Understanding Atmospheric Chemistry using Graph-Theory, Visualisation and Machine Learning.

Dan Ellis

March 2020

Veritatem inquirenti, semel in vita de omnibus, quantum fieri potest, esse dubitandu:
In order to seek truth, it is necessary once in the course of our life, to doubt, as far as possible, of all things.

- Descartes, Rene, *Principles of Philosophy*

Contents

Chapter 1

Applying Visual Analytics to the Atmospheric Chemistry Network

“ I have a notion that when the mind is thinking, it is simply talking to itself, asking questions and answering them. ”

- Socrates, *The collected dialogues of Plato*

1.1 Introduction

?? viewed the importance of a carefully selected visualisation/metaphor in the representation of scientific data. One such category is that of relational data, where we have a set of items, joined by a chosen relationship. Historically this type of problem has often been solved through the use of sociographs to show a set of items and the links between them.

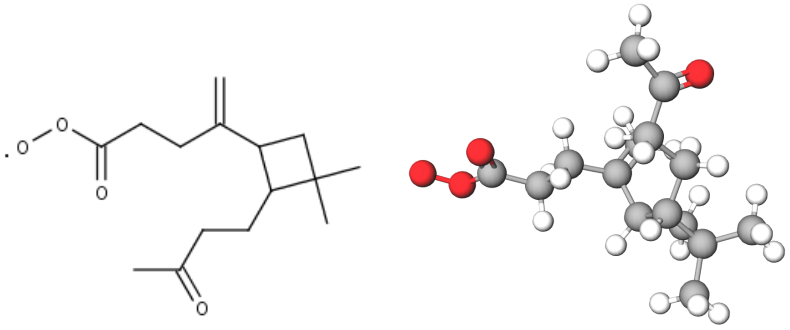
This chapter begins by looking at the use of sociograms in chemistry (Subsection 1.1.2) and the different ways in which these can help convey information to the reader (Section 1.2, Section 1.3). These sections find the force-directed graph to be the most suited for representing the chemical reactions within a mechanism, and therefore this shall be applied to the network of reactions representing the chemistry within an urban environment - Beijing (Section 1.4).

1.1.1 Networks And Their Role In Visual Analytics

Networks are present everywhere - this ranges from interactions within social media to bank transactions, internet routing, genetics to epidemiology [Martin Grandjean, 2016; Staples et al., 2013; Needham and Hodler, 2019; Baronchelli et al., 2013; Sangers et al., 2019; Kohlbacher et al., 2014; Archambault et al., 2014; Schreiber et al., 2014]. This is because the sociogram (or graph) structure may be applied to any set of items which contain one or more relationships between them. In visualisation, these 'items' are often referred to as nodes/vertices, and their relationships as edges/links [Kerren et al., 2014]. These terms will be used interchangeably throughout this thesis.

1.1.2 Graphs In Chemistry

Node-link representations have been at the core of chemistry for many years. They have been used to show the bonds between atoms and are integral to the representation of molecules - both physically (with the aid of molecular model kits) or pictorially to show various structural properties (Figure 1.1). These graph-like analogies provide a pseudo-physical representation of the molecules and their reactions in a way that is intuitive to the user. Subsubsection 1.1.2.1 shows how the sociograph structure is used to represent reactions within the troposphere, however this method of representation is not contrained to atmospheric science - for example Figure 1.3 depicts the biochemical metabolic pathways of the human body. This is an example of another complex chemical network that benefits from this method of representation.



(a) 2D 'classic' (b) 3D

Figure 1.1: **The molecule $C_{141}CO_3$ shown in both 2D and 3D node-link structures.** This is a the result of a series of inorganic species reactions and a desocciation from BCARY - the only sesqueterpine in the MCM. 3D visualisation by [Bergwerf, 2019].

1.1.2.1 Using Sociograms To Describe Reactions

A collection of reactions representing the chemistry of a region is called a mechanism. The Master Chemical Mechanism [**?**] provides a collection of equations describing the gas-phase chemistry which exists within the troposphere (**??**). In its use in policy, and the evaluation of Air Quality Models ([Dick Derwent, 2010]), it is often useful to understand the degradation process different VOCs undergo. In general, this may be done using a series of interconnected reactions in the form of a reaction cycle (Figure 1.2). This type of sociograph shows the directional nature of chemical reactions and the relationships between different species. This has many similarities to a conventional directed graph, except that species (nodes) are sometimes duplicated (for example OH, HO_2, O_2 in Figure 1.2) to aid in the clarity of the figure.

This is an excellent example of how the flow-like nature of a sociogram aids in the understanding of a potentially complex chemical system of 171 organic species and 600 reactions. Evolutionary traits, including the genetic predisposition to interpret shapes faster than text ([Harari, 2015]) make the graph structure a much better method for representing such a system.

1.1.3 Modeling Chemistry As A Directed Graph.

Historically it is shown that the graph format has proven to be an efficient means of understanding the reactions within a mechanism. Traditionally these are constructed manually, with the designer making a series of choices on how best to place, and simplify the chemistry based on their application. As our understanding of chemistry improves and we have started to progress into automated and semi-automated mechanism construction. This makes the construction of mechanisms with tens of millions of species and billions of reaction possible ([Aumont et al., 2005]) and is the point where the manual design/simplification of reaction networks becomes infeasible.

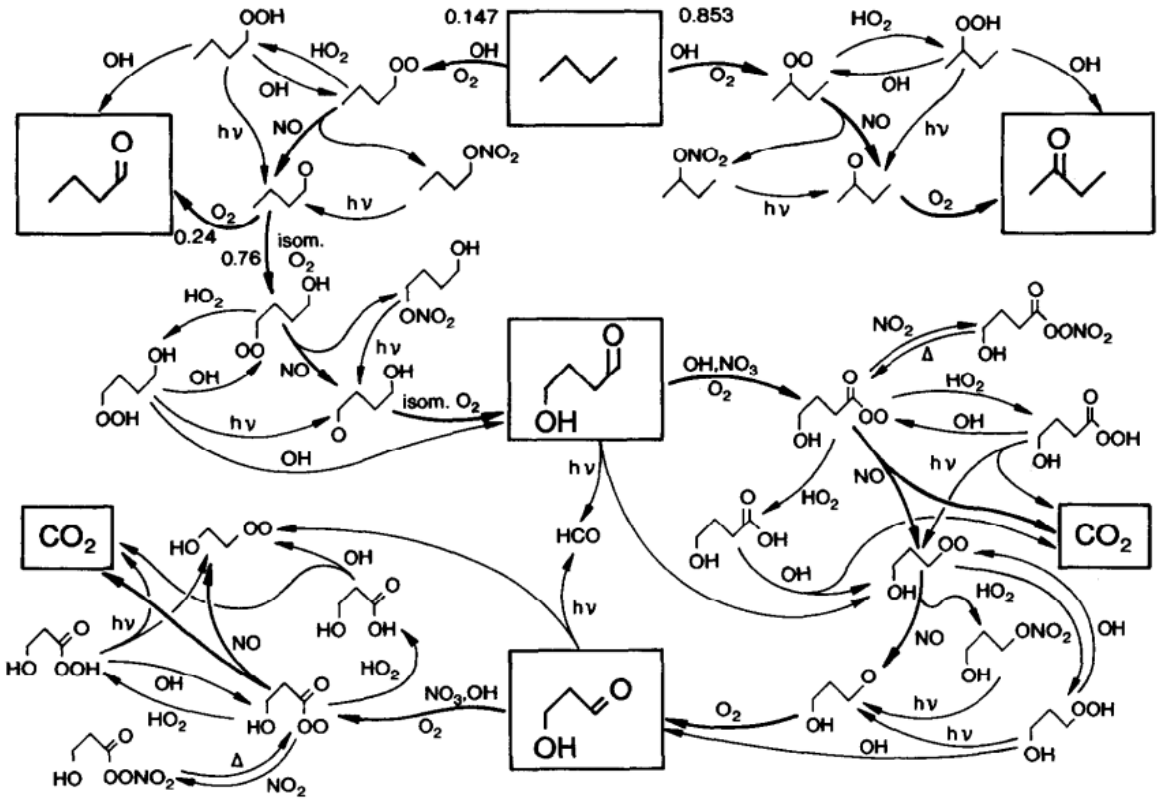


Figure 1.2: **A systematic representation of the degregation of butane.** Using this we are able to see the process C_4H_{10} undergoes before its ultimate demise as carbon monoxide and water. Source: [Jenkin et al., 1997]

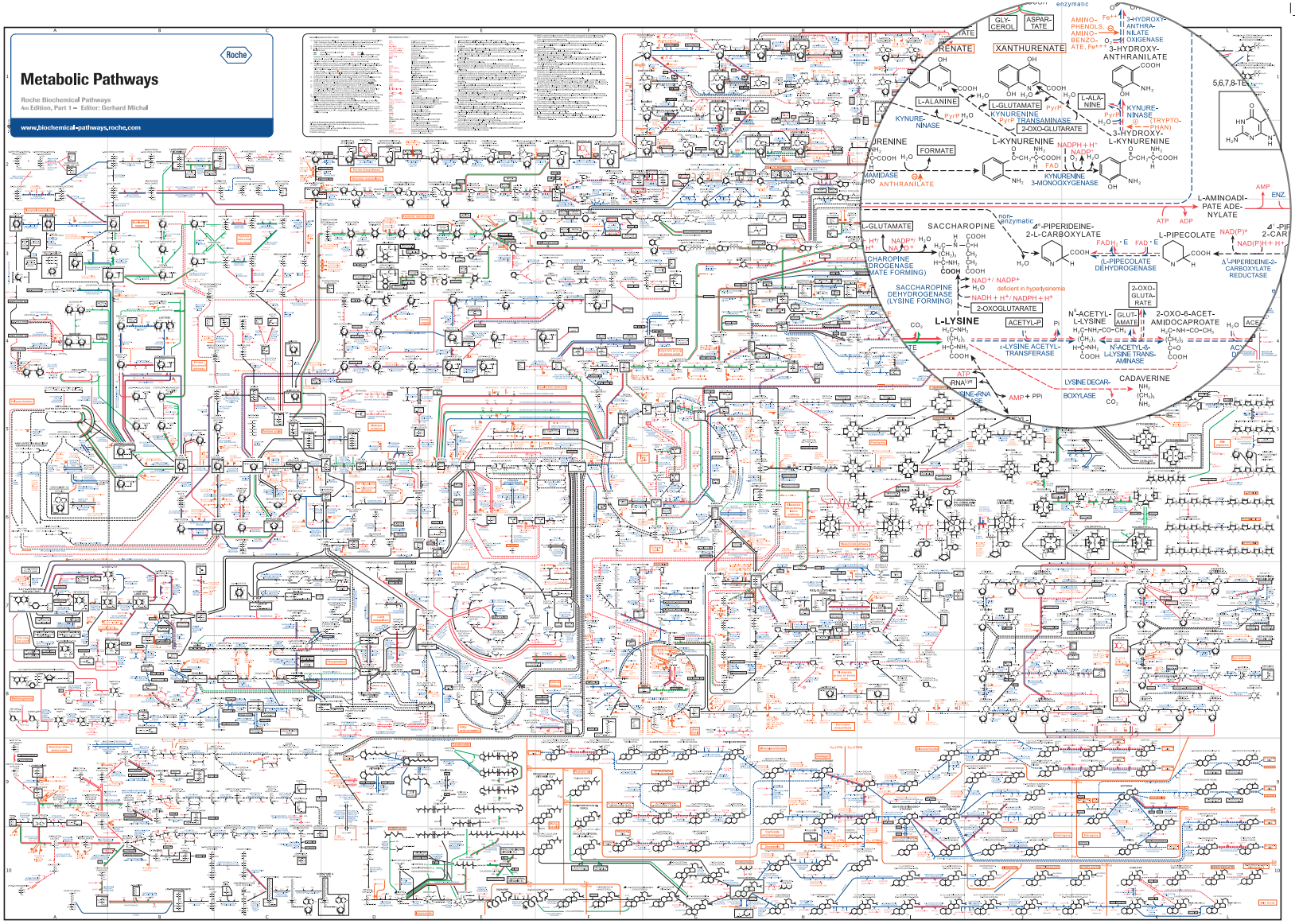


Figure 1.3: **The Roche Metabolic Pathways of the human body.** This example demonstrates the ability to manually represent the complex chemistry of the body using a graph structure. (Original A0 version is available at the source). Source: [Michal, 1965]

Today automatic graph layouts allow us to generate multivariate and complex graphs quickly [Muelder et al., 2014] -This means that, much like in the construction of a mechanism, we can rely on computer-aided design to generate a directed graph representation of the chemistry. Montañez [2016] states that "The beauty of a good information graphic is that it can tell a whole story in a single unit of visual content". This is particularly true for the use of directed graphs in chemistry where we can compare different mechanism subsets,(**??**) or model simulations (**??**).

However, several problems emerge from the complete automation of a task. Firstly real-world data very rarely reacts how it is expected to. Here networks of high edge density often obfuscate the graph data and produce what is only described as a 'birds nest', 'hairball' or 'ball of yarn' within the literature [Roberts et al., 2014]. Although such problems can be shown as moments of turbulence, they encourage a greater understanding of the graphic design process and can catalyze to merge unique ideas into an effective visualisation [Johnson, 2010] - much like the composite metaphors in **??**.

Having established that a graph network ties in both modern and historical methods for representing relational data, we now look at how to present the graph, both in syntax (Section 1.2) and semantics (Section 1.3).

1.2 Graph Syntatics

Syntactic representation considers how best to distribute information on a page for maximum impact. This can be seen between the force-directed graph (top) and geographical location (bottom) layouts in Figure 1.4. Although the geographical layout gives a more accurate representation of the distances between unconnected nodes (airports), a force-directed graph provides greater insight into the relationships (flights) between each airport. This highlights the importance of choosing a suitable syntactic representation to highlight the features of interest. The remainder of this section discusses the syntactic choices required for the visualisation of a complex chemical mechanism.

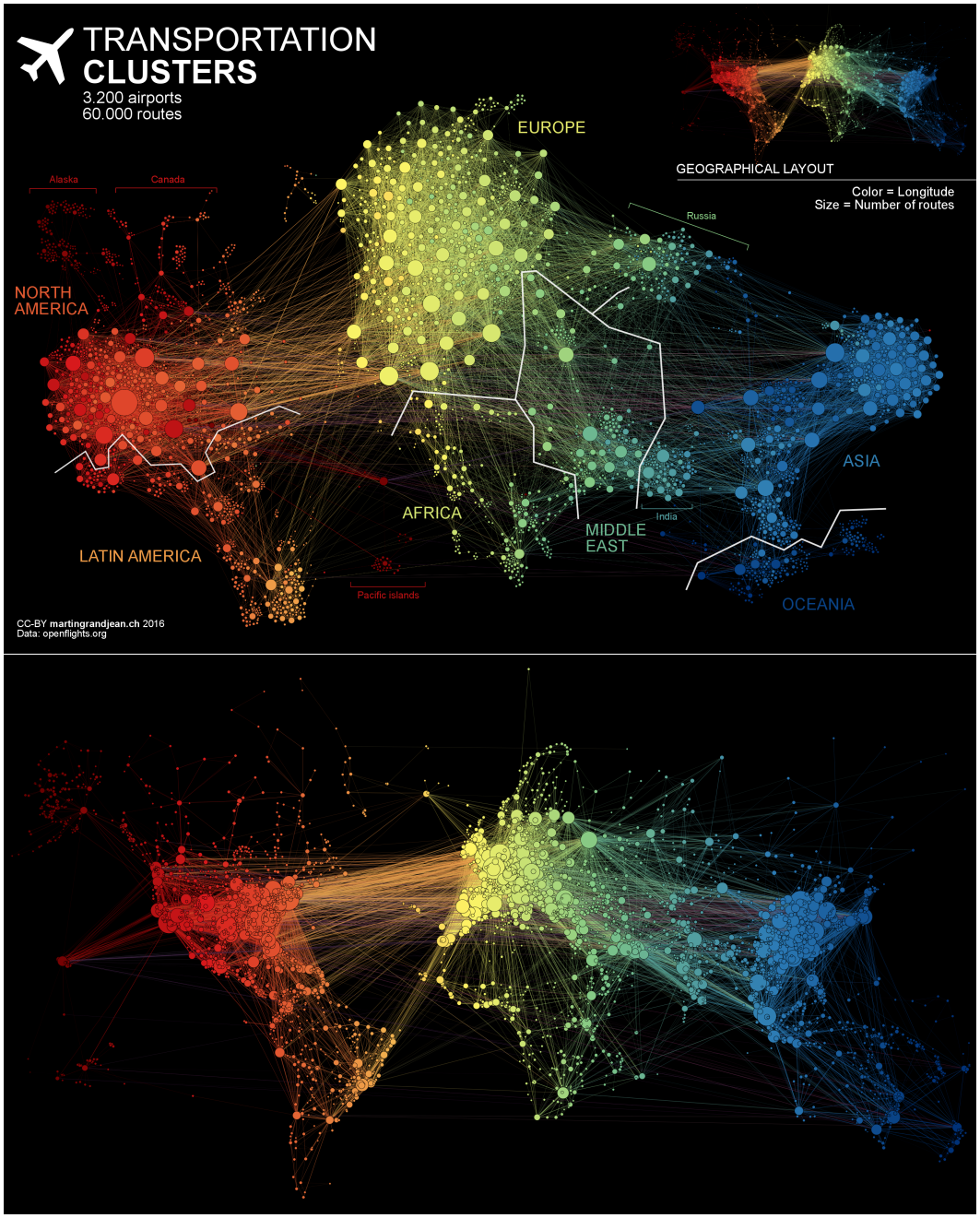


Figure 1.4: **Comparison of different representations of flight data by [Martin Grandjean, 2016].** The top figure shows the data represented by a force-directed graph layout (described below) and a Geo-layout showing each point at its location on the Earth.

1.2.1 Selecting The Correct Evaluation Criteria.

As chemical networks provide a wealth of information on the reactions within a system, this can prove challenging to user cognition and computational resources [Kerren et al., 2014]. In selecting the best possible graph layout, there are many metrics designed around the improving of visualisations aesthetics [Purchase, 2002] however, these have often only been evaluated with a handful of criterions in mind. Such metrics can make it difficult to accurately quantify the changes in user-readability,

especially if they are not treated as originally intended [Pohl et al., 2009].

Edge Crossing

One of the greatest limitations to understanding a graph is the number of overlaping (crossing) edges [Purchase, 1997], especially since users often spend most of their time looking at the edges of a graph in order to understand it [Pohl et al., 2009].

There exist several type of graph layout algorims which aim to reduce the amount of overlapping edges in a graph. The two most common ones are force-directed and orthogonal. Orthogonal designs are those of straight edges at 90 degree angles, such as in architectural or circuit schematics (**??**). Force directed graphs (Subsubsection 1.2.2.3) are a graph layout is designed to simulate a phsical system, where node positions are the result of the push and pull of the edges between them. In the task selecting nodes from a specific path, users were twice as more accurate using this layout than the orthogonal one [Pohl et al., 2009].

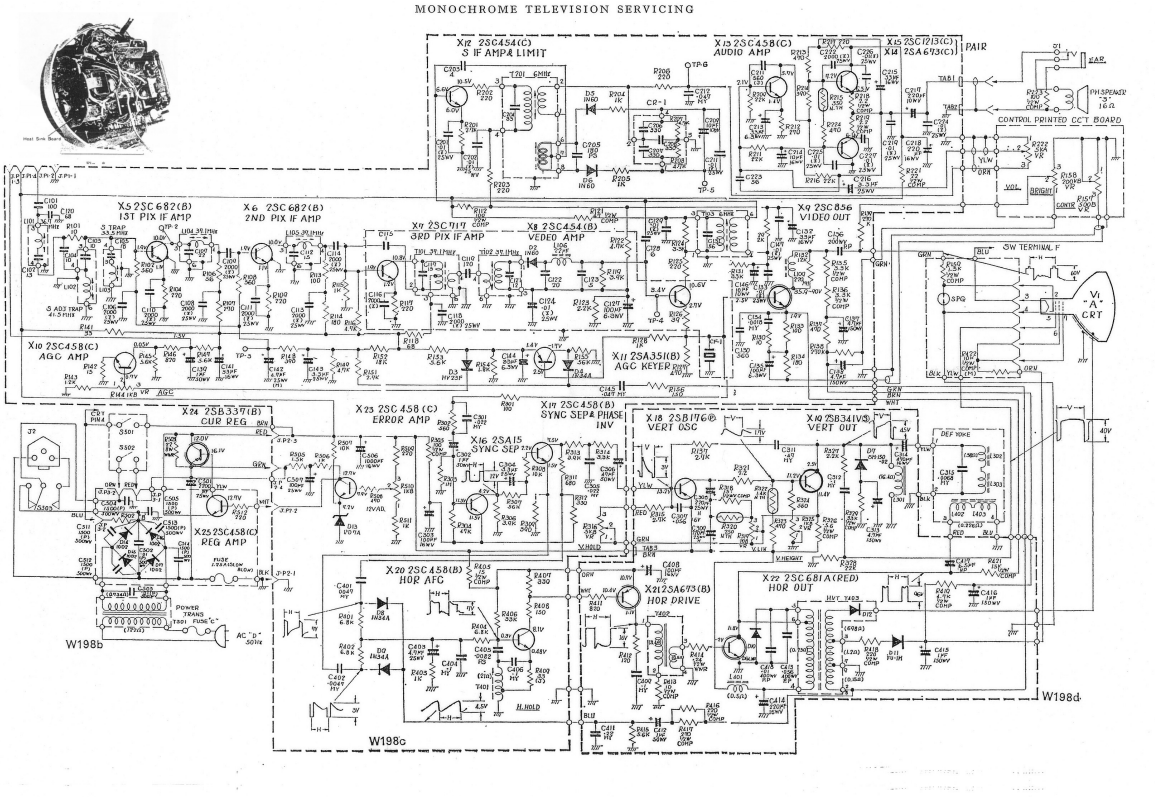


Figure 1.5: **An orthogonal circuit schematic of the Model 3 240 portable cathode ray tube television.** The circuit schematic of the television (top left) shows a much simpler representation of how different components within the television are connected. Source: [**?**]

Node Distribution And Overlap

The distribution of nodes across the page can both hinder or increase the readability of a graph - especially since larger nodes may obfuscate smaller ones in the same location. Generating a graph of an equal node distribution with medium edge length was found to greatly imporve the 'flow' of the graph. Coupling this with graphical-symmetry, this forms the second most important user-ranked preference [Purchase et al., 2003].

In addition to the selection of a layout with a better distribution of nodes, there are several methods in which overlapping nodes may be removed. Although many algorithms aim to reduce this, the treatment of nodes as 'point masses' make it difficult to separate points in a nearby location [Dwyer et al., 2006c]. Dwyer et al. [2006b] explains that there are usually two methods for reducing the number of overlapping nodes in a graph, these are:

1. Create a layout design capable of taking node size (e.g. [Friedrich and Schreiber, 2004]) into consideration. These designs tend to be layout specific and not absolute in removing all overlap between nodes.

2. This requires a level of post-processing in the form of a 'layout adjustment'. Here we reposition nodes after a chosen layout has finished computing. The drawback of this method is that information contained in the graph's shape may be degraded. This can be done through the use of collision detection, or moving nodes to the centre of the vernouli cells [Lyons, 1992].

1.2.2 Automated Graph Drawing Layouts

In their design and evaluation, automatic graph drawing algorithms are created to minimise a specific criterion. This section compares a number of graph drawing algorithms to determine which of these is best suited for the representation of an atmospheric chemistry model. For this task, an MCM subset representative of the VOCs in Beijing (outlined in **??**) is used to providea real-world case study which may be simulated in a chemical model. In this subsection we start with manual hand-drawn, and map inspired graph layouts (**??**) and end at automated force-directed graphs (Subsubsection 1.2.2.3), describing the merits of each layout.

1.2.2.1 Replication Of Hand-Drawin Methods

With the rise of computation, many traditional visualisations adapted for the computer-aided generation. Fields of architecture and circuit design adopted computational software to alleviate some of the difficulties presented by large or complex designs. Similar ideas such as the use of automatically

generated transit maps can be used to link chronological or topological items such as ideas [Foo, 2019]. Figure 1.6 shows all the possible paths for the oxidation of methane to produce carbon dioxide (and water), using the MemoryMap algorithm Foo [2019]. Although such methods can be useful in showing isolated pathways, they provide a convoluted representation of large interconnected systems and require some manual intervention.

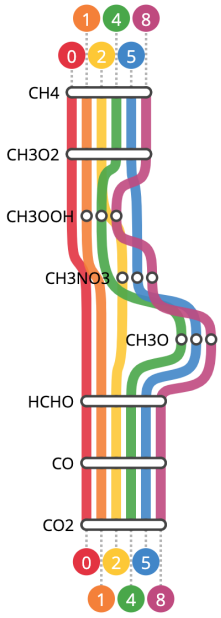
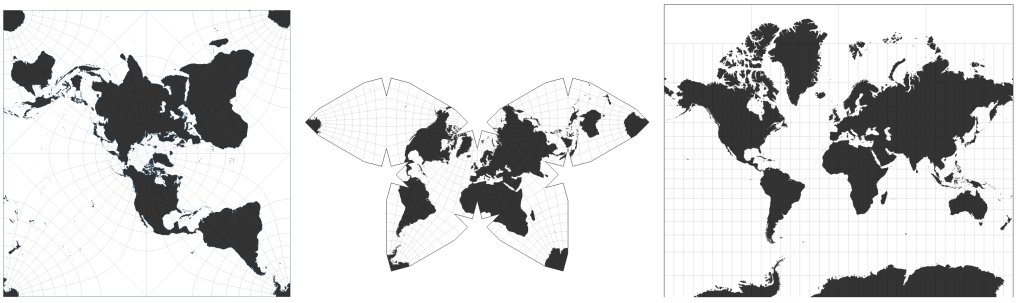


Figure 1.6: **A transit map showing all the possible routes from methane to carbon dioxide.** This was drawn using MemoryMap [Foo, 2019] and uses a version of the MCM methane subset, where carbon dioxide has been introduced.

1.2.2.2 Projection Based

One of the oldest fields of data visualisation fall in the realm of cartography. Here the shapes and distances between points on the surface of the earth (an oblate spheroid) are mathematically mapped onto a 1D plane for graphing purposes [Thomas, 1952]. Since the process of dimensionality reduction will produce inherent distortions within the final product, we end up with a range of map projections, with each striving to achieve a different aim (Figure 1.7). The Pierce Quincuncial, for example, is a conformal mapping technique mapping the surface of a sphere to a square with minimal deviation in scale and the ability to be tessellated in all directions. The Mercator, on the other hand, is a cylindrical projection which grew in popularity due to its unique ability to represent any course of constant bearing[1] as a linear segment within the shipping and navigation industry. Finally the waterman butterfly presents the globe as a truncated octahedron. This allows for the reconstruction of a 3 dimensional word from a 2D plane (ie a printed sheet).

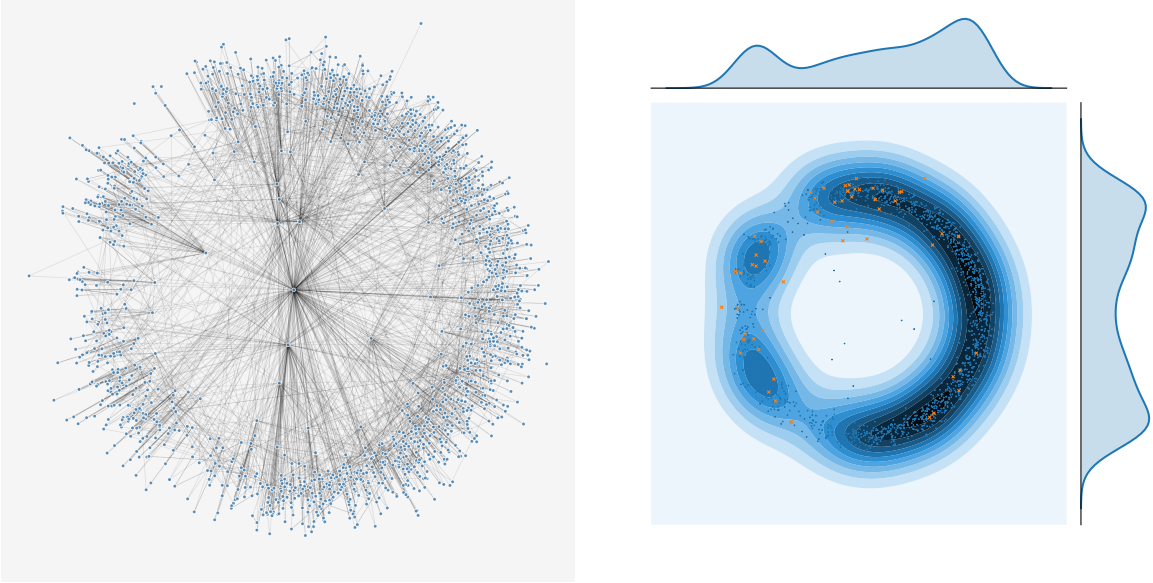
[1]Also known as a 'rhumb', or 'loxodrome', and consists of an arc crossing all meridians of longitude at the same angle.



(a) Pierce Quincucial (b) Waterman Butterfly (c) Mercator

Figure 1.7: **A selection of map projections.** These have been created using DataDrivenDocuments ([**?**]) and show a range of methods for mapping the spheroid shape of the Earth onto a 2D plane.

More recently, the mathematics of mapping a large dimension onto a simpler one has been applied to the problem of graph representation. [García-Pérez et al., 2019] uses the latent hyperbolic geometry of the Mercator layout to provide a 2D embedding for complex real-world networks. This produces a polar representation (r and θ) of the system, where relationships of related species are of the same angle (θ), with nodes of a high degree are closer to the centre (low r value, where r is the radius from the centre). Using the chemical mechanism from the APHH Beijing campaing (described above), this produces a layout, (Figure 1.8) where (a) shows the graph-based representation including links, and (b) shows the density distribution for all nodes. Figure 1.8b shows that primary emitted species (orange dots) are uniformly (radially) distributed for angles and Figure 1.8a reveals that influential nodes with a high degree (highly connected) are located close to the centreof the graph. Although the Mercator embedding does reduce the 'hairball' problem experienced by other layouts, it does not take edge weight/direction or self-loops. This means that it works well for the representation of the general network layout, but cannot be used for advanced data exploration concerning simulation results.



(a) The Mercator graph.

(b) Mercator species distribution

Figure 1.8: **The Mercator Projection.** (a) represents output from the mercator graph layout algorithm. (b) provides a kernel density analysis of the node distribution within this. Here (a) shows graph structure by revealing the density of connections between different nodes, whilst (b) reveals the density of nodes at a specific location.

1.2.2.3 Force-Directed

Force-directed graph layouts are the results of the Spring-Electrical model. This was first introduced by [Eades, 1984] and further improved by [Fruchterman and Reingold, 1991]. Force-directed layouts are in essence a simple physics simulation of like-charged particles represnting the nodes. These particles act similarly to protons which experience Coulomb repulsion and try to get away from each other. If there is a relationship between two nodes, a spring-like attractive force is introduced, drawing the nodes back together.

In the case of a weighted graph (where each link (or relationship) has a value associated with it, we can adjust the spring coefficient of the attractive force to reflect this. This results in a layout where strongly connected objects are drawn together, and weakly connected ones further away. Uses for this type of representation have been shown biology, social networks, and with this thesis atmospheric chemistry [Muelder et al., 2014; Kohlbacher et al., 2014].

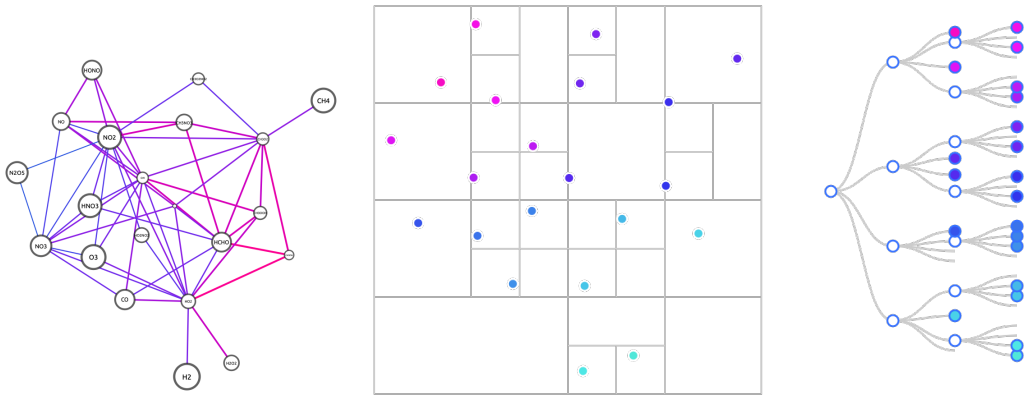
Next we describe the Barnes-Hut algorithm, a mapping algorithm which builds a hierarchical tree of the data by splitting a plane into quartiles. This is used within the many force-directed graph layouts, including those of Force Atlas 2 and Yifan Hu, described shortly. Once this has been done a selection of four different layout algorithms shall be discussed.

Barnes Hut Algorithm

Since calculating the attractive/repulsive forces for each node of a large graph can be computationally intensive, many force-directed layouts rely on the Barnes-Hut approximation. This solves the N-body problem of pairwise reactions between nodes, $O(n^2)$, by approximating long-range reactions by grouping such nodes and applying a single action on their centre of mass- reducing the computational time to $n \log n$.

To do this, first, a spatial index of each node is constructed (see below). This can either be done using a quadtree (2D) or octree (3D).Followign this we calculate the centre(s) of mass, allowing us to aproximate the repulsive forces of a force-directed graph.

Quadtree Construction: A quadtree is the recursive partitioning of two-dimensional space into a set of quadrants (a set of 4 squares). This process is repeated, with each square then being divided into four itself, until there is only a single point within a cell[2]. This converts a network, into a hierarchical tree representation of the nested quadrants in which each point resides (a quadtree), Figure 1.9.



(a) Methane Graph (b) Quadtree constructed from Figure 1.9a

Figure 1.9: **Demonstration of the formation for a quadtree from a force directed graph of Methane (including inorganics)**. (a) shows the force directed graph of Methane from which the quadtree has been constructed- edge colours represent the flux between species. Here we partition the area into 4 and start at the top-leftmost side. This cell is then partitioned into 4 itself, in a recursive process until there is only one point in the cell. At this point we repeat the process to any remaining cells in a clockwise manner (b). The hierarchical tree (b right) shows the containign structure for each node. Here the colours represent the order in which nodes have selected (starting at pink and ending in blue).

Having defined this we move on to looking at the graph layouts.

[2]See **??** for a complete evolution of the methane quadtree.

Force Atlas 2

The force atlas two [Jacomy et al., 2014] algorithms is a force-directed layout designed primarily for scale-free[3] network spatialization. It is primarily designed for the use of networks consisting of 10 to 10,000 nodes and uses barns-hut approximation for the calculation of forces. Attractive forces are derived from the spring-electric model ($F_a = -k.d$), where k is the spring constant and d is the distance between the two nodes. Optional features for the graph include dissuasion by degree (separating nodes with a high number of total links/reactions), logarithmic attraction forces, adjustable gravity (attraction the centre of mass of the system to prevent disconnected components from drifting away) and collision detection to prevent overlapping nodes. Finally an adaptive cooling scheme is applied, where the overall energy of a system is gradually decreased, allowing the nodes to settle into a low energy states.

Yifan Hu

The Yifan Hu graph layout [Hu, 2004], is a multi-level graph drawing algorithm which uses the Barnes-hut algorithm with an octree layout. As with the force atlas algorithm, Yifan Hu also has an adaptive cooling aspect to it - meaning that as the algorithm is run its energy is progressively reduced, allowing the system to settle within a low energy state.

[3] A network whose degree distribution follows a power law (7 degrees of separation). This is described in Chapter 1.

The main difference within the algorithm, however, is the use of the multilevel approach. This has been applied to graph partitioning [11,12,23], matric ordering [24] and the travelling salesman problem [5]. This works by graph coarsening (coalescing neighbouring nodes and weighting them), running the algorithm on the course graph, prolongation and then refining the results. This produces an algorithm that runs faster than the Force Atlas, however, is constrained to only working on un-directed edges.

OpenOrd

A force-directed graph algorithm capable of scaling to very large graphs [Martin et al., 2011]. OpenOrd uses simulated annealing (see below), which has 5 distinct phases. These are each run for a fraction of the total number of iterations and mimic the different states experienced when heating/cooling a physical object (liquid, expansion, cool-down, crunch and simmer) - hear each state describes the amount of energy asigned to the nodes within the force simulation. In additon to this the OpenOrd algorithm applies a degree of edge-cutting to remove a percentage of edges experiencing the most stress within the physical system. This allows the network to open out into a more aesthetically pleasing layout.

Simulated Annealing

Most iterative layouts are updated interactively from some initial configuration in attempt to reach the lowest energy state of the system. In most cases this results in a minimum configuration; however this is generally a local minimum rather than the desired global minimum [Davidson and Harel, 1996]. To overcome this, the work of Metropolis et al. [1953], which was later formulated in general terms by [Kirkpatrick et al., 1983] was used to lay the foundation for simulated annealing algorithms.

Annealing is usually used to describe the slow cooling applied to liquids for them to reach a crystalline (totally ordered, minimum energy) form. Relating this to the spring-electrical model, it can be shown that if the atoms(nodes) are cooled too rapidly (losing energy quickly and coming to a quick stop), they will form amorphous structures representing the local minima, as opposed to the desired global one. If cooled slowly, our graph is allowed to find a thermal equilibrium at every temperature. Working from this idea, a slow cooling constant is applied, whilst occasionally supplying the system with short bursts of energy, that may allow it to overcome local minima.

tsNET

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction techneque which mimics the style of a force-directed graph (this is discussed in **??**). tsNET[4] is a graph drawing algorithm which levarages the non-linear dimensionality reduction capabilites of the t-SNE algorithm [Maaten and Hinton, 2008a]. This works by first computing the shortest-parth distances between all nodes to produce a distance matrix. This distance matrix is then used to construct a cost matrix which consists as the sum of three terms:

1. A measure of the divergence between picking paris of low- and high-dimensional datapoints.
2. A compression factor known to reduce the t-SNE optimisation time, taken from [Maaten and Hinton, 2008b].
3. A repulsion term.

Node positions are then determined by the mimimisation of the cost matrix using gradient decent - an optimisation algorithm used to minimise a function by iteratively moving in the direction of the steepest descent.

Although tsNET makes for a good alternative to classical graph layouts, it does not take link direction into account.

1.2.2.4 Layout Selection

Subsection 1.2.1 explained the importance of removing overlapping edges and Figure 1.2.1 the desire of having a well distributed graph layout. This subsubsection builds on those criterions, assessing all the graph layouts described within this section (Mercator, Force Atlas 2, Yifan Hu, OpenOrd and tsNET). These all use the chemical mechanism representing species within the APHH campaign in Beijing [**?**]. Here we look at the distribution (Subsubsection 1.2.2.4) and density (Figure 1.2.2.4) as they affect a users ability to isolate the shortest path (fastest flux).

Force-directed graphs place a greater emphasis on node positions,

Criteria, such as the ability to isolate the shortest path (in this case the fastest flux), are essential in determining the usefulness of a graph. Comparing different layouts [Pohl et al., 2009] found 68% of user-chosen routes to reflect the shortest path between them.

This is due to the force-directed layout placing a greater emphasis on node positions and distance than other layouts. For comparison, the same study found this to be 40% for hierarchical layouts and

[4] A play on t-SNE and network.

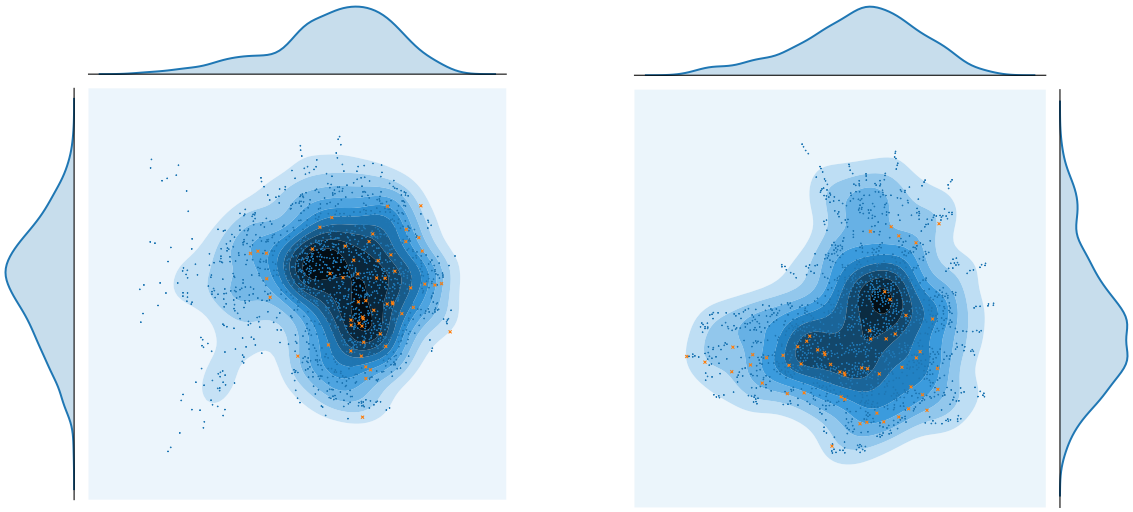
only 2% for orthogonal ones. In this subsection, I look at the use of different graph layouts, and their effect on the user readability of a graph.

Node Distribution

It is known that in partitioning the screen into quartiles with equal numbers of nodes (homogeneity) considerably improves the usability of a graph and increases symmetry [Purchase et al., 2003]. The main problem with node-link diagrams is that in representing complex data using an algorithmic layout can often result in regions of dense, indecipherable links, called hairballs [Ma and Muelder, 2013]. Hairballs obscure nodes and edges within a region, making it impossible to read. Methods such as the pruning of edges [Dianati, 2016] can be applied to networks as a means of reducing the complexity. This may be applied post computation (syntactic representation), which results in the loss of information, or during the algorithmic approximation in the OpenOrd algorithm, to produce clearer node positioning, with the edges re-introduced at the visualisation level.

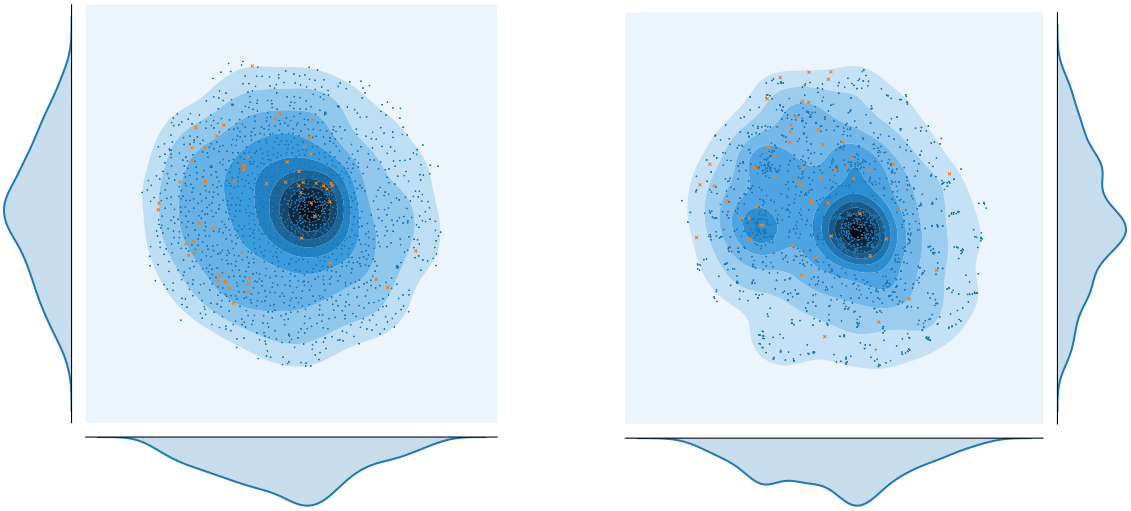
In deciding which layout algorithm produces the best graph-node homogeneity, a kernel density approach is used to compare node distributions across 2D space in Figure 1.8b and Figure 1.10. Here small localised areas of higher density, surrounded with sharp changes in density (shown by the contour lines) is preferable. Such a distribution would highlight the modularity of a graph and allow for the distinction between groups of species with many reactions between them, but few in another group. Graphs with a high homogeneity can be determined through the use of x and y kernel density plots. Here a homogeneous graph will have a uniform distribution across both axes. However, as we also wish to locate regions of chemistry with high modularity (clustering), a uniformly distributed graph would not suffice. Instead, we look for a near-uniform oscillatory distribution with an equal amplitude for each peak and trough. Using these criteria the Mercator (Figure 1.8b), tsNet (Figure 1.10d) and Force Atlas (Figure 1.10b) score the highest, with OpenOrd and Yifan containing a gaussian-like distribution across both axes which is conducive to producing a hairball.

Next, we apply prior knowledge about the graph we are trying to visualise. Here we know that the chemistry within a mechanism is determined by the oxidation of a set of primary emitted VOCs. It, therefore, follows that for an ideal graph layout, each primary emitted species should belong to its area of high density, and not entwined within the hairball. Immediately this notion eliminates the Yifan Hu (Figure 1.10a) and Mercator (Figure 1.8b) layouts since these both contain a high density of primary emitted species (orange crosses) within a single dense region. Using this criterion, the tsNET graph (Figure 1.10d) provides the best representation, followed by the OpenOrd and ForceAtlas layout.



(a) Yifan Hu

(b) Force Atlas 2



(c) OpenOrd

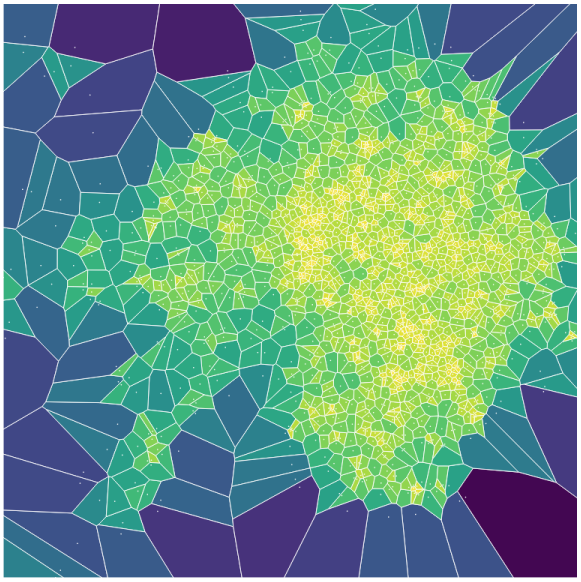
(d) tsNET

Figure 1.10

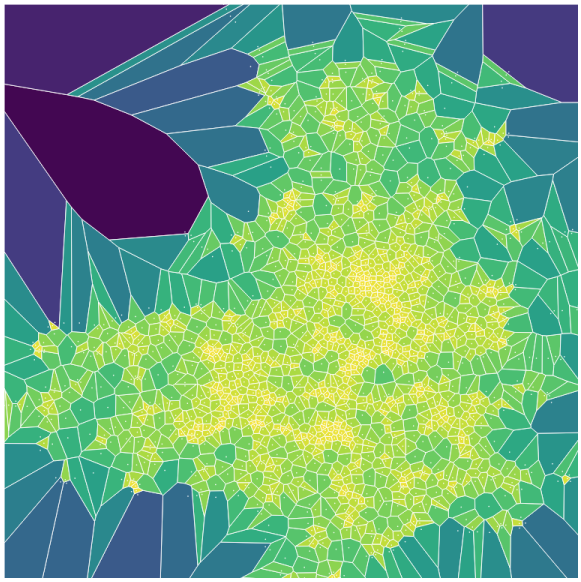
Node Density

Having explored the spatial distribution of nodes within a graph, it is important to determine which layout produces the best node density variation not only across the x and y directions. Here we desire a degree of regular anisotropy to produce 'clusters' of densely connected nodes sparsely separated in space. To calculate the distribution between dense and sparsely packed nodes, it is possible to use Voronoi tesselation. Here each node acts as a seed, and the plane is partitioned into a series of n cells, where n is the number of nodes. Each cell or polygon is calculated such that a polygon boundary is determined by all the points which lie closer to its source seed than any other- mathematically this would be defined as the perpendicular bisectors of the lines between all points. The result is somewhat

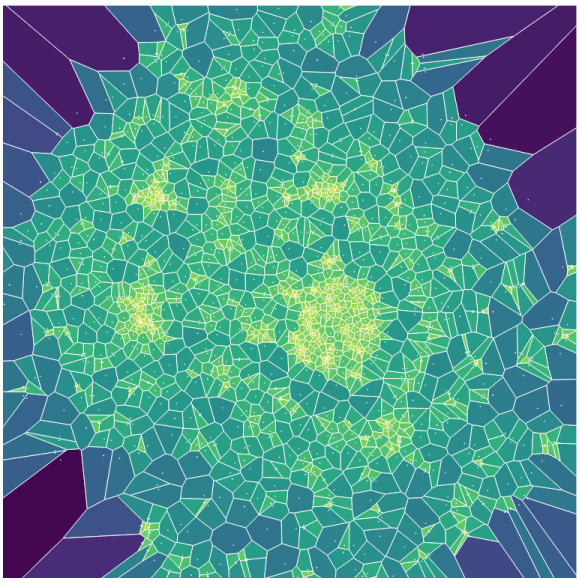
similar to a box full of bubbles, where each bubble fills the largest area it can before meeting another. Next, the area of each polygon is calculated and saved to produce a dataset representative of the complete density distribution between nodes. Here larger areas represents a species with distant neighbours (spatially), and a small one, an area of high density. The method of using vernouli teselation for the calculation of density has been used in the study of neurones [Duyckaerts and Godefroy, 2000] and areas of fixation when viewing images [**?**]. The last part of this process involves colouring the based on the normalised polygon area values and plotted within Figure 1.11. This allows for the clear location of layouts with high isotropy (**??**,**??**), which only contain many cells of a similar size, and consequently only exhibit a slight colour gradient difference between points. Although such layouts are spatially efficent, they do not reveal any additional information about the network structure. The colouring can also reveal the the spatial modularity of the graph. Here it is shown that the mercator, despite having a high $x - y$ node distribution, still contains large areas of unoccupied space due to its non linear density distribution. Under this criterion the ForceAtlas and YifanHu layouts (**??**,**??**) perform best, with distinct modules of high density appearing to be distributed across the graph.



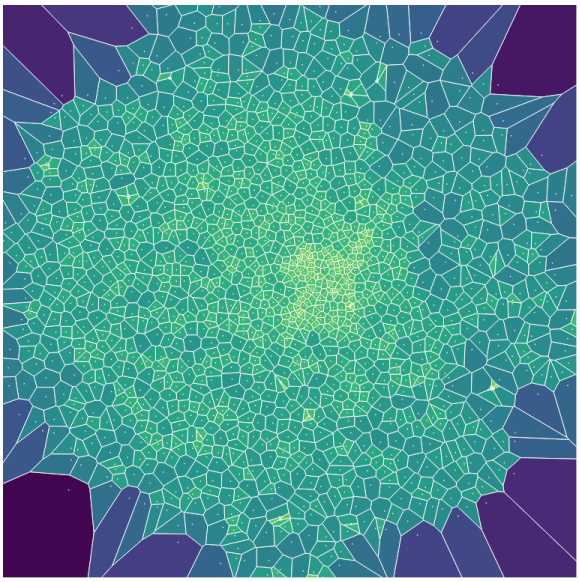
(a) Yifan Hu



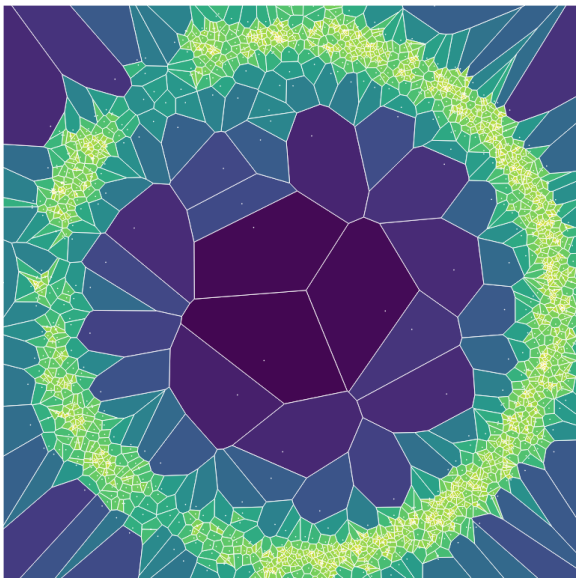
(b) Force Atlas 2



(c) tsNet



(d) OpenOrd



(e) Mercator

Figure 1.11: Comparing the density of nodes for different layouts using voronoi cell areas.

Mathematical Analysis and Layout selection

In addition to the qualitative approach through visualisation, it is also possible represent the polygon areas for each layout in the form of several boxplots, Figure 1.12. The interquartile (IQR) range for each layout represents the range of polygon areas. A large IQR signifies a greater distribution between low and high density areas. In addition to this we are interested in having a higher ratio of smaller area polygons to larger ones. Within the boxplot, this would be represented by having a median which is closer to, or approaching the 25^{th} quartile (the lower box boundary).

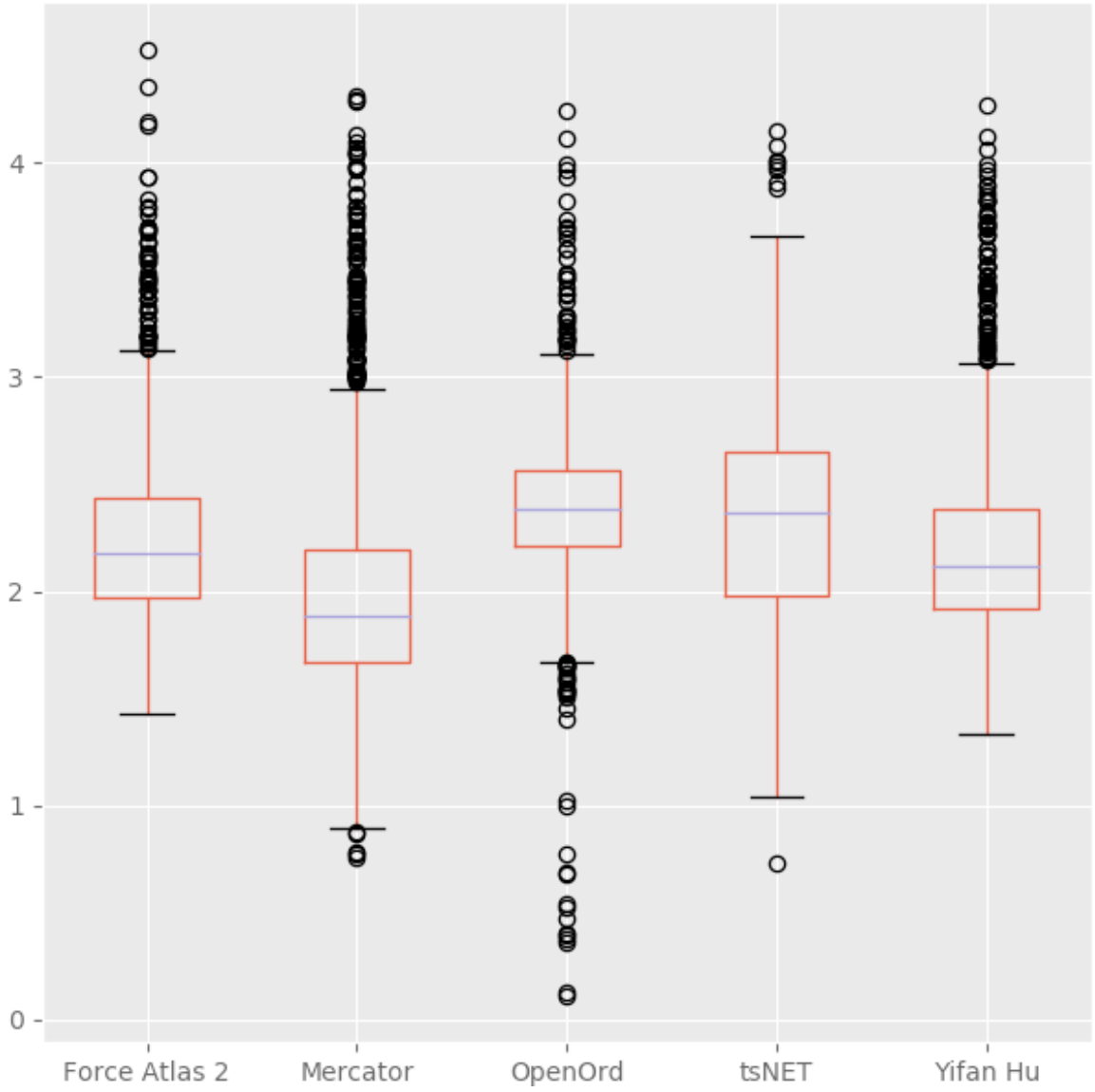


Figure 1.12: Voronoi $\log_{10}$(Area) BoxPlot for all plots in Figure 1.11

Applying these citerions to Figure 1.12, shows Mercator to provide the best result. However in combining this with our previous observations, is is noted that although the ratios approach our ideal range, its radial shape is not conducive to the general represenation of modularity within a network.

The layout with the largest IQR is produced using tsNET algorithm. Although this produces a well distributed algorithm, its inability to handle directed edges and high median rule it out as a possible candidate. The OpenOrd layout can reduce the number of hariballs within a graph through the use of simulated annealing and edgecutting, however it is also this property which in this case has resulted in a homogenious isotropic node distribution (as shown by the small IQR with a sizeable median value). Unfortunately this is not shown as the most effective at hilighting the underlying structure of the chemical mechanism.

This leaves the Force Atlas 2 and Yifan Hu layouts. Out of these the Yfan Hu layout fares better with regards to the box plot, yeilding an overall lower box, with a similar IQR and median ratio. Here its lower median suggests more high density nodes, with a similar distribution to the Force Atlas. This makes sense, since the two algorithms share many similarities, however once again the inability to handle directed edges makes it unsuitable for our application.

This leaves the Force Atlas as the preffered layout for the visualisation of chemical mechanisms. Its directed nature coupled with intuitive design make it applicable and easy to explain, whilst still maintaining an ability to produce a clear representation of any underlying structure. In addition to this, its more uniform spatial distribution (Subsubsection 1.2.2.4) makes it a better candidate than the Yifan Hu graph, which scored the highest in the boxplot test.

1.3 Graph Semantics

Deciding the correct semantic representation for a visualisation is often just as important as the selecting the correct syntatic style. Semantic features are often applied post generation [Bennett et al., 2007] and have uses in the encoding of additional information and clarifying any results within the data. As a means of achieving both an aesthetically pleasing outcome, and an easy to understand visualisation, we must first consider what features we, or the reader, are most interested in. Once this has been decided, we begin to explore various methods for representing them.

1.3.1 Limitations

When selecting visualisation semantics, there are several limitations that we must consider.

Visual

When it comes to Visual analytics the most significant bottleneck is due to the resolving power of the eye - this is known as an acutie. Acuities are a measure of the angle of an observed object with the

viewer's eye using arcs (one arc equates to $\frac{1}{60}^{th}$ of a degree). This provides a unit of measurement for the total amount of information density we can feasibly perceive [Ware, 2013].

In ophthalmology there exist four types of acuities:

- **detection**: The smallest size an object can be whist still being shown
- **recognition**: The smallest size an object can be to be recognised
- **resolution**: The smallest distance between two objects before they begin to merge
- **localization**: The smallest amount of visual change that can be measured between two objects

These provide a set of considerations which may be used to assess a visualisation. Depending on what encoding we use, it is possible to improve/hinder the readers ability to percieve information, Figure 1.13. An example of this would be that for a Macbook Pro retina screen[5], where at 87 pixels/cm[6] we can display at most 2 million resolvable nodes. If we wished to add links between nodes, the total resolvable items is reduced to one million [Jankun-Kelly et al., 2014].

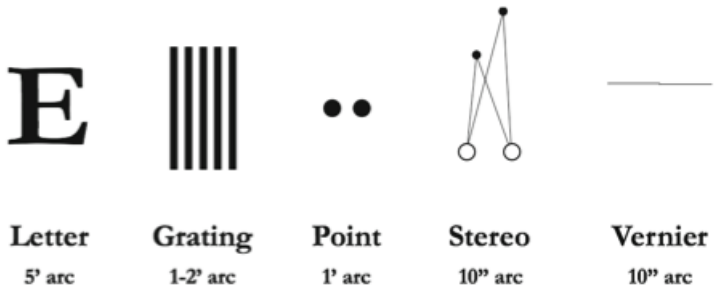


Figure 1.13: Important acuities in visualisation, Source: [Jankun-Kelly et al., 2014; Ware, 2013]

Cognitive

Although it may be possible to distinguish 1 million nodes and links visually, interpreting and understanding these presents another problem. The visual thinking laboratories [VTL, 2019], have a range of publications exploring how presentation can improve though, cognition and communication between info-graphic and reader. [Steven Franconeri, 2018], explains that the time required to interpret a visualisation is directly related to the encoding used to hi-light the data within it. Also problems of 'intentional blindness' and misinterpretation are problems which are often occurred with poorly thought out encodings.

[5] A retina screen, is half the maximum possible resolution of the human eye at a 30cm distance. Additionally the operating system interpolates in sets of 4 pixels, such that the image displayed may not be at full resolution.

[6] at 57cm from the screen

In considering the cognitive load of a visualisation [Norman, 2005] provides a list of three categories which should be explored:

1. Firstly we have the visceral level, a subconscious process where decisions are made rapidly based on sensory inputs to the body. This is usually due to our inherent ability to locate patterns and changes due to semantic properties which shift the focus of the user.

2. Next follows the behavioural level (mostly subconscious). These are often learn reaction to changes noted as part of the visceral level. Here reactions may be honed on and influenced by past experiences and events.

3. Finally we reach the reflective level. Here the user collates all sensory input from the previous two levels and makes an informed conclusion about the underlying data. Conclusions drawn here can be used to bias the methods used within the behavioural level in future events.

Technological

In addition to human limitations, there may be restrictions due to the medium a visualisation is created/presented on. In addition to monitor resolution issue earlier, much scientific research is constrained by the size, resolution and colour quality of the presentation mediums used for talks, printing or posters. [Ware, 2013] explains that a printer capable of producing 1200 dots per inch squared, can only do this for black/white binary images. If for instance 256-greyscale is used, the resulting resolution is then at-least 10 times smaller. This is because printers a Monet style approach to create shading and colour. It therefore follows that at full CYMK, the output resolution will be worse.

It is also essential to have a graph fitting the same overall shape of the canvas on which it is presented [Taylor and Rodgers, 2005]. This not only makes optimal use of any space available, but also reduces the visual complexity as it minimises the number of distinct shapes available to the user.

1.3.1.1 Node Encoding

Within a graph, the nodes represent the set of items we are exploring. Each of these often contain a multitude of features and properties relating directly to them, be it the user details for a retail/fraud network, or the chemical composition and concentration of a species in the MCM. Features of a node describe and additional properties and may be used to determine its interaction with other nodes[7] [Aumont et al., 2005]. It is for this reason that graph convoluted neural networks [Klicpera et al.,

[7]This is further explored in Chapter 4

2018], require a 'feature matrix' describing each node, in addition to the network structure and edge weightings. Within a visualisation, a node may be represented in a range of ways.

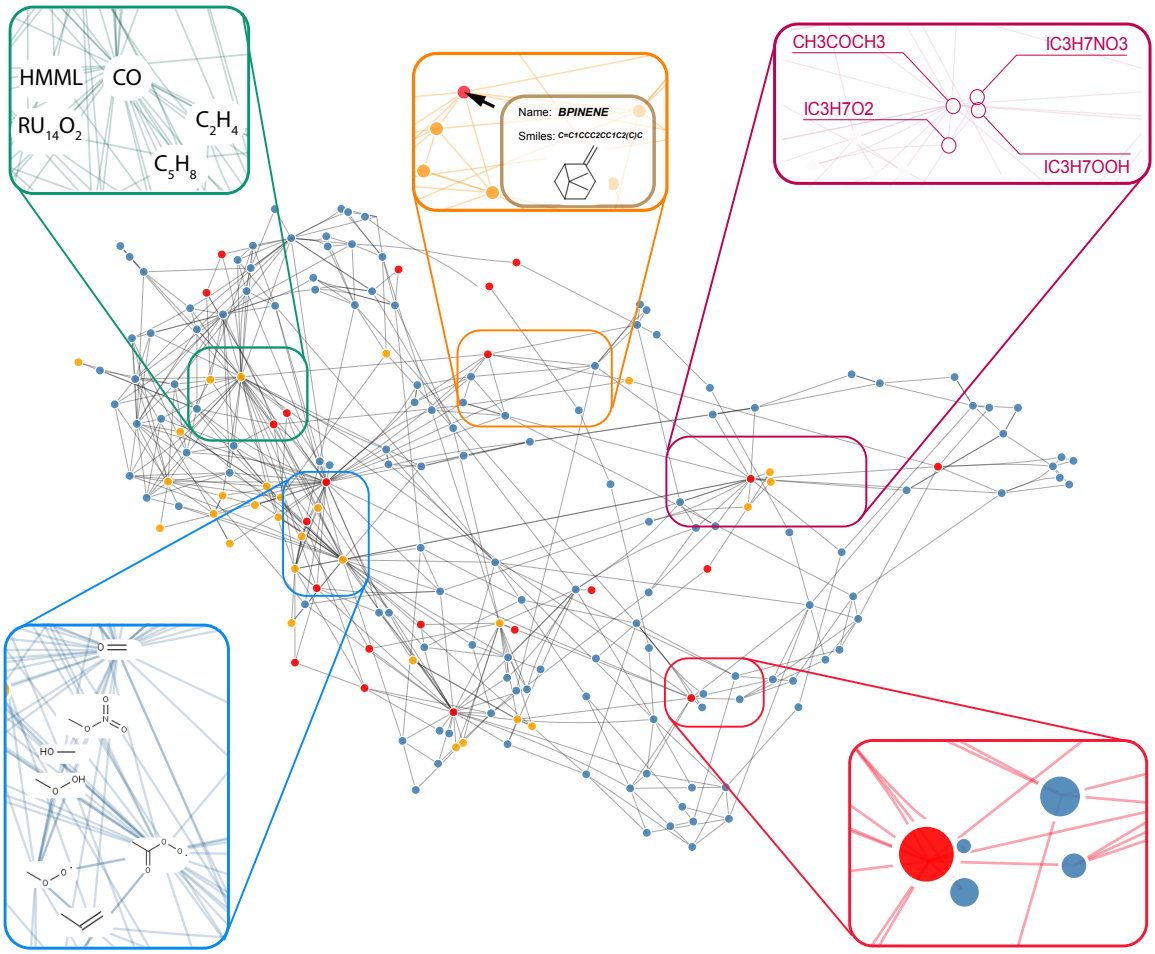


Figure 1.14: **A graph showing 5 different node encoding methods.** These are Circle Atrributes (red), Chemical structure (blue), Species Name (green), External Labels (maroon) and interactive selection (orange). The network shows the Common Representative Intermediate species [Jenkin et al., 2008] mechanism. Node colours represent primary emitted VOCs (red), MCM species (orange) and lumped CRI-only species (blue).

Circle Attributes

The simplest of these range from the use of colour, shape, size, thickness and stroke (outline) to indicate a group. Here it is possible to provide information such as a species concentration based on its size, its importance with its colour, its degree with its opacity and its category with its stroke colour [??]. Such descisions depend on what properties you are trying to show. For instance red species in Figure 1.14 are primary emitted VOCs, orange species exist between both the MCM and the CRI (see figure caption) mechanism, and blue ones are lumped species which do not appear as part of the MCM.

Chemical Structure

Traditional chemical diagrams use the chemical structure to depict the types of reaction that occur , Figure 1.2. This make it intuitive to extract information about functional group and bond changes within species. Such a method of representation, is indeed useful, however when visualising hundreds, if not thousands of nodes on a page, it results in occulsion, or labels too small to resolve visually.

Species Name

Much like the chemical structure, a species name is proven useful in explaining to the user its chemical properties (often due to prior knowledge, or the ability to look this up). Unfortunately since names have differing lengths, this can cause problems, especially with large numbers of closely located nodes. A solution to this may be to adjust the font size to fit in within the circle radius of the node. However this does come with its problems - for instance tiny nodes may have text smaller than a pixel, or the misleading notion that longer names are less important, since they are represented by a smaller font.

Interactivity

Ben Shneiderman's famous mantra goes: *'overview first, zoom and filter, details on demand'* [Shneiderman, 1996]. This goes hand in hand with the philosophy used within the design of an interactive visualisation.

For complicated systems, interactivity plays a vital role in unraveling complexity and reducing clutter [Shneiderman, 1997]. It allows the user to actively query only the items that they are interested in whilst still displaying all the information in a single location [Görg et al., 2007].

A comprehensive list of all available interaction types and styles are provided in [Wybrow et al., 2014]. Some examples of interaction are:

Hi-lighting

- Hovering
- Brushing and Linking
- Magic Lenses (see hidden objects)

Navigation

- Pan / Zoom
- View Distortion (fisheye)

Visual Structure-Level Interaction

- Selection
- Changing layout/mapping attributes
- Changing representation

Data Level Interactions

- Adding / Filtering
- Search / Query

Table 1.1: **A selection of interactive methods.**

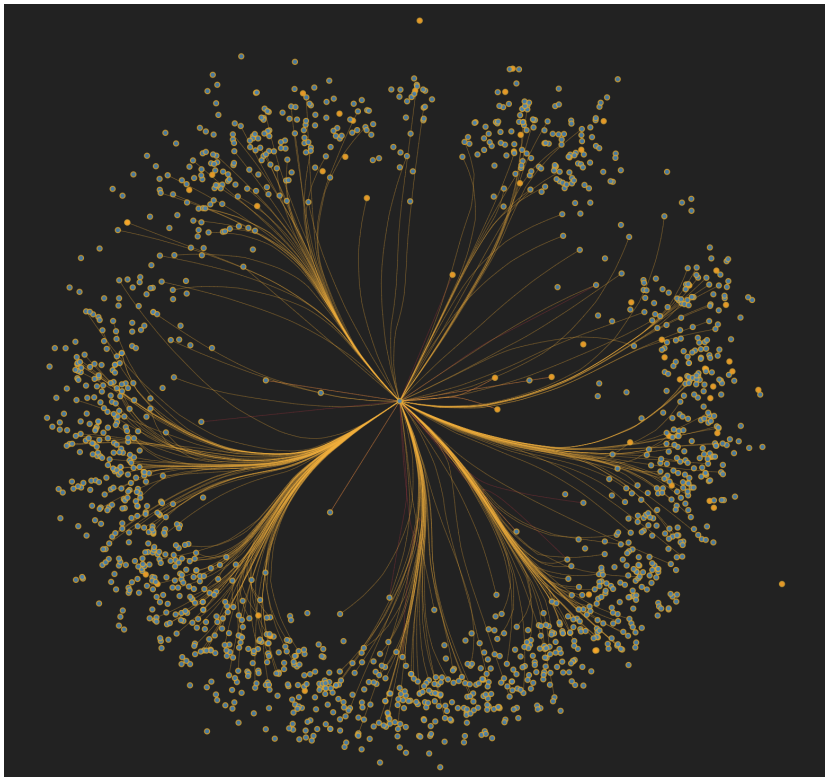


Figure 1.15: **Using mouseover edge-selection to hilight all links related to a node.** This figure shows how in using interactivity it is possible to reduce clutter and filter the information presented by a densely populated graph. In this case the mercator projection (Subsubsection 1.2.2.2) is used, with reactions relating to Carbon Monoxide (centre) hilighted. Orange lines represent reactions producing CO whist the red (some of which may be hidden) are of reactions with CO.

External Labeling

In cases where interactivity is not possible, such as papers, books and this thesis, an alternative approach to data selection has to be employed. Here nodes which are central to the explanation of a certain point are filtered by the author, and displayed through the use of external labels. It is found that having links at 45 and 90 degree angles (such as in transport maps) lead to a clearer layouts and

better distinction from the links already within the graph. Automatically generated labels within the thesis are made using [Lu, 2019].

1.3.1.2 Edge Properties

Defining the purpose of graph-energy models as: a means for creating a visualisation from which the viewer can infer properties of the data [Noack, 2004], it can be shown that this criterion is easily met in small and sparse graphs. However non-planar examples with high edge density (lots of links) can easily result in tangled results with impractical running times [Kumar and Garland, 2006]. In most cases attaining an optimal solutions here seems to be computationally infeasible [Davidson and Harel, 1996]. This is generally because graphs primarily focus on hi-lighting a specific purpose or following a set of aesthetic heuristics [Pohl et al., 2009].

Butane model

Muti-variate edges

Since there are multiple relationships between species, it is important to decide if simplifying the network would be of benefit. Although it is possible to Figure 1.16.. this may cause unecessary clutter for larger networks. Instead it is often useful to simplify the graph, and encode the edge properties within the vector object. This allows the user to retreave any additional information by hovering over the edge or connecting nodes, as required. Should the topic of interest require a specific property, then it would also be possible to remove, or hide, all edges which do not contain it. This produces an interactive graphic containing all the required information, as and when needed, without the unecessary clutter of having every reaction shown.

Figure 1.16: **Multiple edges coloured by type of reaction.** Using the overarching categories of reaction type (see fig wuclan) each type of reaction between two nodes can be visualised using the multi-link format. Photolysis (Bright green), Radical/Other (Amazonite / Teal) Decomposition (Honolulu Blue), RO2(Space Cadet / Purple)

Edge Direction

When using a directional graph it is convention to use arrow heads to represent this. However in high density regions it is often found that arrow heads take up precious real estate in the drawing area [Dwyer et al., 2006a]. As an alternative, colour and line-type can be used to represent the direction instead. This example can be shown in the routing networks presented by [Di Battista et al., 2004]. One example applicable for chemistry would be the use of dashed lines to represent mono-directional relationships, and continuous lines for bidirectional ones.

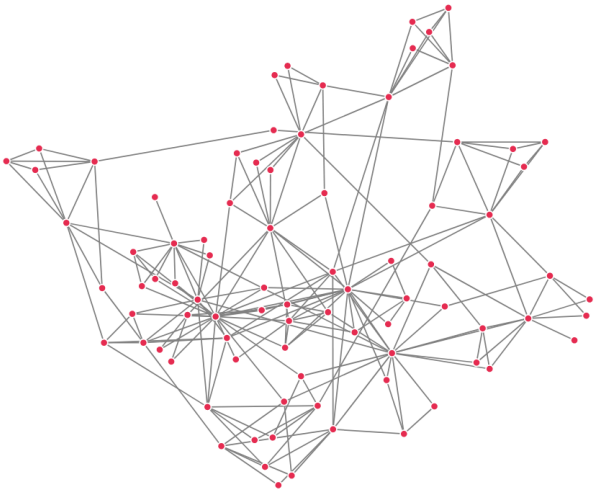
Edge Shape

Edge shape is essential, as it is the medium we use to represent relationships within a graph. For orthogonal graphs, poly-line curved edges are used to provide a layout which is simpler and easier to read [Di Battista et al., 1994]. For asymetric graph drawings circular Lombardi-style curves and cubic brezier lines have been used to reduce the clutter in high edge-density drawings [Chernobelskiy et al., 2012; Goodrich and Wagner, 1998]. Figure 1.17 shows a selection of different edge types for the Butane MCM subset.The liear network (Figure 1.17a) consists of straight lines betwen nodes.

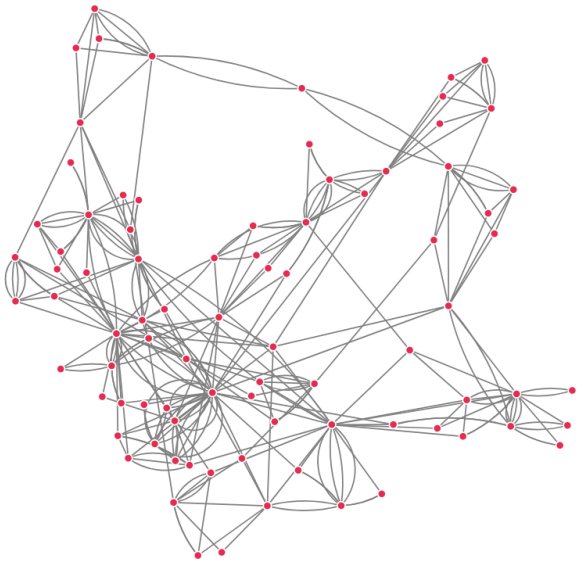
If a multi-edge graph is required, it is impossible to represent this as all edges between two nodes follow the same path. To improve on this a quadratic arc (Figure 1.17 b) can be used. This presents a symmetric representation where each edge is revealed. Finally bezier curves (described below) can be used to show an asymetric representation of the multi-edge graph (Figure 1.17c). Both sets of curved representation rely on a set of control points, allowing the designer to control the curve shape, steepness and asymmetry.

Bezier Curves

Bezier curves are named after Pierre Bezier who used them in the bodywork design of Renault cars in the 1960s [Hazewinkel, 1997]. Since then they have been widely used in graphs,computer graphics, font design and animation/interactivity response [Goodrich and Wagner, 1998; Hazewinkel, 1997; Mortenson, 1999]. Bezier curves come in a range of possible dimensions, cubic beziers are the most commonly used within network visualisation. These contain four control points respectively which can be used to determine the shallowness of the curve through design. In general relatively shallow curves are prefered, as these do not introduce unecessary edge crossing or abrupt changes, which have been shown to hinder a users ability to isolate items of interest [Purchase et al., 2003].



(a) Linear (single-edge)



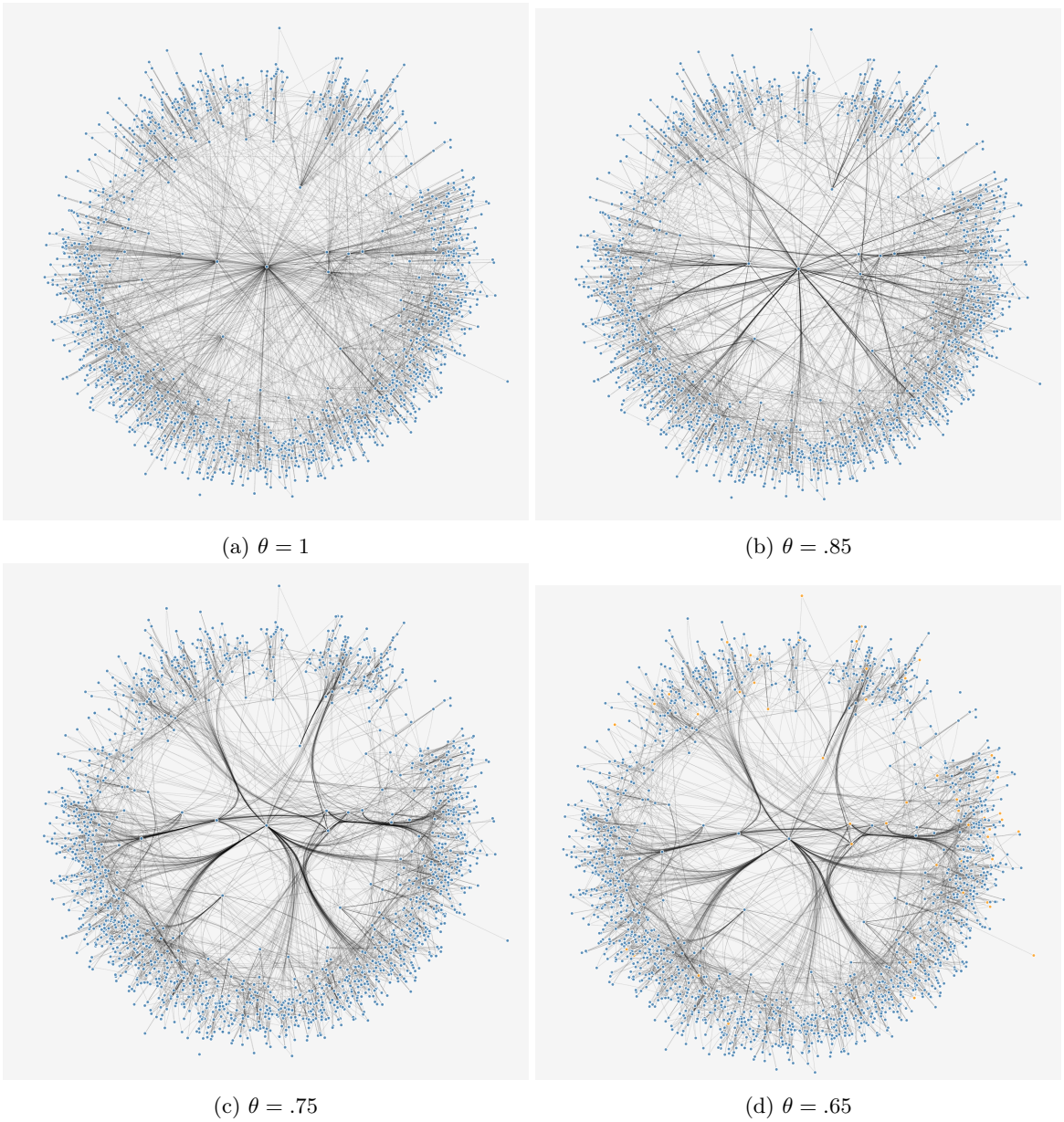
(b) Quadratic (multi-edge)

(c) Bezier (multi-edge)

Figure 1.17: **A selection of edge shapes for the butane network.**

Edge Bundling

Pioneered by [Holten, 2006], edge bundling techniques are an effective way to reduce visual clutter. Much like a force graph, edges are represented as a string of lined points. This allows for edges to be pulled together (attracted to one another) and produces a visualisation akin to moving water droplets on a hydrophobic surface. Figure 1.23 shows how in changing the amount of atraction between edges, it is possible to reduce clutter in a visualisation.



(a) $\theta = 1$ (b) $\theta = .85$

(c) $\theta = .75$ (d) $\theta = .65$

Figure 1.18: **How the compatibility threshold affects edge bundling.** In increasing the amount edges are attracted it is possible to improve the clarity of a graph. However there reaches a point where this distortion can worsen the result, confusing the reader, or creating a false positive. For this reason, I generaly use only a slight bundling value > 0.7.

Power, Routing and Confluence graphs.

Confluent graphs use a graph drawing method in which edges are not drawn as individual distinguishable geometric objects, but rather as a crossing free system of arcs and junctions. [Förster et al., 2019]. Their design is similar to that of the edge bundling algorithm, except that rather than bundling edges spatially (a design which may introduce ambiguity), the bundling is done based on connectivity and can help reduce clutter by grouping multiple edges where the all target nodes are also connected to all the source nodes, Figure 1.19,[Bach, 2020].

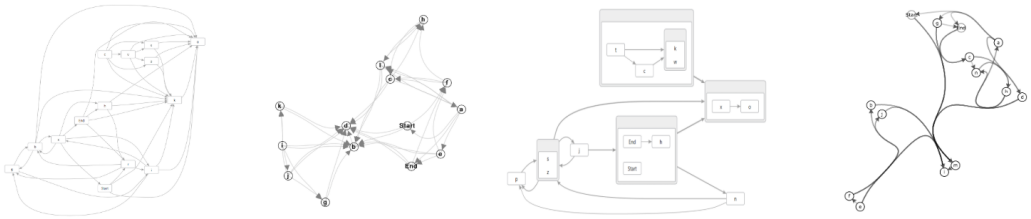


Figure 1.19: **An example of confluent bundling.** From left to right - A traditional network, Edge bundling, Power Graph and Confluent graph representations. Source: [Bach, 2020]

Using butane as an example the construction of a confluent shall be covered. The first step in the process is to create a power graph of our network. Power graphs are a representation of complex networks where sets of items identical source and target links are lumped or grouped within a single item. This is then converted into a routing graph, Figure 1.20. To do this multiple edges which may be bundled have a 'routing' node added to guide them. Next basis-splines, using the routing nodes as control points, are used to map the graph[8], Figure 1.21. Finally crossing links are removed, leaving the confluent graph, Figure 1.22.

Confluent drawings have been found to have many applications (e.g. the ego-centric author network and social interaction graph), they generally perform best in sparce networks with locally dense clusters of a tree like structure [Bach et al., 2017]. Although sparse, the cyclic nature of atmospheric chemistry does not allow for a sufficient reduction in complexity to make them a suitable improvement over traditional graphs. The use of very close fitting basis-splines in addition to a routing graph (confluent graph with crossing artifacts), may however help to simplify specific layouts or mechanism subsets with a certain amoung of tweaking.

[8]These are similar to bezier curves but require a degree, p, $n+1$ control points, and a knot vector of $m+1$ points. Note: Knots are the things that make the curve continuous

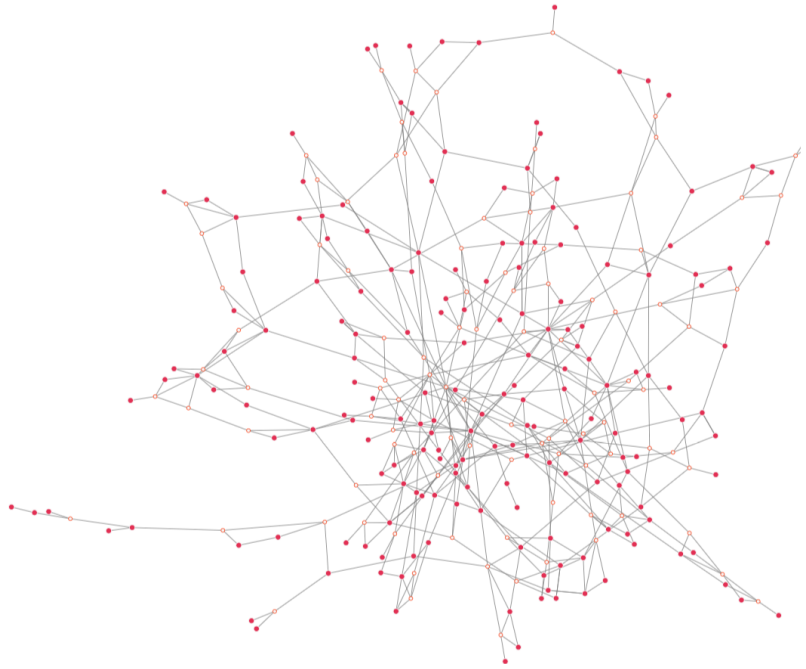


Figure 1.20: **The routing graph of the butane mechanism.** Here paths which contain two or more bundles have an extra 'routing' node introduced (orange stroke)

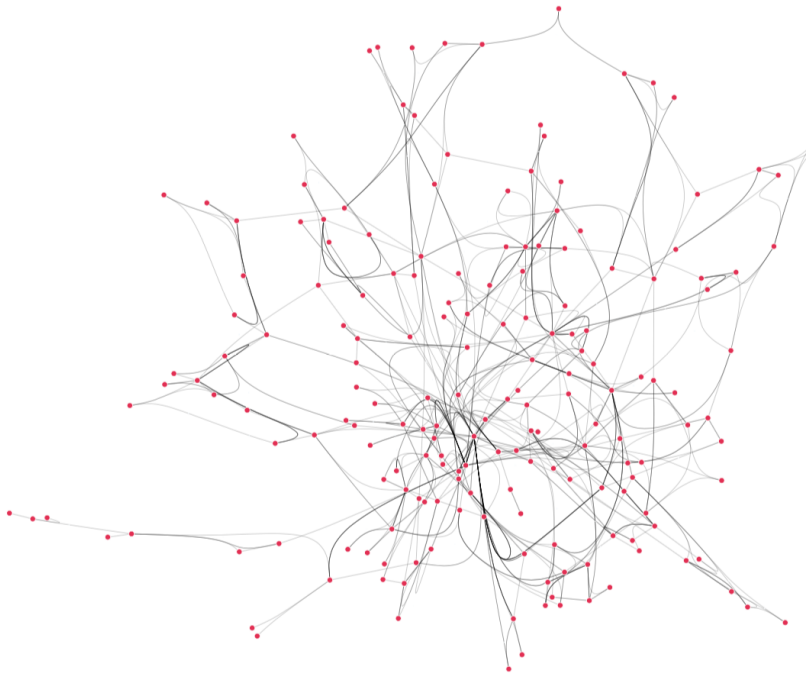


Figure 1.21: **Confluent graph with crossing artifacts.** The routing graph with the addition of basis-splines using the orange routing nodes in Figure 1.20 as control points.

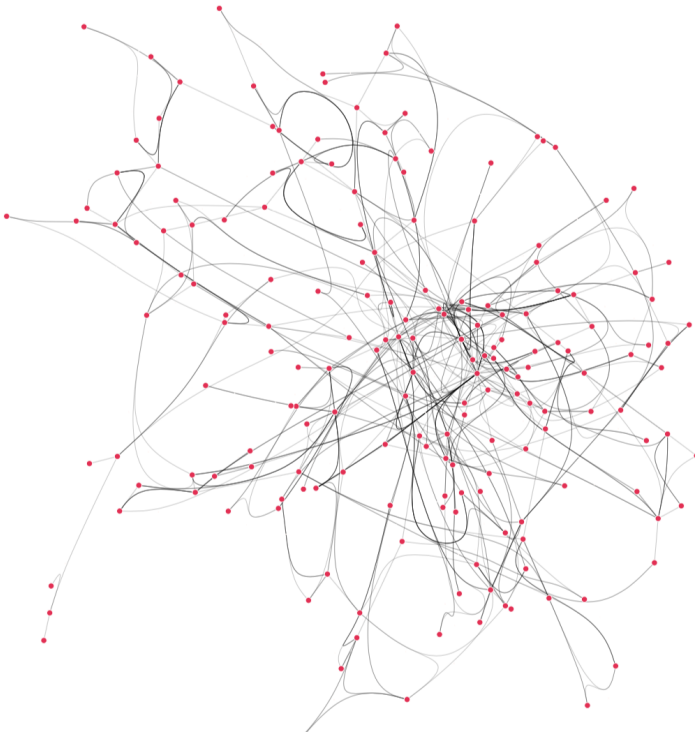


Figure 1.22: **Confluent graphs without crossing artifacts.** The remaining confluent graph with crossing edges removed.

Edge Angle / Continuity

Visual representation utilises our conscious and unconscious pattern recognition and intuition abilities [Dixon, 2012]. To avoid apophenia (finding patterns where they do not exist), careful consideration has to be placed in the design of a graph layout. Although edge corssing is ofthen though of as the most import aesthetic metric, finding a continuity between inward and outbound edges of a node was found to be if equal importance [Ware et al., 2002].

Reducing the angle between related edges increases readability and allows the behavioural process to infer information about a graph correctly. This process can be compared to predicting the direction of turbulent vs laminar flow. In addition to this edges should be spaced evenly around node, maximising the minimum-edge-angle between all edges of a node [Bennett et al., 2007].

1.3.2 Temporal Projection

Story-telling has been an effective method to convey information, experience and cultural values for almost as long as people have been around. Many real-life physical processes occur over time and thus allow the use of a story-telling analogy. [Gershon and Page, 2001] provides a generic structure which begins with creating a general overview of the subject. Events are then animated in order of occurrence and defined as we go along. Finally any remaining conflicts and uncertainty is addressed, and these

are rectified. Using this as a template for our graphs, we find that the content is usually given in the form of a title or figure description, the evolution as the visualisation, and finally the reflection and resolution through the use of user interaction (e.g., node hi-lighting, zoom or animation).

Since very few graph layouts support dynamic time-varying graphs [Kumar and Garland, 2006], several methods of visualising temporal events have been developed. Although storylines can be useful for drawing the evolution of simple systems, these break down when dealing with large numbers of dependant variables. Force-directed layouts may be adapted, to suit these better, whereupon the initial positions of the previous node endpoints are used as the initial positions for consequential simulations. Three methods of representing these are shown in Figure **??**.

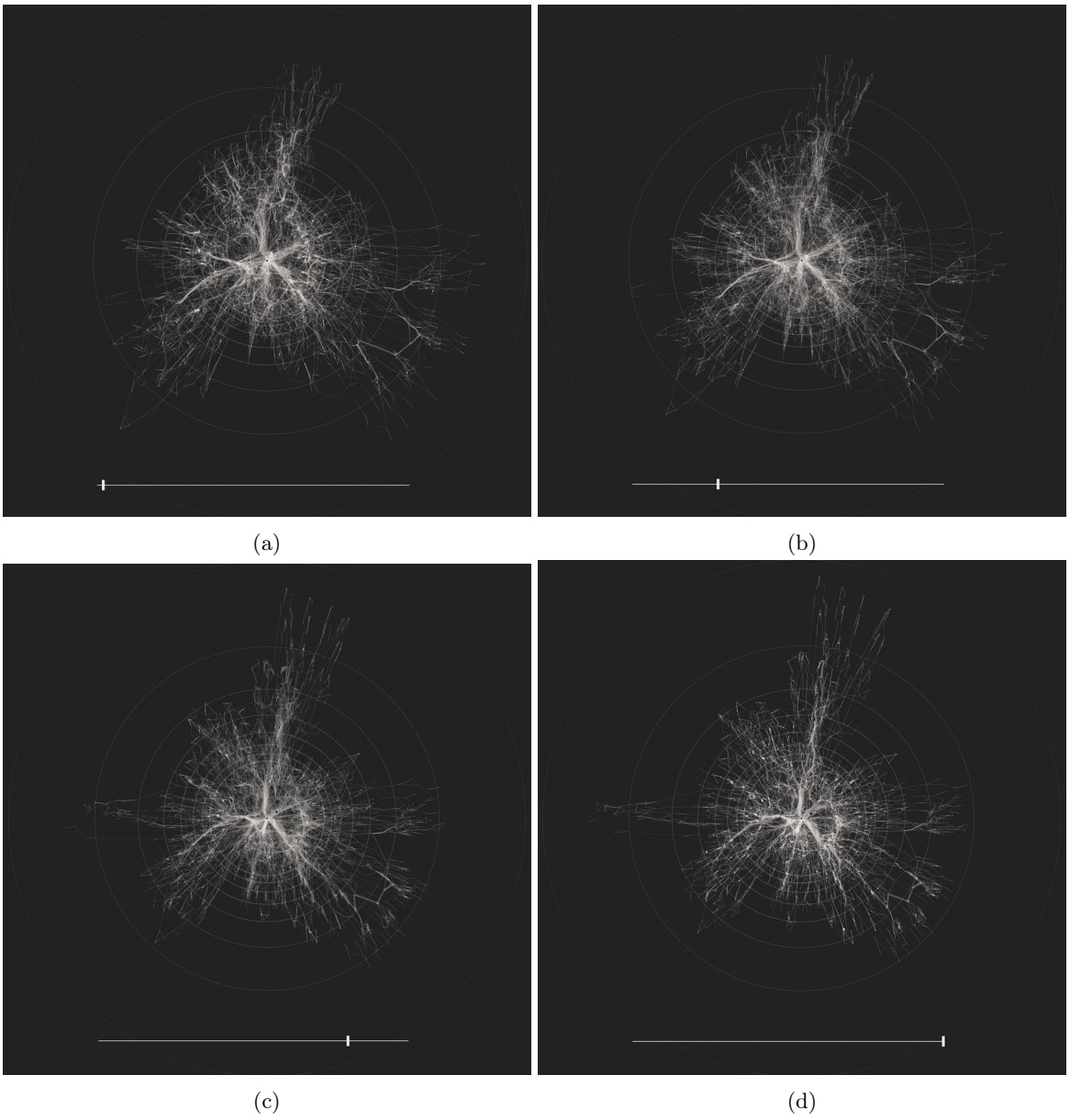


Figure 1.23: **Film style representation of temporal changes in a network.** Showing the temporal changes from a model simulation of the beijing atmosphere. (a) shows a weighted graph at midnight. With the addition of daylight, the chemistry speeds up causing the force graph to contract, changing the overall network shape (the faster reactions have a stronger attractive force).

Finally, user-interaction such as hi-lighting key nodes/links, zoom and animation[9] may be used to clarify information at the reflection stage.

1.3.3 Additional Dimensions

Additional dimensions can be used to emphasise certain aspects of our graphs. For instance multiple layers may be used in a directional graph to separate the importance of the nodes [Dwyer et al., 2006b]. **??** shows the first, second and third generation species of a mechanism containing isoprene in three dimensions. Such a visualisation may be explored interactively, with the aid of a computaional input device (a mouse, keyboard or device gyroscope), or with the aid of red-cyan 3D glasses (for non-interactive mediums such as print).

Different layers can be used to separate of primary VOCs, from species which result in their production (+1 layers) and loss (-1 layers). Temporal data, such as that in **??** can also be presented in this format. The only drawback is the high possibility of obfusciation which may result from many layers of overlapping information.

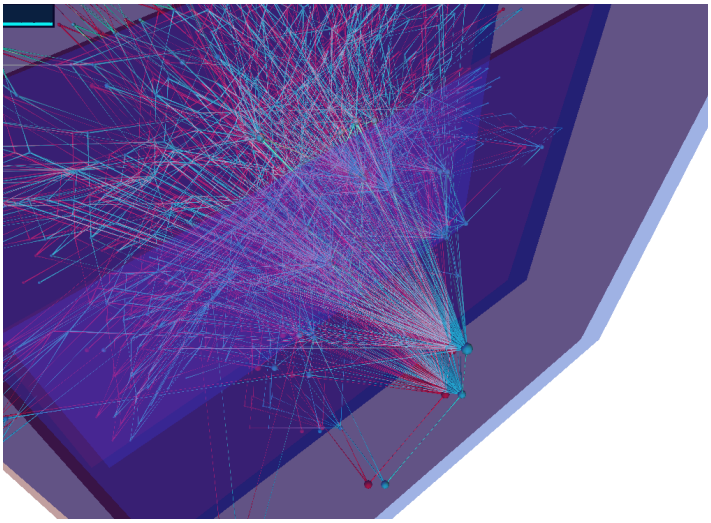


Figure 1.24: **A 3D representation of a graph to hilight certain features.** The first, second and third generation species of isoprene shown as an interactive 3D anaglyph.

1.4 A Chemistry Case Study

To conclude we apply many of the tools described above to a simple case study. We select a MCM subset containing methane as the only primary emitted species, and run it throught the Dynamically simple model of atmospheric chemical complexity (DSMACC) [ref] using the initial conditions of XYZ.

[9] [Archambault et al., 2014] notes that animation poses high demands on the users visual memory, and that snapshots are likely to miss underlying patterns. For this reason an interactive techniques that can allow retrospective selection of timesteps allows for a good compromise between these.

We run this forwards to steady state and extract the flux between species on noon. The edge weight is the net flux (product of the species concentration * the rate of reaction for all reactions), normalised to a value between 1 and zero.

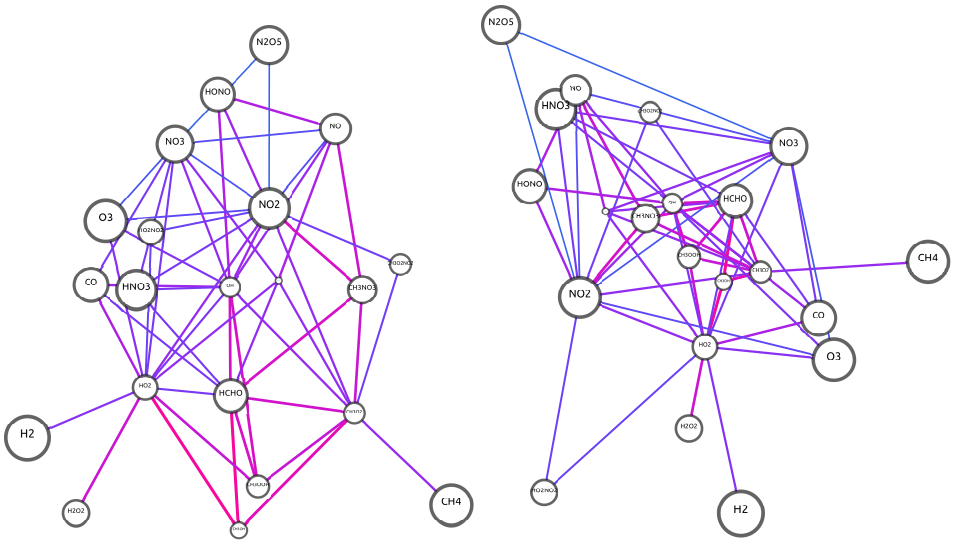
This allows a simplified view of the different properties which affect the visualisation of the graph produced.

1.4.1 Syntatic Representation

Since we shall be using simulation data, we require a layout which deals with both direction and edge weights. In the spirit of zero and Protagoras[10], we opt of the spring-like description presented by the Force Atlas 2 algorithm. This feature hi-lights fast reactions by bringing nodes together. Such a property has been observed to help users select the shortest path within a network [Pohl et al., 2009]. Here users picked the shortest path an average of 68% for force directed graphs, compared to 40% for hierarchical and 2% for orthogonal layouts. Such properties can help us locate any trends in fast reactions which may control the chemistry within a system.

1.4.2 Semantic Representation

Since the graph presented contains only a handful of species, our screen real-estate allows the listing of names for each node. Node sizes are scaled to represent the concentration of each species at that time point, and edges are coloured to represent the strength of each relationship between them.



(a) Connected unweighted (b) Connected $\hat{\Gamma} = flux$

Figure 1.25: Basic steps within in the the graph production process

[10]Famous for the phrase 'man is the measure of all things' suggesting that we are constrained by our experiences

1.5 A Model Of Beijing

Using a spun up model initiated from the campaign results XX Beijing (Where did I get these?) we compare the distribution of links within a model. In f[FIG XX] we see the graph shape change due to the presence of photons.

To perform a sensitivity study on the initial positions of nodes within the force atlas algorithm, a graph consisting of links and weightings is constructed using a box model simulation of the Beijing summer environment at mid-day and feed it the gephi software [Bastian et al., 2009] - an open source software designed for the exploration of networks. We then scrip the java code to perform the functions in Figure 1.26. As part of this, nodes are initiated with a random position, the force atlas 2 layout is then run and then the graph is rotated and translated such that it is centred around carbon monoxide and has a 45 degree angle between this and formaldehyde. This step constrains the general orientation of the graph, allowing us to analyse the generated graphs for global and local minima. The final step is to save a copy of the generated graph layout and repeat to generate a data set, a subset of which is shown in Figure 1.27

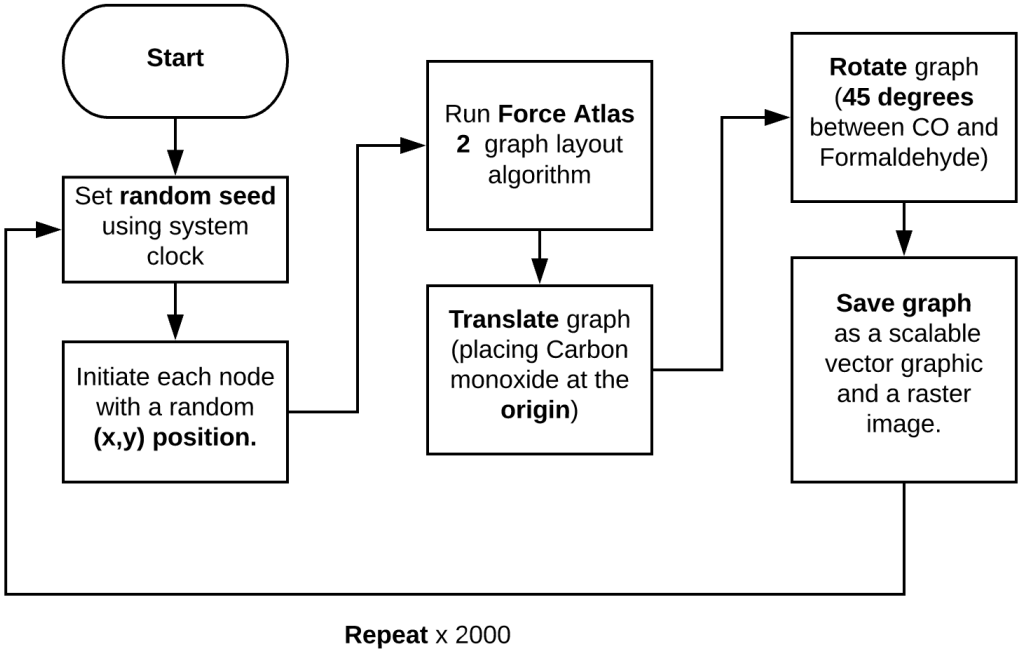


Figure 1.26: A flow chart of the process performed by the custom gephi script used to generate the data set

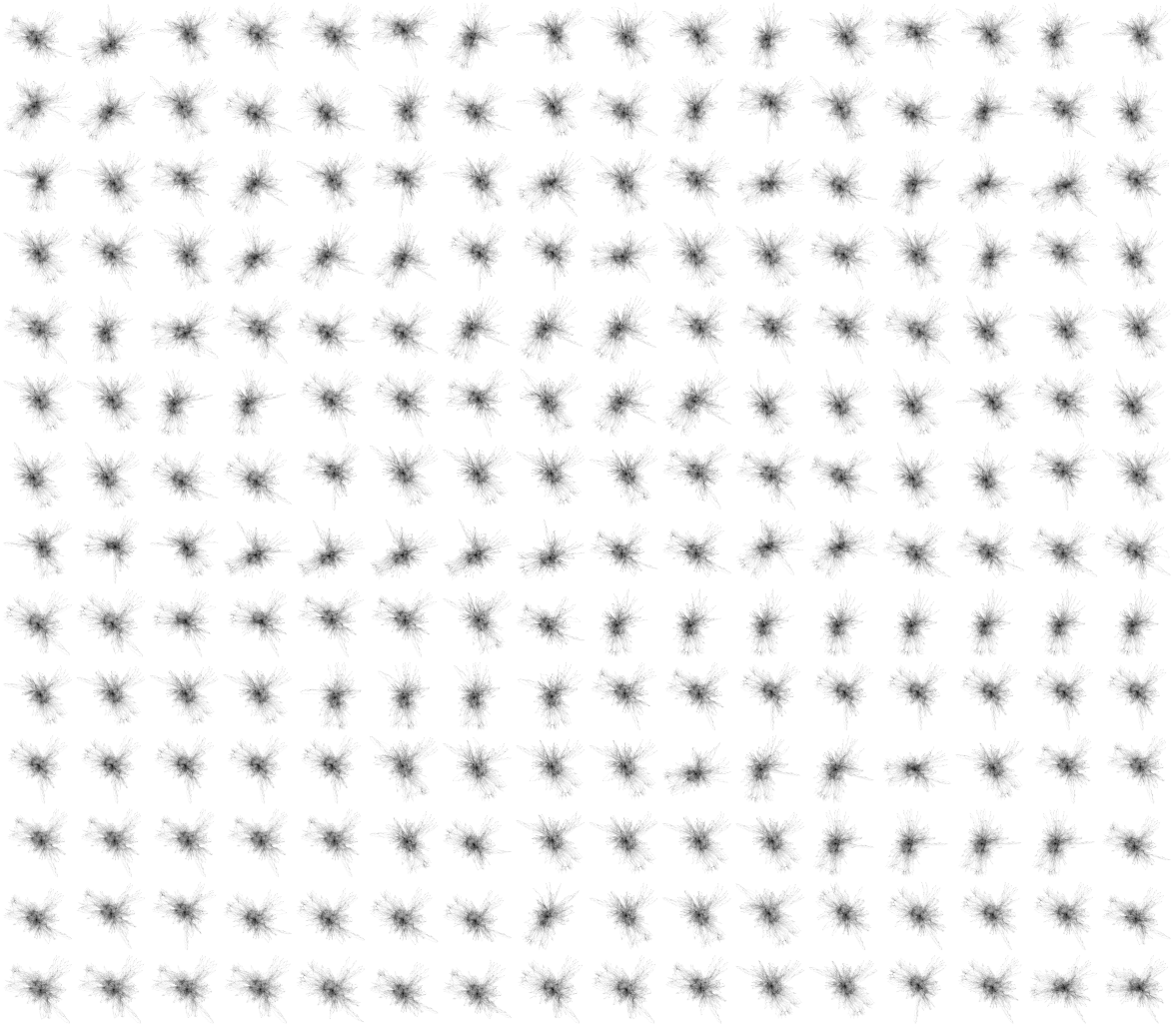


Figure 1.27: A sample of the 900 graphs generated using the force atlas 2 algorithm for the simulation output representative of the summer beijing chemistry at noon.

1.5.0.1 Trends In The Chemistry

Due to the construction protocols of the master chemical mechanism, **??**, primary emitted compounds are oxidised to produce a cascade of species, ultimately ending at carbon dioxide[11] and water. As this process is central to the construction of the mechanism, it follows that they may be used to explain any features uncovered using the network layout.

Network shape

Using Figure 1.27 the pattern recognition capabilities of the human mind identify a certain shape associated with many of the networks. Upon closer inspection it may be hypothesized that the chemistry is split into three main branches. Figure 1.28 categorises all the primary emitted species,

[11]The MCM conserves the number of carbons, allowing CO_2 to be introduced.

and then uses voronoi tessellation[12] to colour neighbouring nodes and their links by the classification of the closest primary emitted species. Using this it is possible to separate the MCM network into an aromatic branch, a terpene branch, an alkane and straight chain alkene branches. Such branches not only help us identify changes of chemistry due to biogenic or anthropogenic sources, but also emphasise the path taken to carbon dioxide and water. Since the MCM does not contain CO_2 we see all the different groups converge on Carbon Monoxide (white, centre). Using this format, we may now compare the orientation of the many automatically generated layouts.

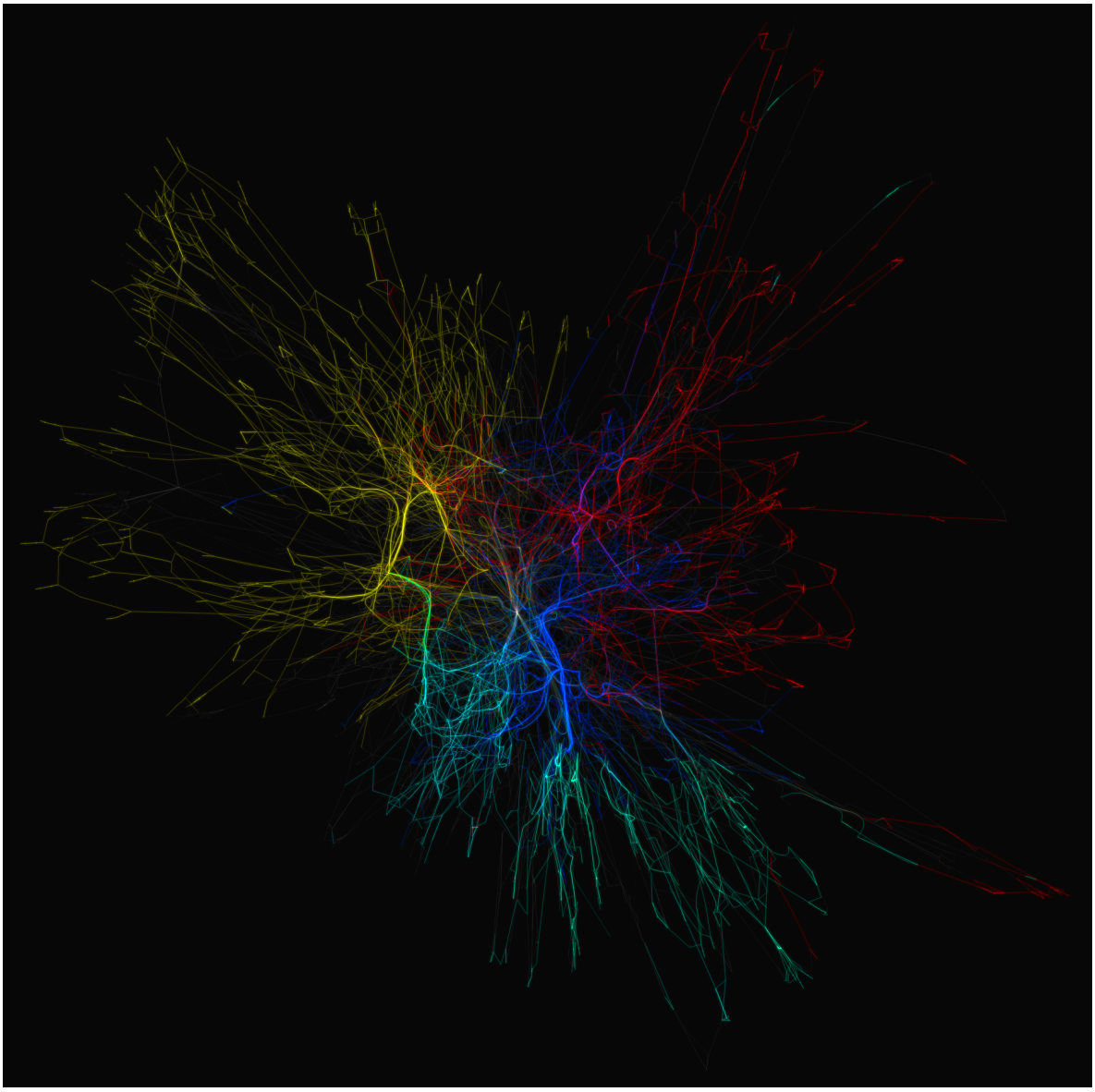


Figure 1.28: Hi-lighting the groups of species, and their products within the MCM network graph. These are Aromatics (gold) , terpenes (turquoise) and Alkane/Alkene carbon chains (red/blue)

[12]see chapter xxx for an example of this

Pattern Matching using t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction technique used in automatic categorisation of images or photographs [Stefaner, 2020; Sangkloy et al., 2016]. This is the same process as referenced in EARLIERREF and described in detail within Chapter...

To compare the generated networks, we flatten the pixel matrix for each centered image in the dataset, and assign the output list to each filename. The resultant dataframe is then fed into the t-SNE algorithm in the Scikit Learn package [Pedregosa et al., 2011]. This reduces the logical list of pixels for each image into a two dimensional representation of their similarity. We plot each file, for its (x, y) coordinate, and isolate clusters of similarity using density contours in

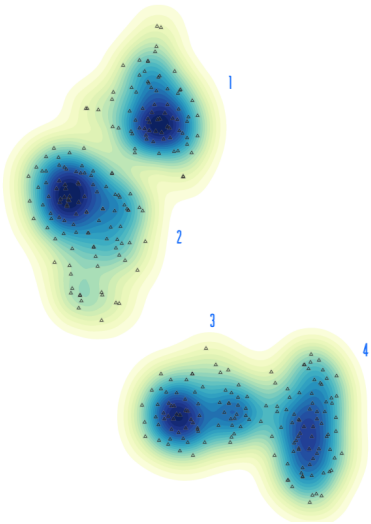
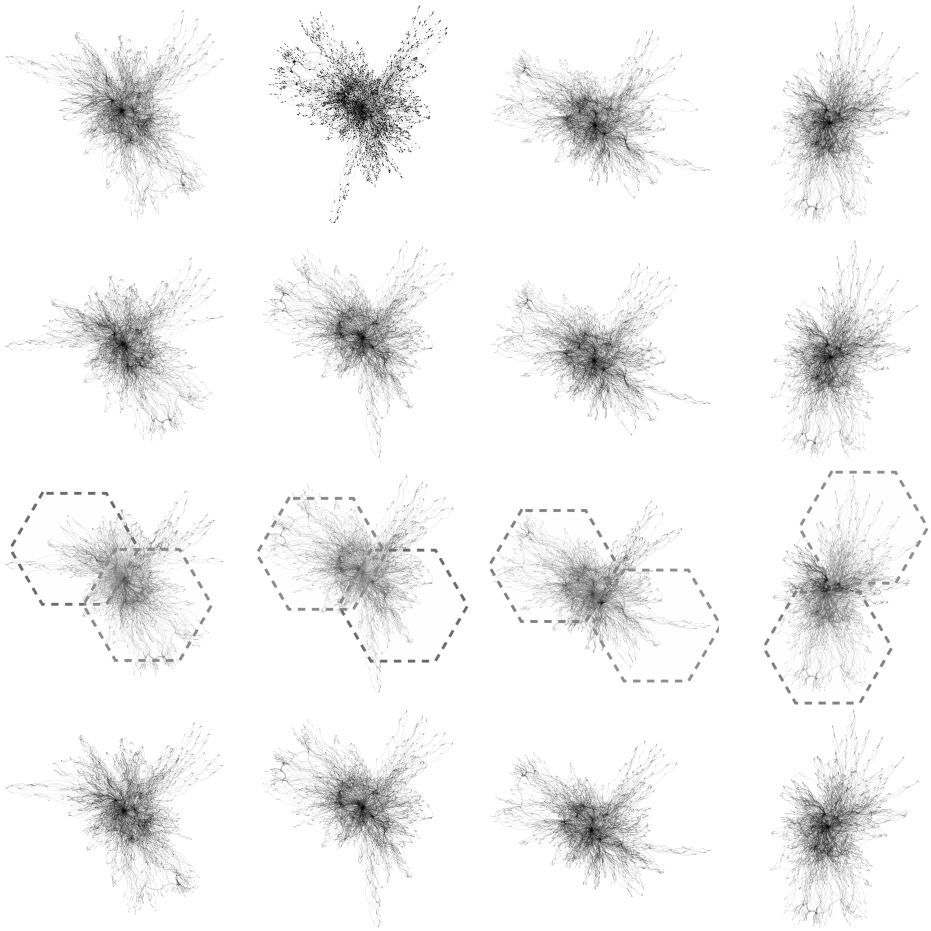


Figure 1.29: **A normalised scatter plot of 2D space produced by the t-SNE algorithm.** Each triangle represents a different file, and the colours/density contours show the regions in which we find similar images/graphs.

Using interactivity and/or vector cluster detection techniques it is possible examine which files contribute to an area of high density. Figure 1.30 shows a sample of four graphs from each corresponding cluster. Although individual node locations may vary, patterns on the macro scale start to emerge, with similar groups exhibiting symmetrical symmetry, e.g. groups 1/2 and 3/4. This suggests a constraint in the overall degree of freedom can be attributed solely to the network structure, and consequently the chemistry which forms this. The non-random nature of the produced graph layouts mean that it would be possible to juxtapose a variety or mechanisms using the force atlas 2 layout.



(a) Group 1 (b) Group 2 (c) Group 3 (d) Group 4

Figure 1.30: **A selection of graphs corresponding to the labeled clusters in Figure 1.29**. These reveal that symmetric similarity between like-positioned points within the t-SNE output.

1.6 Summary

Representing data in a visual format can be used to (utilise) the pattern recognition side of the human brain and alleviate the cognitive strain produced by numerical data. This is a technique used by the Samaritans (YEAR) with the use of cuneiform, and proved useful throughout.

In designing a visualisation it is important to use storytelling and select metaphors familiar to the reader. This should be paired with the correct encoding, as to reduce the time spent trying to comprehend a figure, and increase the knowledge transfer. [ref chapter 1]

When considering relationships, one such analogy lies in the ball and stick analogy. Much like holding hands, this symbolises a similarity between connected items and is the basis of a mathematical graph, or network. Such representations can be applied to the chemical complexity shown in species within the atmosphere.

In representing the chemistry within a mechanism as a graph we may visualise it with the use of a

force-directed layout. These are in essence a simple physical simulation, whereupon each graph node is repelled (like-charge), and connected nodes joined by a spring-like attractive force. It is found that the force atlas 2 algorithm not only produces the best visual aesthetic, but also conceptual understanding. Using this it is possible to see patterns such as the the partitioning of each network into aromatic, terpene and straight chain chemistry.

Although graph layouts have a range of local minima, the overall network structure of the MCM is constrained by its construction protocol (due to the allowed chemical reactions), and thus can be used to produce comparable graphs. This method of visualisation, in combination with interactive querying techniques, can aid in the comparison and understanding of large/complex chemistry simulations. This can be particularly useful in the explanation of specific interactions within a mechanism, or the exploration of temporal changes within a simulation.

In the next chapter, I shall extend the graph metaphor for atmospheric chemistry systems beyond that of just visualisation.

Bibliography

Archambault, D., Abello, J., Kennedy, J., Kobourov, S., Ma, K.-L., Miksch, S., Muelder, C., and Telea, A. C. (2014). *Temporal Multivariate Networks*, pages 151–174. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-06793-3_8.

Aumont, B., Szopa, S., and Madronich, S. (2005). Modelling The Evolution Of Organic Carbon During Its Gas-Phase Tropospheric Oxidation: Development Of An Explicit Model Based On A Self Generating Approach. *Atmospheric Chemistry and Physics*, 5:2497–2517. https://www.atmos-chem-phys.net/5/2497/2005/acp-5-2497-2005.pdf.

Bach, B. (2020). Confluent Graphs. https://aviz.fr/~bbach/confluentgraphs/.

Bach, B., Riche, N. H., Hurter, C., Marriott, K., and Dwyer, T. (2017). Towards Unambiguous Edge Bundling: Investigating Confluent Drawings For Network Visualization. *IEEE transactions on visualization and computer graphics*, 23(1):541–550. http://dx.doi.org/10.1109/TVCG.2016.2598958.

Baronchelli, A., Ferrer-i Cancho, R., Pastor-Satorras, R., Chater, N., and Christiansen, M. H. (2013). Networks In Cognitive Science. *Trends in cognitive sciences*, 17(7):348–360. http://dx.doi.org/10.1016/j.tics.2013.04.010.

Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. *AAAI*. http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

Bennett, C., Ryall, J., Spalteholz, L., and Gooch, A. (2007). The Aesthetics Of Graph Visualization. In *Computational Aesthetics in Graphics, Visualization, and Imaging*. The Eurographics Association.

Bergwerf, H. (2019). Molview. http://molview.org/.

Chernobelskiy, R., Cunningham, K. I., Goodrich, M. T., Kobourov, S. G., and Trott, L. (2012). *Force-Directed Lombardi-Style Graph Drawing*, pages 320–331. Springer Berlin Heidelberg, Berlin, Heidelberg. http://dx.doi.org/10.1007/978-3-642-25878-7_31.

Davidson, R. and Harel, D. (1996). Drawing graphs nicely using simulated annealing. *ACM Trans. Graph.*, 15(4):301–331. http://doi.acm.org/10.1145/234535.234538.

Di Battista, G., Eades, P., Tamassia, R., and Tollis, I. G. (1994). Algorithms for drawing graphs: An annotated bibliography. *Comput. Geom. Theory Appl.*, 4(5):235–282. http://dx.doi.org/10.1016/0925-7721(94)00014-X.

Di Battista, G., Mariani, F., Patrignani, M., and Pizzonia, M. (2004). *Bgplay: A System For Visualizing The Interdomain Routing Evolution*, pages 295–306. Springer Berlin Heidelberg, Berlin, Heidelberg. http://dx.doi.org/10.1007/978-3-540-24595-7_27.

Dianati, N. (2016). Unwinding the hairball graph: Pruning algorithms for weighted complex networks. *Phys. Rev. E*, 93:012304. https://link.aps.org/doi/10.1103/PhysRevE.93.012304.

Dick Derwent, Andrea Fraser, J. A. M. J. P. W. T. M. (2010). Evaluating The Performance Of Air Quality Models. https://uk-air.defra.gov.uk/assets/documents/reports/cat05/1006241607_100608_MIP_Final_Version.pdf.

Dixon, D. (2012). *Analysis Tool Or Research Methodology: Is There An Epistemology For Patterns?*, pages 191–209. Palgrave Macmillan UK, London. http://dx.doi.org/10.1057/9780230371934_11.

Duyckaerts, C. and Godefroy, G. (2000). Voronoi tessellation to study the numerical density and the spatial distribution of neurones. *Journal of Chemical Neuroanatomy*, 20(1):83 – 92. http://www.sciencedirect.com/science/article/pii/S0891061800000648.

Dwyer, T., Koren, Y., and Marriott, K. (2006a). Drawing directed graphs using quadratic programming. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):536–548.

Dwyer, T., Koren, Y., and Marriott, K. (2006b). Ipsep-Cola: An incremental procedure for separation constraint layout of graphs. *IEEE Trans. Vis. Comput. Graph.*, 12(5):821–828.

Dwyer, T., Marriott, K., and Stuckey, P. J. (2006c). *Fast Node Overlap Removal*, pages 153–164. Springer Berlin Heidelberg, Berlin, Heidelberg. http://dx.doi.org/10.1007/11618058_15.

Eades, P. (1984). A heuristic for graph drawing. pages 149–160. cited By 1.

Foo, B. (2019). Memory Underground - Convert Your Memories Into A Subway Map - Home. http://memoryunderground.com/.

Friedrich, C. and Schreiber, F. (2004). Flexible layering in hierarchical drawings with nodes of arbitrary size. In *Proceedings of the 27th Australasian Conference on Computer Science - Volume 26*, ACSC '04, pages 369–376, Darlinghurst, Australia, Australia. Australian Computer Society, Inc. http://dl.acm.org/citation.cfm?id=979922.979966.

Fruchterman, T. M. J. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164. https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380211102.

Förster, H., Ganian, R., Klute, F., and Nöllenburg, M. (2019). On strict (outer-)confluent graphs.

García-Pérez, G., Allard, A., Ángeles Serrano, M., and Boguñá, M. (2019). Mercator: Uncovering Faithful Hyperbolic Embeddings Of Complex Networks. *arxiv*. http://arxiv.org/abs/1904.10814.

Gershon, N. and Page, W. (2001). What storytelling can do for information visualization. *Commun. ACM*, 44(8):31–37. http://doi.acm.org/10.1145/381641.381653.

Goodrich, M. T. and Wagner, C. G. (1998). *A Framework For Drawing Planar Graphs With Curves And Polylines*, pages 153–166. Springer Berlin Heidelberg, Berlin, Heidelberg. http://dx.doi.org/10.1007/3-540-37623-2_12.

Görg, C., Pohl, M., Qeli, E., Xu, K., Ebert, A., and Meyer, J. (2007). *Visual Representations*, pages 163–230. Springer Berlin Heidelberg, Berlin, Heidelberg. http://dx.doi.org/10.1007/978-3-540-71949-6_4.

Harari, Y. (2015). *Sapiens: A Brief History Of Humankind*. Harper. https://books.google.co.uk/books?id=FmyBAwAAQBAJ.

Hazewinkel, M. (1997). *Encyclopaedia Of Mathematics: Supplement*. Number v. 1 in Encyclopaedia of Mathematics. Springer Netherlands. https://books.google.co.uk/books?id=3ndQH4mTzWQC.

Holten, D. (2006). Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748.

Hu, Y. (2004). Efficient, High-Quality Force-Directed Graph Drawing. *web*. http://yifanhu.net/PUB/graph_draw.pdf.

Jacomy, M., Venturini, T., Heymann, S., and Bastian, M. (2014). Forceatlas2, A Continuous Graph Layout Algorithm For Handy Network Visualization Designed For The Gephi Software. *PloS one*, 9(6):e98679. http://dx.doi.org/10.1371/journal.pone.0098679.

Jankun-Kelly, T. J., Dwyer, T., Holten, D., Hurter, C., Nöllenburg, M., Weaver, C., and Xu, K. (2014). *Scalability Considerations For Multivariate Graph Visualization*, pages 207–235. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-06793-3_10.

Jenkin, M., Watson, L., Utembe, S., and Shallcross, D. (2008). A common representative intermediates (cri) mechanism for voc degradation. part 1: Gas phase mechanism development. *Atmospheric Environment*, 42(31):7185 – 7195. http://www.sciencedirect.com/science/article/pii/S1352231008006742.

Jenkin, M. E., Saunders, S. M., and Pilling, M. J. (1997). The Tropospheric Degradation Of Volatile Organic Compounds: A Protocol For Mechanism Development. *Atmospheric environment*, 31(1):81–104. http://www.sciencedirect.com/science/article/pii/S1352231096001057.

Johnson, S. (2010). *Where Good Ideas Come From*. Penguin Publishing Group. https://books.google.co.uk/books?id=3H2Xg5qxz-8C.

Kerren, A., Purchase, H. C., and Ward, M. O. (2014). *Introduction To Multivariate Network Visualization*, pages 1–9. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-06793-3_1.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680. http://science.sciencemag.org/content/220/4598/671.

Klicpera, J., Bojchevski, A., and Günnemann, S. (2018). Predict Then Propagate: Graph Neural Networks Meet Personalized Pagerank. *Arxiv*. http://arxiv.org/abs/1810.05997.

Kohlbacher, O., Schreiber, F., and Ward, M. O. (2014). *Multivariate Networks In The Life Sciences*, pages 61–73. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-06793-3_4.

Kumar, G. and Garland, M. (2006). Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):805–812.

Lu, S. (2019). D3-Annotate. https://d3-annotation.susielu.com/.

Lyons, K. A. (1992). Cluster busting in anchored graph drawing. In *Proceedings of the 1992 Conference of the Centre for Advanced Studies on Collaborative Research - Volume 1*, CASCON '92, pages 7–17. IBM Press. http://dl.acm.org/citation.cfm?id=962198.962200.

Ma, K. and Muelder, C. W. (2013). Large-scale graph visualization and analytics. *Computer*, 46(7):39–46.

Maaten, L. v. d. and Hinton, G. (2008a). Visualizing Data Using T-Sne. *Journal of machine learning research: JMLR*, 9(Nov):2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Maaten, L. v. d. and Hinton, G. (2008b). Visualizing Data Using T-Sne. *Journal of machine learning research: JMLR*, 9(Nov):2579–2605. http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

Martin, S., Brown, W., Klavans, R., and Boyack, K. (2011). Openord: An open-source toolbox for large graph layout. *Proc SPIE*, 7868:786806.

Martin Grandjean (2016). Connected World: Untangling The Air Traffic Network. http://www.martingrandjean.ch/connected-world-air-traffic-network/.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092. http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114.

Michal, G. (1965). Metabolic Pathways. https://www.roche.com/sustainability/philanthropy/science_education/pathways/pathways-ordering.htm.

Montañez, A. (2016). How Science Visualization Can Help Save The World. https://blogs.scientificamerican.com/sa-visual/how-science-visualization-can-help-save-the-world/.

Mortenson, M. (1999). *Mathematics For Computer Graphics Applications*. Industrial Press. https://books.google.co.uk/books?id=YmQy799flPkC.

Muelder, C., Gou, L., Ma, K.-L., and Zhou, M. X. (2014). *Multivariate Social Network Visual Analytics*, pages 37–59. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-06793-3_3.

Needham, M. and Hodler, A. E. (2019). Practical Examples In Apache Spark & Neo4J. *O'Reilly*. https://neo4j.com/neoassets/graphbooks/Graph_Algorithms_Neo4j.pdf.

Noack, A. (2004). *An Energy Model For Visual Graph Clustering*, pages 425–436. Springer Berlin Heidelberg, Berlin, Heidelberg. http://dx.doi.org/10.1007/978-3-540-24595-7_40.

Norman, D. (2005). *Emotional Design: Why We Love (Or Hate) Everyday Things*. Basic Books. https://books.google.nl/books?id=h_wAbnGlOC4C.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pohl, M., Schmitt, M., and Diehl, S. (2009). Comparing The Readability Of Graph Layouts Using Eyetracking And Task-Oriented Analysis. In *Computational Aesthetics in Graphics, Visualization, and Imaging*. The Eurographics Association.

Purchase, H. (1997). *Which Aesthetic Has The Greatest Effect On Human Understanding?*, pages 248–261. Springer Berlin Heidelberg, Berlin, Heidelberg. http://dx.doi.org/10.1007/3-540-63938-1_67.

Purchase, H. C. (2002). Metrics for graph drawing aesthetics. *Journal of Visual Languages and Computing*, 13(5):501 – 516. http://www.sciencedirect.com/science/article/pii/S1045926X02902326.

Purchase, H. C., Colpoys, L., Carrington, D., and McGill, M. (2003). *Uml Class Diagrams: An Empirical Study Of Comprehension*, pages 149–178. Springer US, Boston, MA. http://dx.doi.org/10.1007/978-1-4615-0457-3_6.

Roberts, J. C., Yang, J., Kohlbacher, O., Ward, M. O., and Zhou, M. X. (2014). *Novel Visual Metaphors For Multivariate Networks*, pages 127–150. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-06793-3_7.

Sangers, A., van Heesch, M., Attema, T., Veugen, T., Wiggerman, M., Veldsink, J., Bloemen, O., and Worm, D. (2019). Secure Multiparty Pagerank Algorithm For Collaborative Fraud Detection. In *Financial Cryptography and Data Security*, pages 605–623. Springer International Publishing. http://dx.doi.org/10.1007/978-3-030-32101-7_35.

Sangkloy, P., Burnell, N., Ham, C., and Hays, J. (2016). The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 35(4). https://doi.org/10.1145/2897824.2925954.

Schreiber, F., Kerren, A., Börner, K., Hagen, H., and Zeckzer, D. (2014). *Heterogeneous Networks On Multiple Levels*, pages 175–206. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-06793-3_9.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, pages 336–, Washington, DC, USA. IEEE Computer Society. http://dl.acm.org/citation.cfm?id=832277.834354.

Shneiderman, B. (1997). *Designing The User Interface: Strategies For Effective Human-Computer Interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition.

Staples, J., Nickerson, D. A., and Below, J. E. (2013). Utilizing graph theory to select the largest set of unrelated individuals for genetic analysis. *Genetic Epidemiology*, 37(2):136–141. https://onlinelibrary.wiley.com/doi/abs/10.1002/gepi.21684.

Stefaner, M. (2020). Truth & Beauty - Multiplicity. https://truth-and-beauty.net/projects/multiplicity.

Steven Franconeri (2018). Openvis Conference Proceedings. https://www.youtube.com/watch?v=Jq2Rc0WlYTE.

Taylor, M. and Rodgers, P. (2005). Applying graphical design techniques to graph visualisation. In *Ninth International Conference on Information Visualisation, 06-08 July 2005, London, England: Proceedings*, pages 651–656. IEEE Computer Society. http://kar.kent.ac.uk/14297/.

Thomas, P. (1952). *Conformal Projections In Geodesy And Cartography*. Special publication. Coast and Geodetic Survey. https://books.google.co.uk/books?id=7a60MQEACAAJ.

VTL (2019). Visual Thinking Lab. http://visualthinking.psych.northwestern.edu/.

Ware, C. (2013). Chapter two - the environment, optics, resolution, and the display. In Ware, C., editor, *Information Visualization (Third Edition)*, Interactive Technologies, pages 31 – 68. Morgan Kaufmann, Boston, third edition edition. http://www.sciencedirect.com/science/article/pii/B9780123814647000028.

Ware, C., Purchase, H., Colpoys, L., and McGill, M. (2002). Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110. http://dx.doi.org/10.1057/palgrave.ivs.9500013.

Wybrow, M., Elmqvist, N., Fekete, J.-D., von Landesberger, T., van Wijk, J. J., and Zimmer, B. (2014). *Interaction In The Visualization Of Multivariate Networks*, pages 97–125. Springer International Publishing, Cham. http://dx.doi.org/10.1007/978-3-319-06793-3_6.