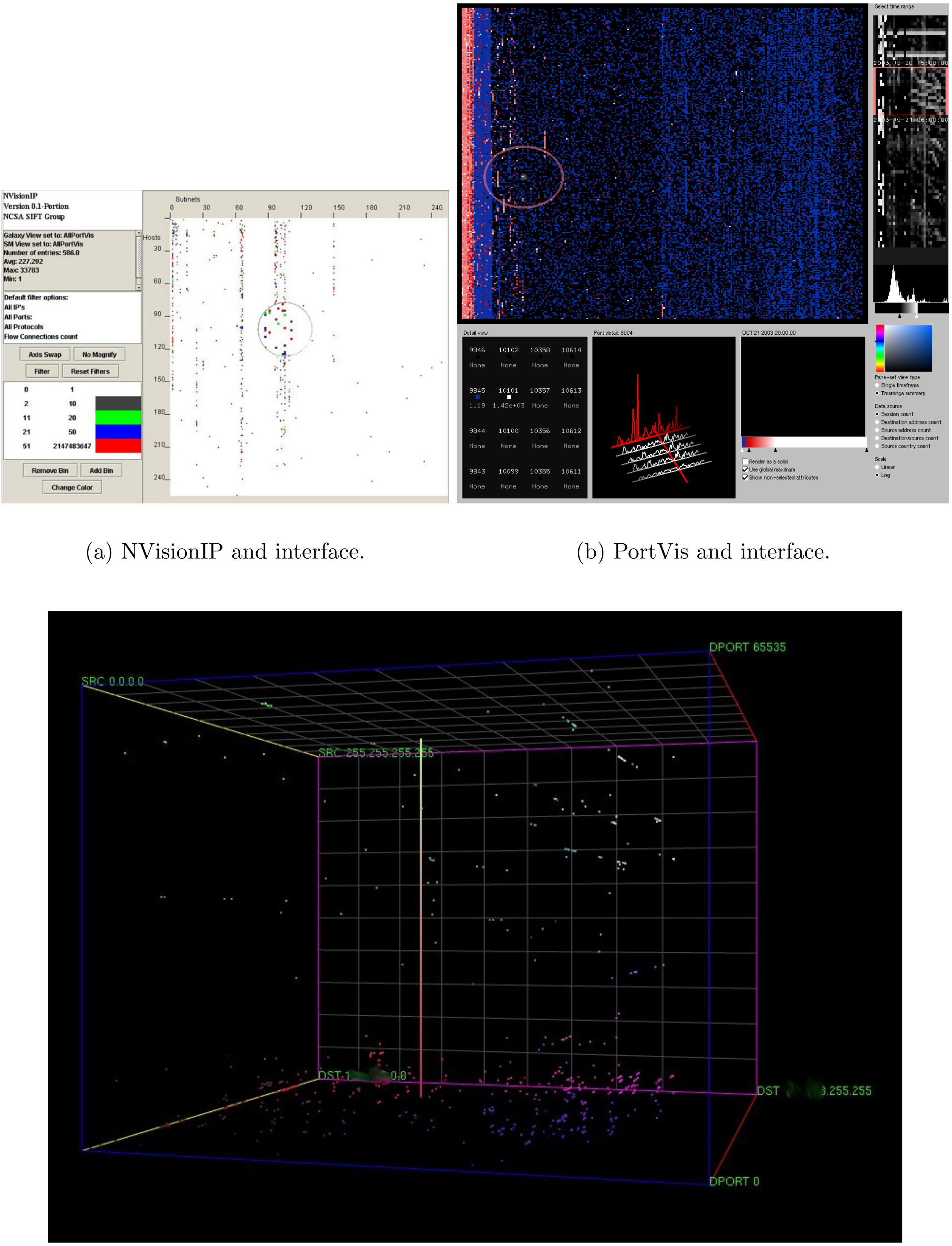
(c) VISUAL and interface.

Figure 2.9: Two parallel coordinate visualizations and VISUAL, all showing network communications over time, reproduced from [75], [44] and [6].

In the second category, Scatterplots render attributes in two-dimensional (2D) or three-dimensional (3D) space using points and/or connecting lines. Ball et al.'s VISUAL [6] incorporated these by placing up to 2500 internal hosts in a grid (hosts within the local area network), up to 10,000 external hosts around the grid (computers on the internet) and connecting lines representing host to host communication in 2D space. Boxes represented external host nodes, which were scaled based on their volume of communication, and grid cells represented internal nodes arranged so that hosts in a subnet occupied the same vertical line. Even with automatic external node positioning, line colours showing traffic direction, interactive filtering and the ability to view different time intervals, produced diagrams can be potentially confusing with connecting lines overlapping external nodes and other lines. We see in Figure 2.9(c) that communications between 1020 internal nodes and 183 external nodes over 80 hours has almost reached a level at which individual lines cannot be discerned.

Lakkaraju at al.'s NVisionIP [40] and McPherson et al.'s Portvis [50] were similar in that they position points in 2D space, colour-coded to an attribute or change in attribute, with the former using host number on the vertical axis and subnet number on the horizontal, and the latter using the high order port and low order port on the horizontal and vertical axes. Both can be seen in Figures 2.10(a) and 2.10(b) to offer varying levels of detail and aggregation over time in multiple windows to help preserve context, and are effective at identifying anomalous activities that produce clusters of points or points that follow regular patterns e.g. lines with sequential port or host scans. Scatterplots however in general perform poorly with low activity (that produce little or no clustering) or covert attacks that follow irregular patterns, and are limited by the number of attributes they can display simultaneously [50].

The third dimension was leveraged in Lau's Spinning Cube of Potential Doom [43] where coloured dots representing IDS alerts were placed in 3D space. It was also well suited to showing similar types of attacks as NVisionIP and PortVis e.g. the straight line produced by a scanning attempt when source and destination IP and destination ports are mapped to the three axes in Figure 2.10(c). There is an added level of complexity though, as with most 3D visualizations an ideal location for the camera in 3D space must first be sought in order to see particular attacks, especially when numerous clustered alerts cause occlusion or ambiguity.



NVisionIP
Version 0.1-Portion
NCSA SIFT Group

Galaxy View set to: AllPortVis
SM View set to: AllPortVis
Number of entries: 586.0
Avg: 227.292
Max: 33783
Min: 1

Default filter options:
All IP's
All Ports:
All Protocols
Flow Connections count

0	1	
2	10	■
11	20	🟩
21	50	🟦
51	2147483647	🟥

Detail view

9846	10102	10358	10614
None	None	None	None
9845	10101	10357	10613
1.19	1.42e+03	None	None
9844	10100	10356	10612
None	None	None	None
9843	10099	10355	10611
None	None	None	None

Port detail: 8004

OCT 21 2003 20:00:00

□ Render as a solid
☑ Use global maximum
☑ Show non−selected attributes

Pane−set view type
◇ Single timeframe
◆ Timerange summary

Data source
◆ Session count
◇ Destination address count
◇ Source address count
◇ Destination/source count
◇ Source country count

Scale
◇ Linear
◆ Log

(a) NVisionIP and interface.

(b) PortVis and interface.

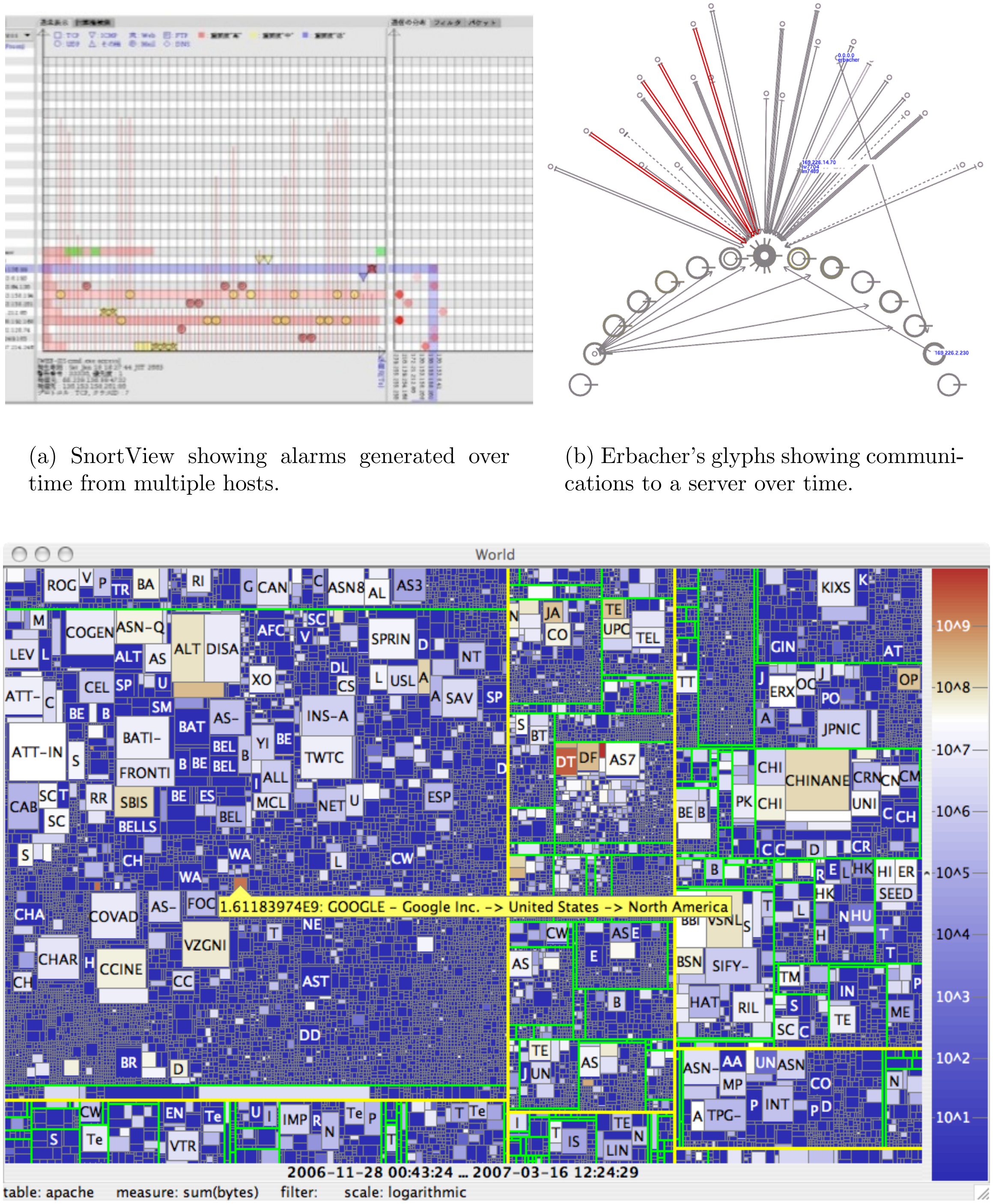
(c) Spinning Cube of Potential Doom.

Figure 2.10: NVisionIP, PortVis and the Spinning Cube of Potential Doom, all showing network communications over time, reproduced from [40], [50] and [43].

In the third category, Glyph-based (informative icons or graphics) visualizations extend scatterplots by mapping multivariable entities to glyphs placed in 2D or 3D space. Koike et al.'s SnortView [38] arranged IDS alerts (which describe abnormal network activity) on a 2D graph by IP address and time, where the shape represented the type of attack and colour represented attack severity. We see a number of severe attacks in red in Figure 2.11(a) over several protocols. Erbacher utilized thick, double, cross hashed or dotted coloured arrows to represent multiple types of communication between hosts e.g. unauthenticated connections, failed password attempts, privileged FTP connections and possible attacks in [24]. The arrows can be thought of as glyphs themselves. Glyphs representing servers were circular, but could show mail activity, system load, disk usage and number of users using multiple concentric circles of varying thickness and radiating spokes. While these glyph diagrams can certainly contain a large quantity of helpful information, we surmise that they could be unnecessarily complex for most users and would require a comprehensive legend and some time to fully understand, as seen in Erbacher's multiple network communications to a single server over time in Figure 2.16(b).

Network security visualizations have also leveraged space-filling techniques. Mansmann's Hierarchical Network Map (HNMap) [48] showed aggregated network traffic entering a host or gateway for a data hierarchy of Continent, Country, Autonomous Systems (AS) and Network, based on the IP addresses of incoming communications in a rectangular interactive treemap. Rendered boxes represented nodes at a particular level in the hierarchy, and children nodes were placed inside the parent boxes. Nodes physically close to each other were rendered closer (e.g. neighbouring countries were adjacent in the diagram), and nodes were sized proportionally to the number of IP addresses contained, with node colour representing any attribute of communication e.g. bytes transferred in all autonomous systems in Figure 2.11(c). As one can move up or down the hierarchy easily, HNMap can effectively show worm propagation or high-activity nodes of interest at any level of the hierarchy, though the amount of information displayed can be overwhelming.

Hilbert Curves are an example of space-filling curves, that were used in [34] to create well defined clusters of similar IP addresses as points, and visualized aggregate communication between them as a continuous fractal space-filling curve.



1.61183974E9: GOOGLE – Google Inc. –> United States –> North America

2006–11–28 00:43:24 ... 2007–03–16 12:24:29

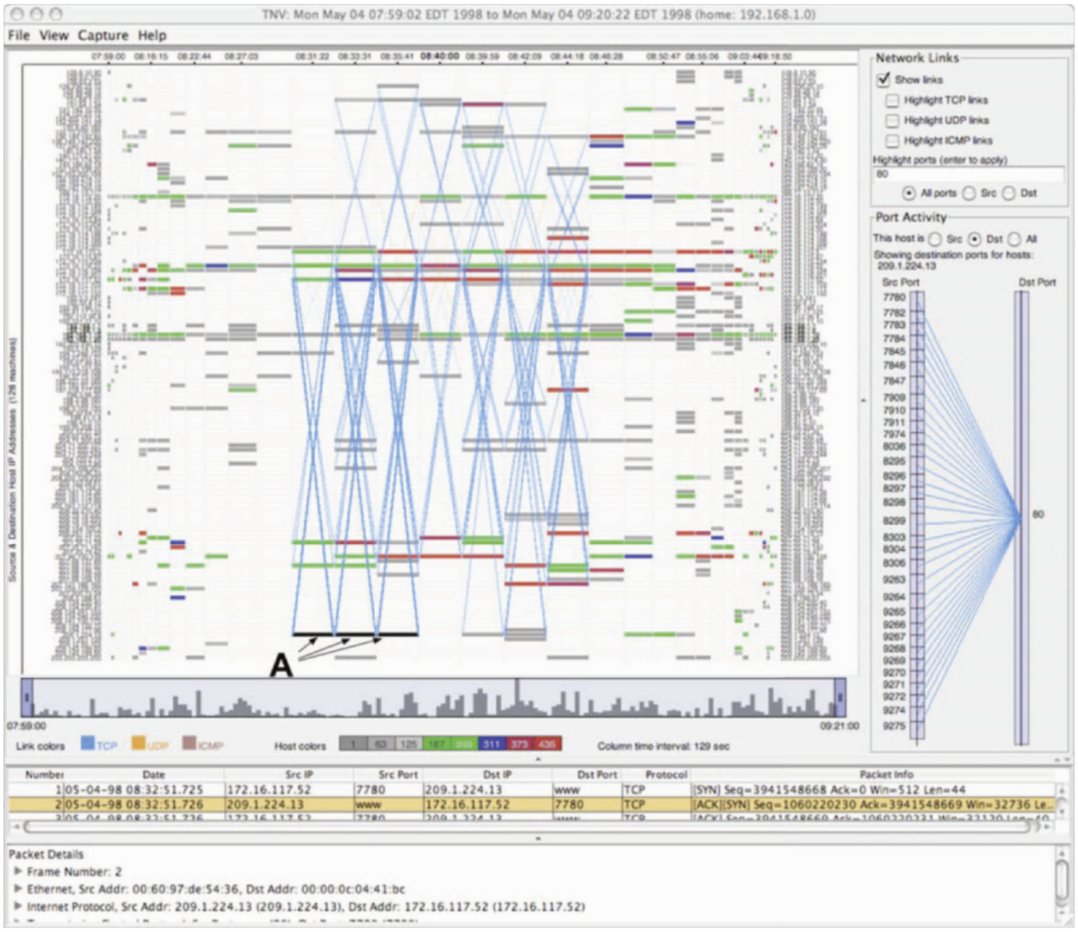
table: apache measure: sum(bytes) filter: scale: logarithmic

(a) SnortView showing alarms generated over time from multiple hosts.

(b) Erbacher's glyphs showing communications to a server over time.

(c) Hierarchical network map showing bytes transferred from external networks.

Figure 2.11: SnortView, Erbacher's Glyphs and the Hierarchical Network Map, reproduced from [38], [24] and [48].



TNV: Mon May 04 07:59:02 EDT 1998 to Mon May 04 09:20:22 EDT 1998 (home: 192.168.1.0)

File View Capture Help

Network Links

☑ Show links

☐ Highlight TCP links

☐ Highlight UDP links

☐ Highlight ICMP links

Highlight ports (enter to apply): 80

● All ports ○ Src ○ Dst

Port Activity

This host is ○ Src ● Dst ○ All

Showing destination ports for hosts:
209.1.224.13

Column time interval: 129 sec

Number	Date	Src IP	Src Port	Dst IP	Dst Port	Protocol	Packet Info
1	05-04-98 08:32:51.725	172.16.117.52	7780	209.1.224.13	www	TCP	[SYN] Seq=3941548668 Ack=0 Win=512 Len=44
2	05-04-98 08:32:51.726	209.1.224.13	www	172.16.117.52	7780	TCP	[ACK][SYN] Seq=1060220230 Ack=3941548669 Win=32736 Le...
3	05-04-98 08:32:51.726	172.16.117.52	7780	209.1.224.13	www	TCP	[ACK] Seq=3941548669 Ack=1060220231 Win=32120 Len=40

Packet Details

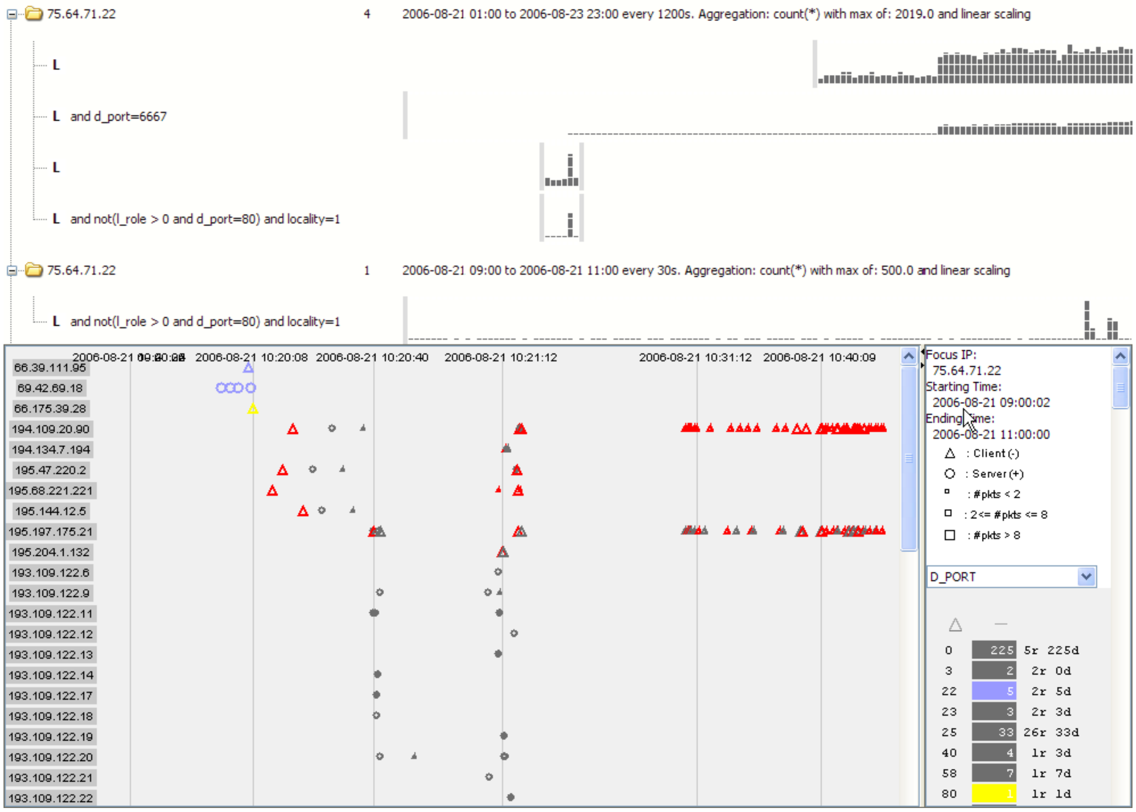
▶ Frame Number: 2

▶ Ethernet, Src Addr: 00:60:97:de:54:36, Dst Addr: 00:00:0c:04:41:bc

▶ Internet Protocol, Src Addr: 209.1.224.13 (209.1.224.13), Dst Addr: 172.16.117.52 (172.16.117.52)

Figure 2.12: TNV showing communicating hosts over time with associated packet details, reproduced from [27].

Many visualizations focus on communication changes over time. Goodall et al.'s Time-based Network traffic Visualizer (TNV) [27], rendered network packets as blocks by IP address and timestamp (on the x- and y-axes respectively), colour-coded by number of packets transmitted, and showed communication between hosts as lines (web traffic in Figure 2.12). It supported the focus + context technique of widening nodes at the time period of interest (near the centre column) and multiple levels of detail (including port activity to the right, and individual packet details to the bottom) for selected hosts and intervals. This allowed for viewing communications in their historical context (via the histogram showing all communications that facilitates navigation to different time periods) and assisted in identifying abnormal activity. However, comparing attributes of multiple hosts in different time periods was not possible.



75.64.71.22 4 2006-08-21 01:00 to 2006-08-23 23:00 every 1200s. Aggregation: count(*) with max of: 2019.0 and linear scaling

L

L and d_port=6667

L

L and not(l_role > 0 and d_port=80) and locality=1

75.64.71.22 1 2006-08-21 09:00 to 2006-08-21 11:00 every 30s. Aggregation: count(*) with max of: 500.0 and linear scaling

L and not(l_role > 0 and d_port=80) and locality=1

Focus IP:
75.64.71.22
Starting Time:
2006-08-21 09:00:02
Ending Time:
2006-08-21 11:00:00

△ : Client (-)
○ : Server (+)
▫ : #pkts < 2
□ : 2<= #pkts <= 8
□ : #pkts > 8

D_PORT

0	225	5r 225d
3	2	2r 0d
22	5	2r 5d
23	3	2r 3d
25	33	26r 33d
40	4	1r 3d
58	7	1r 7d
80	1	1r 1d

Figure 2.13: ISIS showing multiple stages of an investigation and the resulting event plot of communication over time, reproduced from [55].

Phan et al.'s ISIS [55] also highlighted temporal relationships by using progressive multiples of timelines and event plots, where analysts progressively viewed timelines of network activity for hosts that were related in some way to an individual focus IP (75.64.71.22 in the example given in Figure 2.13 with five timelines in two queries). Timelines showed packets transferred against time, and all timelines were visible at once to provide historical context.

The event plots were scatterplots with glyphs for different events, and allowed the analyst to determine activity patterns across hosts and how they changed over time. The event plot in Figure 2.13 shows communications from the compromised focus IP where the intruder logged in via SSH (the traffic represented in blue in the first two rows) and installed an unauthorized IRC server which became active soon after (the large volume of traffic in red).

As ISIS could only render single attributes over time per host, multiple attribute comparison from a host e.g. comparing activity on different ports, was not possible.

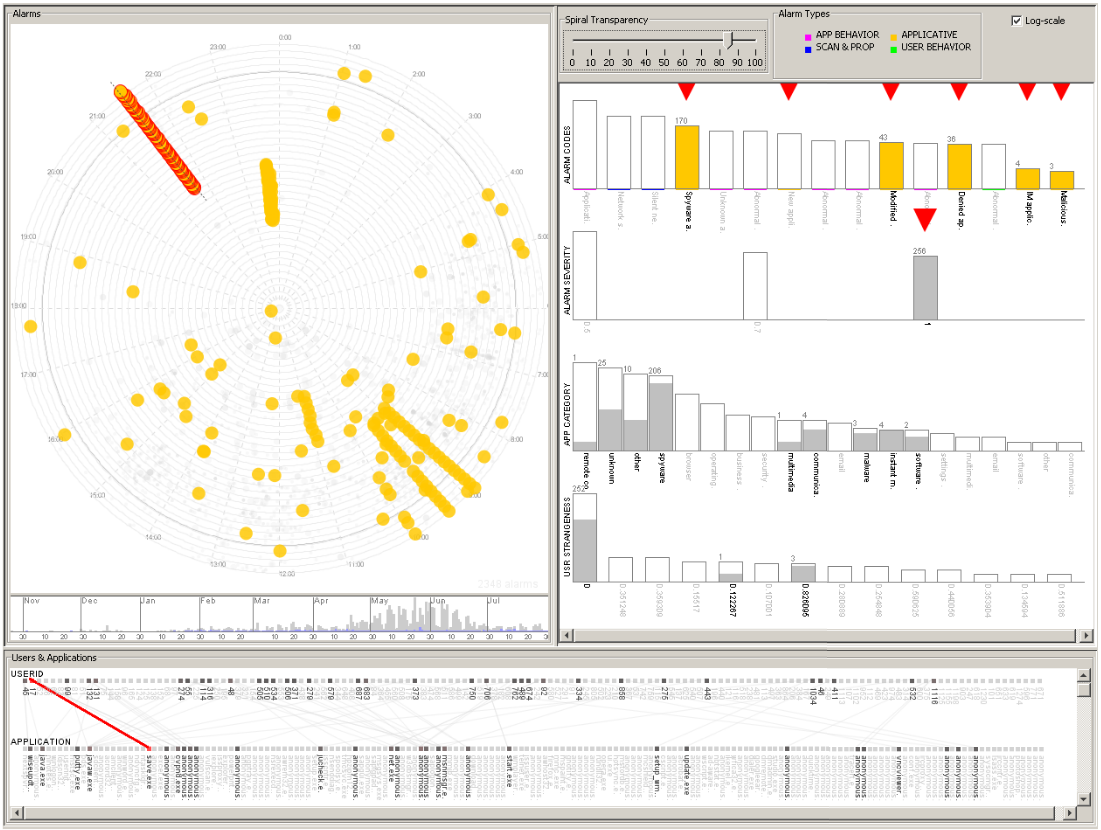


Figure 2.14: Spiralview showing a number of alarms generated over time together with associated data, reproduced from [9].

Bertini et al.'s SpiralView [9] emphasized not only how IDS alarms were generated over time, but how they were related to network resources (users, hosts or services). Its primary goal was visualizing the evolution of a changing network and assisting in network administration. Alarms within a 24 hour period were rendered in a circle (those occurring at midnight at the topmost point, and 6pm at the lowermost), and time periods were placed in chronological order, with older periods being closer to the diagram centre and newer periods closer to the outer diameter. This allowed for periodic behaviour to be seen in circle sectors, which is useful for alarms happening at particular times of day.

Alarm type and severity were communicated with node colour and size respectively, and the visualization allowed for zooming, dynamic alarm filtering and selection of visible alarms. A subdiagram showed relationships between any pair of users, hosts and applications, and highlighting a group of alarms updated related histograms

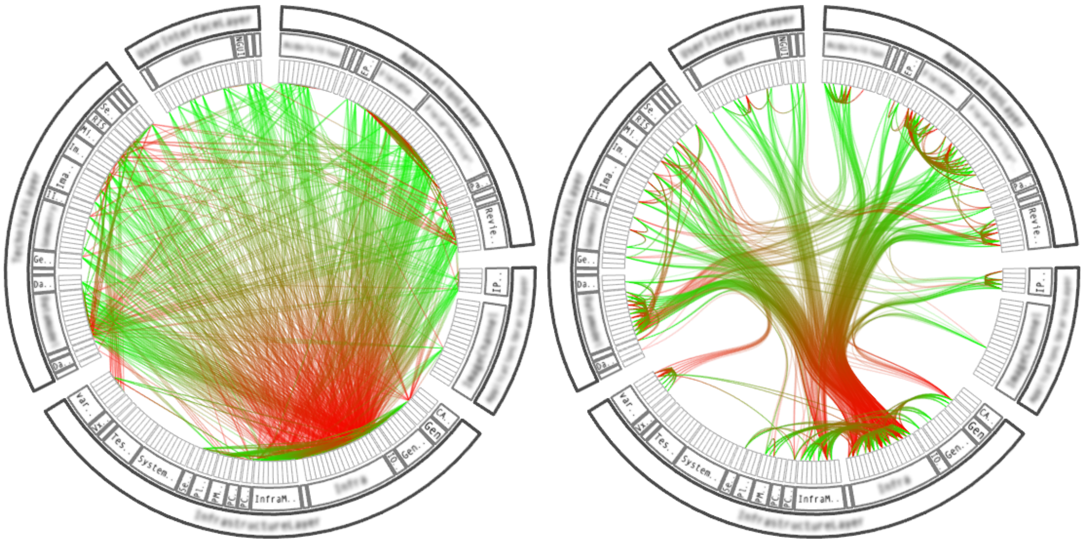


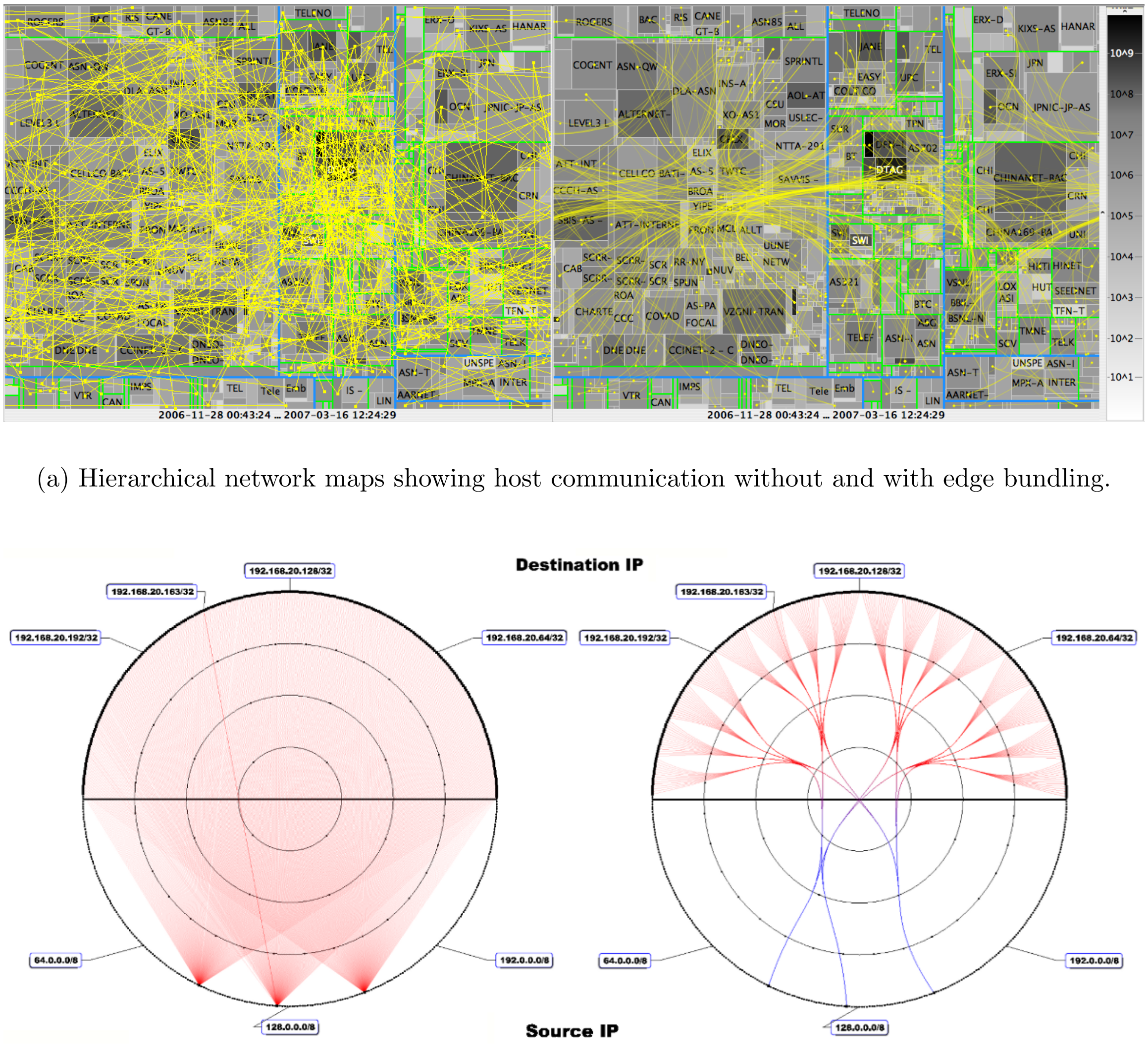
Figure 2.15: Edge bundles where connecting lines represent procedure calls between modules, reproduced from [32].

and user, host, or application pairs.

We see in the SpiralView of Figure 2.14 a large number of alarms characterized by the many gray dots, and a particular alarm type occurring frequently, and at particular times in a day (shown in yellow with the spikes around 9am and 9:30pm). Related categories for this alarm type are also shown in the histograms to the right in yellow. Highlighting the subset recurring at 9:30pm shows that they were all generated by the same user and application (denoted by the single red line in the user-application line graph). Alarm occlusion was unfortunately possible in this spiral diagram.

Many visualizations display entity relationships using connected lines, a process that has been shown to produce confusing results when multiple lines overlap, especially with basic parallel coordinate plot visualizations e.g. Ball et al.'s VISUAL [6]. Holten's edge bundling [32] solved this problem by modeling B-spline curves on the polylines produced by the vector data points. This resulted in lines connected to similar entities being bundled together, in the same manner as cable ties. Visual clutter then can be substantially reduced while emphasizing relationships between groups of related nodes. Bundling lines representing function calls between software modules highlight those modules that are called most often, as seen in Figure 2.15.

This technique has been used to great effect in many visualizations where relationships are overlaid onto existing diagrams (e.g. Mansmann et al.'s HNMap was augmented to show host communications between continents, countries, autonomous systems (AS) and networks [47] as shown in Figure 2.16(a) where high volume AS to AS communications are highlighted), and where hosts in two groups were aligned around the half-circumferences of a circle and network flows between the groups were represented as edge-bundled lines in Taylor et al.'s FlowBundle [69] (line thickness and colour were used here to represent activity volume and direction, and nodes were arranged on the edge of the circle, as seen in Figure 2.16(b)).



(a) Hierarchical network maps showing host communication without and with edge bundling.

(b) FlowBundle showing host communication with and without edge bundling.

Figure 2.16: Two examples showing edge-bundled network communications, reproduced from [69] and [47].

2.3 Limitations of Prior Approaches

We have seen how many research efforts solve specific analysis problems using various visualization techniques. In conducting this survey, it became apparent that some shortcomings were not sufficiently addressed, which could possibly hinder the use of those visualizations in other areas, or force analysts to use third-party or in-house tools to preprocess their data before it is visualized.

While it is difficult to say whether the surveyed visualizations are generalizable beyond a particular intended purpose, we surmise that it would be mentioned if such were the case (either explicitly by describing them as generalized tools, or implicitly in discussing their use with other types of data).

We have also seen instances where limited drill-down abilities are available and analysts may not have access to base data that would offer insight into a group of aggregate data. Content layout and label readability are often areas of concern as well, where item occlusion and incorrect sizing can severely reduce the effectiveness of the visualization.

Multiple focus + context techniques were also described, ranging from the simplest (SolarPlot's sector zooming) to the more complex (multiple regions of interest via edge dragging with InterRing), with one focusing strictly on leaf nodes of interest (Sunburst). A simpler mechanism requiring fewer mouse clicks should suffice in most cases, such that the RSF diagram properties and context are preserved.

And finally, the majority of visualizations surveyed are limited to showing just two attributes at a time (with attempts at showing a third in the Spinning Cube of Potential Doom unnecessarily increasing the amount of interaction required), and in those cases did not incorporate any present data hierarchy. Concurrent visualization of up to three attributes in the context of the data hierarchy would enable greater insight to be obtained in a shorter period of time.

Chapter 3

DataBurst

The DataBurst visualization builds on traditional radial space-filling diagrams by facilitating novel methods of data exploration and providing not only a high-level view of all input data at once, but the relationships between up to three associated numeric attributes. It was developed in C++ as a plugin to the open source FloVis framework [69] using freely available libraries including wxWidgets, OpenGL and FreeType [1].

This chapter describes the DataBurst visualization pipeline in depth, beginning with the input data format, followed by a discussion on how data attributes can be visually represented and compared, and ending with descriptions of available methods of working with large volumes of data.

3.1 Motivation

Three main issues have motivated the development of DataBurst from the shortcomings observed in the literature review.

The first is the need for new interactive techniques when working with large volumes of data such that an analyst can move smoothly from an overview of all data to a more detailed view of a subset of the data, without losing overall context in a radial space-filling diagram. This will be accomplished through the novel implementation of Hyperbolic Distortion and TearAway subdiagrams.

The second issue is the fragmented and highly specialized visualizations available for data analyses thus far, which have been created for specific problem areas and make use of particular datasets. A flexible and extensible tool is therefore needed, which can visualize potentially any constant-depth hierarchical data for the purposes of analysis.

[1] `www.wxwidgets.org`, `www.opengl.org` and `www.freetype.org` respectively

The third and final issue is the need to view three numeric data attributes simultaneously in the context of the data hierarchy - a limitation that forces the analyst to visually and mentally compare multiple visualization instances (or worse yet, from memory) in order to investigate more than two attributes.

3.2 Input Data Format

The underlying input data rendered by DataBurst takes the form of the classic tree structure representation, which contains a single root, and all nodes are the parent of one or more child nodes (except the leaf nodes at the lowest level). Using graph theory terminology, it is a connected and directed graph where all nodes save the root have one in-edge, and all nodes save the leaves have one or more out-edges.

Only constant depth trees are currently considered, as these constitute the majority of use cases where we would like to compare and contrast a number of leaf edges that are all at the same level i.e. they are all related to the same constant number of parents. It should be noted that a variable depth tree can be simulated by assigning a number of dummy parent nodes to all leaf nodes not at the lowest level until all leaves are at the same level.

Currently, input data is submitted via flat, pipe-delimited text files, where each line contains a fixed number of alphanumeric elements. If any one line of data contains more or fewer elements, this would produce a variable depth tree, which is not natively handled by DataBurst. It is up to the user to ensure that the datatypes of the elements in each column are all the same type.

The example data used in this section originated from census data of three major english speaking countries - Canada, United States of America and the United Kingdom, and were sourced from the freely available online resources: Statistics Canada, the US FBI's Uniform Crime Reporting Program and the UK's Office of National Statistics [2]. The data chosen in this example contain the unemployment rates, population densities (per square kilometer) and crime rates (per 100,000 persons) in major areas of those countries.

One would be motivated to capture data of this kind in order to compare different statistics at both the country and area levels, and in this case we are interested in

[2] `www.statcan.gc.ca`, `www.fbi.gov/ucr/ucr.htm` and `www.statistics.gov.uk` respectively

any relationships between unemployment and crime. Other interesting statistics like household income, education level and economic activity would be useful ones to include here, but are not easily included for several reasons (listed below).

It should be noted that an analysis of this kind should not be taken lightly as only like statistics should be compared e.g. crimes considered should all be of the same type (all recorded violent non-household-related crimes in this case), and recording practices, sampling frequencies and classification schemes may vary greatly from country to country and even area to area within a country. The unemployment or crime rate may also be more indicative of the population in an area than the number of unemployed or crimes committed there e.g. one area's crime rate can be 10 times lower than another if they both have the same level of criminal activity and the former is 10 times more populous.

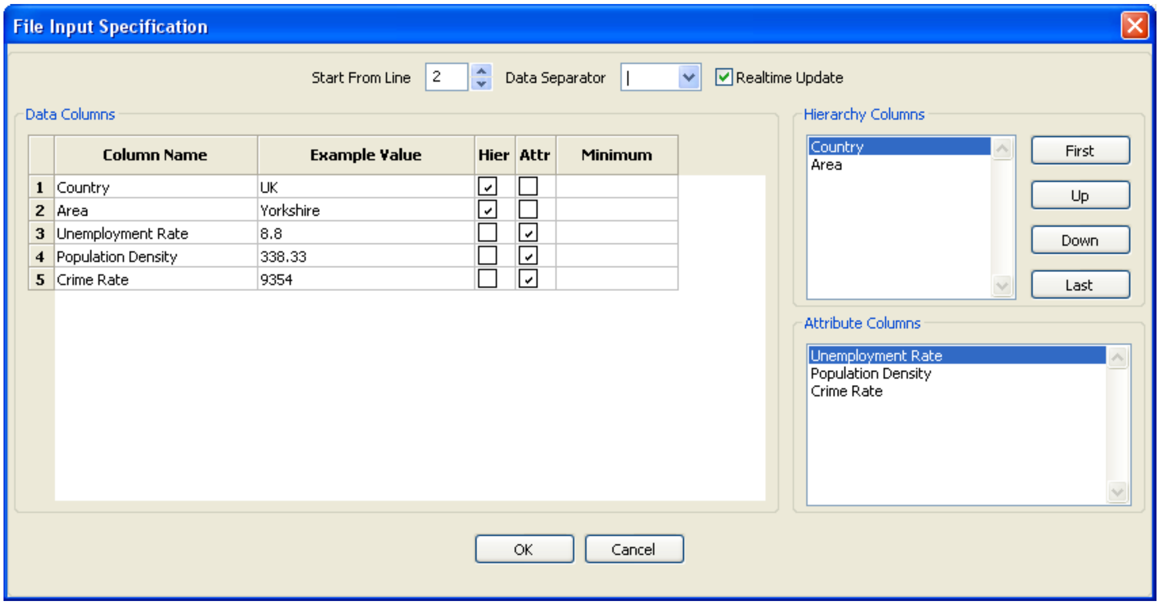
These reasons, together with the fact that some data were only available for 2008 while others only for 2009, allow this data to be useful in demonstrating the visualization, but not accurate or complete enough for a case study and thorough analysis. We make the assumption that statistical changes between the two consecutive years are not drastic.

We aggregate some areas only to ensure that nodes are large enough to produce legible labels in the sample diagrams solely for the purposes of demonstration e.g. averaging the results of North East and North West England into a single North England entry, and doing the same for all 50 of the United States into the four geographic regions used by the United States Census Bureau [3]. One may potentially not want to lose such detail in a real-world analysis.

A sample of the census input data is included below:

```
Canada|Atlantic|11.4|13.615|887.25
|Quebec|8.5|5.76|756
|Ontario|9.0|14.29|756
USA|Midwest|5.14|40.2|343.1
|Northeast|5.47|200.41|282.4
|South|5.5|281.9|565.53
|West|5.23|23.05|394.08
```

[3] http://www.census.gov/geo/www/us_regdiv.pdf



Start From Line 2 Data Separator | ☑ Realtime Update

	Column Name	Example Value	Hier	Attr	Minimum
1	Country	UK	☑	□	
2	Area	Yorkshire	☑	□	
3	Unemployment Rate	8.8	□	☑	
4	Population Density	338.33	□	☑	
5	Crime Rate	9354	□	☑	

Hierarchy Columns: Country, Area

Attribute Columns: Unemployment Rate, Population Density, Crime Rate

Figure 3.1: DataBurst's file input specification dialog.

Elements in each line of input data are categorized as hierarchy elements or data attributes. The former are used to place the node represented by that line of input, within the defined hierarchy, while the latter are positive numbers associated with that node. Empty elements in a particular line of input are interpreted as a repeated value from the last entered element in a previous line (in this case, the Canada value would be carried down to the following two lines). This reduces the size of input data files with complex hierarchies, and speeds the loading of this data at startup.

The column labels and data hierarchy are defined when the user first loads the input data into the visualization, using the dialog shown in Figure 3.1. The user provides column labels, selects the columns that are to be included in the hierarchy or as data attributes and enters lower limits of numeric values to be visualized in the Data Columns section e.g. to only show unemployment rates above 5%. Different lines of input data can be viewed in the Example Value column using the Start From Line spinner at the top of the dialog. The unique list of non-alphanumeric characters in the first line of input data are presented to the user in the Data Separator combo box, so that any one can be selected if the default pipe-separator was not used.

Selected hierarchy columns can then be dynamically reordered to produce the desired hierarchy in the Hierarchy Columns section. In Figure 3.1 we see the first two columns are included as hierarchy elements, as we would like to see all areas within each country. The remaining elements are included as data attribute elements, and are listed in the Attribute Columns section. The data hierarchy and data attribute definition are collectively referred to as metadata.

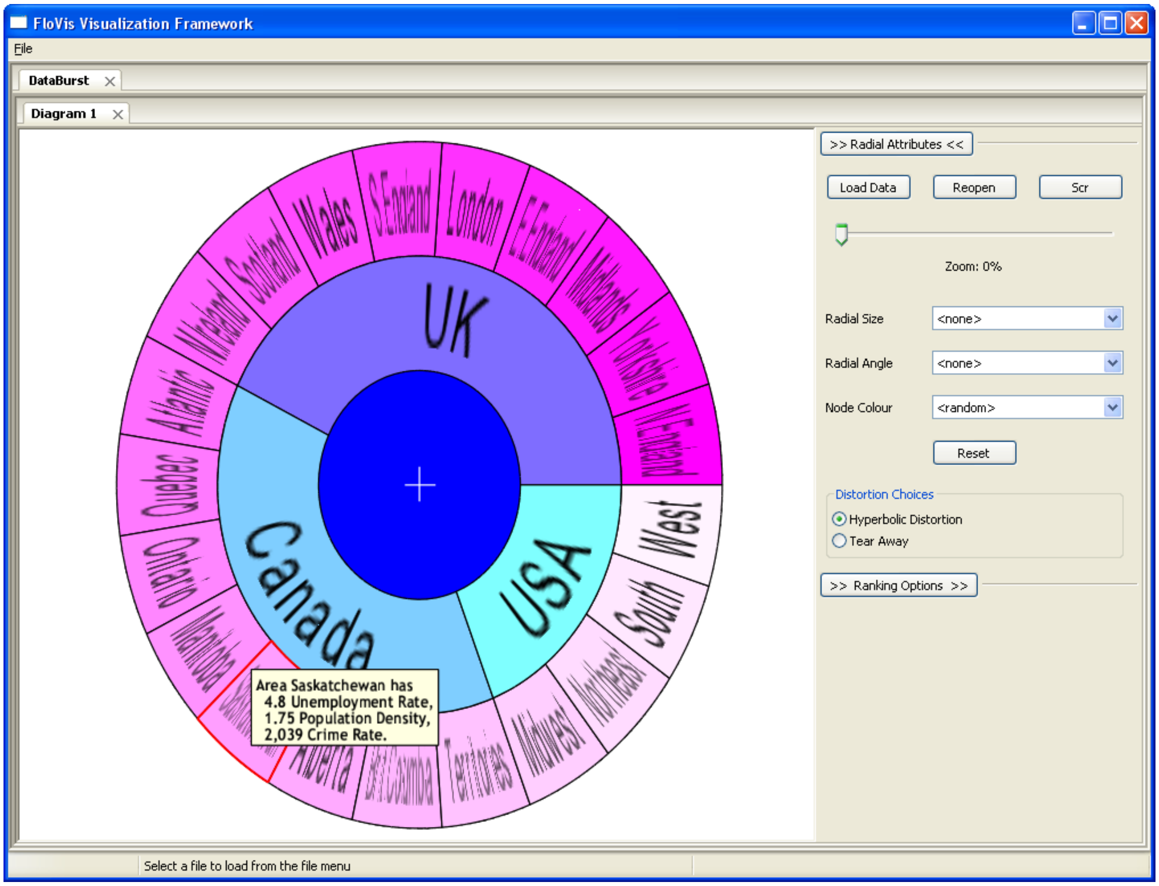
As in most instances a large volume of data would be present in the input file, the metadata is maintained by DataBurst and stored in a separate file (with the same name and the extension txt_meta) so that it can be dynamically updated from the interface without having to reprocess the input data in any way. An update that is often performed is the reorganizing of the defined data hierarchy e.g. visualizing the data grouped by country and then year to see individual changes over time, or by year and then country to see aggregate changes over time.

In instances where summary data are visualized, the detailed data can be stored in another file (again with the same name but with the extension txt_detail), such that the component elements of particular groups of interest identified in the visualization can be retrieved e.g. clicking on the Yorkshire node could load statistics for Yorkshire cities into the clipboard if present. This can be critically important in many analyses where certain groups of data are outliers and analysts need to investigate the base data to determine the cause.

3.3 Basic Radial Structure and Interface

The default rendering of a DataBurst places all leaf nodes on the outermost ring. Leaf nodes are assigned the same radial angle (equal to *360/n*, where n is the number of leaf nodes, such that they completely fill the outermost ring), and the same radial size (100 pixels in this case). The former is an angular measurement around the diagram centre, and the latter is a distance measurement from the diagram centre. Parent nodes are then rendered such that their radial angle is equal to the sum of the radial angles of their children, and their radial size is the average of the radial sizes of their children, and they are one level above their children (and therefore closer to the diagram centre). Note that the default colour scheme simply differentiates diagram nodes and does not initially communicate any meaning.

Render order takes place counterclockwise, starting by default from the zero degree



Area Saskatchewan has
4.8 Unemployment Rate,
1.75 Population Density,
2,039 Crime Rate.

Radial Size <none>

Radial Angle <none>

Node Colour <random>

● Hyperbolic Distortion
○ Tear Away

Select a file to load from the file menu

Figure 3.2: DataBurst interface and diagram for given sample data. Saskatchewan highlighted.

angle (the horizontal line that moves to the cardinal direction east from the diagram centre, or the positive x-axis using graph terminology). Figure 3.2 shows the rendered output of the example census data described in the previous section, where the N. England and West USA nodes are first and last nodes rendered in the outer ring.

The user can move the mouse cursor over any node and receive pertinent metadata, input data values and individual or aggregate attribute values. The first describes the type of data at the particular level (depending on the defined hierarchy, Area in the highlighted case above), and the last is either the list of all attribute values for a leaf node, or the total of all attribute values for a group of related nodes. The former is shown in Figure 3.2 while the latter would be seen for any parent node (statistics at the country level). The first attribute is always the total group count for parent nodes (USA would have a group count of 4), while the remainder are the data attributes selected by the user (the three statistics in this case).

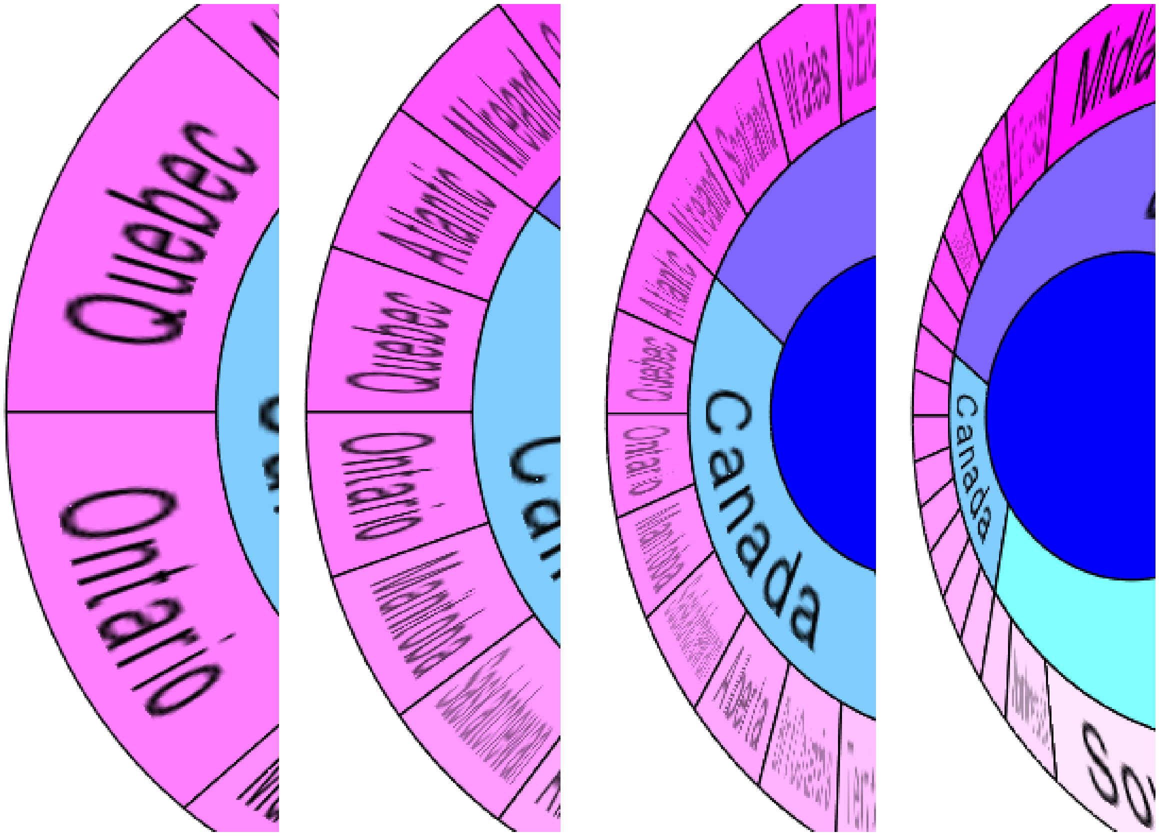


Figure 3.3: Rendering labels of nodes with differing radial angles (using hyperbolic distortion in this case).

Node labels are rendered character by character as textures, always along the length of the node and are scaled down when their width exceeds their node's width. As texture operations are the most time consuming stage in the rendering process, a node label is only rendered if its width is 1/4 or more of its original value (ensuring that only readable labels are rendered). A node label's opacity is also reduced when its size is reduced, allowing it to progressively fade out the more it is deemphasized, until it eventually disappears. Node labels are also flipped if their centres are in the upper half of the diagram to increase readability.

These rendering techniques are shown in Figure 3.3 using hyperbolic distortion (formally defined in the Focus and Context section) which allows us in this case to increase the number of nodes within one particular area. Aliasing issues with clusters of node edges are handled by making those edges of nodes having a radial angle of less than 0.5 degrees transparent.

3.4 Leveraging Radial Attributes

While RSFs show parent-child relationships effectively, many analyses require the comparison of particular attributes of data in the context of the data hierarchy. Using the census data example, we may not only want to quickly compare the crime rates between areas, but also between countries.

Data attributes can be individually assigned to any of the three available radial attributes (radial size, radial angle and node colour). Assignments are made from any of the three respective combo boxes in the Radial Attributes section of the main interface (see Figure 3.2).

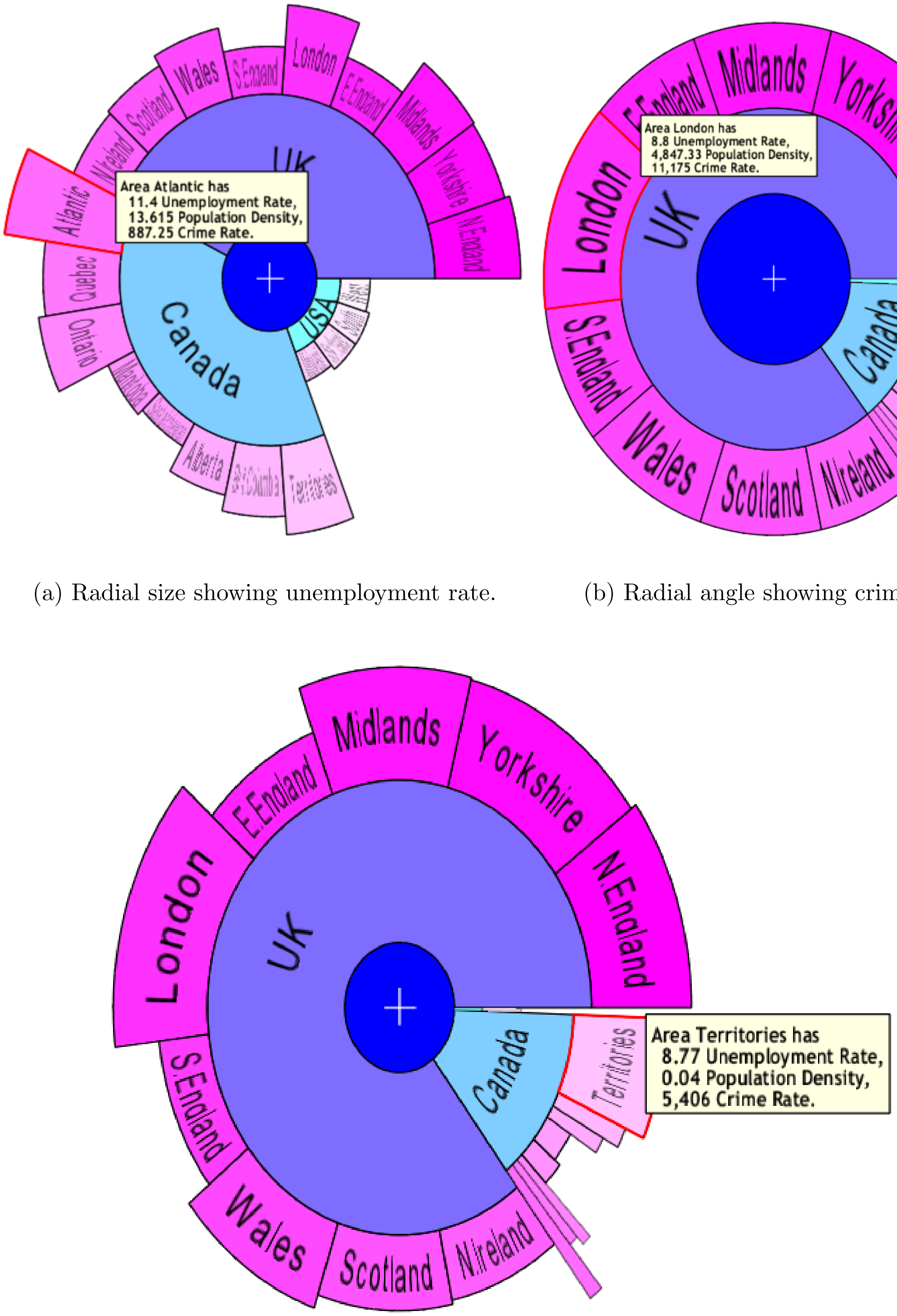
3.4.1 Radial Size

When attribute values are assigned to radial size, their relative differences are visualized by scaling node radial sizes (the pixel distance between the closest edges to and furthest edges from the diagram centre) based on their attribute values. This is done in two passes, where the radial sizes of all leaf nodes are set to their attribute values, and then scaled such that the lowest value becomes 50 and the largest becomes 150 (ensuring that text is not clipped within smaller sized node).

The radial sizes of all parent nodes are then calculated as the average of the radial sizes of their immediate children. This assignment increases the prominence of nodes with larger attributes, as those nodes are pushed towards the outer diameter of the diagram and become more separated from the other nodes closer to the centre.

Unemployment rate is assigned to radial size in Figure 3.4(a), and we see that Atlantic Canada has the highest unemployment rate, followed closely by the Canadian territories and many parts of the UK. Many areas appear to have low scores including Saskatchewan and Manitoba in Canada, and all areas of USA.

The low USA per-region unemployment statistics here however have arisen from the aggregation practice used to reduce the number of nodes in that country from 50 to 4. On checking the per state USA statistics, we see several with high unemployment rates (e.g. Michigan stands at 8.3% in the Midwest, which has 5.1% overall), but they are grouped with other states having sufficiently low unemployment rates (e.g. South Dakota, also in the Midwest, at 3.0%).



(a) Radial size showing unemployment rate.

(b) Radial angle showing crime rate.

(c) Both radial size and angle showing unemployment rate and crime rate respectively.

Figure 3.4: DataBurst diagrams for given sample data when assigning data attributes to radial size and angle.

3.4.2 Radial Angle

When attribute values are assigned to radial angle, their relative differences are visualized by scaling node radial angles based on their attribute values. This is done in two passes, where the base radial angles of all leaf nodes are incremented by their attribute values, and then scaled such that their sum equals 360. The latter process can be thought of as a type of angular normalization that ensures the radial structure is preserved. The radial angles of all parent nodes are then calculated as the sum of their children radial angles, which can be thought of as rolling the attribute values up the hierarchy (effective at increasing the label readability of nodes of interest).

Crime rates are assigned to radial angle in Figure 3.4(b), and we see they are consistently high in all areas of the UK, peaking in London with over 11,000 crimes committed per 100,000 persons. The Canadian Territories have the highest crime rates in that country, and the four US areas rank the lowest overall and are barely visible. This result is verified as per state crime rates in the US are all below 1,000.

We see that leaf nodes having large radial angles emphasize those leaf nodes in addition to their parents i.e. the overall ranking of the 3 countries can be quickly done. Note that while this notion supports many types of analysis (where we want to visually identify nodes with large values and can safely ignore low values), it is currently not possible to do the converse, where small values produce large nodes and vice versa. We can still view these small nodes with the interactive techniques discussed in the next section, but this can be emulated by inverting the numerical data in the data preprocessing stage before visualization. Measures to handle this within the visualization will be considered in future work.

Nodes having both large radial sizes and angles are then doubly emphasized, as the area they occupy can be larger than through any single data attribute assignment (larger area here implies a node with two large data attributes). Figure 3.4(c) shows the two previous assignments simultaneously and highlight that unemployment and crime rates can be quite high in some areas e.g. London UK and Canadian Territories dominate the smaller nodes of lesser interest, while some places suffer only high unemployment (e.g. Ontario, Quebec and Atlantic Canada).

Interestingly we see several areas with low amounts of each rate (e.g. the four US regions), and no instances where crime rates are high while unemployment rates are low. Looking strictly at these two statistics then would not allow reliable identification of any causal relationship between them - we would like a way to visualize all three statistics simultaneously (or perform a quantitative analysis such as linear regression).

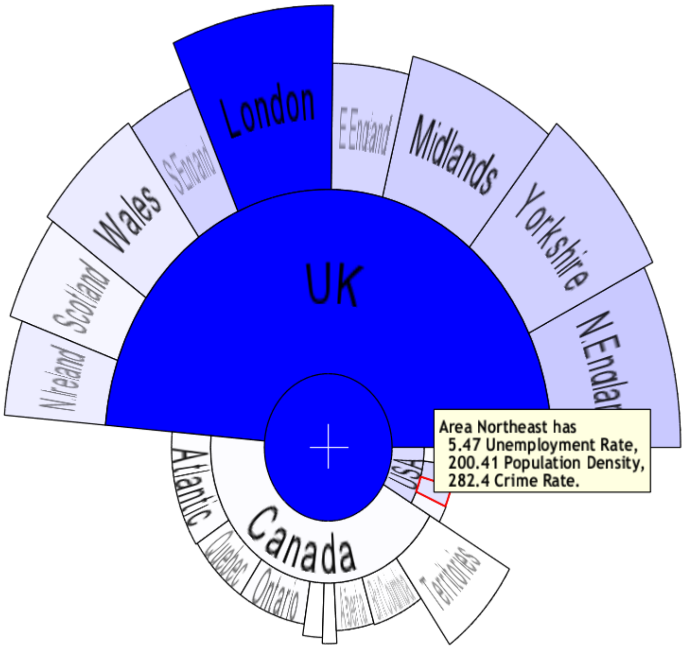
3.4.3 Node Colour

DataBurst initially assigns node colours so that they are different shades of blue and red depending on the ring the node is in, and its position within the ring. Colours are simply chosen such that nodes can be more easily differentiated from their neighbours and do not connote any meaning with respect to the input data.

When attribute values are assigned to radial colour, their relative differences are visualized by scaling node colours based on their attribute values. This is done in two passes, where the blue opacity for all nodes is set to their attribute values, and then scaled such that for the maximum value the opacity is highest (darkest blue), and lowest for the minimum value (no node colour). Note that this is done at all levels such that it is possible for the darkest nodes at the leaf level to have parents that are not the darkest at the next level. This assignment is particularly effective at identifying parent nodes of interest (those parent nodes having larger attribute values than their siblings at that level).

Leveraging all three radial attributes enables the comparison of three data attributes simultaneously, such that relationships between any pair or all three can be visualized. Figure 3.5 shows how crime rates, unemployment rates and population densities are related after they have been assigned to radial size, radial angle and node colour respectively (note that we swapped crime rates and unemployment rates from Figure 3.4(c) to produce a diagram where nodes in the USA are more visible).

We see that population densities are highest in the UK (peaking in London) and the South and North East regions of the USA (all above 200), with Canadian population densities being quite low in comparison (peaking at 14.3 in Ontario). In the absence of any other data, this may lead one to conclude that there exists a relationship between higher population densities and higher crime rates, and that unemployment rates do not affect crime. A more exhaustive investigation into other measurable statistics for additional time periods would be necessary to draw any reasonably accurate conclusions.



Area Northeast has
5.47 Unemployment Rate,
200.41 Population Density,
282.4 Crime Rate.

Figure 3.5: DataBurst diagram for given sample data when assigning crime rates, unemployment rates and population densities are assigned to radial size, radial angle and node colour. UK and London have the highest population densities in their respective rings, and have the highest opacities.

3.5 Focus and Context

When working with large volumes of data, the ability to focus on smaller groups of data without losing overall context is essential. To this end we provide two novel means of interaction that are simple to perform and allow analysts to look more closely at the parts of the diagram of interest, while maintaining context within the original hierarchy. The first makes use of hyperbolic distortion, which can informally be described as an interactive fisheye lens technique [59], and the second allows the user to partition input data while still preserving radial attribute assignments. Note that both interactions are independent in this implementation and currently cannot be performed at the same time.

3.5.1 Hyperbolic Distortion

Hyperbolic distortion is done through the physical movement of dragging the diagram centre away from the nodes of interest, which visually emphasizes the nodes of interest (the nodes furthest from the diagram centre have their radial angles increased), while

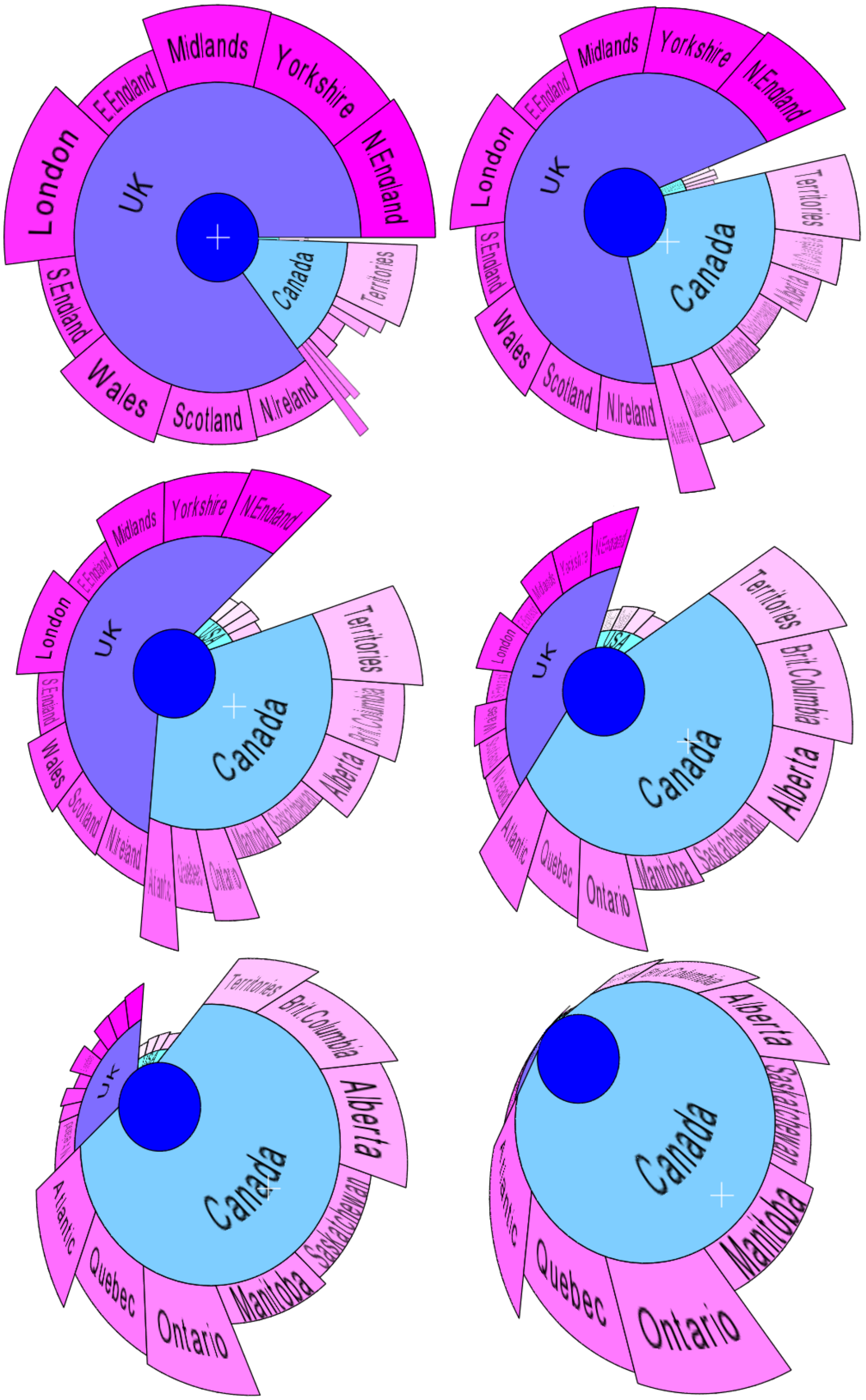


Figure 3.6: DataBurst diagram for given sample data using hyperbolic distortion to emphasize areas within Canada.

the nodes now closest to the diagram centre are deemphasized (their radial angles are reduced). Figure 3.6 shows this with the radial centre being moved progressively towards the node representing London, UK, allowing the nodes representing the various regions of Canada (and USA to some degree) to be examined.

Mathematically this effect is produced by adding the values of a Gaussian curve to the base radial angles of the leaf nodes, where the node (or two nodes) of highest interest take on the peak value, the resulting radial angles are scaled to sum to 360, and parent angles are recalculated. The start angle of the diagram, α in Figure 3.7, is also updated to seamlessly produce the desired effect (Figure 3.6 shows the diagram's start angle being progressively increased).

The displacement of the diagram centre controls the intensity of the hyperbolic distortion i.e. the height of the Gaussian. Maximum distortion is obtained when the diagram centre is moved to the outermost ring (the limit of allowed movement, as seen in the last subdiagram of Figure 3.6).

This focus method does not preserve radial angle assignments as distortion is increased i.e. when the centre is half displaced, the radial angle assignments are accounted by their base angles and the hyperbolic distortion amount equally, while at maximum distortion radial angles take their values completely from the Gaussian. This ensures that nodes of interest that dominate all others due to a radial angle assignment can be investigated together with their neighbours of lower interest when distorted i.e. that smaller nodes can still be investigated.

As high magnification can negatively impact a user's ability to track fast moving nodes [29], this distortion also ensures that focus-targeting is not diminished as a user can drag the diagram centre along the circumference to quickly browse through all nodes, but can reduce the speed of movement by moving the diagram centre inwards.

Note that radial size and node colour assignments are preserved during hyperbolic distortion, and this focus method offers the inherent ability of increasing readability of nodes of interest.

Informally, the application of the Gaussian and determination of the start angle of the diagram is given by the following C-like pseudocode, where

- x and y are the coordinates of points along the Gaussian curve (the value of x is between 0 and 1 and the peak value of y occurs at x=0.5 and is mostly determined by the displacement of the diagram centre during distortion i.e. the variable *centre_displacement*),

- zoom allows for fine control of the distortion without moving the diagram centre (ranges between 1, the default value and lowest zoom, and 100), and is controlled by a slider in the Radial Attributes section of the main interface,
- the base angle is the non hyperbolic-distorted angle of a node (equal to the normalized sum of *360/n* and the radial angle attribute, if any, where *n* is the number of nodes in the diagram) while the node angle is the distorted value,
- the angle of the diagram centre is the angle between the positive x-axis and the line, *l*, joining the original diagram centre to the new diagram centre,
- *min_emphasis_node* is the node closest to the new diagram centre, and
- all angular calculations take place counterclockwise from the positive x-axis, and the last function call splits *min_emphasis_node* into two by the line *l*, with the angle representing the portion closest to the x-axis being returned.

```
fish_eye(){
  x = base_angle_of_mid_point_of(the_first_leaf_node)/360
    + angle_of_diagram_centre()/360;
  for all leaf nodes n {
    y = gaussian(x, zoom) * centre_displacement;
    n.setangle (n.baseangle + y);
    x += base_angle_of_mid_point_of(n)/360
       + base_angle_of_mid_point_of(n.next)/360;
    if (n is closest to the diagram centre)
      min_emphasis_node = n;
  }
  normalize_leaf_angles();             // so that they all sum to 360
  diagram_start_angle = angle_of_diagram_centre()
     - fraction_of_node_closest_to_x_axis(min_emphasis_node)
     - min_emphasis_node.start_angle;
}
```

The gaussian function is defined with the variance (which controls the width of the bell portion of the Gaussian), σ^2=0.0025 as

$$gaussian(x, z) = \frac{z}{\sqrt{2\pi\sigma^2}} e^{-z(x-0.5)^2/2\sigma^2} \tag{3.1}$$

The simple example illustrated in Figure 3.7 demonstrates the algorithm given, with the line, l, described previously, seen in the DataBurst diagram on the left. We add the calculated Gaussian values to the radial angles of each of the four nodes based on the angle and displacement of the hyperbolic distortion. After normalization, the resulting distorted angles of the nodes 1 through 4 are roughly 54, 52, 56 and 198 in that order. We see that the emphasis node, node 4, has the largest angle, while the node on the opposite side of the diagram, node 2, is assigned the least.

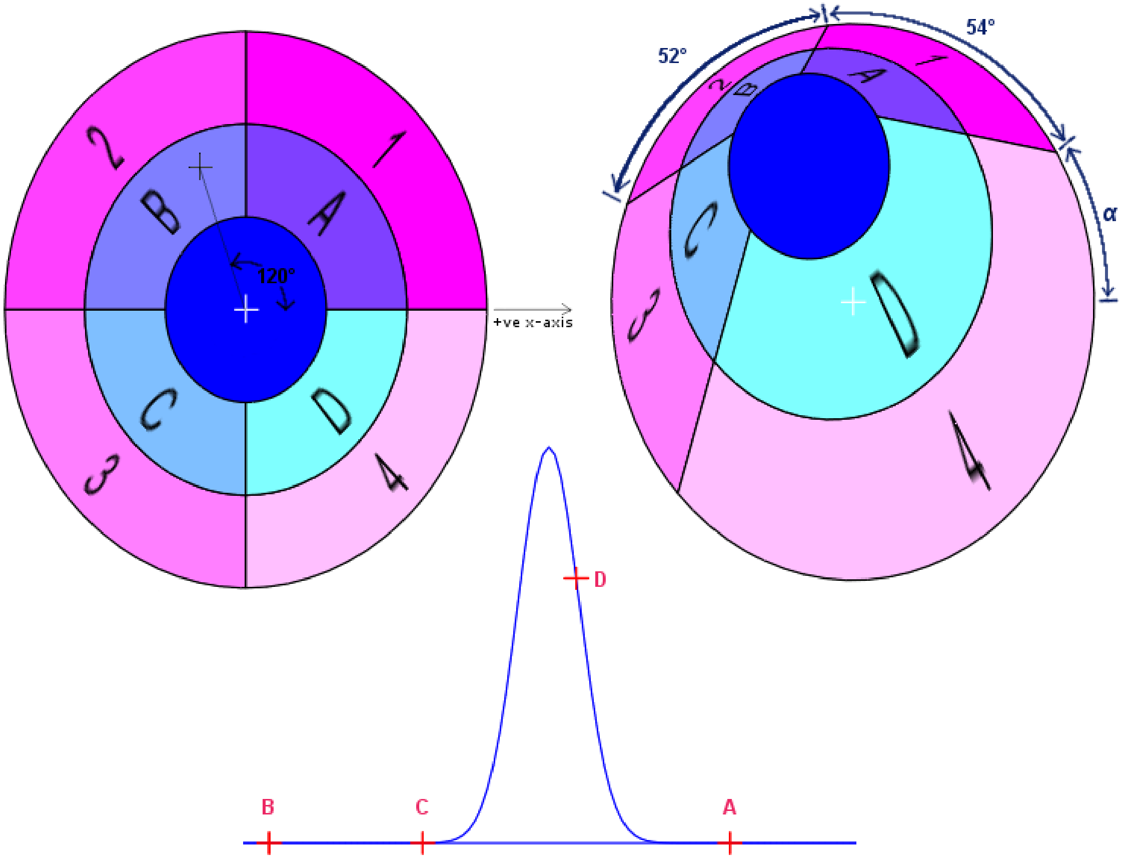


Figure 3.7: Calculations involved in performing hyperbolic distortion.

The choice of x values in the algorithm can be viewed as one that keeps the relative horizontal distances between nodes along the x-axis of the Gaussian graph proportional to the nodes' base radial angles, with the position of the 1st node being influenced by its radial angle and the distortion angle (120° in this case).

The diagram start angle is $\alpha = 120 - 30 * (52/90) - 54$ in this example.

3.5.2 TearAway SubDiagrams

In instances where particular groups of data are to be handled separately from others, a typical analysis may involve producing sets of input data and visualizing each separately. This may lead to unintentional issues such as delays in regenerating data based on multiple partitioning strategies, missing or repeated data if partitioned/merged incorrectly, and the loss of important relationships between the data across multiple partitions.

DataBurst allows analysts to work on their single, large dataset, and dynamically tear off groups of data that are not required for a particular analysis or stage of an analysis, or to increase node visibility with multiple diagrams. This increases node label readability and allows particular subgroups to become the focus of an analysis, while still enabling relative data attributes to be visualized across the complete input data through the radial diagram attributes.

The TearAway procedure is performed in real-time by clicking on any node of a diagram, and dragging away from that diagram's centre. The new torn away node or nodes are then radially expanded to create a new DataBurst subdiagram. The original diagram is radially expanded as well. This animated transition occurs gradually as the new subdiagram is moved away from the original, and both expansions continue until the new subdiagram is sufficiently far away such that no overlapping occurs and the openings in each diagram are closed.

This is demonstrated in Figure 3.8, where nodes representing USA's four regions are removed from the main diagram. It should be noted that the three radial attributes are preserved at all times in this process, and the relative sort order of nodes within a subdiagrams is the same as the original diagram.

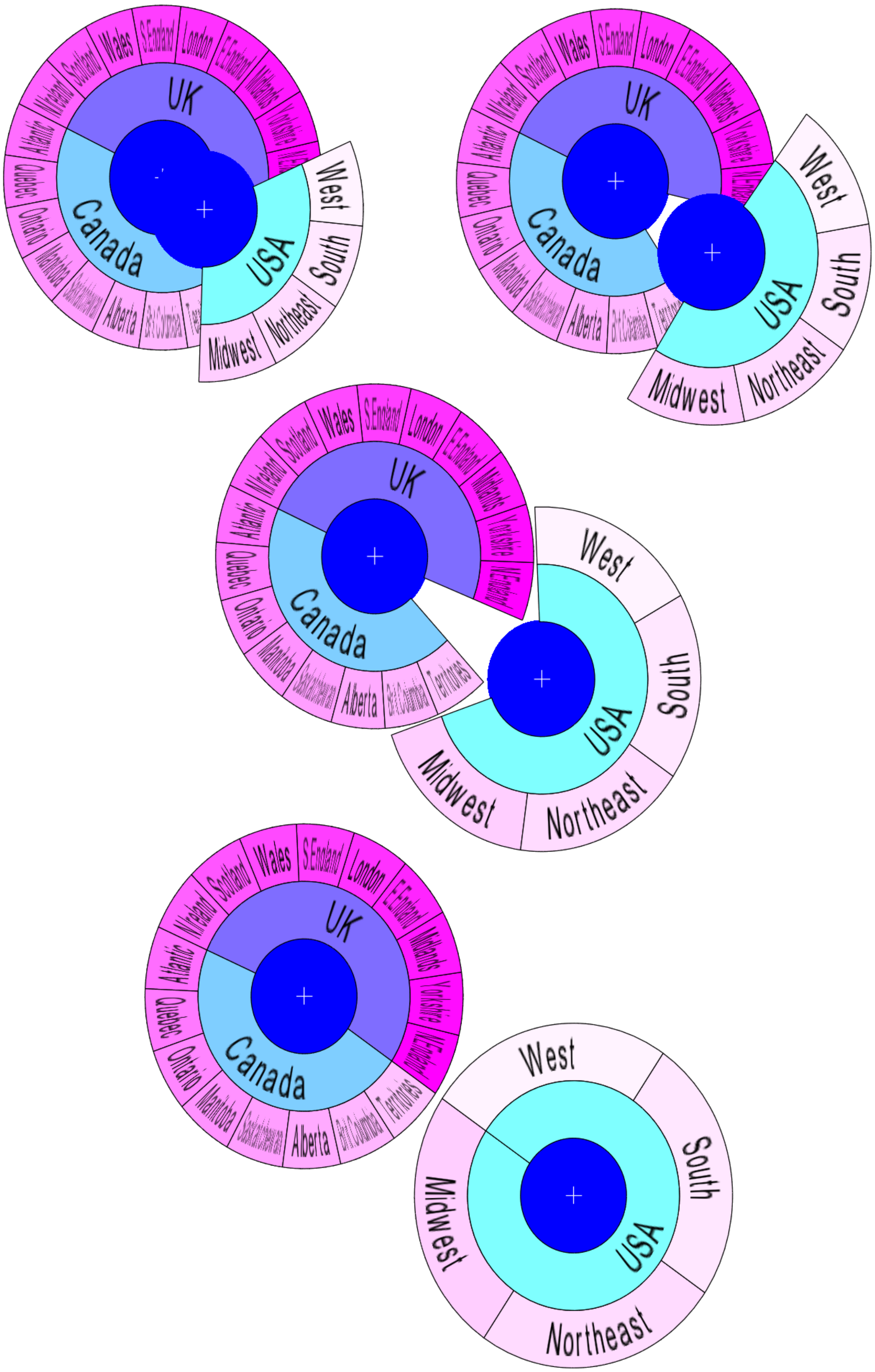


Figure 3.8: DataBurst diagram where the four nodes representing USA regions are progressively torn away from the main diagram.

Chapter 4

Case Study - Visualizing Network Flow Data

In this chapter we present detailed investigations into flow data in the area of network security, where security analysts are tasked with identifying network attacks (as they happen or in a forensics analysis) or precursor activity that may indicate an impending attack - necessary activities due to the rising cost incurred from increasingly complex intrusions. We will introduce some network security basics, discuss associated problems, and investigate how the DataBurst diagram facilitates data analysis.

4.1 Network Security

The need for continuous vigilance in the realm of computer security was noted by many, including Denning [18] who observed that numerous security flaws exist in systems that cannot all be easily identified or rectified, relying on less secure systems may be easier and less costly than updating to more secure ones, and insider information can often enable the misuse of even the most secure system. As the creation of a completely secure system is not achievable, we then look to local and network usage patterns to identify abnormal and malicious activity.

Attack identification can be generally done in either of two ways. The first involves creating a model of normal network activity such that anomalous activity falls outside that model, and the second maintains signatures of all known network intrusions and attacks, and flags any activity that matches, or is closely related. Unfortunately, both methods suffer from low accuracy as some normal activity can be flagged as intrusive, and direct attacks may go undetected due to their similarity to normal activity or dissimilarity to known attacks [3].

For further details, the background, advantages and shortcomings of many past intrusion detection systems (IDS) are discussed by McHugh in [49] and more recently in the comprehensive discussion on intrusion detection and prevention systems (IDPS) by Scarfone and Mell in [60].

4.2 Network Flow Data

We examine network flow data as input data in this case study, or more commonly, netflow data. Netflow refers to the proprietary Cisco network protocol which enables the collection of IP traffic information at the router level [1]. Netflows offer an effective means of monitoring network activity for the purposes of network security and management, which often cover the needs of most security analyses. Netflow records, or flows, are by default stored in binary data files (not human readable) to facilitate quick storage and retrieval.

Netflow data are typically retrieved by selecting the desired flows (using the rwfilter program), aggregating them if necessary (using the myriad available tools e.g. rwuniq, rwgroup, rwcount, rwsort) and returning the data in delimited, columnar text format, which can be stored in flat files, spreadsheets or uploaded to databases for future or continuing analysis. These programs are part of the SiLK [2] (System for Internet-Level Knowledge) toolset developed by CERT Network Situational Awareness Team (CERT NetSA). More detailed descriptions of these tools are given by Gates in [26].

Some sample data are listed in Table 4.1, which were generated by filtering the communications between two anonymized IP addresses for a particular date and time, collected in a live network. The final listing was produced using multiple calls to the rwfilter and rwsort programs, and using rwmatch to pair query flows to response flows i.e. to determine conversations between these hosts.

A single netflow record represents a unidirectional sequence of IP packets that share the same source and destination IP address and port (the first four columns), IP protocol (the 5th column, which differentiates TCP, UDP and ICMP protocols among others), ingress interface and IP type of service within a particular time period.

Associated with each netflow are a number of flow attributes, including the number of bytes and packets transmitted (columns 6 and 7), TCP flags (column 8) and timestamps (start time and duration in seconds or milliseconds). Different router manufacturers may choose to include these and other attributes based on the router's intended use.

[1] http://www.cisco.com/web/go/netflow

[2] http://tools.netsa.cert.org/silk/

sIP	dIP	sPort	dPort	Pro	Pkts	Byts	Flags	Start Time	Dur
IP1	IP2	1719	3306	6	3	128	FSA	2006/06/29 20:03:20	1
IP2	IP1	3306	1719	6	4	242	FSPA	2006/06/29 20:03:20	4
IP1	IP2	137	137	17	158	4629		2006/06/29 20:03:21	924
IP2	IP1	137	137	17	181	1411		2006/06/29 20:03:21	924
IP1	IP2	4892	3306	6	4	168	FSA	2006/06/29 20:03:23	3
IP2	IP1	3306	4892	6	4	242	FSPA	2006/06/29 20:03:23	3
IP1	IP2	1719	3306	6	2	80	R	2006/06/29 20:03:24	0
IP1	IP2	2485	3306	6	4	168	FSA	2006/06/29 20:03:29	6
IP2	IP1	3306	2485	6	3	202	FSPA	2006/06/29 20:03:29	3
IP2	IP1	3306	2485	6	2	154	FPA	2006/06/29 20:03:35	0

Table 4.1: Sample netflows between two hosts showing source and destination IP and ports, protocol, packets and bytes transmitted, TCP flags, start time and duration.

Important uses of netflow attributes are in identifying hosts involved in communications (using columns 1 and 2), quickly determining the type of communication (using columns 3, 4 and 5 e.g. web traffic is typically transmitted via port 80) and overall behaviour of the communication (using the remainder columns). Note that user-generated data within the conversation is not specifically tracked within a netflow as this would involve full packet capture (packets however can be captured separately and linked to related netflows if necessary).

Capturing netflows at ingress and egress points in a network is then an effective means of monitoring network activity, while ensuring privacy is maintained. This is analogous to examining the frequency, source and destination addresses of emails passing through an email server during an investigation, where the email contents need not be analyzed. By aggregating netflows between hosts (both internally and externally) one can gain an understanding of communications. Numerous research efforts have examined netflow usage patterns to identify both known and previously unseen network attacks.

Some issues are inherently associated with netflow data, primarily stemming from the large volume of network data that can pass through a single collection point in a particular time interval. Network communication speed increases have historically exceeded processing and storage speed increases, where the latter is described by the ubiquitous Moore's law (processing speed and memory capacities double approximately every two years) and the former by Butters' Law of Photonics [3] (where the cost of transmitting a bit over fiber optics is halved every nine months). This causes two issues to arise, the first being the router sampling data and the second being a separation of collection, storage and analysis.

Sampling occurs when the router is under heavy load, and passes on incoming packets without processing them, allowing overflow attacks to be possible (where the router does not record enough of the attack to identify it). The modular design of a typical system using netflows also may not have the infrastructure required to allow real-time monitoring, as the processing and transferring of data from the collection point for analysis can take some time due to the volume of data. This could translate into security reports being updated in particular time intervals (e.g. every 15 or 30 minutes), during which a network intrusion could be well underway.

As these reports can be strongly time-oriented, their data can be easily communicated using line graphs, which can sometimes be difficult to accurately read for long periods of time due to compression along the time axis, and is of course limited to two or three dimensions. We argue that radial visualizations are a good fit in this instance as they can communicate multiple attributes simultaneously in addition to relationships between groups of data.

4.3 The Problem

Network flows arguably capture the minimum amount of the most important aspects of network communications. A detailed analysis of netflows can identify numerous instances of anomalous activities, and many commercially successful intrusion detection systems (IDS), e.g. Snort [39] and Bro [54], leverage this protocol in identifying a wide range of attacks, including those seeking to obfuscate their presence (e.g. by taking place over a sufficiently long enough period of time).

Network administrators therefore depend heavily on tools developed by experienced third-party network security companies. While this may appear to reduce the workload of the network administrators in an enterprise environment, updated attack signatures may not be received quickly enough with fast-spreading worms, and

[3] `http://www.tmcnet.com/articles/comsol/0100/0100pubout.htm`

an IDS may produce a prohibitively large list of suspected attacks (most of which may be false positives due to incorrectly identified normal activity) that make routine exhaustive investigation impossible.

A network administrator then is often tasked with the role of a security analyst, where a complete understanding of hardware and software available on the network is not only required, but a competent understanding of basic network attacks and best practices to perform during anomalous activities e.g. denial of service attacks, is also necessary. Network security analysts may then build islands of knowledge particular to their environment and specific tools that enable required analyses based on their experience [49]. These tools usually take as input heavily processed data that originated from netflow data.

The problem then is that with a wide range of tools existing for multiple environments, and the varying knowledge level of security analysts, it is difficult to produce a single tool with specific functionalities that satisfy all possible needs in any situation. A modular, general purpose approach is therefore necessary.

4.4 The Solution

Given the varying needs and knowledge levels of security analysts, one solution is to provide an intuitive data visualization and flexible interface that allows any type of problem to be formally defined, data extracted and processed using available tools at hand, and the results interpreted in the context of the analysts knowledge base. We propose that the combination of DataBurst and the SiLK tools can enable network security analysts to leverage their knowledge to search for and identify anomalous activities and attacks.

A complete DataBurst solution may entail a Unix server that stores all netflow data captured at a router, which are then processed at intervals by custom SiLK scripts on an analysis server, which would upload all aggregate data to a database. The analyst would then render numerous DataBurst diagrams from his terminal based on previously developed data extraction scripts, which would describe the current state of all network traffic or highlight particular instances of anomalous activity.

Once potentially suspect hosts are identified, analysts would retrieve relevant historical data in another DataBurst diagram and draw their own conclusions. With minor modifications, DataBurst can be modified to automatically update itself continuously for a defined period of time from an input data stream.

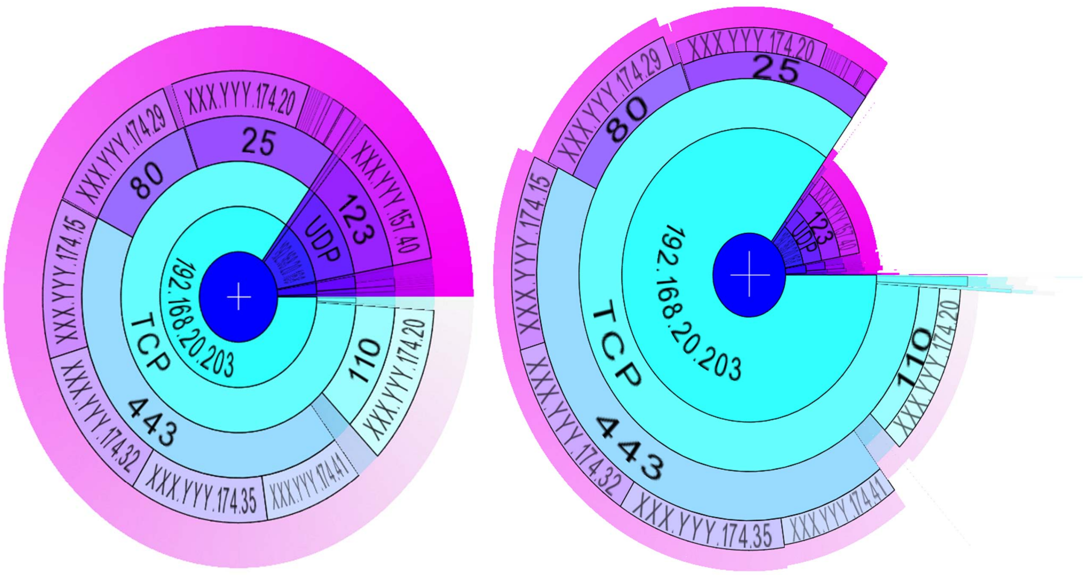
The cases described in the following two sections do not cover the gamut of responsibilities of network security analysts, but deal with many of the activities that may be performed in a typical analysis. In both cases, a single netflow repository serves as the source from which input data are initially obtained, and contains flows captured from a live /22 network (the first 22 bits of each IP address describes the network number, and the remaining 10 bits describe the host number, allowing for 2^{10} hosts in this network) with internet connectivity over a period of several months.

The SiLK tool suite was used to retrieve and aggregate all netflow data such that external IP addresses are anonymized for privacy reasons (using SiLK's rwrandomizeip tool, which ensures that every individual external IP address is replaced with the same randomly generated IP address), and the total number of bytes, packets and flows were calculated for each tuple of source and destination IP and port (and protocol when necessary).

4.5 Example 1 - Network Overview

We first demonstrate DataBurst's ability to present an overview of a large amount of netflow data, by investigating one hour's worth of network activity. This subset of data, shown in Figure 4.1, shows outgoing communications from nine internal hosts to several external hosts over a wide range of ports and protocols. As this is a complete overview, this input data shows the bytes, packets and flows transferred between each 5-tuple of source IP, protocol, destination port, destination IP and source port, resulting in about 12,500 lines of input data. When searching for specific behaviour, an analyst may only need to visualize the flows of a particular protocol e.g. only TCP flows.

One may be tempted to initially conclude that a single particular host (192.168.20. 203, referred to here as host A) dominates all communication in the data rendered, but this would be misleading as the diagram simply says that in this case host A communicates with a number of other hosts (about 300). This by itself does not indicate anomalous activity and we can use the diagram to further investigate this host.



(a) Network overview (b) With radial size showing bytes transferred

Figure 4.1: DataBurst diagrams showing one hour's worth of netflow data aggregated by source IP, protocol, destination port, destination IP and source port with about 12,500 leaf nodes.

We then note that the majority of outgoing flows from host A are through four TCP ports. Two of them (80 and 443) indicate general web browsing activity [4] with approximately 21 hosts. This activity does not appear to be anomalous as the outgoing requests for web services are all roughly the same size (indicative of normal browser activity), though a security analyst should ensure that users on host A have permission to access web resources and that incoming web activity to this host are within normal parameters.

Of the remaining high-activity destination ports, we note that many flows are transmitted to port 25 to 42 hosts (upper portion of the diagram) and from port 25 to approximately 260 hosts (between destination ports 443 and 110 in the lower right of the diagram, which could be emphasized using hyperbolic distortion). This tells us that this host also functions as an SMTP mail server which is communicating both to other email servers (the first group of flows) and email clients (the second larger group of flows). This is verified with the outgoing port 110 flows, indicating POP3

[4] `http://www.iana.org/assignments/port-numbers`

activity. The activity again is most likely normal as well due to its uniformity.

We also see that the majority of UDP traffic consists of single flow communications to and from port 123 to a limited number of external hosts. Further investigation into the source flows shows that these are generated at regular time intervals. Researching the Network Time Protocol (NTP) shows that this server is likely either updating its system clock with that of external time servers or is responsible for synchronizing the clocks of other external hosts. While the anonymized external IP addresses makes this difficult to verify, the low volume and frequency of activity suggests it is normal.

At this point experienced security analysts would then modify their SiLK scripts to filter off normal outgoing flows e.g. web requests, NTP or email activity of a certain size, so that they can focus on those things that do not immediately appear normal. We emulate this by tearing away those nodes that we assume to be normal, which results in Figure 4.2.

Having removed many flows from the original listing, we have a better view of activity in remaining hosts. Expected amounts of low volume activity are present e.g. additional web and email activity (192.168.20.69 appears to be another multipurpose server), general network management activity (several hosts on this network offer domain name and NetBios services) and SSH and Microsoft SQL services.

The large quantity (in terms of bytes, packets and flows transferred, shown in Figure 4.3) of ICMP activity may give the impression that anomalous activity is occurring, but the subset of ping requests (ICMP port 2048 [5]) are seen to be going to the list of external time servers previously communicated with. This is not a documented feature of the protocol, but it is likely that they represent application-specific service-availability checks, and we may safely ignore them.

A number of ICMP replies however indicate that some external hosts were repeatedly attempting to connect to hosts or ports that were either unreachable (ICMP port 771) or prohibited (ICMP port 781). The DataBurst diagram in Figure 4.3 highlights these in blue as its radial colour is assigned to flows transferred.

One would expect that an unauthorized external user would not continue to attempt to access resources when the correct privileges are not had or the resources do not exist. This was indeed true for many external hosts, but several were found that

[5] http://www.iana.org/assignments/icmp-parameters

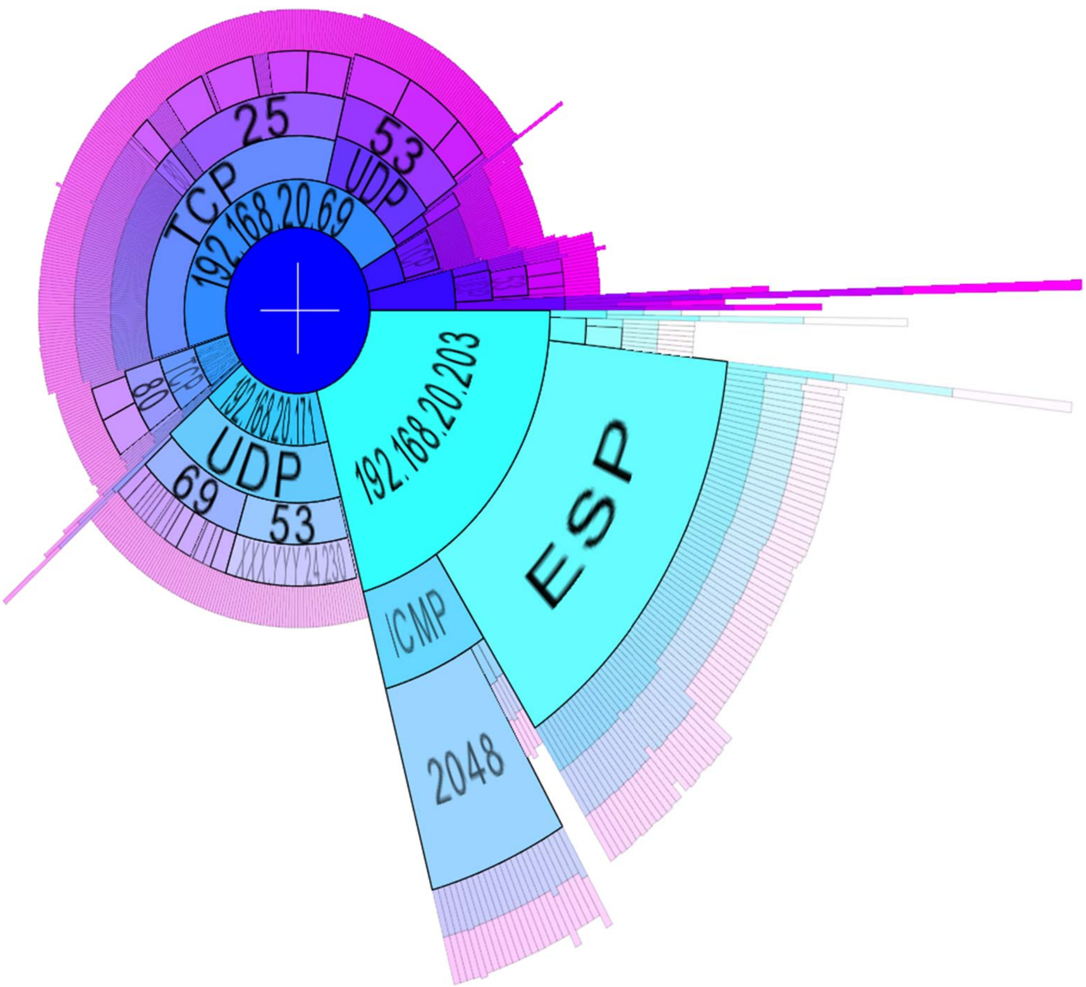


Figure 4.2: DataBurst diagram showing one hour's worth of netflow data aggregated by source and destination IP and port, and protocol excluding some identified normal activity and mapping radial size to bytes transferred.

attempted this many times within the hour (e.g. XXX.YYY.174.20, examined using hyperbolic distortion in Figure 4.3, did this several hundred times).

As the activity of these hosts is clearly not normal, these hosts are likely not legitimate users. A security analyst's next step may involve rendering DataBurst diagrams for previously captured flows of outgoing communications to these external hosts (possibly indicating compromised internal hosts) and blocking all future incoming communications from them. Behaviour and usage patterns could also be gleaned such that future or similar activity could be identified before security breaches occur.

Finally, both Figures 4.2 and 4.3 highlight that two internal hosts are communicating with external hosts via the Encapsulating Security Payload (ESP) protocol

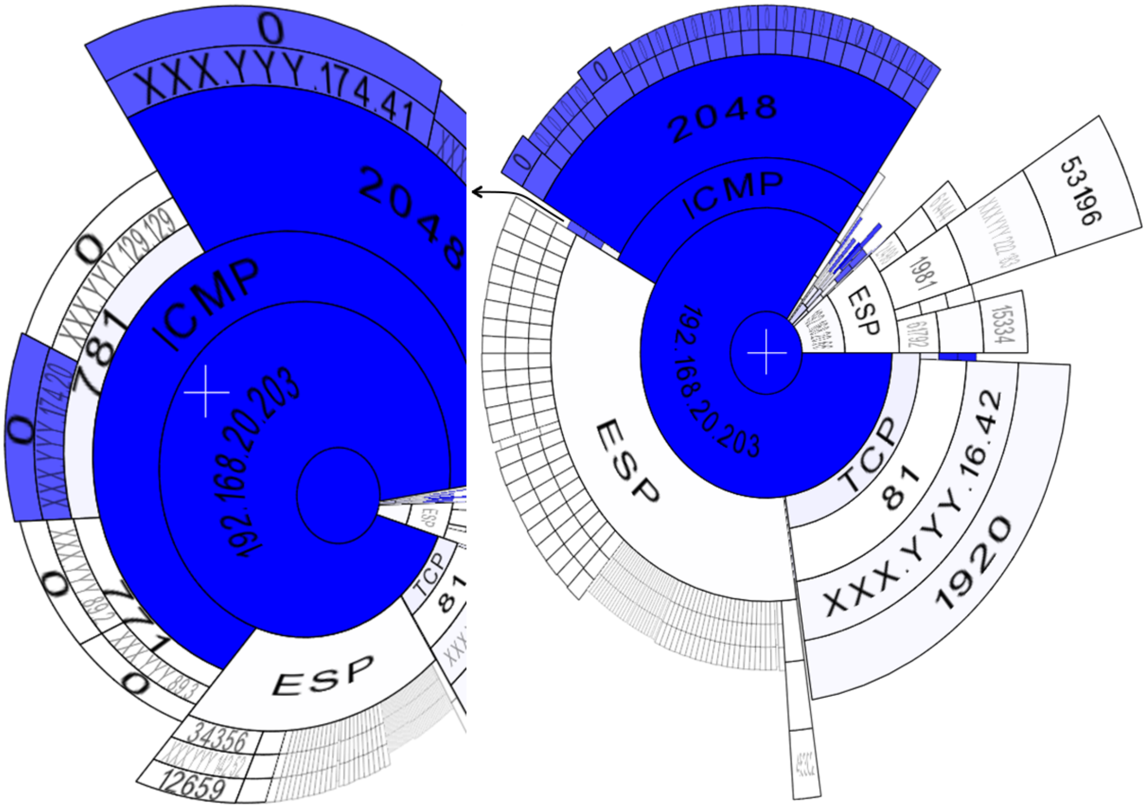


Figure 4.3: DataBurst diagrams showing one hour's worth of netflow data aggregated by source and destination IP and port, and protocol excluding some identified normal activity, and a closer view of a node of interest using hyperbolic distortion. Radial size, angle and colour are assigned to bytes, packets and flows transferred.

which is a component of the IPSec protocol suite [6] facilitating the authentication and encrypting of individual IP packets, usually in Virtual Private Networks (VPNs). It is difficult to accurately say if communications on this protocol are normal or not (due to their use of encrypted packets and dynamic ports), but the security analyst can keep a list of the IP addresses of known VPN clients and servers and investigate those hosts communicating on this protocol that are not on the list. Note that the observed outgoing web traffic from host A could be generated by the many VPN clients.

4.6 Example 2 - Game Server

In this example, we investigate the behaviour of a legitimate server based on a 3 hour period of outgoing flows from the first day of flow capture, February 2nd 2006, and

[6] `http://www.ietf.org/rfc/rfc2401.txt`

show how DataBurst can quickly communicate particular network events to analysts. In hindsight we note that the server was compromised before capture began, but include related findings as a good example of an analysis of a network intrusion resulting in unauthorized modification of at least one server.

A sample of initial outgoing netflows is visualized in Figure 4.4, where radial size, angle and colour are assigned to packet counts, flow duration and bytes transferred. Some regular web activity is seen on several clients, but two hosts are seen to have sent a large number of packets, with one doing so for the majority of the time period being analyzed.

The first, 192.168.22.73, has several outgoing flows with high numbers of packets sent in small periods of time, and others where few packets are sent in longer time periods. As the outgoing ports are non-standard, the security analyst should investigate the running processes on this host to determine what exactly is generating the activity.

The internal host of interest in this example though, is 192.168.20.165. It transmits a large volume of data through port 27015, a well known port through which dedicated servers are run for the video game Half Life. This was confirmed as network game activity is known to be bursty (generating a large number of packets) and can lead to lengthy flows being logged for communications between hosts (seen with long durations).

In such situations, a network security analyst would check to ensure that no other internal hosts are displaying similar behaviour (no other hosts generating large numbers of packets from or to similar ports), and would also delve into the log of previous flows captured, to determine exactly how the internal host was compromised e.g. a high level of FTP activity from an unauthorized external host. The particular information, the vulnerability that was exploited, was unfortunately not made available.

On February 25th after several weeks of investigation, the game server was permanently removed and the host was secured. In looking at daily activity over time, we create a hierarchy of source IP, date and source port, and assign radial size to bytes transferred by the compromised server (see Figure 4.5). We see that outgoing traffic steadily increases at the beginning of the investigation, and stops abruptly on February 25th. The single large node in each day represents port 27015, indicating

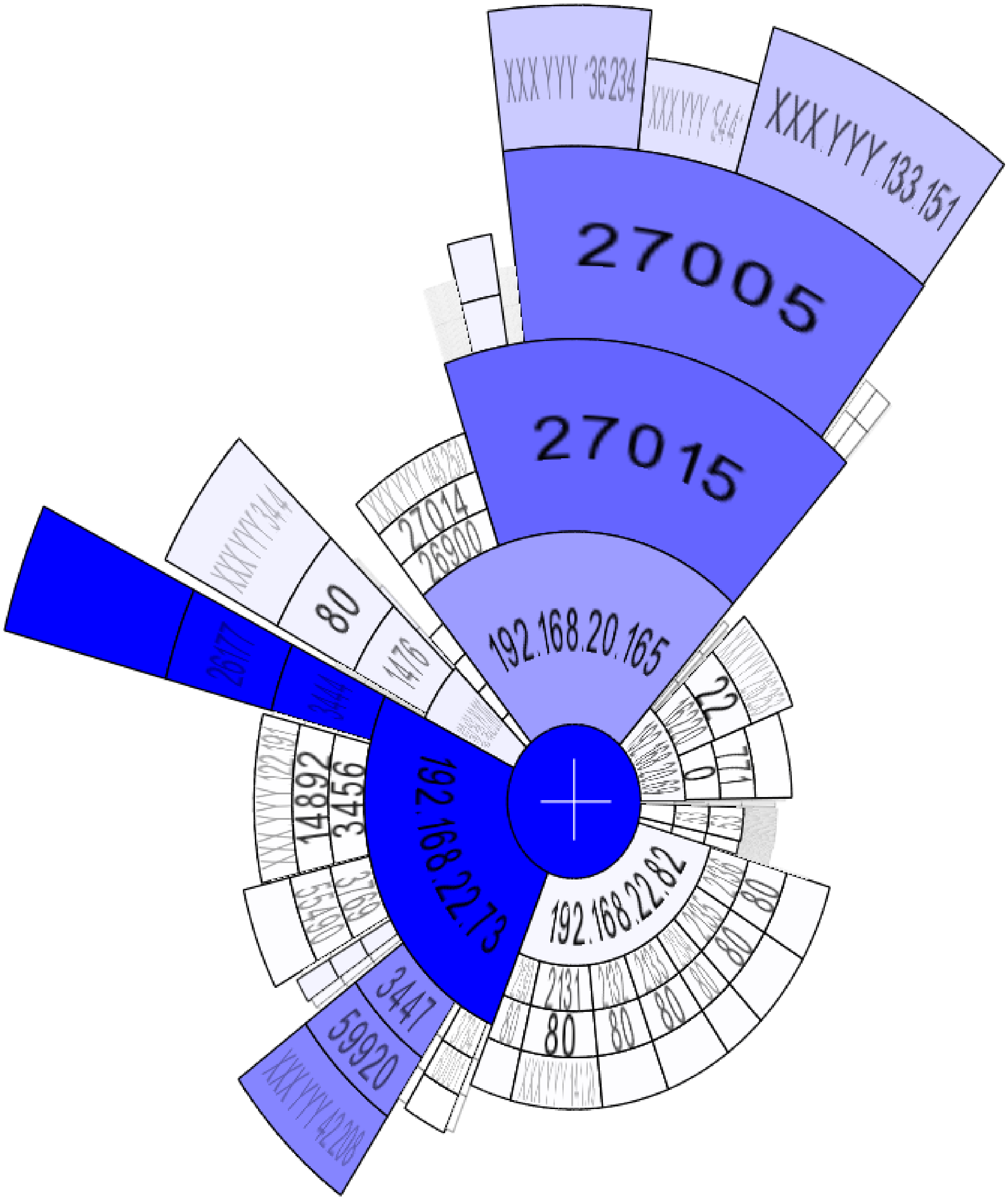


Figure 4.4: DataBurst diagram showing six hour's worth of netflow data aggregated by source IP, source and destination port, and destination IP and mapping radial size, angle and colour to packet counts, flow duration and bytes transferred.

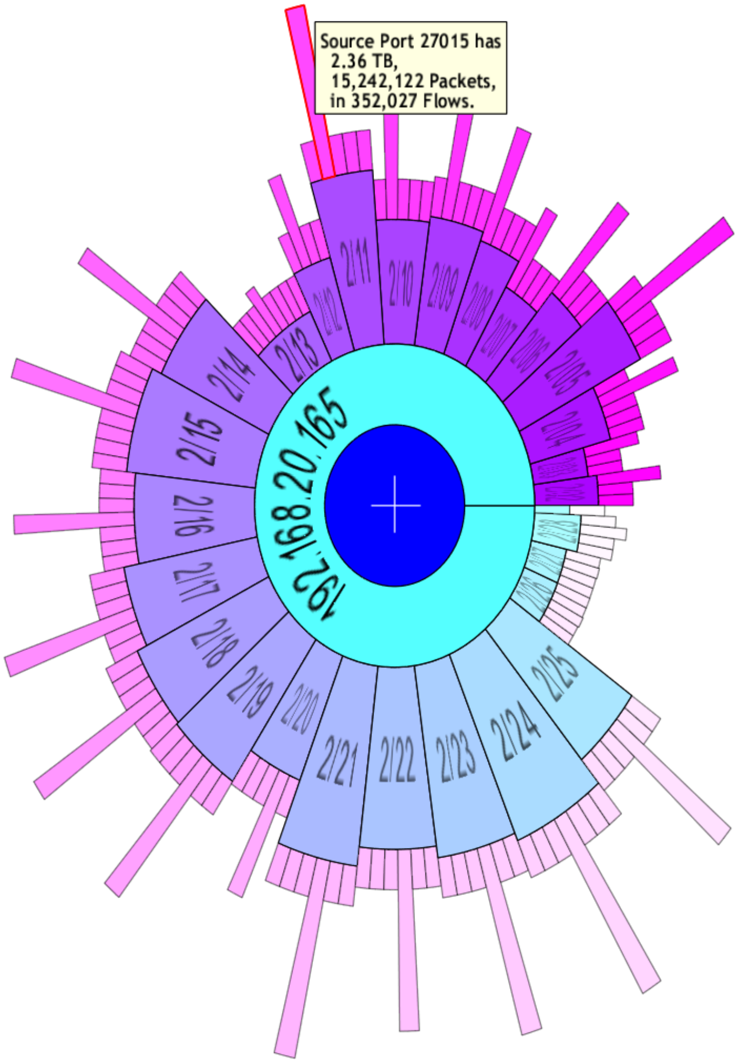


Figure 4.5: DataBurst diagram showing six hour's worth of netflow data aggregated by source IP, date, and source port and mapping radial size to bytes transferred.

that it is the highest activity port.

An interesting observation is the dip in outgoing activity on the 12th and 13th (from 2.3TB to 0.2GB) while the corresponding incoming activity (not visualized) did not change. This may represent an attempt to disable the game server or some hardware event that slowed its ability to respond to incoming traffic.

Chapter 5

Case Study - Visualizing Genetic Data

In this chapter we present detailed investigations into genotype data pertaining to two proteins of the Influenza A virus and their relationships to two phenotypes (any observed characteristics or traits). The DataBurst visualization facilitates analyses of this type as they typically involve hierarchical data and a handful of key attributes that are to be compared. We will begin by defining necessary concepts and terminology used in this chapter, then discuss associated problems, and show how the DataBurst diagram facilitates data analysis.

5.1 Background

Bioinformatics, the application of statistics and computer science within biology, is used to investigate many complex biological interactions at the molecular level up to the ecological level. An organism's genome is what defines its properties and governs related biological interactions, and its constituent genes are inherited from its ancestors. A cell's genes are made from a DNA molecule containing genetic information made up of 4 nucleotide bases (Adenine, Cytosine, Thymine or Guanine), representing the cell's genotype, and are responsible for creating protein molecules in the cell.

Proteins typically perform a single job, e.g. in cell division, defence, movement, which together all generate the organism's phenotypes. Proteins are chains of amino acid monomers that fold into complex structures based on the amino acid sequence, where the structure determines the job performed. Different parts of the protein can also perform unique roles e.g. identifying and binding with another protein through intermolecular forces, or overall molecular stability.

Replication is an important process that occurs at the molecular levels in cells, where a copy of the original DNA is passed on, and errors can randomly occur producing genetic variations, or mutations, that may be successfully carried on in future cell divisions. They can be categorized as insertions, deletions, or point mutations (where for example an A changes to a G). These mutations can result in the new cells having altered genes (or alleles) which produce proteins that could be very functionally different from those of the parent, giving rise to the genetic diversity seen throughout evolution. Genetic changes produced can be advantageous (where the organisms gain a competitive advantage allowing them to proliferate in a particular environment, a process known as natural selection), deleterious (where the changes remove a beneficial attribute or make the organism more susceptible to negative influences, reducing its fitness if inherited) or neutral (the organism is unaffected, and the change may be perpetuated or not).

Varying amounts of genetic change can give rise to a range of genes (called alleles of the original gene) and we can define evolutionary relationships between gene sequences by their relationships with ancestors and past genetic variations. Homologs are descended from a common ancestor e.g. the four limbs of tetrapods, and those that are separated by speciation events are of particular interest as they help us to identify patterns of genetic divergence and determine similarities between organisms [2].

We obtain the one-dimensional sequences of biological molecules through gene sequencing, which produces a string representation of that gene (with alphabet A, C, T and G) and the corresponding encoded protein (with a 20 character alphabet). This allows us to classify and compare homologous genes and proteins from separate individuals such that functional, structural and evolutionary relationships between them can be gleaned from regions of similarity. As we have seen above, homologous sequences can differ due to point mutations and can be of varying lengths due to insertions and deletions, so sequence alignment is used to introduce gaps into sequences such that homologous characters are aligned vertically within the produced tabular listing [31].

A sample of five protein sequences were aligned using ClustalW [1], which is based on [42], and the results are presented in Figure 5.1 . It shows the first thirty and last twenty residues with conserved gap characters removed, where amino acids are highly conserved at a position in an alignment if they have similar biochemical properties. Gap characters are shown here by empty spaces, and particular amino acids of a

[1] http://www.ebi.ac.uk/clustalw

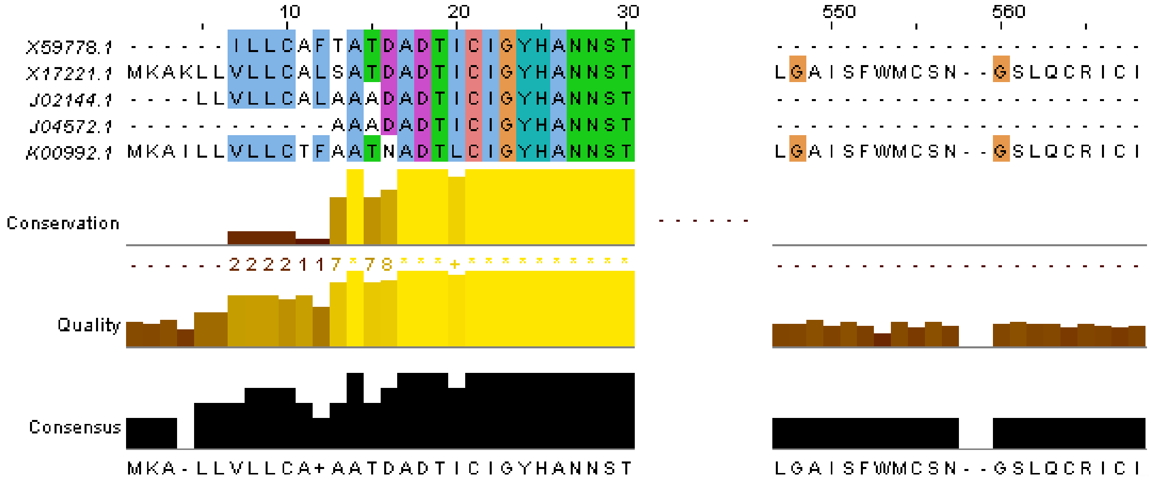

10
20
30
550
560
X59778.1 ------ILLCAFTATDADTICIGYHANNST ----------------------
X17221.1 MKAKLLVLLCALSATDADTICIGYHANNST LGAISFWMCSN--GSLQCRICI
J02144.1 ----LLVLLCALAAADADTICIGYHANNST ----------------------
J04572.1 ------------AAADADTICIGYHANNST ----------------------
K00992.1 MKAILLVLLCTFAATNADTLCIGYHANNST LGAISFWMCSN--GSLQCRICI
Conservation
------2222117*78***+*********** ----------------------
Quality
Consensus
MKA-LLVLLCA+AATDADTICIGYHANNST LGAISFWMCSN--GSLQCRICI

Figure 5.1: Aligned sequences visualized using ClustalW. High (above 90%) and low (below 50%) consensus colours are shown in red and blue respectively, with conserved gap characters removed. The '+' consensus character at position 16 indicates that more than one amino acid has the highest observed frequency.

column are colour-coded if the column meets minimum criteria (defined in [42]). The three histograms measure, from top to bottom, the conservation of physicochemical properties, the alignment quality based on observed substitutions, and the amino acid most frequently observed (together with the frequency at which it was seen) at each column of the alignment.

Sequence alignment therefore shows which residues are homologous between sequences, which is difficult to deduce given the numerous known and unknown genetic changes (e.g. mutations) that relate them to their common ancestors. Relating homologous genes helps to infer evolutionary relationships between species and is the basis for the field of phylogenetics, where closely related sequences have a much smaller evolutionary distance between them (fewer genetic changes separate the sequences). We can also assess whether a given trait has emerged multiple times in the evolution of a group, or can be traced back to a single common ancestor.

A phylogenetic tree for the sample sequences, generated using on online implementation of PhyML and TreeDyn [2] is shown in Figure 5.2, where the third and fourth sequences (J02144.1 and J04572.1) are the most closely related, followed by the first and second (X59778.1 and X17221.1), with these two groups together being more closely related to each other than the fifth sequence (K00992.1). Note that the

[2] http://www.phylogeny.fr/version2_cgi/simple_phylogeny.cgi

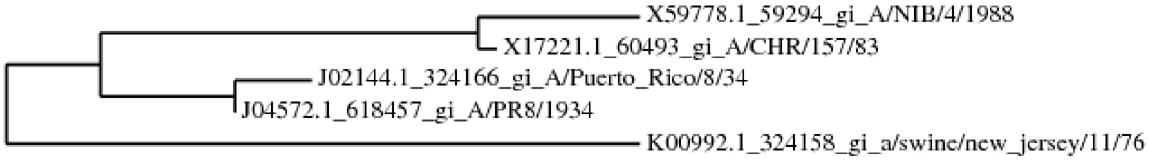

X59778.1_59294_gi_A/NIB/4/1988
X17221.1_60493_gi_A/CHR/157/83
J02144.1_324166_gi_A/Puerto_Rico/8/34
J04572.1_618457_gi_A/PR8/1934
K00992.1_324158_gi_a/swine/new_jersey/11/76

Figure 5.2: A phylogenetic tree produced from the sample sequences shown in Figure 5.1.

leaves of a phylogenetic tree represent observed sequences while ancestral nodes are typically not represented by observed individuals e.g. prehistoric man.

Biologists are interested in knowing phylogenetic signals affect shared genes i.e. whether similarity is due to genetic makeup or phenotypic traits. We use the following example with the sequences in Figure 5.2 to describe this, where in the first case a phenotype is observed in the first pair of sequences, and in the second case it is observed in the first and last sequences. The first case is in itself not necessarily interesting as phenotypic similarity is generally expected in the presence of genetic similarity, as the common ancestor of the first pair of sequences would be responsible for obtaining and passing on the phenotype. The second case however is less expected, and much more interesting, as no single common ancestor could have been responsible, and the single phenotype's multiple origins are independent.

The emergence of particular phenotypes cannot be easily discerned as the complete list of evolutionary ancestry between sequences is not known. Examining genotype-phenotype relationships helps answer this question.

5.2 The Problem

The rapidly dispersing 2009 influenza pandemic [51] forced the scientific community to look closely at the constantly mutating annual seasonal flu virus and the pandemic H1N1 influenza A virus. The latter is likely a result of genetic re-assortments between human, avian and swine influenza viruses [76], to which little to no pre-existing immunity exists in humans. With this lack of immunity, its resistance to adamantane drugs, and a high infection rate, the pandemic virus spread quickly, resulting in hundreds of deaths worldwide. Costly and slow production of effective vaccines further compounded this problem [14].

As the pandemic and seasonal flu viruses are genetically and antigenically quite different [25] (immune response with the latter), it is expected that their phenotypes would differ (which manifest as differing symptom patterns [68] and antiviral resistances). Many research efforts focused on the resistance of the new pandemic strains to existing antiviral drugs, together with the search for novel, effective therapeutics.

Particularly, relationships between amino acids at one or more particular positions in the influenza protein sequences and the strains' observed phenotypes was sought, the most important of which are resistances to antiviral drugs e.g. the zanamivir drug has historically been effective against pandemic influenza, but some resistant strains have been recently identified with a novel mutation in their neuraminidase protein [33].

This is not a trivial problem however as the solution is based on protein sequences captured over time that only represent a subset of all genetic variations that occurred, some of which are only partially sequenced, and is compounded by the fact that the number of sites contributing to the observed phenotype is unknown. Uncontrolled random factors, bias and unknown variables arising from differing experimental procedures, human error and environmental influences further impact this analysis.

We therefore need a method of identifying those amino acids and their positions that are most highly correlated with the observed phenotype, and an interactive way of choosing the best candidates when a large number of similar sites are identified. We use mutual information [16] to identify the desired sequence positions. We also perform this analysis by reducing the effect of phylogenetic signals (as discussed in Section 5.1) by investigating two distinct strain groups in a phylogenetic tree of influenza strains.

5.3 Data Acquisition and Preprocessing

The data used in this analysis were originally obtained from the open access Influenza Virus Resource at the National Center for Biotechnology Information's (NCBI) website[3] which is built from direct submissions from individual laboratories and bulk submissions from larger sequencing centres. Our data are comprised of aligned H1N1 human-host influenza A sequences of the two surface proteins, hemagglutinin (HA) and neuraminidase (NA). Note that the influenza A genome contains eleven genes in

[3] http://www.ncbi.nlm.nih.gov/genomes/FLU/SwineFlu.html

total (HA, NA, NP, M1, M2, NS1, NS2, PA, PB1, PB1-F2 and PB2) in eight RNA segments (which have a total length of 14,000 nucleotides) [52]. As the HA and NA proteins enable the influenza virus to invade and reproduce in living cells, inhibiting these proteins can potentially curtail its spread. These data were processed and uploaded to the data warehouse and web application, SeqMonitor [45] where 4,723 HA and 3,618 NA records are available as of April 14th, 2010.

SeqMonitor can incrementally accept sequence data and isolate metadata through an automated pipeline, where new data is parsed to obtain geographic, temporal and antiviral resistance information (whether user-verified, explicitly stated, or inferred from context e.g. the isolate name), and multiple sequence alignment is performed using Muscle version 3.7 [22]. Detailed information on the dataset can be found in the original authors' papers in [53] and [45].

Note that gaps occur often near the start and end of aligned sequences as many strains are only partially sequenced, due to cost or time constraints, such that their sequence lengths are much shorter than their alignment lengths.

The input data, obtained from SeqMonitor, to be visualized in DataBurst take the form of aligned H1N1 human-host influenza protein sequences of the HA and NA genes, together with metadata associated with each sequence. These are obtained in flat CSV-type files, with the former in the popular FASTA format (with key information on a line prefixed with ">", and one or more subsequent lines containing the protein sequence) and the latter containing tabular data (where the first line contains header information, one column contains the key information - accession number in this case, and any of the other columns containing categorical data to be used as candidate phenotypes).

The metadata describes whether the strain is of swine origin or not (2009 pandemic or not), together with the sample's location, from the continent to the geographic coordinate level. The two classes of antiviral resistances considered here are adamantane and oseltamivir, where the former refers to a pair of drugs (amantadine and rimantadine) having the same active ingredient that interfere with the function of the M2 protein, and the latter is a neuraminidase (NA) inhibitor that slows the spread of the influenza virus by preventing it from leaving its host cell. The HA and NA sequences have alignment lengths of 568 and 470 positions respectively.

The unrooted, bifurcating phylogenetic tree was then inferred using the aligned data and RAxML version 7.0.4 [63] and takes the form of hierarchically nested "Accession:Branch Length" tuples which associates nested pairs of sequences, with each sequence placed at the leaf level. The tree was visualized using the external application FigTree[4], which facilitates selection of sequence groups with common ancestory.

5.4 Mutual Information

We use mutual information to determine the degree of association between individual amino acids and the observed phenotypes (the two classes of antiviral resistances). Numerous instances of prior work have been performed by other authors on H1N1 data (e.g. [25], [67], [11]), with the most well documented observation, the mutation of the H amino acid at position 275 to Y (or H275Y) in the NA protein conferring resistance to oseltamivir, being used to verify one of the analyses conducted here. It should be noted that no other widely known and accepted mutations are attributed to antiviral resistance in the HA and NA genes of H1N1. This is the first application of mutual information in exploring this particular dataset to this author's knowledge, as well as the first visual analysis of this nature.

Numerous similar methods exist and have been widely used in the literature. One such group of methods involves supervised machine learning, which includes support vector machines [7], artificial neural networks [70], and decision trees [8] and are useful in the presence of high-dimensional data and/or noisy data. Another popular group is the regression analysis (e.g. linear regression) and variance analysis (e.g. ANOVA) methods, which are tailored for continuous or numerical variables. Statistical correlation tests are also used e.g. t-test, Pearson's chi-square test, Fisher's exact test etc., but are not generally flexible enough to be used in this context as they are essentially linear distance measures between like entities, and are sensitive to outliers [17]. Methods based on categorical variables have also been used e.g. discriminant analysis (requires normally distributed independent numeric variables) and logistic regression. No literature was found comparing/contrasting the effectiveness of these techniques with genetic data.

[4] http://tree.bio.ed.ac.uk/software/figtree/

Mutual information is ideal in this context in that it is more applicable to the data being analyzed i.e. it makes no assumption of data distribution or dependence, can be used for both categorical and numerical data (using various binning or integral techniques), and does not require large samples of data to produce reliable results yet can still be calculated quickly and efficiently with large datasets. It has been used often within the literature, primarily in investigating the relationships between genes (e.g. within a genome as in [66] and [17]), and between sites in a group of DNA, RNA or protein sequences (e.g. in ribosomal RNA with [12] and [28], and the proteins of related types of influenza viruses [73]). The method used here differs from these approaches as the dependence of individual amino acids of aligned protein sequences will be calculated against known phenotype data, as was similarly done in several papers investigating drug resistances (e.g. [7] and [71], but only at particular sites and not across the entire sequence).

Mutual information is an information theoretic measure that describes the mutual dependence between two discrete random variables, as described by [61] and [16]. More generally, it is the amount by which we have reduced uncertainty in a variable having known about another. Shannon's similar and more efficient interpretation is more widely utilized, and makes use of Shannon entropy.

The Shannon entropy H(X) is defined as

$$H(X) = -\Sigma p(x_i)log(p(x_i)) \tag{5.1}$$

where x_i is the i^{th} possible state in a system X, $p(x_i)$ represents the probability of the occurrence of state x_i, and the summation is done across all states in the system. The joint entropy, H(X,Y) is calculated in the same way using joint probabilities $p(x_i, y_j)$ for all pairs of states in the system. Shannon entropy measures state distribution in a system and lies between 0 and 1 inclusively (0 when X and Y are completely conserved, and 1 when all states are equally distributed). Mutual information is then calculated as I(X;Y) = H(X) + H(Y) - H(X,Y), and can vary between 0 and 1 inclusively (a value of 1 is possible, though often unlikely with real-world data as mutual information is capped at the minimum entropy of either variable).

5.5 The Solution

Using the data sourced from SeqMonitor (as described in detail in the previous section), the analysis performed consisted of the following steps:

1. Merge the protein sequences with the metadata (by the accession key) to obtain the required sets of protein sequences and phenotype pairs. The calculations described below are run for each attribute of the metadata chosen as a phenotype in a single pass through the data files. Protein sequences having null phenotypes (where antiviral resistance or sensitivity is unknown) are ignored.

2. For each sequence,

   - Determine the counts of each phenotype in the dataset, P_k (where k represents each phenotype value) e.g. for oseltamivir resistance, $P_{resistant}$ is the count of resistant sequences, and $P_{sensitive}$ the count of sensitive sequences.
   - For each site in the sequence, determine (a) the count of each amino acid, $A_{i,j}$, and (b) the count of each combination of amino acid and phenotype, $AP_{i,j,k}$ (where i represents the amino acid character in the j^{th} site, and k represents the phenotype value).

3. Calculate the mutual information score at each site using equation 4.1 as follows:

   - Define X as the random variable amino acid and Y as the random variable phenotype, and $H(X_j)$ as the Shannon entropy of X at the j^{th} site, using equation 5.1,
   - $H(X_j) = \Sigma\ p(X_{i,j})\log(p(X_{i,j}))$, for all i amino acids at the j^{th} site, where $p(X_{i,j})=A_{i,j}/N_s$ and N_s is the total number of sequences,
   - $H(Y) = \Sigma\ p(Y_k)\log(p(Y_k))$, for all k phenotypes, where $p(Y_k)=P_k/N_s$,
   - $H(X_j,Y) = \Sigma\ p(X_{i,j},Y_k)\log(p(X_{i,j},Y_k))$, for all i amino acids at the j^{th} site for all k phenotypes, where $p(X_{i,j},Y_k)=AP_{i,j,k}/N_s$,
   - $MI_j = H(X_j) + H(Y) - H(X_j,Y)$

4. Return a list of sites sorted by mutual information in descending order. Allow the user to select a subset (usually the top n) of sites to be rendered in DataBurst.

5. Render the n sites in the order selected in DataBurst (i.e. sites are placed higher than other sites in the hierarchy if their mutual information score is higher, such that any particular level in the hierarchy has a lower score than the one immediately above). Allow the user to bring in additional metadata columns (e.g. year and location) into the hierarchy, depending on the analysis being performed.

Important points to note with this approach are that all mutual information scores are calculated in a single pass prior to each analysis. Multiple lists of mutual information scores can be kept for each identified phenotype for the single pair of input files, through which an analyst can select different single sites or combination of sites (from a single list) for visualization with DataBurst.

The choice of the highest-scoring n sites can be thought of as a measure of statistical filtering, where we look at the fraction of the many hundreds of available sites that are most responsible for an antiviral resistance.

Visualization of individual phylogenetic groups is currently accomplished by filtering those sequences into another input file, which is then used as input into Step 1 above in a subsequent iteration. As Figtree does not support exporting of group selections, the filtering process involves highlighting the subtree made up of those sequences within Figtree to determine the size of that group, n, then using an external custom C++ application to process the tree and output the n-1 neighbouring sequences of any member of the group.

5.6 Results

Of the 3,618 NA and 4,723 HA records available, counts of strains' resistances are listed in Table 5.1 and computed mutual information scores are listed in Table 5.2.

	NA		HA	
	Adamantane	Oseltamivir	Adamantane	Oseltamivir
Resistant	348	282	349	270
Sensitive	397	429	476	421
Unknown	2,873	2,907	3,898	4,032

Table 5.1: Strain counts by adamantane and oseltamivir phenotypes.

	NA				HA			
	Oseltamivir		Adamantane		Oseltamivir		Adamantane	
	MI	Pos	MI	Pos	MI	Pos	MI	Pos
1	0.654886	275	0.649080	267	0.669077	558	0.613226	206
2	0.346510	432	0.645588	188	0.538634	147	0.606886	147
3	0.317056	393	0.641962	367	0.401988	207	0.600325	146
4	0.313543	188	0.638242	393	0.365123	146	0.599973	53
5	0.313447	367	0.591308	287	0.362411	210	0.591358	211
6	0.312938	267	0.588178	45	0.328808	203	0.570764	163
7	0.308960	78	0.585631	78	0.319040	201	0.548393	526
8	0.308204	45	0.567161	331	0.316505	206	0.547727	421

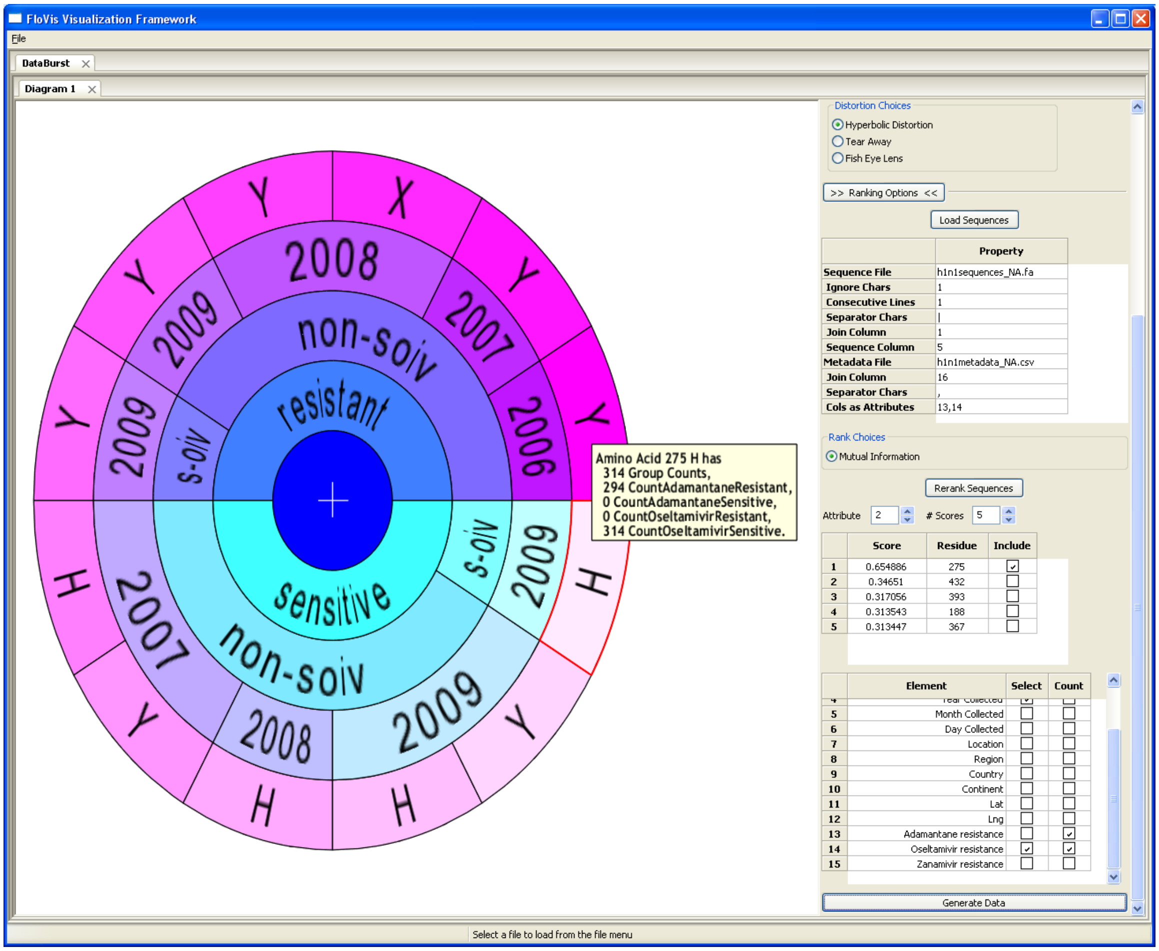
Table 5.2: The top 8 mutual information scores (MI) of positions (Pos) in the NA and HA sequences for oseltamivir and adamantane resistances, in descending order.

Table 5.1 shows that of all tested strains, the ratio of sensitive to resistant strains is far greater with respsect to oseltamivir than adamantane (implying that more strains may be resistant to adamantane), and the resistances of a large number of strains are not known (their proteins were sequenced, but the originating strains were not tested for antiviral resistance). Table 5.2 lists the top eight mutual information scores for the NA and HA protein sequences compared against the adamantane and oseltamivir resistance phenotypes. Further scores were not included as they differ by less than 0.01 to their next highest or lowest score.

While it is immediately clear that one or two sites of both proteins are highly correlated to oseltamivir phenotype, the scores obtained with the adamantane phenotype do not paint as clear a picture. With adamantane scores and the NA protein, about 18% of sites generate a mutual information score of 0.5 or greater and the remaining positions average approximately 0.02 (standard deviation 0.03). A similar distribution is seen with the HA protein against adamantane resistance, where 18% of sites score 0.5 or greater, and the remaining positions average about 0.05 (standard deviation 0.08).

To gain insight from the genotype data present, we can visualize selected amino acids at the sites of interest, together with other available metadata, in the Ranking Options panel of the DataBurst diagram.

Figure 5.3 represents the collection of all NA protein sequences grouped by oseltamivir resistance, source (swine origin or s-oiv, and non-swine origin or non-soiv),



Amino Acid 275 H has
314 Group Counts,
294 CountAdamantaneResistant,
0 CountAdamantaneSensitive,
0 CountOseltamivirResistant,
314 CountOseltamivirSensitive.

Distortion Choices

● Hyperbolic Distortion
○ Tear Away
○ Fish Eye Lens

>> Ranking Options <<

Load Sequences

	Property
Sequence File	h1n1sequences_NA.fa
Ignore Chars	1
Consecutive Lines	1
Separator Chars	|
Join Column	1
Sequence Column	5
Metadata File	h1n1metadata_NA.csv
Join Column	16
Separator Chars	,
Cols as Attributes	13,14

Rank Choices

● Mutual Information

Rerank Sequences

Attribute 2 # Scores 5

	Score	Residue	Include
1	0.654886	275	☑
2	0.34651	432	☐
3	0.317056	393	☐
4	0.313543	188	☐
5	0.313447	367	☐

	Element	Select	Count
4	Year Collected	☑	☐
5	Month Collected	☐	☐
6	Day Collected	☐	☐
7	Location	☐	☐
8	Region	☐	☐
9	Country	☐	☐
10	Continent	☐	☐
11	Lat	☐	☐
12	Lng	☐	☐
13	Adamantane resistance	☐	☑
14	Oseltamivir resistance	☑	☑
15	Zanamivir resistance	☐	☐

Generate Data

Select a file to load from the file menu

Figure 5.3: DataBurst of all NA sequences grouped by oseltamivir resistance, source, year, and amino acid at position 275, generated from the Ranking Options panel.

year collected, and amino acid at position 275, rendered with the default DataBurst settings. It is immediately apparent that the majority of strains that are resistant to oseltamivir have a Y at position 275, and all swine-origin strains are sensitive to oseltamivir when the amino acid at that position is H, and resistant when this amino acid is Y. This insight however is not enough to conclude that particular mutations confer the resistance in question. We next continue the analysis by incorporating resistance counts.

In assigning the counts of oseltamivir-resistant (Figure 5.4(a)) and sensitive (Figure 5.4(b)) strains to radial size, we see that Y275 non-swine-origin oseltamivir-resistant strains and H275 swine-origin oseltamivir-sensitive strains dominate all other samples. The association however, is not perfect, as we also see the presence of an

X275 non-swine oseltamivir-resistant strain (recorded during sequencing to show that an amino acid exists at that position, but it was not identified) and two single Y275 non-swine oseltamivir-sensitive strains (possibly due to genetic variation at other sites). The former may be excluded from this analysis if desired.

The picture becomes clearer then, as we now see that:

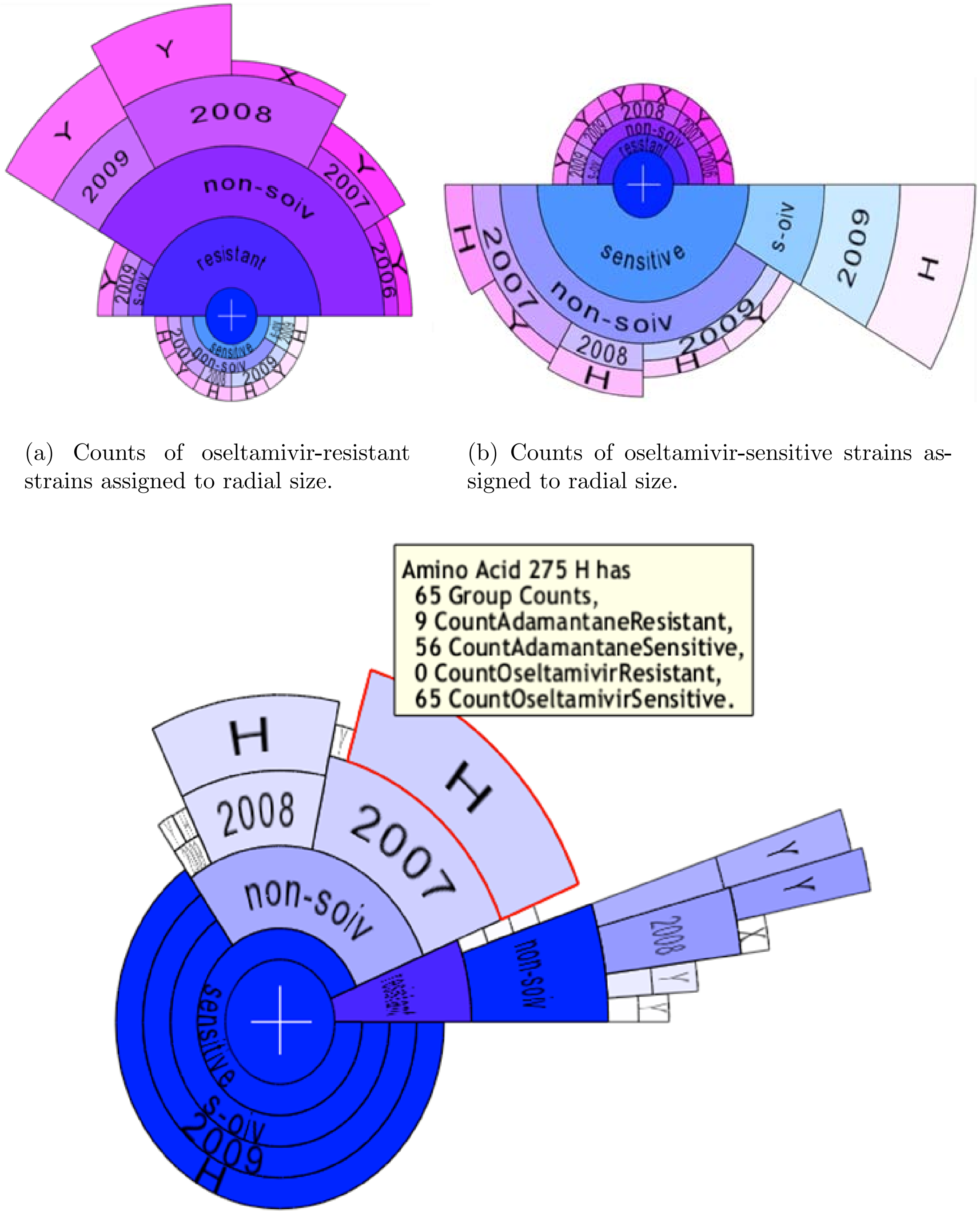
1. There is a strong association between amino acids at position 275 and oseltamivir resistance, and the former may likely be responsible for conferring this resistance (the vast majority of H275 strains are sensitive while the vast majority of Y275 strains are resistant),

2. The majority of non-swine-origin influenza strains are resistant to oseltamivir (this resistance increased markedly in 2008, and then decreased in 2009, possibly around the time of swine outbreak, with a decrease in the number of sensitive strains from 2007 to 2009), and

3. The vast majority of swine-origin influenza strains are sensitive to oseltamivir (with seven swine-origin Y275 resistant strains in 2009 otherwise).

By leveraging the final radial attribute, radial colour, we can compare up to three data attributes simultaneously. When assigning counts of adamantane-sensitive strains, oseltamivir-sensitive strains and total group counts to radial size, angle and colour respectively in Figure 5.4(c), we can quickly identify those groups that are sensitive to both antiviral drugs in the overall context - two groups totalling 86 strains in this case. Further exploring the diagram shows 11 swine-origin strains sensitive to both antiviral vaccines.

In contrast, the HA protein shows a shortcoming of the mutual information score calculation against oseltamivir resistance, and the sequence alignment used. Position 558 should provide better correlation to the HA protein than position 275 with the NA protein given that its score is slightly higher (0.669077 vs. 0.654886), but when investigated it was found that of the 691 HA sequences that were assigned an oseltamivir resistance, only one had an amino acid of G at position 558, while the remainder were gap characters. This is mostly expected due to its proximity to the end of the aligned sequence (where there would be more gap characters due to partial sequences), but highlights the possible need for additional heuristics in selecting



(a) Counts of oseltamivir-resistant strains assigned to radial size.

(b) Counts of oseltamivir-sensitive strains assigned to radial size.

Amino Acid 275 H has
65 Group Counts,
9 CountAdamantaneResistant,
56 CountAdamantaneSensitive,
0 CountOseltamivirResistant,
65 CountOseltamivirSensitive.

(c) Counts of adamantane- and oseltamivir-sensitive strains assigned to radial size and angle respectively with group counts assigned to colour.

Figure 5.4: DataBurst diagrams showing relationships between counts of strains' resistances at amino acid position 275 of the NA protein.

appropriate sites to be scored. Using the next highest-scoring position in the HA protein, we again see a clear separation as the amino acids at position 147 are either K for swine-origin strains, or either T or a gap character for non-swine origin, with the oseltamivir-sensitive swine-origin group dominating all other groups.

We continue the analysis by next looking at adamantane resistance. The mutual information scores with both proteins do not clearly identify agreement between a small number of sites and this resistance. As such, the analyst can interactively investigate which sites are relevant with the DataBurst diagram.

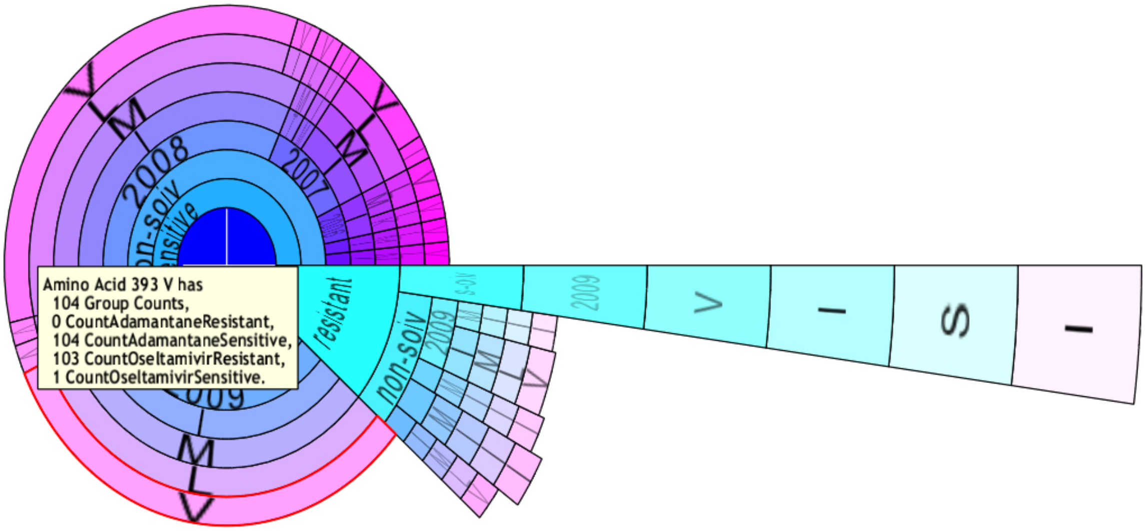

Amino Acid 393 V has
104 Group Counts,
0 CountAdamantaneResistant,
104 CountAdamantaneSensitive,
103 CountOseltamivirResistant,
1 CountOseltamivirSensitive.

Figure 5.5: Top-ranking residues in strains of the NA protein that are sensitive (shown with radial angle) and resistant (shown with radial size) to adamantane.

The results shown in Figure 5.5 are for the 4 highest-ranking positions in the NA protein (sites 267, 188, 367 and 393) with respect to adamantane resistance, where the hierarchy is defined similarly to previous examples, and radial angle and size are mapped to counts of adamantane-sensitive and resistant strains respectively.

Many observations can be made. The sequence groups with amino acids IMLV (in the order of sites given above) dominate the adamantane-sensitive data, increasing in size up to 2008 and then decreasing in 2009 (similarly with oseltamivir-sensitive non-swine data in the NA protein sequences), but the presence of other amino acid combinations (not currently visible e.g. IMIV, IKLV, TMLV) indicate that some point mutations occurred for strains at these sites without changing their phenotype.

With the resistant sequences there is a dominant combination (VISI) together with 5 IMLV sequences (one in 2006, remainder in 2009), and 25 MIII sequences over 2007, 2008 and 2009. It is unclear whether the non-VISI sequences that correspond to a resistant phenotype represent stages in mutation that led up to the presence of resistant sequences, or otherwise. The diverse geographic locations of these sequences does not make the picture any clearer. This diagram also shows that all swine-origin sequences are resistant to adamantane.

Even though these high scoring sites in the NA protein, with respect to adamantane resistance, may appear to be randomly dispersed along the protein sequence, it may well be possible that their positions in the protein's three-dimensional structure are quite close to each other. The required data and software to investigate this however, was not available to us during the analysis.

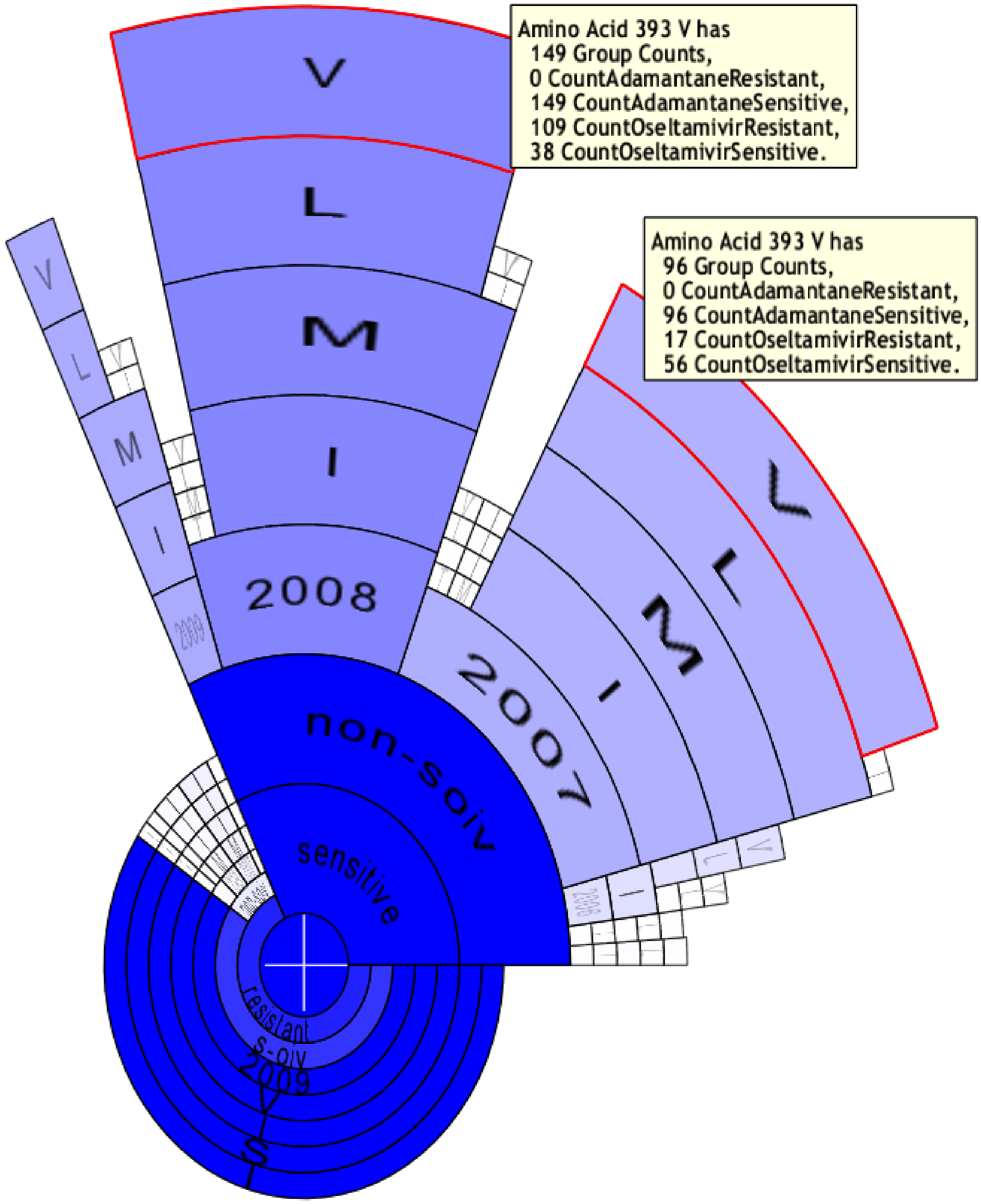
We note as well that as adamantane antivirals are purposed to interfere with the function of the M2 gene [56], the possibility exists that the genetic composition of the NA and HA genes may not directly affect adamantane resistance.

Comparing group counts of sequences that are both adamantane-sensitive and oseltamivir-sensitive (by assigning both to radial size and angle respectively), we see in Figure 5.6 that only IMLV sequence groups in 2007 and 2008 show high counts of both. In 2009, the adamantane-sensitive count fell somewhat, and only one oseltamivir-sensitive sequence was present. Comparing counts of adamantane-resistant and oseltamivir-resistant groups showed that the two are independent.

When looking at distantly related groups of NA protein sequences within the phylogenetic tree (visualized in Figure 5.7 using FigTree, and annotated in an image editor), we obtain the mutual information scores in Table 5.3 based on the two selected groups (1,677 strains in Group 1, in red, and 1,711 strains, in Group 2, in blue). Group membership was chosen such that the evolutionary distance between groups is large, the evolutionary distances between sequences within a group are minimized, and all strains tested for resistance to either or both drugs are present.

Note that in Figure 5.7 the common ancestor of each group is highlighted with a circle in the respective group colour, and all counts of resistant and sensitive strains given are for those assessed strains (i.e. the statement ”318 strains resistant to adamantane” does not imply that the remainder are sensitive).

The metadata statistics are interesting in their own right. We see that while



Amino Acid 393 V has
149 Group Counts,
0 CountAdamantaneResistant,
149 CountAdamantaneSensitive,
109 CountOseltamivirResistant,
38 CountOseltamivirSensitive.

Amino Acid 393 V has
96 Group Counts,
0 CountAdamantaneResistant,
96 CountAdamantaneSensitive,
17 CountOseltamivirResistant,
56 CountOseltamivirSensitive.

Figure 5.6: Top-ranking residues in strains of the NA protein sensitive to adamantane and oseltamivir.

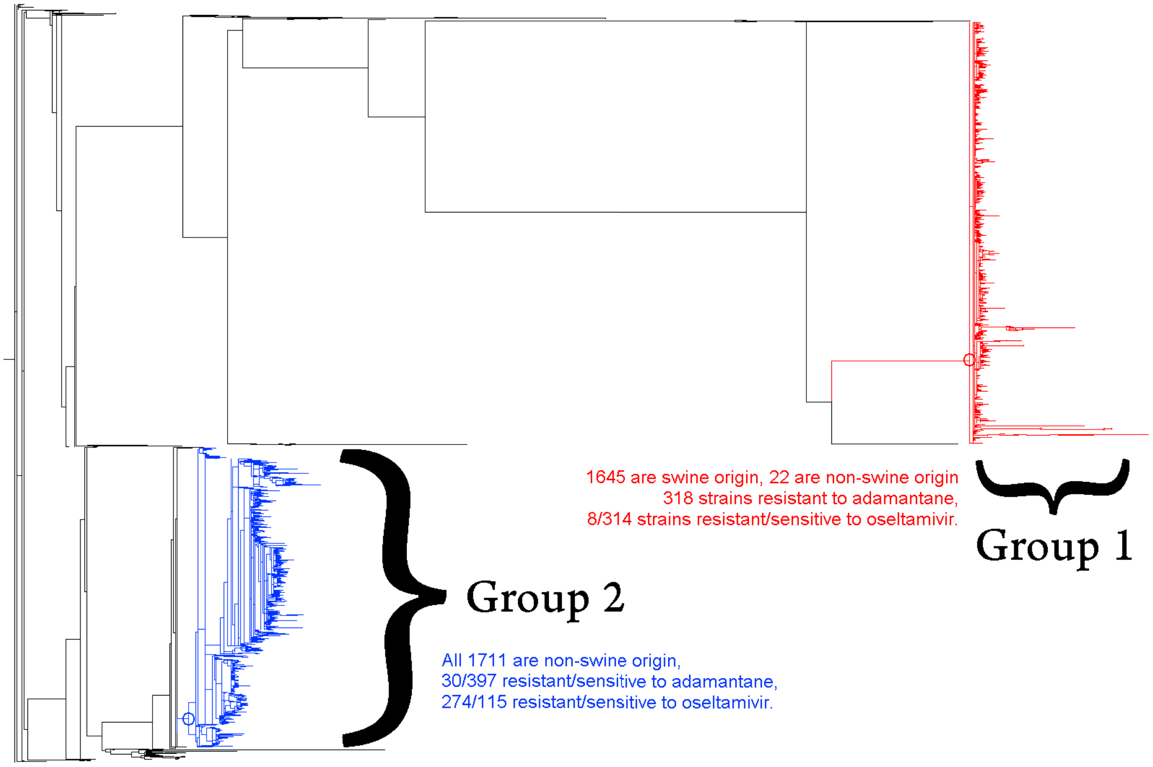

1645 are swine origin, 22 are non-swine origin
318 strains resistant to adamantane,
8/314 strains resistant/sensitive to oseltamivir.
Group 1
Group 2
All 1711 are non-swine origin,
30/397 resistant/sensitive to adamantane,
274/115 resistant/sensitive to oseltamivir.

Figure 5.7: Phylogenetic tree of the NA proteins of all strains, with two distinct groups highlighted in red and blue, with appropriate metadata statistics given.

Group 1 comprises very closely related sequences, with 22 non-swine sequences in this group, and conversely the distinct set of sequences in Group 2 are all of non-swine origin. Unfortunately, the 22 non-pandemic sequences in Group 1 were not tested for either resistance, though we believe that this group is characterized by having more sequences resistant to adamantane and sensitive to oseltamivir, based on the converse resistance ratio of Group 2, which has more sequences sensitive to adamantane and resistant to oseltamivir.

As the Group 1 sequences tested for adamantane are all resistant, the first column of mutual information scores in Table 5.3 tells us nothing. The 8 oseltamivir-resistant strains do not affect the low mutual information scores by much, but the impact of the H275Y mutation is still seen where the score at that position is ten times the next highest. Group 2 sequences show a better separation of oseltamivir-related scores, due to the greater number of oseltamivir-resistant and sensitive strains present, reiterating that position 275 is highly correlated to the oseltamivir resistance phenotype.

Interestingly, position 275's score is lower in this group than overall (either due

Group 1				Group 2			
Adamantane		Oseltamivir		Adamantane		Oseltamivir	
MI	Pos	MI	Pos	MI	Pos	MI	Pos
0	470	0.116337	275	0.18146	267	0.576679	275
0	469	0.011673	336	0.18146	188	0.330728	354
0	468	0.011673	257	0.18143	453	0.101199	28
0	467	0.011673	127	0.18140	130	0.059102	453
0	466	0.005881	82	0.17469	82	0.058287	393

Table 5.3: The top 5 mutual information scores for the two distantly related strain groups against adamantane and oseltamivir resistances.

to the majority of oseltamivir-sensitive sequences being separated into Group 1 or the minimum entropy of the variables has been lowered), and the 2nd runner up in this list (position 354) is 93rd in the previous overall listing (with a score of 0.0477), implying that much genetic variation may be present in these sequences in positions other than 275.

Adamantane-related Group 2 scores are much lower (again possibly due to the two reasons given above), and we still see positions 267 and 188 at the top of this list, but only by a marginal amount.

Interpreting these results from an evolutionary standpoint, we can see that swine-origin strains have diverged considerably from the majority of the non-swine strains with respect to the NA gene. We also see that all tested pandemic samples are resistant to adamantane and most are sensitive to oseltamivir, and that the converse is true for most of the non-swine strains. We therefore see the phenotype information mostly matching up with the phylogenetic data (produced from the genotypic data) at hand.

Mutual information scoring emphasizes that changes at position 275 of the protein sequences of the NA gene are highly associated with changes in the oseltamivir-resistance phenotype even after removing phenotypic signals. These scores may have also shortlisted a number of sites that may be similarly associated to the adamantane-resistance phenotype, though further statistical analysis is required to say with surety.

Chapter 6

Conclusion

Few visualization tools offer the ability to analyze and investigate large volumes of multi-dimensional, constant-depth hierarchical data, flexibly defined according to the varying needs of the analysis. Of the many interesting approaches to doing so, radial space-filling diagrams are effective in rendering a data hierarchy such that parent-child relationships can be easily gleaned.

There are also few interactive methods available for working with large volumes of data in an RSF that are simple to use and maintain context when performed. And finally, the majority of research efforts are limited to comparing two attributes simultaneously forcing manual comparisons to be done when investigating more than two attributes.

DataBurst was developed to address these shortcomings, and renders any given constant-depth hierarchy of data according to the needs of the analyst in an RSF diagram. Its contributions are the two unique methods of focus + context in Hyperbolic Distortion, where we drag the diagram centre to emphasize some nodes and deemphasize others, and TearAway subdiagrams, where we can create subdiagrams having nodes whose attributes can still be compared across multiple diagrams.

Another key contribution is DataBurst's leveraging of the radial attributes of node size, node angle and colour, in order to effectively compare up to three numeric attributes associated with the data hierarchy. This dynamic assignment of data attributes to radial attributes, and the ability to dynamically reorder the data hierarchy, allows analysts to view their data from multiple perspectives. The focus + context methods further enable data exploration without losing overall context.

We have shown that DataBurst is a useful tool in exploring both network security data and protein sequences, and highlighting particular network events and genotype and phenotype relationships. In the first case study, an overview of network activity was quickly gleaned and further detailed investigations were performed on individual nodes and their component flows. DataBurst assisted the identification and filtering out of benign activity as dictated by the analyst's knowledge, and the growth and subsequent deactivation of a game server was visualized.

In the second case study, DataBurst supported the identification of sites highly associated with known phenotypes and gave the quantitative insight allowing such assertions to be made with confidence by mapping appropriate counts to radial attributes and bringing large outliers immediately to an analyst's attention while filtering out infrequently occurring data (useful when working with large amounts of data).

The H275Y mutation which confers oseltamivir resistance to swine-origin influenza strains was effectively highlighted by the DataBurst diagram, both when looking at all sequences at once, and looking at two groups of sequences that were sufficiently distant from each other, yet contained highly related sequences (identified using the phylogenetic tree).

In the other instances where mutations are not exactly clear, DataBurst helps to identify relevant data that would suggest how a particular phenotype is related. This was seen in the adamantane resistance analysis on the NA protein (with very similar results obtained using the HA protein), where some strains with sequences containing a particular combination of amino acids (amino acid combination IMLV at positions 267, 188, 367 and 393 in the example) appear to be sensitive to adamantane, while others having another combination of amino acids are resistant.

We are aware that adamantane's function is most predominantly seen in the M2 protein, but the evidence presented here shows that some relationship exists between these two proteins and the adamantane phenotype - the cause of which is difficult to deduce from the data at hand. This relationship is therefore not a causative one.

We have established DataBurst's utility through these case studies, indicating that the DataBurst visualization will prove helpful to future analyses involving large volumes of hierarchical data.

6.1 Future Work

In the limited time frame available for implementation of the visualization and execution of the case studies, some improvements were identified that could further enhance the DataBurst interface and others that would produce better results in the case studies.

6.1.1 On the DataBurst Diagram

The DataBurst interface can be augmented to allow for runtime changes to be made to the data e.g. interactively hiding, permanently removing or sorting nodes within the hierarchy, so that changes to the input data file can be effected from within the visualization's interface. Allowing for non-numeric attributes (where interactive transformation, of text data for example, to numeric data is allowed) and aggregate attributes (where a pseudo numeric attribute can be created as the sum of other numeric attributes for example, or multiple single amino acids can be concatenated into a single attribute) could also reduce the work involved in data preparation prior to each analysis.

More complex node selection techniques would also assist in allowing data analysts to obtain custom lists of nodes of interest from the visualization. Currently individual selection is allowed, but more complex structure-aware brushing, rectangular or polygonal selection capabilities could prove useful.

It was hoped that the area within each individual node could be leveraged, by producing rectangular visualizations within them such that nodes of interest (those that are wider or longer) could show additional information when expanded. One possible use would have been a line graph that shows the distribution of a particular data attribute over time e.g. bytes transferred per minute within an hour of netflows for each quartet of source and destination IP and port. This would have had the unique effect of aggregating graph values for parent nodes e.g. total bytes transferred per minute within an hour between two IP addresses across all ports. An efficient and real-time implementation of this would be challenging.

As most analyses are never conducted solely by one individual [72], collaborative techniques would greatly assist in allowing the analyst to quickly markup a diagram (to say highlight nodes of interest with questions or comments), save the state of his analysis (the degree and direction of hyperbolic distortion, any torn away subdiagrams and radial attribute assignments), and forward this to a colleague for assistance and feedback. A web-based interface and storage solution for this information would greatly enhance a collaborative workflow of this type.

6.1.2 On the Network Security Case Study

As a lot of the network security data were not filtered initially, many of the produced DataBurst diagrams contained various degrees of normal benign data that could have been safely ignored. It is safe to assume that network security analysts would have sufficient knowledge to remove the majority of such data so that any netflow data that is chosen to be visualized would be questionably anomalous or unknown data.

Future research work in this area would therefore involve more intimate knowledge of all allowed services in netflows captured within a particular network, such that we can enforce filters that are known to reduce the amount of normal data netflow to be rendered.

6.1.3 On the Genetic Data Study

A conditionally weighted mutual information score, as proposed by MacDonald and Beiko in [46], may assist in reducing the list of highly scoring sites to investigate (from the 86 NA sequences and 102 HA sequences having a score above 0.5 with respect to adamantane resistance). This incorporates phenotype and taxonomic information, and would lower scores where the confounding effect of common ancestry is present. Another optional scoring method is Minimum Redundancy Maximum Relevance (mRMR [20]), which chooses sites that are mutually distant from each other while still being highly correlated to a chosen phenotype.

Further analysis would involve the implementation of the above and any other scoring options that can better clarify results obtained. Further investigating the phylogenetic trees of both proteins may also uncover additional insights. The unexplained interesting observations made in the analysis (e.g. the increased score of position 354, and the lower score of position 275 when incorporating phylogenetic data with the NA protein and oseltamivir resistance) will also be investigated.

Another investigation that could be performed would be comparison, possibly using mutual information again, of sites with other sites (as opposed to sites with phenotypes). This may shed some light on why there are Y275 non-swine-origin NA proteins that were sensitive to oseltamivir, given that the H275Y mutation was seen to confer resistance. It will also give a better indication how changes in any of the high scoring positions with respect to adamantane resistance in both the NA and HA proteins are related to each other.

And finally, future analyses could be performed on other proteins of the influenza A virus (multiple substitutions in the transmembrane domain of the M2 protein have made some strains resistant to adamantane [1]) and on the NA and HA proteins with other drugs (while zanamivir has proven highly effective in the past, the World Health Organization has acknowledged unique mutations in the NA gene that have produced zanamivir resistance [33]).

Bibliography

[1] Y. Abed, N. Goyette, and G. Boivin. Generation and characterization of recombinant influenza a (h1n1) viruses harboring amantadine resistance mutations. *Antimicrobial Agents and Chemotherapy*, 49(2):556–559, Feb 2005.

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell, 4th Edition*, chapter 1, The Diversity of Genomes and the Tree of Life. Garland Science, 2002.

[3] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner. State of the practice of intrusion detection technologies. *CMU/SEI-99-TR-028, Carnegie Mellon University*, 2000.

[4] K. Andrews and H. Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. *InfoVis '98. Proceedings of InfoVis Late Breaking Hot Topic Papers*, Oct 1998.

[5] M. Ankerst, D. A. Keim, and HP Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Proceedings of Visualization '96, Hot Topic Session*, San Francisco, CA, 1996.

[6] R. Ball, G. A. Fink, and C. North. Home-centric visualization of network traffic for security administration. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 55–64, New York, NY, USA, 2004. ACM.

[7] N. Beerenwinkel, M. Däumer, M. Oette, K. Korn, D. Hoffmann, R. Kaiser, T. Lengauer, J. Selbig, and H. Walter. Geno2pheno: Estimating phenotypic drug resistance from hiv-1 genotypes. *Nucleic Acids Research*, 31(13):3850–3855, July 2003.

[8] N. Beerenwinkel, B. Schmidt, H. Walter, R. Kaiser, T. Lengauer, and Hoffmann D. Diversity and complexity of hiv-1 drug resistance: A bioinformatics approach to predicting phenotype from genotype. *Proceedings of the National Academy of Sciences USA*, 99(12):8271–8276, June 2002.

[9] E. Bertini, P. Hertzog, and D. Lalanne. Spiralview: Towards security policies assessment through visual correlation of network resources with evolution of alarms. In *VAST '07: Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 139–146, Washington, DC, USA, Nov 2007. IEEE Computer Society.

[10] R.A. Cava, P.R.G. Luzzardi, and C.M.D.S. Freitas. The bifocal tree: A technique for the visualization of hierarchical information structures. In *Proceedings of the*

Workshop on Human Factors in Computer Systems (IHC), Fortaleza, Brazil, 2002.

[11] Centers for Disease Control and Prevention (CDC). Oseltamivir-resistant novel influenza a (h1n1) virus infection in two immunosuppressed patients - seattle, washington, 2009. *MMWR. Morbidity and mortality weekly report*, 58(32):893–896, April 2009.

[12] D. K. Chiu and T. Kolodziejczak. Inferring consensus structure from nucleic acid sequences. *Computing Applications in Biosciences*, 7(3):347–352, July 1991.

[13] M. Chuah. Dynamic aggregation with circular visual designs. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 35–43,151, Los Alamitos, CA, USA, October 1998. IEEE Computer Society.

[14] N. Collin and X. de Radigues. Vaccine production capacity for seasonal and pandemic (h1n1) 2009 influenza. *Vaccine*, 27(38):5184–5186, Aug 2009.

[15] C. Collins, S. Carpendale, and G. Penn. Docuburst: Visualizing document content using language structure. *Computer Graphics Forum*, 28:1039–1046(8), June 2009.

[16] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley-Interscience, 1st edition, August 1991.

[17] C. Daub, J. Weise, R. Steuer, J. Kopka, and S. Kloska. Using b-spline functions to introduce fuzzyness to mutual information based analysis of gene-expression data. *BMC Bioinformatics*, 118(5), 2002.

[18] D. E. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, 13(2):222–232, 1987.

[19] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualizations of Graphs 1st Edition.* Prentice Hall, Upper Saddle River, NJ, 1999.

[20] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *CSB '03: Proceedings of the IEEE Computer Society Conference on Bioinformatics*, page 523, Washington, DC, USA, 2003. IEEE Computer Society.

[21] G. M. Draper and R. F. Riesenfeld. Interactive fan charts: A spacesaving technique for genealogical graph exploration. In *Proceedings of the 8th Annual Workshop on Technology for Family History and Genealogical Research (FHTW 2008).* Brigham Young University, 2008.

[22] R. C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, March 2004.

[23] J.P. Egan. Signal detection theory and roc analysis. Academic Press, 1975.

[24] R. F. Erbacher. Glyph-based generic network visualization. In *Proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference on Visualization and Data Analysis*, volume 4665 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 228–237, January 2002.

[25] R. J. Garten et al. Antigenic and genetic characteristics of swine-origin 2009 a (h1n1) influenza viruses circulating in humans. *Science Express (New York, N.Y.)*, 325(5937):197–201, May 2009.

[26] C. Gates, M. Collins, M. Duggan, A. Kompanek, and M. Thomas. More netflow tools: For performance and security. In *In Proceedings of the 18th Large Installation Systems Administration Conference (LISA 2004).*, pages 121–132, Nov 2004.

[27] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi. Preserving the big picture: Visual network traffic analysis with tn. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 6, Washington, DC, USA, Nov 2005. IEEE Computer Society.

[28] R. R. Gutell, A. Power, G. Z. Hertz, E. J. Putz, and G. D. Stormo. Identifying constraints on the higher-order structure of rna: continued development and application of comparative sequence analysis methods. *Nucleic Acids Research*, 20(21):5785–5795, November 1992.

[29] C. Gutwin. Improving focus targeting in interactive fisheye views. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 267–274, New York, NY, USA, 2002. ACM.

[30] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.

[31] T.C. Hodgman. A historical perspective on gene/protein functional assignment. *Bioinformatics*, 16(1):10–15, January 2000.

[32] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12:741–748, 2006.

[33] A. C. Hurt, J. K. Holien, M. Parker, A. Kelso, and I. G. Barr. Zanamivir-resistant influenza viruses with a novel neuraminidase mutation. *Journal of Virology*, 83(20):10366–10373, October 2009.

[34] B. Irwin and N. Pilkington. High level internet scale traffic visualization using hilbert curve mapping. In *VizSEC 2007*, Mathematics and Visualization, pages 147–158. Springer Berlin Heidelberg, 2008.

[35] M. Kaufmann and D. Wagner. *Drawing Graphs: Methods and Models (Lecture Notes in Computer Science, Vol. 2025).* Springer, January 2001.

[36] D. A. Keim, J. Schneidewind, and M. Sips. Circleview: a new approach for visualizing time-related multidimensional data sets. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 179–182, New York, NY, USA, 2004. ACM.

[37] D.A. Keim, F. Mansmann, J. Schneidewind, and T. Schreck. Monitoring network traffic with radial traffic analyzer. *Symposium On Visual Analytics Science And Technology*, pages 123–128, 2006.

[38] H. Koike and K. Ohno. Snortview: visualization system of snort logs. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 143–147, New York, NY, USA, 2004. ACM.

[39] J. Koziol. *Intrusion Detection with Snort.* Sams, May 2003.

[40] K. Lakkaraju, W. Yurcik, and A. J. Lee. Nvisionip: netflow visualizations of system state for security situational awareness. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 65–72, New York, NY, USA, 2004. ACM.

[41] J. Lamping and R. Rao. Visualizing large trees using the hyperbolic browser. In *Conference companion on Human factors in computing systems: common ground*, CHI '96, pages 388–389, New York, NY, USA, 1996. ACM.

[42] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins. Clustal w and clustal x version 2.0. *Bioinformatics*, 23(21):2947–2948, November 2007.

[43] S. Lau. The spinning cube of potential doom. *Communications of the ACM*, 47(6):25–26, 2004.

[44] C. P. Lee, J. Trost, N. Gibbs, R. Beyah, and J. A. Copeland. Visual firewall: Real-time network security monito. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 16, Washington, DC, USA, 2005. IEEE Computer Society.

[45] N. MacDonald, D. Parks, and R. Beiko. Seqmonitor: Influenza analysis pipeline and visualization. *PLoS Currents: Influenza*, Sept 2009. RRN1040. PMCID: PMC2762774.

[46] N. J. MacDonald and R. G. Beiko. Efficient learning of microbial genotype-phenotype association rules. *Bioinformatics*, 26(15):1834–1840, August 2010.

[47] F. Mansmann, F. Fischer, D. A. Keim, and S. C. North. Visualizing large-scale ip traffic flows. In *Proceedings of 12th International Workshop Vision, Modeling, and Visualization*, pages 23–30, Nov 2007.

[48] F. Mansmann and S. Vinnik. Interactive exploration of data traffic with hierarchical network maps. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1440–1449, Nov 2006.

[49] J. McHugh. Intrusion and intrusion detection. *International Journal of Information Security*, 1:14–35, 2001. 10.1007/s102070100001.

[50] J. McPherson, KL. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. Portvis: a tool for port-based detection of security events. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 73–81, New York, NY, USA, 2004. ACM.

[51] S. B. Mossad. The resurgence of swine-origin influenza a (h1n1). *Cleveland Clinic Journal of Medicine*, 76(6):337–343, June 2009.

[52] P. Palese. The genes of influenza virus. *Medicine*, 10(1):1–10, Jan 1977.

[53] D. Parks, N. MacDonald, and R. Beiko. Tracking the evolution and geographic spread of influenza a. *PLoS Currents: Influenza*, Sept 2009. RRN1014. PMCID: PMC2762414.

[54] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(23-24):2435–2463, 1999.

[55] D. Phan, J. Gerth, M. Lee, A. Paepcke, and T. Winograd. Visual analysis of network flow data with timelines and event plots. In *VizSEC 2007*, Mathematics and Visualization, pages 85–99. Springer Berlin Heidelberg, 2008.

[56] R. M. Pielak, J. R. Schnell, and J. J. Chou. Mechanism of drug inhibition and drug resistance of influenza a m2 channel. *Proceedings of the National Academy of Sciences*, 106(18):73797384, May 2009. PMCID: PMC2678642.

[57] C. Plaisant. The challenge of information visualization evaluation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 109–116, New York, NY, USA, 2004. ACM.

[58] G. G. Robertson and J. D. Mackinlay. The document lens. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 101–108, New York, NY, USA, 1993. ACM.

[59] M. Sarkar and M. H. Brown. Graphical fisheye views of graphs. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 83–91, New York, NY, USA, 1992. ACM.

[60] K. Scarfone and P. Mell. Guide to intrusion detection and prevention systems (idps). In *Technical Report CSRC special publication SP 800-94*. National Institute of Standards and Technology (NIST), 2007.

[61] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, September 1998.

[62] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[63] A. Stamatakis. Raxml-vi-hpc: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, November 2006.

[64] J. Stasko. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computer Studies*, 53(5):663–694, 2000.

[65] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. *InfoVis 2000. IEEE Symposium on Information Visualization*, 0:57, 2000.

[66] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics*, 18(90002):S231–240, October 2002.

[67] S. M. Stroh, J. Ma, R. Lee, F. Sirota, and F. Eisenhaber. Mapping the sequence mutations of the 2009 h1n1 influenza a virus neuraminidase relative to drug and antibody binding sites. *Biology Direct*, 4(18), May 2009.

[68] J. W. Tang, P. A. Tambyah, F. Y. Lai, H. K. Lee, C. K. Lee, T. P. Loh, L. Chiu, and E. S. Koay. Differing symptom patterns in early pandemic vs seasonal influenza infections. *Archives of Internal Medicine*, 170(10):861–867, May 2010.

[69] T. Taylor, D. Paterson, J. Glanfield, C. Gates, S. Brooks, and J. McHugh. Flovis: Flow visualization system. In *In Proceedings of the Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH).*, 2009.

[70] D. Wang and B. Larder. Enhanced prediction of lopinavir resistance from genotype by use of artificial neural networks. *The Journal of Infectious Diseases*, 188(5):653–660, September 2003.

[71] K. Wolf, H. Walter, N. Beerenwinkel, W. Keulen, R. Kaiser, D. Hoffmann, T. Lengauer, J. Selbig, AM. Vandamme, K. Korn, and B. Schmidt. Tenofovir resistance and resensitization. *Antimicrobial Agents and Chemotherapy*, 47(11):3478–3484, November 2003.

[72] J. Wood, H. Wright, and K. Brodlie. Cscv - computer supported collaborative visualization. In *Proceedings of BCS Displays Group International Conference on Visualization and Modelling*. Academic Press, 1995.

[73] Z. Xia, G. Jin, J. Zhu, and R. Zhou. Using a mutual information-based site transition network to map the genetic evolution of influenza a/h3n2 virus. *Bioinformatics (Oxford, England)*, 25(18):2309–2317, September 2009.

[74] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. *Infovis 2002. IEEE Symposium on Information Visualization*, pages 77–84, 2002.

[75] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju. Visflowconnect: netflow visualizations of link relationships for security situational awareness. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 26–34, New York, NY, USA, 2004. ACM.

[76] S. M. Zimmer and D. S. Burke. Historical perspective – emergence of influenza a (h1n1) viruses. *New England Journal of Medicine*, 361(3):279–285, July 2009.